Deep learning for Genome-Wide Association Studies

Deepak Sharma

Computer Science
McGill University, Montreal

December 12, 2021

A thesis submitted to McGill University in partial fulfilment of the requirements of the degree of Master of Science.

Abstract

Genome-Wide Association Studies (GWAS) are a popular tool in statistical genomics that are used to identify genetic variants associated with various diseases. However, their success has been limited, in part because they typically do not incorporate interactions between variants to model target traits. Since Deep neural networks have been successful across domains abundant with complex signals, like speech, language, and vision, they are also popular candidates for modelling interactions between genetic variants. However, their black-box nature is a hindrance to their application for GWAS.

In this thesis, we present a pipeline to train and interpret feedforward neural networks to conduct a genome-wide association study (GWAS). We show that trained deep neural networks can be interpreted using feature-importance techniques to accurately distinguish and rank simulated causal genetic variants. We improve its accuracy by extending the pipeline to the multi-task setting, wherein we simultaneously model two related, simulated traits. We demonstrate the accuracy, reliability, and scalability of our approach by identifying most known Diabetes genetic risk factors found using a conventional GWAS on the UK Biobank.

Résumé

Les études d'association pangénomiques (GWAS) sont un outil statistique important pour identifier des variations génétiques associées aux maladies. Cependant, leur succès est possiblement atténué par l'omission d'effets d'interaction entre les variations génétiques et les phénotypes d'intérêt. Comme les réseaux de neurones profonds se sont avérés efficaces dans plusieurs domaines riches en signaux complexes, comme l'analyse de la parole, du langage et de la vision, ils sont des candidats intéressants pour modéliser les interactions entre les variations génétiques. Cependant leur nature opaque ("black box") est une lacune importante pour leur utilisation dans le contexte des GWAS. Dans cette thèse, nous présentons une méthode pour entraîner et interpréter des réseaux de neurones à propagation vers l'avant dans le cadre de GWAS. Nous montrons que les réseaux ainsi entrainés peuvent être interprétés en utilisant des méthodes d'attribution de l'importance des variables pour distinguer et ordonner des variations génétiques causales simulées. L'entraînement en mode multitâche, où nous modélisons simultanément deux phénotypes reliés, a aussi permis d'améliorer l'exactitude de notre méthode. Nous démontrons l'exactitude, la fiabilité et la performance de notre approche en identifiant à nouveau les facteurs de risque génétique les plus connus pour le diabète dans la UK Biobank.

Acknowledgements

I am extremely grateful to my supervisors, Dr. Audrey Durand and Dr. Joelle Pineau, for giving me the freedom to explore my ideas. Their feedback was invaluable in shaping the key directions of the work in this thesis. I was consistently in awe of their unyielding optimism and patience, which I have resolved to emulate in my career. I am also extremely grateful to Dr. Marc-André Legault, who consistently provided me with his time and his knowledge of genomics, which was necessary in shaping and steering this project, time and time again. I am deeply grateful to Dr. Marie-Pierre Dubé for her constant guidance in framing the bigger picture. I would also like to thank Louis-Philippe Lemieux Perreault and Audrey Lemaçon, for their diligent effort in exploring and testing new ideas.

I wish to thank my parents and my sister. I would not be here without their unconditional love and support — I love you.

Contribution of Authors

In this thesis, we presented a pipeline to train and interpret feedforward neural networks to conduct a genome-wide association study (GWAS) and showed that trained deep neural networks can be interpreted using feature-importance techniques to accurately distinguish and rank simulated causal genetic variants. The work in Chapter 3 and the single-task results in Chapter 5 are presented in the "Interpretability in Machine Learning for Scientific Discovery" workshop at ICML 2020. This is joint work with Audrey Durand, Marc-André Legault, Louis-Philippe Lemieux Perreault, Audrey Lemaçon, Marie-Pierre Dubé, and Joelle Pineau. We also extended our approach and analysis to the multi-task setting, wherein we simultaneously model two related, simulated traits. This is covered in Chapter 4.

The empirical analysis of the accuracy and rank correlation performance metrics on simulated data across Chapters 3 and 4 was conducted with the help of Audrey Durand, Marc-André Legault, and Joelle Pineau.

In Chapter 5, our single-task and multi-task experiments on Diabetes and HbA1c were conducted with help from Audrey Durand, Marc-André Legault, Louis-Philippe Lemieux Perreault, Audrey Lemaçon, and Marie-Pierre Dubé. Finally, Marc-André Legault's help was invaluable in preprocessing the UK Biobank data and conducting the conventional GWAS on the UK Biobank data for both Diabetes and HbA1c .

Contents

List of Figures

List of Tables

1

Introduction

Beginning on October 1st, 1990, an international team of researchers led one of the greatest feats of human exploration. They sought to sequence and map all genes that make up a human being. This historic endeavour is called the Human Genome Project ("Initial sequencing and analysis of the human genome" 2001), and it formally completed in April 2003, opening the first window into nature's genetic manual for a human being. The next challenge is to figure out how to read its contents, understand how they work together to maintain our health, and more importantly how they contribute to human disease. For a brief background on human genetics, see Chapter 2.

A very common starting point is to observe naturally occuring biological differences between people and correlate them to the corresponding differences in their genetic makeup. After ruling out the correlations that could have occured due to chance, or explained by non-genetic factors like age and gender, a final list of genetic markers can be validated in controlled conditions in a lab. This is the goal of the Genome-Wide Association Study (GWAS, Pearson 2008). It trains mathematical models to predict the occurence of a target disease or measurement of a quantitative biological trait over a large population of individuals, implicitly forcing the model to learn to focus on parts of the genetic data that are relevant to explain the occurence of that trait.

GWAS are extremely common; as of 2019, 5687 GWAS across 3567 publications

had been registered in the GWAS catalog (Buniello et al. 2019), including studies of various diseases like Diabetes (Xue et al. 2018), Alzheimer's (Jansen et al. 2019), Coronary Artery Disease (CAD) (Deloukas et al. 2012), and continuous measurements like LDL-cholesterol (Sandhu et al. 2008) and adult human height (Wood et al. 2014). Despite their many successes, GWAS have failed to explain a large proportion of the variability within various traits of the sample populations they are conducted on. This is called the "missing heritability" problem (Eichler et al. 2010). Although there is no clear consensus on the main cause, one of the prevailing views is that non-additive, complex interactions between genetic variants might be "hiding" variants that would be discoverable if this complexity were to be taken into account. Typical GWAS models rely on a single genetic marker or linear combinations of genetic markers to explain trait variation in the study population (see Chapter 2). Thus, GWAS are limited to the identification of genetic variants with strong marginal effects, possibly leaving out a large number of genetic effects due to genetic interactions or other non-linear effects.

The overarching motivation of this thesis is to model and identify interactions between genetic markers that are significantly associated with a target biological trait. This thesis aims to incorporate Deep Neural Networks (Goodfellow, Bengio, and Courville 2016) into a GWAS in order to model complex, non-linear interactions between genetic markers. Deep Neural Networks are a powerful and flexibile modelling technique in Machine Learning, that have been shown to model complex, high dimensional data extremely well. They are at the center of a spate of remarkable advances in automated speech processing (Graves, Mohamed, and G. E. Hinton 2013; Bahdanau, Chorowski, et al. 2016), language generation (Bengio et al. 2000; Mikolov et al. 2010; Devlin et al. 2019), image recognition (Krizhevsky, Sutskever, and G. E. Hinton 2012; K. He et al. 2016), medical image segmentation (Ronneberger, Fischer, and Brox 2015), and molecular biology (Senior et al. 2020) etc, that have occured over the last decade or so.

In order to use Deep neural networks for GWAS, we need to train them on genomic data to predict a trait and identify which genetic markers are important to the model in making its predictions. Prediction on genomic data is challenging for neural networks because genomic datasets do not have enough samples to help the network distinguish patterns amongst the vast number of potential signals in each sample (Nicholls et al. 2020). Furthermore, it is difficult to *interpret* and *explain* predictions of Deep neural networks. This limits our *trust* in our ability to extract novel scientific insights from them. In fact, there is no established theory of the behaviour of deep learning models, nor a systematic comparison of techniques that can be used to interpret deep neural networks, especially in the context of genomic data. In contrast, these aspects are extremely well understood for the statistical tools used by GWAS today (see Section 2.1.3). *Thus, before we can use Deep neural networks to identify useful genetic interactions, we must investigate whether it is possible and practical to interpret them to find genetic markers identified by a conventional GWAS.* This helps us clarify the aim of this work:

Aim Incorporate Deep Neural Networks into a GWAS pipeline that is accurate, reliable, and as scalable as conventional GWAS on modern genomic datasets.

Objectives In order to achieve the achieve the aforementioned goal, it is necessary to:

1. Investigate how we can interpret a Deep Neural Network to measure the importance of the input to the model's predictions.

2. Devise a GWAS pipeline that trains and interprets Deep Neural Networks to identify genetic markers that are important to the model's predictions.

3. Compare its accuracy, reliability, and scalability to a conventional GWAS on a large real-world genomic dataset.

Outline We start with Sections 2.1.1 and 2.1.3 of Chapter 2, which give a brief background on human genetics and how GWAS are most commonly conducted today, including a brief discussion on how their results are interpreted. In Section 2.2, we cover Feedforward Neural Networks and how they are used in Machine Learning to predict a single target. In Section 2.3, we give a brief overview of the current state of research in explainability and interpretability of black-box models, including the motivations and goals of interpretability research. We then give a brief survey of current interpretability techniques for deep learning models but focus more on techniques that operate on trained models (post-hoc), as opposed to advances in deep learning theory or techniques that operate before the model has been trained (ad-hoc). As part of this survey, we introduce the concept of feature importance and discuss some common scalable deep neural network feature importance techniques such as Integrated Gradients, DeepLift, and GradInput (Sundararajan, Taly, and Yan 2017; Shrikumar, Greenside, and Kundaje 2017; Shrikumar, Greenside, Shcherbina, et al. 2016).

Chapter 3 brings these topics together to formulate a pipeline that trains multi-layered feedfoward neural networks and uses feature importance techniques to attribute importance scores to each genetic marker in order to conduct a GWAS. We present accuracy metrics that we can use to compare the different feature importance techniques that we consider for the pipeline. We also discuss the qualities that would be desirable for our pipeline to be "trustworthy" for a GWAS on real-world data, and present aspects from the Machine Learning interpretability literature that correspond to these qualities, namely, attribution Accuracy, Consistency, and Fidelity (see Section 3.2.2 and Robnik-Šikonja and Bohanec 2018). This helps us devise a methodology to evaluate, compare, and select between the various pipelines: conduct a GWAS using each pipeline on a simulated dataset with known causal genetic markers, and test their performance along each aforementioned aspect. Finally in this Chapter, we present positive results that demonstrate that most causal genetic

markers are not only identifiable but that their rankings based on their importance to the model correlate with their rankings based on the magnitude of their effect on the trait. We also observe and discuss some discrepancies between results for binary and quantitative targets, which motivates our work on the multi-task version of our pipeline in the next Chapter.

Since biological traits are often correlated and likely to share causal genetic factors (Visscher, Wray, et al. 2017), we hypothesize that information from predicting one target can improve prediction for another related target. Thus in Chapter 4, we extend our pipeline to the multi-task setting wherein a single model is trained to simultaneously predict a binary and a quantitative trait, both simulated to share a proportion of their causal markers. The results from our experiments indicate that compared to the scenario of predicting a single target trait (Chapter 3), multi-trait prediction does increase the number and ranking accuracy of correctly identified causal genetic markers of both traits. Finally in Chapter 5, we apply our methods from Chapter 3 and 4 to genomic data from the UK Biobank and identify known genetic risk factors for Diabetes and glycated hemoglobin measurements (HbA1c). Chapter 6 concludes this work with a discussion on the challenges of training artificial neural networks to predict biological traits from genomic data, issues related to the simulated setting used to compare different pipelines, and the robustness and reproducibility of our main results.

2

Background

Beginning with a primer on human genetics, this chapter introduces the reader to Genome-Wide Association Studies (GWAS), one of the most popular tools that geneticists have for identifying and studying the link between human genetics and disease, as well as Deep Neural Networks (DNN), a powerful tool that can be trained to spot patterns in data that are too complex for a human being.

2.1 Genome-Wide Association Studies

Every life form has traits that are passed down its generations. The set of traits that result from the interaction of an organism's genetic material and its environment is called its phenotype, for example, visible traits like height, skin colour, eye colour, and hair colour, and hidden traits like average heart beat and blood type etc. The genetic material that is responsible for the organism's development and function is called its genotype. The genotype of every life form on earth consists of a special molecule named DNA, which is short for dioxyribonucleic acid. Aptly named the "the blueprint of life", it affects the development, function, and inheritance of traits between generations. Tracking changes in DNA within a population and relating these changes to bodily functions and diseases, can help us understand how DNA works and personalize treatments to an individual. This chapter introduces a popular tool that

helps us do that, the Genome-Wide Association Study.

2.1.1 A primer on human genetics

The human body contains approximately 10^{13} cells (Bianconi et al. 2013). Each of these cells contains a nucleus, where DNA is organized into 23 pairs of chromosomes. DNA is made up of two long strands that wind around each other into a double helix. These strands are a sequence of four nucleic acids: adenine (A), cytosine (C), guanine (G) and thymine (T). If one strand contains adenine then the opposite strand will always contain thymine, while cytosine always pairs with guanine.

Subsequences of DNA code for protein molecules, which are vital for practically every process of a cell. These sequences are called genes and the set of all genes is called the genome. The first map of the human genome was provided by the Human Genome Project.

About 97% of the human genome is fixed across generations. The remaining portion varies between individuals and potentially contributes to similarity of traits between relatives. Thus it is possible to create a "reference genome" (Auton 2015) and track genomic variation down generations, and within or across populations.

A physical location of a gene or DNA sequence is given by its locus. If there is variation across genomes at a particular locus, then that region is referred to as a variant, and its different versions are called alleles. Single nucleotide polymorphisms (SNPs) are single base pair variants that are sufficiently common in a population (at least 1% or more of the population). For example, Figure 2.1 shows both copies of chromosome 2 at the same locus, in a population of 4 individuals, with a single variant highlighted in yellow. Only individual 2 contains a different set of nucleotides (both copies contain the pair C-G) compared to the rest of the population (one copy contains C-G and the other T-A). Since this is 25% ($> 1\%$) of the population, that site can be called a SNP, and the two different versions of this SNP are its alleles.

Figure 2.1: Both copies of chromosome 2 at the same locus, in a population of 4 individuals, with a single SNP highlighted in yellow. Courtesy: National Human Genome Research Institute

SNPs are the most common form of genetic variation in humans. On average, they occur once in every 1000 nucleotides, resulting in roughly 4 to 5 million SNPs per individual. The allele most common in the population is called the major allele and the second most common is called the minor allele. In Figure 2.1, the SNP has two alleles, with the major allele belonging to individuals 1,3, and 4. The allele found in the reference genome of the population is called the reference allele, and all other alleles of that variant are called alternative alleles.

Depending on their location, SNPs may play a direct or indirect role in diseases. Some SNPs are part of genes, while others are part of DNA between genes, which might affect the degree to which a gene is 'expressed'. Identifying SNPs that correlate with traits or diseases can help us gain insight into disease biology, identify individuals with a higher risk of complex diseases, and identify possible drug targets.

2.1.2 What is a GWAS?

A Genome-Wide Association Study (GWAS) is an observational study that is conducted with thousands of subjects to find genetic variants that are associated with a trait (Marees et al. 2018). Typically, several traits (or phenotypes) and covariates of

interest are measured, and upto a few million SNPs are genotyped.

Independently for each SNP, a linear model is trained to predict the target phenotype using the SNP and the covariates as input. This is typically done using a logistic regression model for binary targets and linear regression for quantitive (continuous) targets. The coefficient for the SNP term is tested for significance via a statistical hypothesis test like the likelihood ratio test or the Wald's test (B. Li and Babu 2019).

The most common approach to modelling SNPs is the additive model (Clarke et al. 2011), in which the number of minor alleles are counted and included as input. For example, assume that for a particular region of DNA, the minor allele is C (Cytosine) and the major allele is A (Adenine). Then the 3 possible combinations at that location are "AA", "AC", and "CC". Since we count the number of minor alleles, these combinations will be coded as 0, 1, and 2 respectively.

In the following section we will briefly describe the methods used to conduct a typical GWAS in order to clarify the motivations behind this thesis.

2.1.3 The traditional GWAS pipeline

As mentioned in Section 2.1.2, for a typical GWAS today, hundreds of thousands to millions of SNPs are genotyped alongwith measurements of several traits and relevant covariates across thousands of subjects. For simplicity, let's consider a single quantitive outcome for N samples, denoted as y_i for $i \in 1 \dots N$ and additively modelled genotype x_{ij} for SNP $j \in 1 \dots M$ and sample i. As mentioned in Section 2.1.2, quantitive phenotypes are analyzed using a linear regression model:

$$y_i = x_{ij}\beta_j^1 + \beta_j^0 + \epsilon_{ij}, \tag{2.1}$$

$$\epsilon \sim \mathcal{N}(0, \Sigma_{M\times M}), \tag{2.2}$$

where β_j^0 is the bias term for SNP j and ϵ are the residuals distributed normally with zero mean and covariance $\Sigma_{M\times M}$. The true effect vector for each SNP j is β_j^1,

with the null hypothesis being that $\beta_j^1 = 0$. The Wald's test is used for significance testing with the ratio $\frac{\hat{\beta}_j^1}{\text{se}(\hat{\beta}_j^1)}$ being the test statistic, $\hat{\beta}_j^1$ being the estimated regression coefficient for SNP j and se being the standard error, which is the square root of the variance of the Maximum Likelihood Estimate (MLE) of $\hat{\beta}_j^1$ (Fahrmeir et al. 2013, Appendix B.4.4).

This gives a p-value, which is the probability of getting a statistic at least as extreme as the computed test statistic assuming that the null hypothesis ($\beta_j^1 = 0$) is true. If the p-value for SNP j is smaller than some threshold then the null hypothesis can be rejected, and the association between the SNP and the phenotype is called significant. If it is larger then the null hypothesis is considered to be true and the alternative hypothesis is rejected. The probability threshold used to determine the significance of a test is called its significance threshold and is typically denoted by α. It is the probability of rejecting the null hypothesis assuming that it is true and is typically set to 0.05. In the context of the GWAS pipeline described in this section, the null hypothesis is rejected if the probability of the observed Wald's test statistic is less than α.

Multiple comparison correction Consider a sample with a million genotyped SNPs ($M = 10^6$ in Equation 2.2). With $\alpha = 0.05$, the likelihood of rejecting the null hypothesis at least once is $1 - (1 - \alpha)^{10^6} \approx 100\%$. This implies that it is practically certain to get a significant test statistic for any statistical analysis considering a large group of independant tests. This is called the Multiple Comparisons's problem (R. G. Miller 1981). A common method to account for the number of independant tests is by dividing α by the number of tests. This is called Bonferroni correction (Fahrmeir et al. 2013, Page 471):

$$\alpha^{corr} = \frac{\alpha}{N}, \tag{2.3}$$

where N is the number of statistical tests being conducted i.e. the number of SNPs in the GWAS. For an initial significance level $\alpha = 0.05$ and $N = 10^6$, the Bonferroni "corrected" significance level (α^{corr}) is 5×10^{-8}. This cut off is commonly considered as the "gold standard" significance threshold for evidence of association in a GWAS (Clarke et al. 2011; Pe'er et al. 2008).

Interpretation of results A significant result in a GWAS does not imply that a SNP is causal for a phenotype. Proof of causation requires an understanding of the biological mechanisms giving rise to the target phenotype and the role played by different alleles of the candidate SNP in that mechanism.

Instead, it is quite likely that the identified SNP is correlated with the true causal SNP. This non-random correlation between SNPs at different loci is termed Linkage Disequilibrium (LD). In fact, a significant result in a GWAS that is true is most likely to be due to LD, i.e. when a GWAS marks a SNP as significantly associated with the target phenotype then that association is most likely indirect due to LD between the associated and causal SNPs. In the case of a spurious association, it is likely that the study has data quality issues in the form of sampling bias or because it has not sufficiently controlled for its confounders.

In GWAS, a confounder is a covariate that modulates the association between a SNP and a phenotype without being a consequence of the SNP. If a model does not take into account the effects of the confounder then the estimate of the effect parameter $\hat{\beta}^1$ is statistically biased or confounded. Common confounders in GWAS are population stratification and admixture. The former refers to differences in frequencies of the alleles of a SNP due to differences in ancestry whereas the latter is due to different patterns of LD in populations consisting of individuals that have mixed genetic ancestry (Clarke et al. 2011).

Confounding in GWAS is minimized via quality control of the samples used in the analysis to minimize population stratification, directly adding common covariates

like age and sex, and finally by including covariates constructed using dimensionality reduction techniques like Principal Components Analysis (PCA, Jolliffe 1986) conducted on the matrix of genotypes of all the samples in the study.

2.2 Deep Neural Networks

An artificial Neural Network (ANN) is a flexible modelling technique in machine learning that is loosely inspired by the structure and function of the brain (Goodfellow, Bengio, and Courville 2016). The primary unit of computation in an ANN is an artificial neuron that combines a linear and non-linear transformation of the input signal (in that order). It first applies a linear mapping on an input vector $x \in R^n$ to predict a scalar $\hat{y} \in R$ using the parameters $\theta \in R^n$, followed by a non-linear function $g : R \to R$:

$$\hat{y} = g(\theta^T x) \tag{2.4}$$

Artificial neural networks are constructed using several artificial neurons that are organized into layers stacked on top of each other as shown in Figure 2.3. There are several variations of these networks that arrange the neurons in different ways to model different modalities of data. Figure 2.3 displays one of the most basic and popular ways these units can be arranged. This arrangement is called a Feedforward neural network because the information flows from the input to the output with no feedbacks between layers (during prediction).

The number of units in a layer is its width, while the number of layers is called the depth of a neural network. These two hyperparameters are key to its representational power. The non-linear function used in each neuron is called the activation function. Designing activation functions is a hot area of deep learning research but the most popular is the ReLU (Nair and G. E. Hinton 2010; Xu et al. 2015). Networks that

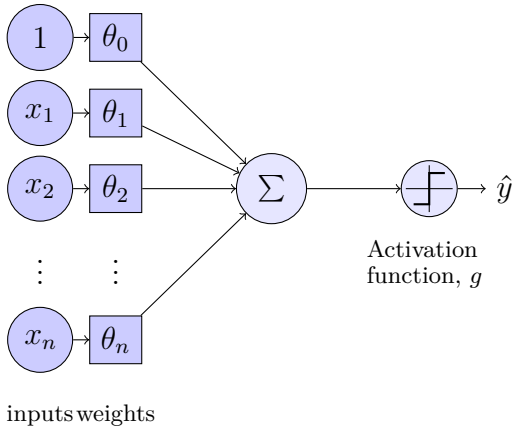

Figure 2.2: This figure illustrates Equation 2.4 in more detail with the addition of the bias term θ_0

have many layers with a large number of neurons are called deep neural networks. They have been shown to possess the ability to learn low dimensional representations from high dimensional input data (Goodfellow, Bengio, and Courville 2016).

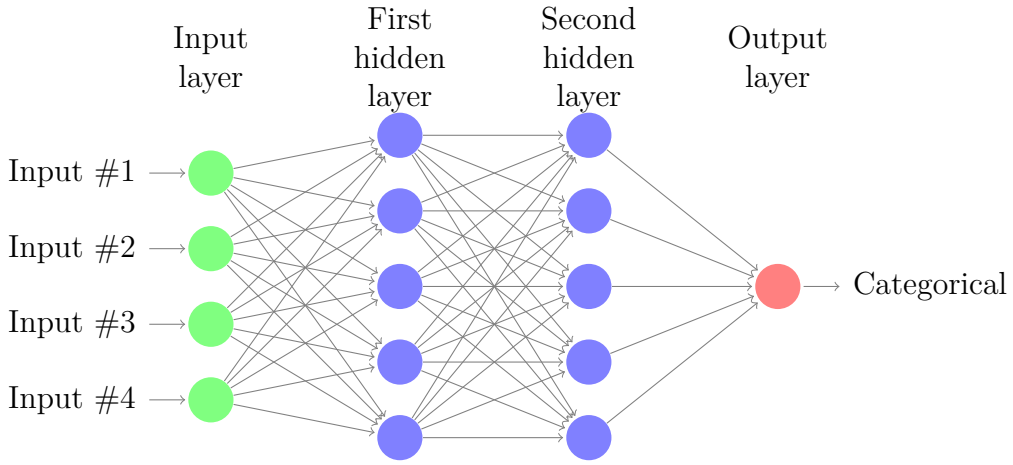

Figure 2.3: A feedforward neural network with one output head

A neural network is trained to learn a function $\hat{f}$ to best approximate the true function that maps some input x to output y. The quality of this approximation is judged by a loss function $\mathcal{L}$ that compares the predictions made by the neural network model to the true value in the dataset. Typically, the loss function is positive when the model makes a mistake and 0 when it is correct. For example, the Binary Cross Entropy function (BCE, Equation 2.5) is most commonly used for binary classification

tasks, whereas the Mean Squared Error (MSE, Equation 2.6) is popular for regression. Thus, one of the basic objectives of any machine learning model is to minimize the loss function computed on the entire training set:

$$\mathcal{L}^{class} = -\sum_i (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)), \tag{2.5}$$

&

$$\mathcal{L}^{regr} = \sum_i 0.5 * \|y_i - \hat{y}_i\|_2^2 \tag{2.6}$$

The loss function ($\mathcal{L}$) is minimized via an optimization procedure called gradient descent, which modifies the parameters of the neural network against the direction of the gradient of the objective function with respect to the parameters. The gradient for each weight is computed by the repeated application of the chain rule for derivatives, starting from $\mathcal{L}$ and backwards to the parameters of the first layer. Along with clever reuse of some intermediate computation, this procedure to calculate the gradients for gradient descent is called backpropagation (Rumelhart, G. E. Hinton, and R. J. Williams 1986b).

Popular synonyms for the loss function are the objective function, the cost function, and the error function (Goodfellow, Bengio, and Courville 2016). However, sometimes the error in prediction is insufficient to fully specify the desired solution. For example, a machine learning practitioner might prefer a model in which the computation of the model's output is easily traceable from the input to the output, in which case each of the weight matrices of the model would have to be sparse. A straightforward way to encourage the learning algorithm to arrive at more sparse solutions for θ is to add the L_1 norm of the weights as an additional penalty to the loss function $\mathcal{L}$ (Goodfellow, Bengio, and Courville 2016):

$$J = \mathcal{L} + \lambda \sum_i |\theta_i|, \tag{2.7}$$

where higher values of λ can be used to put more pressure on the algorithm to sparsify the weights and J is the modified loss function, also called the objective function to distinguish it from the prediction loss $\mathcal{L}$. The modification of $\mathcal{L}$ in order to specify a preference for one solution over another is a form of Regularization, which is a key concept in Machine Learning (Hastie, Tibshirani, and Friedman 2009). Equation 2.7 is an example of parametric regularization via addition of a norm penalty $\Omega(\theta) = \sum_i |\theta_i|$. Another example of a norm penalty used for regularization is the L_2 norm of the weights: $\Omega(\theta) = \frac{1}{2}\|\theta\|_2^2$ (Goodfellow, Bengio, and Courville 2016). Using the L_2 norm penalty for regularization encourages the weights of the network to be closer to the origin (Goodfellow, Bengio, and Courville 2016). A generalized version of the final objective function which contains a regularization penalty is given by Equation 2.8:

$$J = \mathcal{L} + \lambda \cdot \Omega(\theta), \tag{2.8}$$

where λ is a hyperparameter used to balance between the prediction error $\mathcal{L}$ and the penalty $\Omega(\theta)$. Finally, the optimal parameters of the model ($\hat{\theta}$) are obtained by minimizing J as shown in Equation 2.9:

$$\hat{\theta} = \text{argmin}_\theta J \tag{2.9}$$

The explosion in the use of deep learning techniques for machine prediction tasks is due to their competitive performance at prediction in a wide variety of settings as well as their ease of use. The extreme flexiblity that this approach provides and the relative simplicity of the optimization procedure allows practitioners to easily train architectures that are tailored to any input modality e.g. Recurrent Neural Networks

(RNNs, Rumelhart, G. E. Hinton, and R. J. Williams 1986a) for sequential data, and Convolutional Neural Networks (CNNs, LeCun et al. 1989) for image data. However, their black box nature is a double edged sword; it makes it easier to adopt these models for a variety of prediction tasks without worrying about how the model made its predictions. Particularly, it is difficult to understand which aspects of the input data were the most useful to the model's predictions. The incredible speed at which deep learning models are being adopted across various industries has led to renewed interest in research on deep learning explainaibility. This topic is explored in the next Section.

2.3 Interpretability of Deep neural networks

Accurate prediction from massive amounts of data across various modalities is a very attractive feature for any general-purpose prediction technique. But is it all that we ask of systems that are slowly becoming pervasive in society? If an automatic diagnosis system running using a DNN makes a recommendation that could be potentially threatening to a patient's life then the clinician using that system will take it into account if and only if they trust its rationale. But what is the rationale behind the prediction of a neural network? How can we verify it? How do we contextualize it for the clinician? Furthermore, can the clinician trust it?

These are some of the most important questions that motivate a field of machine learning research termed Explainable AI, or XAI (Gunning et al. 2019; Escalante et al. 2018; Barredo Arrieta et al. 2020). The recent explosion in the deployment and research of machine learning systems has given rise to a surge of interest in the field. However, it is not immediately clear how to answer the questions in the preceding paragraph because they are qualitative in nature. In fact, terms like trust

and explanation (or rationale) can have varying definitions across the literature, which makes it difficult to assess the claims of explainability techniques and compare them to each other. A related and an equally nebulous and important term in the field of XAI is interpretability. Research on making models more interpretable is motivated by the desire to explain and possibly extrapolate model behaviour. Thus, explainability may be viewed as a goal of interpretability research.

This section provides an overview of the motivations, goals and methods behind interpretability research of machine prediction in order to help clarify the distinct ideas that constitute the field. While a large number of existing reviews focus on explanations of general AI techniques (Lipton 2018; T. Miller 2017; Adadi and Berrada 2018; Gilpin et al. 2019) we follow Fan, Xiong, and G. Wang 2020 to present a taxonomy focussed on deep learning interpretability.

2.3.1 What do we want interpretable AI to do?

Interpretability refers to the extent to which a model's behaviour can be summarized at different abstraction levels. Unfortunately, the black-box nature of deep learning models limits our ability to understand the decision making process behind a prediction. Based on Lipton 2018 we summarize the real-world goals of interpretability research:

Trust Works like B. Kim (2015a) and Ribeiro, Singh, and Guestrin (2016) suggest that trust in a model cannot be built if the model is not interpretable. But the notion of trust is ill-defined and highly contextual. If accurate prediction is all that is required of an AI technique then its demonstration is sufficient for a model to be deemed trustworthy. Alternatively, if the training and deploment environments differ then trust requires demonstration of the robustness of the model's predictions; even so, for models that could directly impact lives automated crime forecasting systems (Richardson, Schultz, and Crawford 2019), we would require that the model makes

the right predictions for the right reasons. In fact, for groups that already face a disproportionate risk of structural harm (Galtung 1969) the development of trust requires transparency and accountability into the development of the entire model development pipeline (Raji et al. 2020; D. R. Williams et al. 2010).

Causality Most of statistical learning theory that is in practical use today helps algorithms learn statistical dependencies between observations that are useful for prediction (Hastie, Tibshirani, and Friedman 2009). But a purely statistical relationship is fickle and weaker than a cause-effect relationship. This phenomenon of a cause-effect relationship between two quantities is called Causality, and it is much more valuable and more difficult to obtain (Peters, Janzing, and Schlkopf 2017). A practical general-purpose prediction technique holds the promise to greatly accelerate the rate at which scientific discoveries are made if researchers can rely on it to learn an accurate model of the world. A mechanistic understanding of a model's inner workings will allow researchers to generate and test hypothesis about the natural world. Deep neural networks model complex non-linear functions of their inputs using thousands to billions of parameters (T. B. Brown et al. 2020; Kaplan et al. 2020), which makes it challenging to decipher causal relationships between their inputs and outputs.

Robustness Traditional statistical prediction theory assumes that all data are sampled independantly of each other from the same distribution (Hastie, Tibshirani, and Friedman 2009). An important goal of interpreting a model is to judge and improve its robustness in settings where this assumption breaks down. For example, the distribution $P(X)$ of the inputs X might change resulting in covariate shift (X. Chen et al. 2016), or the joint distribution $P(X, Y)$ of the inputs and targets Y might change (Ben-Tal et al. 2013). A robust model is capable of handling these changes in the sampling distribution of the data.

Informativeness A common use of prediction systems is to give domain experts more information before they make a decision. The user in such a scenario uses the model to support their own decisions (B. Kim 2015b; Chouldechova 2017; Fiebrink 2011). For example, Caruana, Kangarloo, et al. (1999) show how to report cases considered to be most similar to the one under investigation by an artificial neural network trained to predict Pneumonia mortality. Case-based reasoning is emphasized during medical training and practice suggesting that case-based explanations could be even more useful for less mission-critical applications.

2.3.2 Interpretability methods

Broadly speaking, interpretability techniques should make the model's inner workings more transparent at different levels of its components. Some common approaches provide a useful picture of how a trained model might be working via mathemtical analysis of the collective behaviour of the trained weights, or by testing the sensitivity of its outputs to its inputs, or indirectly by providing relevant examples that might be similar to the sample in question. Others employ regularization to force the model to learn a more interpretable representation, or explicitly design model architectures to be more modular. We closely follow Fan, Xiong, and G. Wang (2020) and spot three major themes: the theoretical study of the behaviour of neural networks, techniques that seek to obtain more interpretable models by guiding their optimization or by designing architectures that are inherently interpretable, and finally those that analyze models that have already been trained. We borrow terminology from Fan, Xiong, and G. Wang (2020) to name these categories as Theoretical Analysis, Ad-Hoc analysis, and Post-Hoc analysis, respectively, presented as a taxonomy tree in Figure 2.4. We briefly cover Theoretical and Ad-Hoc Analsysis first, before focussing on Post-Hoc analysis.

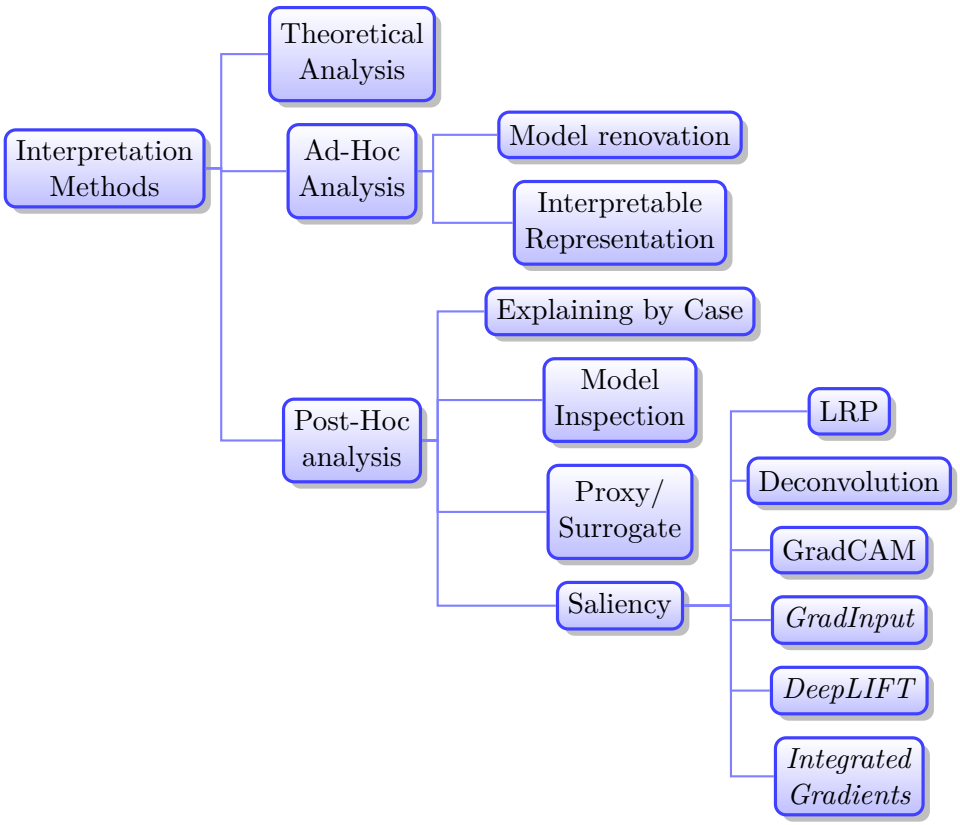

Figure 2.4: Taxonomy of interpretability techniques based on Fan, Xiong, and G. Wang (2020)

Theoretical Analysis Theoretical advances in different facets of a model's optimization trajectory, the shape of the loss function, or in estimations of the generalization ability of a model, can also aid interpretability. For example, Jacot, Gabriel, and Hongler (2018) investigate feedforward neural networks with widths tending to infinity (infinite-width limit) and show that NNs in this regime simplify to linear models with a kernel calleed the Neural Tangent Kernel (NTK), making it easier to study the trajectory of the loss function during gradient descent. This result was closely followed by a slew of papers investigating kernel gradient descent for wide, overparameterized neural networks (number of parameters $\gg$ number of training examples), such as Du et al. (2019) who show that gradient descent on overparameterized NNs perfectly optimizes a quadratic training loss or J. Lee et al. (2018), who show that infinitely wide and deep feedforward networks exactly correspond to Gaussian Processes (Rasmussen and C. K. I. Williams 2005) and devise a pipeline to compute

their covariance function, allowing the computation of exact prediction uncertainties. However, empirical results show that overparameterized neural networks still perform better than infinitely wide networks (Arora et al. 2019), so there is still substantial work remaining to fill in the gap between machine learning practice and theory.

Ad-hoc methods Ad-hoc methods construct models that are inherently interpretable by enforcing biases via the training process or explicitly in the model architecture. The goal of the former is to guide the optimization process to models that generate interpretable representations, while the latter focusses on designing modular architectures consisting of components with clearly defined roles that together make up a larger prediction system. For example, Chorowski and Zurada (2015) impose a non-negative constraint on the weights of a neural network and argue that it is more interpretable since neurons can never cancel each other, making it easier to trace the effects of input features to the final output. Other auxiliary objectives aim to guide the optimization process to solutions that use sparse, or disentangled internal representations. The former can make it easier to explicitly trace paths from a model's inputs to its outputs by zeroing redundant weights, while the latter makes it easier to factor the representation into parts that are uniquely affected by changes in the input (Locatello et al. 2019). Techniques under Model Renovation seek to obtain more explicit explanations of the model's decisions by extending or modifying model architecture, e.g. Chu et al. (2018), who propose using piece-wise linear functions as activation functions to help obtain closed-form solutions to a network's predictions, or works like H. Liu, Yin, and W. Y. Wang (2019) and Camburu et al. (2018), which generate descriptions by augmenting the model DNN architecture with a language model fine tuned together to provide natural language explanations of the model's decisions. The descriptions are simple and comprehensible but they rely on augmenting the training data with textual explanations to train the language model that generates them.

While ad-hoc methods change the model architecture or its optimization, Post-Hoc Analysis methods are generally employed on trained models:

2.3.2.1 Model Inspection

Model inspection methods directly analyze the weights or layer activations of the neural network to find patterns of activations that are specific to a certain class of inputs. For example, Zhou, Khosla, À. Lapedriza, et al. (2015) show that CNNs trained on ImageNet learn object detection filters without being given any object-level training objective, or Y. Wang et al. (2018), who develop a technique named Distillation Guided Routing (DGR) to find paths in the network (starting from the input nodes to the output nodes) that are critical to preserve the model's prediction performance, called Critical Data Routing Paths (CDRPs). Another example is the Concept Activation Vector (CAV, B. Kim et al. 2018), which is defined as the vector orthogonal to the hyperplane that linearly separates the activations of inputs containing a specific feature/concept (e.g. animals with stripes) from the activations of inputs without that concept (animals without stripes), at a particular neural network layer l. This allows the user to associate a concept at the input level (e.g. stripes) with the vectors at each layer of the network. Furthermore by slowly perturbing the activations at layer l in the direction of the CAV, users can measure the sensitivity of the model's predictions to the input concept associated to the CAV.

2.3.2.2 Proxy / Surrogate

These methods construct a simpler and more interpretable model from the larger deep network. The process of transferring the knowledge of one model (defined in terms of its predictive behaviour) to another is called model distillation and was first defined in G. Hinton, Vinyals, and Dean (2015), where a smaller DNN was trained to mimic the behaviour of an ensemble of DNNs. Today, model distillation refers to a whole host of techniques that construct proxies to approximate trained DNNs. The

transferred knowledge can be from a subset of the input space, for example a subset of training examples with similar features, or it can be more global in which case the goal is to translate the entire model.

Perhaps the most popular technique in this category is Local Interpretable Model-agnostic Explanations (LIME, Ribeiro, Singh, and Guestrin 2016). In order to explain a model's prediction for a specific sample, LIME uses inherently interpretable models such as decision trees or linear regression to approximate the outputs of the target model in a small neighborhood of the sample. The proxy model ingests an interpretable version of the data, for example a binary input to indicate the presence or absence of specific words. The main disadvantage of local proxy models is that they usually make simplistic assumptions to find inputs that are used to train the proxy and it is unclear how they perform in regions that do not satisfy that assumption. This may result in explanations that break down for complex unstructured datasets such as text and images, where finding similar inputs that differ only by the presence or absence of specific concepts can be difficult. Another drawback is the instability in their explanations to minor perturbations in the input: Alvarez-Melis and Jaakkola (2018) show that feature importance values generated by LIME for the features in a two dimensional dataset vary widely for a two layer NN, within a small neighborhood of the input space.

2.3.2.3 Explaining by Case

Case-based explanations employ case-based reasoning (Kolodner 1992) which involves associating the case being investigated with previously seen examples that are better understood. This can involve finding individual examples from the dataset or a composition of existing samples that are most similar to the sample being investigated. Two samples can be compared for similarity via a similarity metric applied on their corresponding hidden representations in the neural network. Samples can also be compared to "prototypes" (Bien and Tibshirani 2011), which are typically a composition

of samples that best summarize the characteristics of the samples in the entire dataset. Although not a post-hoc analysis method, a notable prototype based technique is the Prototypical Part Network (ProtoPNet, C. Chen et al. 2019), which dissects images into prototypical parts before classifying the image. On the other hand, Wallace, Feng, and Boyd-Graber 2018 use Deep K-nearest-neighbours (DKNN, Papernot and McDaniel 2018) to construct a model uncertainty metric called conformity-leave-one-out, which measures the drop in the proportion of training examples that are similar to the test example after it has been perturbed.

2.3.2.4 Saliency

This subcategory includes many popular methods that compute a form of feature importance score for each input and are typically visualized via a saliency map. Saliency maps are especially popular in interpreting predictions of image classification models. Typically, the importance (or saliency) of each feature is overlayed on the original image to present and compare the relative importance of each feature. Therefore, these techniques are also called Feature Attribution or Feature Importance techniques.

Feature attribution methods attempt to quantify the relevance of a feature to a model's prediction. A simple, direct way is to train different models for each combination of input while comparing the model performance for each combination of inputs. This involves training a different model for each combination and can be used to compute the marginal relevance of an input to the output. This is a form of a perturbation-based post-hoc interpretabilty method. Obviously, this is a very time consuming process, especially for models with a large number of parameters. Alternatively, gradient based methods compute the sensitivity of a trained model to its inputs. If a model is highly sensitive to changes in an input, then that input is likely to be a key component in the model's internal decision making process. A key benefit of such methods is that only a single model needs to be trained and interpreted.

An attribution, A, is a credit assignment on each individual input feature x_i of a

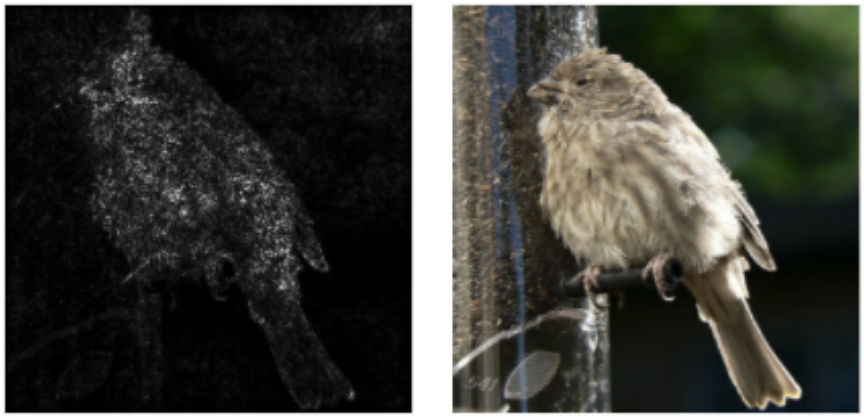

Figure 2.5: A heatmap obtained by Sturmfels, S. Lundberg, and S.-I. Lee 2020 to visualize pixel-wise importance scores generated by running integrated gradients on the Inception V4 (Szegedy, Ioffe, and Vanhoucke 2016) image classification network, trained on ImageNet (J. Deng et al. 2009)

single sample input x (e.g. each pixel in an image, or each character or word in text, or each SNP in GWAS) that measures how sensitive the model's prediction y is to changes in x_i. It is often presented as a heatmap overlaid on the original data (saliency map). The attribution heatmap visually indicates which aspects of a particular data example have the greatest influence on the model's prediction of the target y. For example, Figure 2.5 contrasts an attribution heatmap to the original image (in the validation set) for a popular image classification model named Inception V4 (Szegedy, Ioffe, and Vanhoucke 2016) trained on ImageNet (J. Deng et al. 2009).

More concretely, let x_n denote an M dimensional input feature vector of the n'th input to a feedforward neural network f with L hidden layers, with a corresponding output being $f(x_n)$. Let the reference (or baseline) input and output be denoted by $\overline{x}$ and $f(\overline{x})$ respectively. We define $\Delta x = x - \overline{x}$ and $\Delta f = f(x) - f(\overline{x})$ to be the difference with reference input and output values respectively. The attribution vector A_n is an $M \times 1$ dimensional sample-specific credit-assignment for each individual feature of the input x_n, providing a quantitative measure of the importance of each feature to the output $f(x_n)$. The importance of a feature can simply be the sensitivity of f to x at x_n:

$$A_n = \nabla f(x)|_{x_n},$$

but some methods compute a modified form of the equation above or define importance with respect to the reference inputs and outputs $\overline{x}$ and $f(\overline{x})$. We now describe and compare such sample-specific feature attribution methods. Note that for the remainder of this Section, we will carry forward the notations used for attribution scores A, the function being interpreted f, and the input x of a single sample n with dimensions $M \times 1$ but drop the sample indexing subscript n for convenience.

GradInput, Input $\times$ Gradients (IXG) (Shrikumar, Greenside, Shcherbina, et al. 2016) Given the deep network f and input x, the attributions are calculated as the element-wise product of x and the gradient of f calculated at x:

$$A = x \odot \nabla f(x) \tag{2.10}$$

DeepLIFT (Shrikumar, Greenside, and Kundaje 2017) DeepLIFT decomposes the difference in the output prediction of a neural network and a reference output by propagating activation differences layer by layer via a modified backpropagation rule. The attribution score of each feature indicates the degree to which that feature helped change the model's output from a reference output to the observed output. It assigns a score $C_{\Delta x_m \Delta f}$ for the input x s.t.

$$\Delta f = \sum_{m=1}^{M} C_{\Delta x_m \Delta f}, \tag{2.11}$$

where $C_{\Delta x_m \Delta f}$ can be thought of as a weight assigned to the input x_m in proportion to its contribution to the difference Δf. Shrikumar, Greenside, and Kundaje (2017) provide multiple rules like the Linear, Rescale, or Reveal-Cancel rules to compute $C_{\Delta x_m \Delta f}$. We follow Ancona et al. (2018) and only cover the Rescale rule due to its

connections with ϵ-LRP, which we cover later. We use a_j^l to denote the output value of the j'th input neuron of layer l, and w_{ij}^l to be the parameter that weighs the output value of the i'th neuron in layer l into the j'th neuron in layer $l+1$. This weighted activation value, $w_{ij}^l a_j^l$, will be denoted by z_{ij}^l. The contribution score of the neuron i in layer l, denoted by $C_{\Delta a_i^l \Delta f}$, is calculated using Equation 2.12 below:

$$C_{\Delta a_i^l \Delta f} = \begin{cases} f(x) - f(\overline{x}), & \text{if } l = L \\ \sum_k \frac{z_{ik}^l - \overline{z}_{ik}^l}{\sum_{i'} z_{i'k}^l - \sum_{i'} \overline{z}_{i'k}^l} C_{\Delta a_k^{l+1} \Delta f}, & \text{otherwise} \end{cases} \tag{2.12}$$

and finally, A_{x_m} is equal to the contribution score $C_{\Delta x_m \Delta f}$, which can be computed using Equation 2.12 by treating x_m as a_m^0, i.e. $C_{\Delta x_m \Delta f} = C_{\Delta a_m^0 \Delta f}$.

LRP (Bach et al. 2015) Layer Relevance Propogation (LRP) propagates the relevance of a neuron's output starting from the output layer in which the relevance is set to the output of the target neuron. Let r_i^l denote the relevance of the neuron i in layer l of the network. Then LRP starts by setting r^L to be the output of the neuron at the output layer L. r^L is redistributed to the previous layers via the propagation rule defined in Equation 2.13, where z_{ji}^l is the weighted activation of the neuron i in layer l onto the neuron j in the next layer $l+1$, b_j^{l+1} is the bias term of neuron j, and ϵ is a small scalar added to the denominator to avoid numerical instabilities.

$$r_i^l = \begin{cases} f(x), & \text{if } l = L \\ \sum_k \frac{z_{ik}^l}{\sum_{i'}(z_{i'k}^l + b_j^{l+1}) + \epsilon \cdot sign(\sum_{i'}(z_{i'k}^l + b_j^{l+1}))} r_k^{l+1}, & \text{otherwise,} \end{cases} \tag{2.13}$$

with the attribution score A_m of input feature x_m being the relevance value r_m^0, computed using Equation 2.13. Ancona et al. (2018) show that if $f(0) = 0$, with no additive biases ($b_j^l \ \forall j$ and l), and $\overline{x} = 0$, then the contribution and relevance scores produced using Equations 2.12 and 2.13 respectively, are the same. Further-

more, Shrikumar, Greenside, Shcherbina, et al. 2016 show that absent the ϵ term in Equation 2.13, LRP and GradInput produce the same attribution scores.

Integrated Gradients (Sundararajan, Taly, and Yan 2017) This technique computes the path integral of the partial derivative of the output with respect to an input feature along the line segment joining a reference value to a sample in the dataset. Given the deep network f, input x, and baseline input $\overline{x}$, the attribution score A_m of feature x_m is:

$$A_m = (x_m - \overline{x}_m) \int_{\alpha=0}^{1} \frac{\partial f(\overline{x} + \alpha(x - \overline{x}))}{\partial x_m} \, d\alpha, \tag{2.14}$$

where α is associated with the path from x to $\overline{x}$, and is smoothly distributed in range $[0, 1]$. The R.H.S of Equation 2.14 accumulates the sensitivities of f to changes in feature x_m as the input is varied along the straight line connecting $\overline{x}$ and x. Therefore, it is the line integral of the gradient of the model w.r.t the input features, along the straight line path between $\overline{x}$ and x. Intuitively, x_m should have increasing relevance if gradients are large between a baseline point $\overline{x}$ and x along the m'th dimension.

CAM and GradCAM (Zhou, Khosla, A. Lapedriza, et al. 2016 and Selvaraju et al. 2017) A class activation map (CAM) indicates the regions in an input image used by a CNN for classification. Zhou, Khosla, A. Lapedriza, et al. 2016 uses a global average pooling layer on the activation map of the final convolutional layer to generate attributions for a specific class. GradCAM (Selvaraju et al. 2017) extends the CAM method by using the gradients of the network output with respect to the last convolutional layer to achieve the class activation map. It should be noted that these methods are specific to CNNs.

Deconvolution (Zeiler and Fergus 2014) This is a feature visualization technique for CNNs that reverses the convolution operations via separate deconvolution layers.

The technique reconstructs patterns at each convolution layer that caused the highest activation in the model.

In this Chapter, we briefly described the structure and role of DNA in human biology and disease, and introduced genetic variants called SNPs, which could be useful for the localization of genetic causes of various traits and diseases. Subsequently, we introduced the Genome-Wide Association study and covered the basics of the statistical analysis pipeline of a conventional GWAS. We then introduced Artificial Neural Networks, a flexible and popular machine learning modelling framework that we can leverage to learn to spot and utilize complex patterns from large amounts of data. We introduced gradient descent as the optimization framework that is used to train most neural networks, as well as Regularization with its most popular examples. We discussed the disadvantages of the black box nature of Deep Learning, particularly the difficulty of comprehending the patterns used by deep learing models to make predictions. This led us to explore the topics of Interpretability and Explainability in Artificial Intelligence. After describing the relatively ambiguous nature of the topics, we listed some practical goals of interpretability research. Finally, we proceeded to summarize an extensive taxonomy of interpretability techniques for deep learning models, mostly focussing on feature importance techniques that can be applied on a trained model, since they do not require training multiple models and can be used to compare the relative importance of a feature to the model's predictions. We now have a powerful technique at hand to model complex relationships between data as well as the tools to interpret the relationships that are learnt. We attempt to do so on genetic data in subsequent chapters.

3

Phenotype prediction and interpretation using Deep Neural Networks

Conventional GWAS fail to model interactions between SNPs, nor do they consider non-linear signals between genetic variants to model the target trait. Indeed, GWAS are limited to the identification of genetic variants with strong marginal effects, possibly leaving out a large number of genetic effects governed by SNP-SNP interactions or other non-linear effects. Since deep networks are known to be able to model arbitrarily complicated non-linear functions of their inputs (Goodfellow, Bengio, and Courville 2016), they are well suited to model more complex interactions between SNPs. However, we still need to decipher the maze of a neural network's weights to find inputs that are helping the network make its predictions. Therefore, in this chapter, we present a pipeline to conduct a GWAS using Deep Neural Networks and validate it on simulated data to show that it is effective at distinguishing known causal SNPs.

3.1 Related Work

While DNNs have been previously applied to analyze GWAS datasets (Libbrecht and Noble 2015; Ching et al. 2018), most of them have been applied to risk prediction, wherein the model predicts a risk score based on the genotype. For example, Montaez

et al. (2018) use deep models to improve classification of polygenic obesity, but the model uses SNPs from loci that were obtained from a prior association analysis. In contrast, Waldmann (2018) use 1 layer NNs in order to be able to average the model's weights and directly calculate the regression coefficient for each SNP. Using 1 layer NNs is equivalent to conducting the GWAS conducting logistic regression and fails to model complex non-linear interactions between the SNPs.

Romagnoni et al. (2019) present a thorough comparison of approaches relying on three different classes of models: logistic regression, gradient boosting decision trees (GBT; Hastie, Tibshirani, and Friedman 2009), and deep neural networks, in terms of their prediction ability and SNP association identification on a Crohn's Disease dataset. In order to interpret DNNs, they used permutation feature importance, a model-agnostic approach that involves breaking the relationship between a single feature and the target, and then measuring the decrease in the model's performance. This is especially useful for opaque models like DNNs but in modern GWAS datasets, this will involve separately permuting millions of SNPs, making this approach computationally expensive. Tran and Blei (2018) conduct maximum-likelihood-ratio tests for every input SNP by comparing the predictive performance of a neural network trained on all input SNPs against a very similar neural network trained on all but the target SNP. However, this approach does not scale to datasets with a large number of SNPs since the number of neural network models trained scales linearly in the number of SNPs, which could be in order of hundreds of thousands to millions.

Evidently, the key challenge is the absence of methods that can determine whether features used by a model are novel biomarkers or spurious correlations.

3.2 Filtering and ordering causal SNPs

Let's consider a typical, probabilistic setup used to model genotype populations. Suppose that there are N samples, each with M measured SNPs. As described in Section 2.1.2, SNP i in sample j is denoted by $x_{ij} \in 0, 1, 2 \ \forall \ i \in 1, 2, \ldots, M$ and $j \in 1, 2, \ldots, N$. Collectively, the SNPs for all samples are denoted by $X_{N \times M}$, which is called the genotype matrix of the sample population. As mentioned in Section 2.1.2, the genotype of every population is affected by the different ancestries of the samples in a GWAS, reflected by attributes such as their race and ethnic origin. This results in a complex hidden structure in the base frequencies of the alleles of each SNP. Since this structure is unobserved, we shall denote it by the latent variable z_j for sample j. Let the base allele frequency for SNP i, denoted by π_i, be a function of $z_j \ \forall \ j \in 1 \ldots N$, that is SNP i has different base allele frequencies $\pi_i \left(z_j \right)$ corresponding to each population sample j. By clustering the values of z, we will model a sample population which stratifies into different population substructures, each with their own genetic ancestries. Additionally, we model admixture by varying the proportion to which an individual sample j belongs to each cluster. We collect the frequencies of different SNPs across the sample population into the matrix π, with π_{ij} being the individual-specific allele frequency of SNP i conditioned on the ancestry of sample j.

We will denote the quantitative trait by $y_j \ \forall \ j \in 1, 2, \ldots, N$, which we will collectively denote by $Y_{N \times 1}$. As mentioned in Section 2.1.2, a common assumption of most GWAS is that the trait under study is a linear function of the genotype and various other covariates like ancestry and sex etc (Clarke et al. 2011). Thus, we will assume that this trait is a linear function of the genotype matrix X with a genetic effect vector $\beta_{M \times 1}$, and non-genetic factors that we will collectively denote by $\lambda_{N \times 1}$ (Equation 3.1).

$$Y = \beta X + \lambda + \epsilon, \tag{3.1}$$

where ϵ is random noise. In order for population structure to be a confounder, we allow λ and ϵ to be a function of z.

In order for a deep learning model to be a useful prediction tool in GWAS, we need to design a pipeline that accurately predicts Y from X, but more importantly, accurately estimates β. But as we reviewed in Section 2.3.2, the field of Explainable AI currently lacks a unified and comprehensive picture of the importance of a NN's features to its predictions. Nevertheless, the past few years has seen the development of different feature importance techniques (Section 2.3.2) that generate importance scores for a model's features to try to quantify their importance to a model's predictions.

These scores can be used to rank SNPs and localize regions of high importance to the model. Moreover, as mentioned in 2.1.2, GWAS summary scores are used to narrow the loci that harbour SNPs with non-zero effects on the target phenotype, which suggests that it might be possible to relax the initial goal of accurately estimating β to simply ranking genomic regions by the effect size of their constituent SNPs.

More precisely, what is the correlation between the rank of SNPs, ordered by their true effect size (β), and the feature importances generated by feature importance techniques? If the rank correlation is consistently high, then a GWAS pipeline based on prediction using deep networks can be used to localize SNPs that are associated with the target phenotype by ranking them according to their feature importances.

3.2.1 Algorithm

In this thesis, we present Algorithm 1, a pipeline for GWAS that uses deep learning models for prediction and ranks input SNPs by a summary of importance scores that are generated using some of the feature importance techniques that we covered in

Section 2.3. We now describe the algorithm, including our choice of feature importance techniques, in more detail:

Consider a dataset $\mathcal{D}$ of N samples (individuals) with M SNPs each ($X_{N\times M}$), and their corresponding covariates $Z_{N\times 1}$ and target phenotypes $Y_{N\times 1}$, that has been split into three separate subsets $\mathcal{D}_{train}, \mathcal{D}_{valid}, \mathcal{D}_{test} : \mathcal{D}_a \cap \mathcal{D}_b = \emptyset \; \forall \; \mathcal{D}_a, \; \mathcal{D}_b \in \{\mathcal{D}_{train}, \mathcal{D}_{valid}, \mathcal{D}_{test}\}$ and $\mathcal{D}_{train} \cup \mathcal{D}_{valid} \cup \mathcal{D}_{test} = \mathcal{D}$. We first optimize multi-layer feedforward neural networks on $\mathcal{D}_{train}$.

Algorithm 1: Our GWAS pipeline that uses feedfoward neural networks to predict a single trait and feature importance methods to attribute a global importance score to each SNP

Input: A dataset $\mathcal{D} = (X, Y, Z)$ of N samples with M SNPs each, consisting of the genotype matrix X, phenotypes Y, and covariates Z. Feedforward Neural Network $f_\theta : \Re^M \times \Re^K \rightarrow \Re$, a feature importance method `Attribute()`, objective function $J = J^{cat}$ if Y is categorical and J^{cont} if it is continuous, and a validation metric $h : \Re^N \times \Re^N \rightarrow \Re$

Result: Optimized neural network model $f_{\hat{\theta}}$ and an attribution vector $\hat{R}_{M\times 1} = (r_j)$ s.t. $r_j \geq 0 \; \forall \; j$

begin
  Split the dataset tuple (X, Y, Z) into training $D_{train} = (X_{train}, Y_{train}, Z_{train})$, valid $D_{valid} = (X_{valid}, Y_{valid}, Z_{valid})$, and test $D_{test} = (X_{test}, Y_{test}, Z_{test})$
  for $t \leftarrow 1$ **to** T **do**
    Optimize f using gradient descent on J, calculated on the training dataset D_{train}, to yield f_{θ^t}
    if $h(Y_{valid}, f_{\theta^t}(X_{valid}, Z_{valid})) \leq h(Y_{valid}, f_{\theta^{t-1}}(X_{valid}, Z_{valid}))$ **then**
      $f_{\hat{\theta}} \leftarrow f_{\theta^{t-1}}$
      break
    else
      $f_{\hat{\theta}} \leftarrow f_{\theta^t}$
    end
  end
  R $\leftarrow$ `Attribute`$(\mathrm{f}_{\hat{\theta}}, (X_{test}, Z_{test}), [\ldots])$
  $\hat{R} \leftarrow mean(|R|)$
  return $f_{\hat{\theta}}$, $\hat{R}$
end

Step 1: Train a feedforward model We train a feedforward model to output a prediction $\hat{y}_n$ given input SNPs $X_i \in \mathcal{D}_{train}$, using typical loss functions like the Mean Squared Error (Equation 2.6) for regression or Binary Cross Entropy (Equation 2.5) for classification. This is repeated for all elements of $\mathcal{D}_{train}$ and the resulting model is then evaluated on the validation data $\mathcal{D}_{valid}$ using the metric h. For classification, h can be the mean conditional log likelihood of the phenotypes in the validation set given their corresponding genotypes, under the trained model, which is equivalent to $\mathcal{L}^{cat}$ in Equation 3.2. For regression, a popular choice for h is Explained Variance (EV= $1 - \frac{Var(y-\hat{y})}{Var(y)}$), which is the proportion of the variance in the target phenotypes that is explained by the variance in the residuals. We halt training as soon as this metric starts decreasing between consecutive rounds of the subroutine Train in Algorithm 1, or if we've exhausted our computing budget.

Since we expect a sparse subset of the input SNPs to be causal, we can force the models to pick a handful of SNPs by adding an L1-penalty (Equation 2.7) to the first layer weights. This modifies the final objective function of training a model for categorical traits to J^{cat} in Equation 3.2 and quantitative traits to J^{cont} in Equation 3.4. We denote the first layer weights using θ_1 for both objectives. The final optimized model is denoted by $f_{\hat{\theta}}$.

$$\mathcal{L}^{cat}(X, y; \theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)), \tag{3.2}$$

$$J^{cat}(X, y; \theta) = \mathcal{L}^{cat}(X, y) + \lambda|\theta_1|, \tag{3.3}$$

&

$$\mathcal{L}^{cont}(X, y; \theta) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.4}$$

$$J^{cont}(X, y; \theta) = \mathcal{L}^{cont}(X, y) + \lambda|\theta_1| \tag{3.5}$$

Step 2: Attribution We now compute importance scores for each SNP j in each sample i contained in the test dataset $\mathcal{D}_{test}$ using feature importance techniques described in Section 2.3.2. As mentioned previously, we cannot use CAM, GradCAM, and Deconvolution since our model does not contain any convolution layers, and although LRP is applicable, Shrikumar, Greenside, and Kundaje (2017) show that it is equivalent to GradInput without the ϵ term in Equation 2.13, while being unstable relative to the other techniques. This means that we are left with a choice between 3 techniques, namely GradInput (IXG), DeepLIFT (DL), and Integrated Gradients (IG).

The Attribution subroutine in Algorithm 1 describes the general pipeline of the procedure of generating an aggregated importance score $R = r_j$ for each SNP $j \in 1 \dots M$. Since our three candidate attribution techniques (GradInput, DeepLIFT, and Integrated Gradients) all require slightly different hyperparameters, we describe them here so that they may be appropriately used in Algorithm 2. While GradInput simply needs a genotype sample x_i used to make a prediction, DeepLIFT and Integrated Gradients also require reference inputs $\overline{x}$ and $\overline{z}$ to compute $\overline{y}$. In addition, Integrated Gradients needs t, the number of steps taken to reach the sample x_i from $\overline{x}$ along the line segment connecting those two points.

In order to improve robustness of the final importance scores of all SNPs, we recommend averaging R over multiple random seeds, essentially running Algorithm 1 multiple times with a different random initialization of the model weights and dataset splits.

Our pipeline is still incomplete since we haven't picked the feature importance method. Moreover, we need to specify the baseline inputs for both DeepLIFT and Integrated Gradients ($\overline{x}$ and $\overline{z}$) and the parameter t for Integrated Gradients, in Algorithm 2. They are specified as part of the experiments used to evaluate our pipeline. We first list and describe the criteria used to evaluate it.

Algorithm 2: Computing attributions for every SNP in a set of inputs. The square brackets on Line 4 indicate optional arguments that are to be included if DeepLIFT or Integrated Gradients are used.

Input: A dataset $\mathcal{D} = (X, Y, Z)$ of N samples with M SNPs each, consisting of the genotype matrix $X_{N\times M}$, phenotypes $Y_{N\times 1}$, and covariates $Z_{N\times K}$, where K is the dimension of the covariates. Feedforward Neural Network $f_\theta : \Re^M \times \Re^K \to \Re$, a reference input genotype $\overline{x}$ and reference input covariates $\overline{z}$, and an integer parameter $t \in \mathbb{Z} : t \geq 5$

Result: A matrix $A_{N\times(M+K)}$

1 `Attribute(`f, (X, Z), $(\overline{x}, \overline{z})$, t`):`
2 $n \leftarrow 0$, $A \leftarrow \overrightarrow{0}_{N\times(M+K)}$
3 **for** *each row* x_i *of* X **do**
4 $A_i \leftarrow attr(f_{\hat{\theta}}, (x_i, z_i), [(\overline{x}, \overline{z}), t])$
5 **end**
6 **return** A

3.2.2 Evaluation

As mentioned before, the pipeline needs to be accurate at identifying the causal SNPs if the model is using them to make accurate predictions. If we known how many there are, we can simply count the number of causal SNPs present at the top after ranking the SNPs. For example, we could compute the number of causal SNPs that are included in a set of SNPs ranked at the top when ordered by their summarized importance scores (i.e. top-K accuracy for each causal SNP where K is the number of causal SNPs). However, this ignores how the model ranks causal SNPs between each other as well as the ranks of any causal SNPs that are outside the top K SNPs. This leads us to consider rank correlation metrics like the Kendall Tau (Kendall 1938). But the Kendall Tau does not give more weight to mistakes in ranking more important items (causal SNPs with larger β). In order to compute rank correlations that take into account β, we also report the Weighted Kendall Tau rank correlation (Vigna 2015), which is a generalization of the Kendall Tau that is designed to deal with ties and weigh differences in rankings according to a weight for each rank. This allows us to increase the penalty of incorrectly ranking causal SNPs with larger effects, giving us a more precise measure of a pipeline's ranking accuracy. Thus, we report three different

metrics for each attribution method: the Top-K accuracy (Top-K) with K being the number of causal SNPs, the Kendall-Tau rank correlation (τ) after ranking the causal SNPs by their causal coefficient β and the attribution scores, and the corresponding Weighted Kendall Tau (τ_β) to penalize mistakes in ranking causal SNPs with higher values of β.

With a set of accuracy metrics in hand, we now describe the qualities that we believe an attribution method should possess to make it ideal for use in Algorithm 1, as well as how we are going to measure them:

Accuracy The pipeline needs to be accurate at identifying the causal SNPs if the model is using the causal SNPs to make accurate predictions. If a model is right for the right reasons (Doshi-Velez and B. Kim 2017) then the pipeline needs to be able to show that. This is a necessary condition for the pipeline to be useful at all.

Consistency The accuracy of the pipeline should be consistently high for accurate models. We measure this by measuring the mean and standard deviation of the pipeline's accuracy over the top 10% performing models.

Fidelity Robnik-Šikonja and Bohanec (2018) define Fidelity as how well explanations reflect the behaviour of the model. If the model is right for the right reasons then the accuracy of the pipeline should positively correlate with the model's peformance. We capture fidelity by measuring the correlation between model performance and each attribution accuracy metric, for all trained models.

We now proceed to describe the generation procedure for the datasets used to train and evaluate our pipeline, and subsequently the experiments used to conduct the evaluation.

3.3 Experiments

We'd like to run experiments in order to evaluate how accurate our GWAS pipeline is at identifying and ranking known causal SNPs (Accuracy), as well how to investigate how the pipeline's accuracy is affected by the predictive performance of the underlying model. The latter goal requires assessment of the pipeline's attribution accuracy when the underlying model's prediction performance is high (Consistency) as well as when it is low (Fidelity). In order to judge any of these qualities of the pipeline, as well as to compare the different attribution methods, we'll need to know the ground-truth causes. Thus, it makes sense to perform experiments on simulated data. We first describe the procedure used to generate the datasets used for our experiments, before moving on to the details of the experiments with the pipeline. We generate 4 datasets of 10,000 individuals with 10,000 SNPs, including 10 causal SNPs, for both a categorical and a continuous trait.

3.3.1 Simulation

As mentioned earlier, in order to be able to assess our pipeline quantitatively, we need to be able to simulate the genotype matrix, the population level confounders, and most importantly the ground-truth causal effects of the SNPs. Esssentially, In this section, we describe the procedures used to generate the aforementioned data and design the classification and regression tasks.

We follow Hao, Song, and Storey (2016) to simulate genotypes with a population structure that is reflected by a sample's spatial position in the population. The genotype matrix $X_{N\times M}$ is simulated by sampling from a Binomial of a matrix of allele frequencies π.

$$x_{ij} \sim \text{Binomial}(2, \pi_{ij}), \text{where } \pi \text{ is a matrix of allele frequencies} \tag{3.6}$$

As depicted in Equation 3.7, π is constructed from the product of matrices $\Gamma_{M\times P}$ and $S_{P\times N}$, where the former maps the structure of the sample population to the allele frequencies of each SNP, the latter represents the position of each sample in the population (the population structure), and P determines the number of subpopulations or clusters that the samples can be clustered into. We set $P = 3$, with the last row fixed to 1. This means that the first 2 rows of S will determine the position of each sample on a unit square. The constants 0.9 and 0.05 (Equation 3.7) were picked so that $\pi_{ij} \in 0.05 \ldots 0.95$.

$$\begin{aligned} \pi &= \Gamma S, \\ \Gamma_{mk} &\sim 0.9 \times \text{Uniform}(0, 0.5), k \in 1, 2, \\ \Gamma_{m3} &= 0.05, \\ S_{kn} &\sim \text{Beta}(a, a), k \in 1, 2, \\ S_{3n} &= 1 \end{aligned} \tag{3.7}$$

As shown in Equation 3.8, binary traits are sampled from Bernoulli distributions with parameters that are a linear function of a random effect vector, the SNPs, and the spatial position of each sample.

$$\begin{aligned} \epsilon_n &\sim Normal(0, \sigma_n^2), \\ y_n &\sim Bernoulli(\sum_{m=1}^{M} \beta_m x_{mn} + \lambda_n + \epsilon_n), \end{aligned} \tag{3.8}$$

with β_m defined as below:

$$\beta_m \sim \begin{cases} \mathcal{N}(0, 0.5), & \text{if } m \in m^{causal} \\ 0, & \text{if } m \notin m^{causal}, \end{cases} \tag{3.9}$$

where m^{causal} is the list of causal indices. For both prediction tasks, we set $m^{causal} = 2000, 2200, 2400, \ldots, 3800$, which evenly spreads the causal SNPs in the middle of the

first half of the genotype and sets the effect parameter for non-causal SNPs to zero. We simulate quantitative (or continuous) traits using the linear logit function from Equation 3.8, as described in Equation 3.10 below:

$$\begin{aligned} \epsilon_n &\sim Normal(0, \sigma_n^2), \\ y_n &= \sum_{m=1}^{M} \beta_m^{cont} x_{mn} + \lambda_n + \epsilon_n \end{aligned} \tag{3.10}$$

As per Hao, Song, and Storey (2016), we simulate λ_n and σ_n as follows:

1. Assign each sample j to a partition obtained by running K-mean clustering on the columns of the sample frequency matrix S, with $K = 3$. Let the partitions be denoted by S_1, S_2, and S_3
2. $\lambda_j = k$ for all $j \in S_k$
3. Draw $\gamma_1^2, \gamma_2^2, \gamma_3^2 \sim$ InverseGamma$(3, 1)$ and set $\sigma_j^2 = \gamma_k^2$, $\forall j \in S_k$.

The confounding of the causal effect is controlled by the $\mathcal{B}$ parameter a in Equation 3.7, which controls the sparsity of the sample population. If $a = 1$, the samples are placed uniformly within a unit square. As a gets smaller, the samples separate into clusters more easily. In order to address the confounding, we compute and provide the top 3 principal components of the genotype matrix as input to the classifier.

3.3.2 Training, Attribution, and Ranking

Training and Model Selection We run experiments over the space of hyper parameters listed in Table 3.1, with L1 regularization hyperparameter $\lambda \in \{0.01, 0.1, 1, 10\}$. This results in 9 different model architecures, with 4 different configurations of the L1-penalty, each of which was trained on each spatial configuration of the datasets, over 5 seeds, for both classification and regression tasks. We train each model on

50% of the samples, early stop on 25% of the samples, and validate the model architectures using the remaining 25%. The different dataset settings were characterized by the values of the spatial parameter $a \in \{0.01, 0.1, 0.5, 1\}$. All the classification datasets have an average case-control ratio of 0.3.

As mentioned in Step 1 of Section 3.2.1, we force the models to pick a handful of SNPs by adding an L1-penalty to the first layer weights. We use J^{cat} as the objective function for the classification task, and J^{cont} for the regression task. The metric h used to select models is the validation set likelihood $\Pi_i P(y_i|x_i;\hat{\theta})$ for classification, and validation set EV for regression. For each combination of spatial configuration a, architecture, and λ, we select models with the highest average value of h.

Table 3.1: Width of the 2-layer feedforward NNs trained on the simulated single-task data

		Number of hidden units in layer
Layer #	First	{32,64,128}
	Second	{128,256,512}

Attribution We then apply the attribution method to the predictions of the trained model on the test set to compute attribution scores for each input SNP of each test sample. These scores are summarized by averaging their absolute value. For DeepLift and Integrated Gradients, the reference input is set to be the mean genotype of all samples. We min-max scale the attribution scores to [0, 1], in order to be able to compare and combine scores for the same input across experiments of a spatial configuration.

Ranking As mentioned in Subsection 3.2.2, we compute 3 different metrics; the number of causal SNPs in the top K highest ranked SNPs (Top-K), with K being the number of causal SNPs, the kendall tau rank correlation for the causal SNPs (τ), and the weighted kendall tau rank correlation for the rankings of the causal SNPs (τ_β). Since we know that we have simulated 10 causal SNPs, we select 10 SNPs with

the highest attribution score and count the number of causal SNAPS in that set to compute Top-K. The rank correlations are computed between a ranking computed on the causal coefficient β and a ranking computed on the summarized attribution scores.

The Weighted Kendall Tau rank correlation metric gives a weight to each rank and combined the weights for every pair of ranks being compared. For example, if the weight of ranks i and j ($i < j$) is w_i and w_j respectively, then the mistake of ranking j above i is weighted by combining w_i and w_j. Thus, we need to decide what w_r should be for each rank r, as well as how different weights are to be combined. We follow suggestions made by Vigna (2015), and set the weight for rank r to $\frac{1}{r+1}$ and combine rank weights by adding.

3.4 Results

We report negative control results as the attribution and model accuracy for untrained models on the classification dataset, in Table 7.1 of the Appendix. As expected, the untrained models on the classification dataset have zero model accuracy and the corresponding attribution tasks report zero causal SNPs.

We now report accuracy and consistency results for classification and regression separately, in that order. Finally, we report results for the correlation between model and attribution performance, together for both tasks. Higher values are better for all reported attribution and model performance metrics.

3.4.1 Accuracy and Consistency

Table 3.2 lists the target likelihood, PR AUC, and ROC AUC, on the test set for the best performing model on each of the four datasets. The PR and ROC AUCs are above 0.85 for all models while the likelihood is at least 75%.

Table 3.2: Model performance and architecture of the best model on the classification tasks

a	Architecture	λ	$\Pi_i P(y_i\|x_i;\hat{\theta})$	PR AUC	ROC AUC
0.01	[64,512]	0.1	0.77	0.87	0.94
0.10	[64,512]	0.1	0.79	0.88	0.94
0.50	[64,512]	0.1	0.84	0.91	0.97
1.00	[64,512]	0.1	0.80	0.90	0.95

Table 3.3 lists the attribution metrics discussed in Section 3.2.2 for the models in Table 3.2. All models identify at least half of the causal SNPs (Top-$K \geq 5$) while consistently ranking most causal SNPs correctly ($\tau \geq 0.78$ for at least one attribution method per a). GradInput (IXG) clearly performs worse in terms of ranking the SNPs and if the causal coefficient β for each causal SNP is taken into account (τ_β), then its performance is even worse. This implies that although GradInput is able to filter in as many causal SNPs as DeepLIFT (DL) and Integrated Gradients (IG), it is unable to rank them correctly.

Table 3.3: Model performance of the best model and corresponding attribution accuracy on the classification tasks

		Top-K			τ			τ_β		
a	$\Pi_i P(y_i\|x_i;\hat{\theta})$	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.77	5.33	5.33	5.0	0.87	0.84	0.67	0.90	0.89	0.73
0.10	0.79	6.00	6.00	6.0	0.90	0.90	0.75	0.96	0.96	0.67
0.50	0.84	9.00	9.00	9.0	0.90	0.90	0.70	0.93	0.92	0.70
1.00	0.80	6.33	6.00	6.0	0.76	0.78	0.66	0.79	0.81	0.70

Tables 3.4 and 7.2 (in Appendix 7.1) list the mean and standard deviation in the test set likelihood and attribution accuracy computed over the top 10% models. We notice that the mean attribution performance is lower than the best model, suggesting that the pipeline's accuracy improves for models with better predictive performance. The deviation in Top-K attribution accuracy is higher as a percentage of their corresponding mean in Table 3.4, which might be because the metric relies on a hard cutoff, resulting in a less smooth distribution.

Table 3.4: Model performance and attribution accuracy, averaged over the top 10% models on the classification tasks.

		Top-K			τ			τ_β		
a	$\Pi_i P(y_i\|x_i;\hat{\theta})$	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.69	4.77	4.77	4.60	0.70	0.70	0.59	0.81	0.82	0.69
0.10	0.69	5.30	5.33	5.23	0.79	0.80	0.70	0.87	0.87	0.67
0.50	0.74	7.67	7.67	7.50	0.83	0.83	0.68	0.87	0.87	0.69
1.00	0.69	5.57	5.53	5.33	0.68	0.69	0.62	0.71	0.72	0.67

Table 3.5 shows the explained variance reported by the best model on each of the regression tasks, along with their attribution accuracy metrics. We observe that Top-K accuracy is much better than during classification while the ranking metrics Top-K and τ_β are lower across the board. This implies that more causal SNPs were filtered out but they were not correctly ranked. A potential reason for this could be that most β coefficients for the causal SNPs in the categorical datasets were mostly negative, resulting in the number of cases ($y = 1$) being at most a third of the number of controls ($y = 0$) in the dataset. As mentioned in Section 3.3.1, this was by design to improve resemblance with real-world GWAS datasets. In order to improve the predictive performance of the model on each classification dataset, the cases were oversampled over the controls, which would have forced the models to rely on the SNPs with positive causal coefficients potentially at the cost of a fraction of the causal SNPs with negative coefficients.

Table 3.5: Model performance of the best models and corresponding attribution accuracy on the regression tasks

		Top-K			τ			τ_β		
a	EV	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.79	9.0	9.0	9.0	0.87	0.85	0.61	0.84	0.82	0.63
0.10	0.75	9.0	9.0	9.0	0.81	0.84	0.76	0.79	0.81	0.74
0.50	0.69	7.0	7.0	7.0	0.44	0.44	0.53	0.49	0.50	0.55
1.00	0.68	9.0	9.0	10.0	0.81	0.78	0.67	0.75	0.72	0.63

Table 3.6 lists the mean EV and attribution metrics computed over the top 10%

models for the regression task, and Table 7.3 (in Appenfix 7.1) lists the corresponding standard deviation. We notice that the Top-K accuracy is very stable since there is almost no deviation across the models and almost all values are identical to Table 3.5. Likewise, the ranking metrics Top-K and τ_β are also very similar to the best model. This suggests that model performance is not correlated with the ranks of the causal SNPs. We investigate this while testing for Fidelity in the next Section.

Table 3.6: Model performance and attribution accuracy, averaged over the top 10% models on the regression tasks.

		Top-K			τ			τ_β		
a	EV	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.75	9.0	9.0	9.00	0.78	0.77	0.58	0.76	0.76	0.60
0.10	0.70	9.0	9.0	9.00	0.81	0.82	0.73	0.79	0.80	0.66
0.50	0.64	7.0	7.0	7.00	0.52	0.52	0.51	0.54	0.55	0.51
1.00	0.64	9.0	9.0	9.93	0.82	0.81	0.63	0.77	0.77	0.60

3.4.2 Fidelity

As mentioned in Section 3.3, if an attribution method is true to a model's performance, then its attribution performance should directly correlate with its prediction performance. In Table 3.7, we report the Pearson correlation between the performance metrics of all trained models and their corresponding attribution accuracy metrics, averaged over all datasets, for each attribution method. We find that in the classification task, correlation for DeepLIFT and Integrated Gradients is consistent for all datasets, whereas for GradInput, the correlation between model performance and the ranking metrics is much lower with larger standard deviation.

In contrast, the correlation between EV and any attribution accuracy metric is much lower across the board for the regression task. In fact, there is almost zero correlation between model performance and the ranking metrics for Integrated Gradients and DeepLIFT, and slightly negative correlation for GradInput. This partially

validates the low mean rank correlation in Table 3.6 and low standard deviation in Table 7.3.

Table 3.7: Mean Spearman correlation and standard deviation between the test set model performance and attribution accuracy. Test likelihood ($\Pi_i P(y_i|x_i;\hat{\theta})$) was used as the performance metric for classification tasks, and EV for the regression tasks.

Task	Method	Top-K	τ	τ_β
Classification	DL	0.75 ± 0.10	0.75 ± 0.06	0.73 ± 0.04
	IG	0.74 ± 0.10	0.75 ± 0.05	0.72 ± 0.02
	IXG	0.72 ± 0.13	0.56 ± 0.14	0.35 ± 0.28
Regression	DL	0.65 ± 0.40	-0.03 ± 0.41	-0.09 ± 0.31
	IG	0.88 ± 0.02	-0.05 ± 0.40	-0.09 ± 0.30
	IXG	0.62 ± 0.44	-0.26 ± 0.31	-0.27 ± 0.33

3.5 Conclusion

The most common approach to discover SNP to phenotype associations is to test single SNPs at a time. Furthermore, common approaches try to capture any interaction use linear models; completely ignoring any complex (non-linear) interactions between them. Deep learning models have been successful in a wide variety of domains, especially with complex unstructured data. Although they have been frequently used for disease risk score prediction from genotype, their use and success as a scientific discovery tool in GWAS has been limited thus far. One of the key reasons for that is the lack of understanding and consensus on explaining neural network model predictions. In particular, there are no established methods that can help identify whether the features used by a model are novel biomarkers for a phenotype or simple spurious correlations. This would require a measure of importance of a feature to a model's predictions as well as a measure of its significance. Additionally, there has been no empirical study to assess the strengths and weaknesses of prevailing deep learning feature importance techniques in the context of GWAS.

In this chapter, we designed a pipeline to conduct a GWAS using deep learning for phenotype prediction and gradient based feature importance methods to construct summarized importance scores of the input SNPs. We considered three pipelines that are all distinguished by the feature importance method used; DeepLIFT, GradInput, and Integrated Gradients. We conducted a series of tests to evaluate the capabilities of each pipeline in terms of the number of correctly identified known causal SNPs and their rankings on simulated datasets that were constructed with known ground truth causal coefficients. The accuracy and ranking metrics reported for the best models in Section 3.4.1 clearly show that all 3 pipelines can capture the effects of at least half of all causal SNPs and arrange almost all of them in correct order if the phenotype is categorical. In contrast, the proposed pipelines correctly identify almost all causal SNPs in the regression task albeit with low rank correlation with their causal coefficients.

We also show that the pipeline is robust to changes in model architecture as evidenced by the low variance in attribution performance of each pipeline. Considered together, these results show that at least on a simulated GWAS dataset, known causal SNPs can be clearly distinguished from non-causal SNPs by the proposed pipelines. Based on the fact that pipelines using DeepLIFT and Integrated Gradients perform similarly and often better than GradInput on the classification task, we can conclude that following experiments should employ Algorithm 1 with the DeepLIFT or Integrated Gradients attribution methods.

The relatively low Top-K accuracy for attribution on classification tasks and the low rank correlation metrics (τ and τ_β) for attribution on regression tasks do point to some deficiencies in our pipeline. These issues can be exacerbated in real world GWAS datasets that can have far more SNPs with many more peaks of significant association. Furthermore, the case-control imbalance can be much larger in real-world datasets making optimizing the model much harder. This lack of a predictive signal can be tackled by either oversampling data from the minority category, penalizing

mistakes made for cases more heavily, or adding auxiliary prediction tasks for related targets. We explore the latter in the form of multi-task prediction and the application of our pipeline to real-world data in subsequent Chapters.

4

Multi-task prediction and attribution with Deep Neural Networks

Genetic association studies often test multiple traits. Their analysis typically consists of testing each trait individually and then integrating the evidence for association for a particular SNP across traits (Galesloot et al. 2014). However, this approach is inconsistent with biology (Chavali et al. 2010), wherein the same SNP(s) can impact multiple related traits, as well as ignoring any potential in improving predictive power contained in predicting related traits together. Joint prediction of the target disorder with related phenotypes can act as a domain specific regularization that can help generalization. Predicting multiple related targets is a form of multi-task learning that exploits the commonalities or differences between related tasks in order to improve performance at each individual task (Caruana 1993). Multi-task learning can improve generalization at each prediction task by providing a domain specific inductive bias during training. Multi-tasking can also aid interpretability of the final model by narrowing the set of potential solutions that the model converges to. In this Chapter, we explore whether it is possible to extend our pipeline from Chapter 3 to leverage information between related traits in order to improve performance on each trait.

4.1 Related Work

The standard approach in genetic association studies is to analyze a single trait. This ignores the opportunity to integrate phenotypic information of related traits. It is well known that the genetic architecture of complex disorders involves common variants with small effect sizes (Visscher, M. A. Brown, et al. 2012), necessitating studies with large samples sizes in order to increase the power to detect said variants. The vast majority of common genetic variants for most traits have a markedly lower effect than 1% (Visscher, M. A. Brown, et al. 2012). Interestingly, many GWAS have highlighted loci that affect multiple traits, which potentially increases the evidence for pleiotropy in human disease (Solovieff et al. 2013). Pleitropy is the phenomenon of the same variant affecting multiple traits either directly (biological pleiotropy) or via another trait (mediated pleiotropy).

The joint analysis of multiple phenotypes has recently become popular for improving statistical power to detect novel associations. Solovieff et al. (2013) provide a detailed summary of techniques that analyze multiple phenotypes and broady classify them into two groups: multivariate techniques that directly model the association between a single SNP and multiple traits in a single cohort and univariate techniques that analyze the test statistics of multiple genotype-phenotype tests (single SNP and single trait). Most multivariate techniques require that the all target phenotypes are measured on each individual, which might increase the scope of the study, thus complicating its approval. This makes them less feasible for studies of rare diseases. However, if information for all phenotypes is available then these techniques have the advantage of being able to investigate the correlations between the phenotypes in addition to testing associations between them and each variant. For example, Ferreira and Purcell (2008) use canonical correlation analysis (CCA, Hotelling 1936) to find a linear combination of SNPs and another of phenotypes such that the cross-correlation between the two combinations is maximized. However, CCA assumes that the geno-

types be normally distributed, and may result in a higher false positive error rate if this assumption is violated. MultiPhen, developed in O'Reilly et al. (2012), performs logistic regression to predict the genotype at a single SNP from multiple phenotypes, thus finding a linear combination of phenotypes that are associated with each SNP. Another method, MANOVA (Warne 2014), which is the multivariate generalization to the Analysis of Variance (ANOVA), is equivalent to this procedure when the dependant variable (the genotype) is normally distributed. However, both CCA and MANOVA have an increased false-positive error rate when this assumption is violated. All 3 approaches model linear combinations of the phenotypes and test a single SNP at a time.

Univariate techniques combine results from multiple, single-phenotype and single-variant tests to identify variants that are associated with multiple phenotypes. Although the single-phenotype tests may can be conducted from the same sample of phenotypes, the key advantage of this category of techniques is their potential to systematically examine results from several GWAS (Panagiotou et al. 2013) in order to improve power to detect smaller genetic effects without the need to share individual-level data. Several techniques have been proposed to pool associations across multiple single-trait GWAS to test for presence of associations between the genotype and multiple traits (Bhattacharjee et al. 2012; Sluis, Posthuma, and Dolan 2013; Bolormaa et al. 2014; Zhu et al. 2015; Turley et al. 2018).

For a systematic comparison of some of the multi-trait GWAS methods mentioned above, we refer readers to Porter and O'Reilly (2017), who conduct a comprehensive evaluation of multi-trait GWAS methods, including methods that use sample-specific phenotype data (Ferreira and Purcell 2008; O'Reilly et al. 2012 and MANOVA) as well as those that inspect summary statistics of individual single-trait single-SNP GWAS (Sluis, Posthuma, and Dolan 2013; Zhu et al. 2015).

The subfield of machine learning called multi-task learning provides an interesting and potentially useful approach to developing techniques for multi-trait GWAS.

Multi-trait prediction is a natural application area for single-input multi-output multitask learning (Thung and Wee 2018), if multiple targets (correlated biological traits) are predicted from a single input (single genetic population). The key idea behind multi-task learning is being able to leverage information between the tasks to improve performance at each task. This has been used across various application areas like computer vision (Girshick 2015), speech processing (L. Deng, G. Hinton, and Kingsbury 2013), and natural language processing (Collobert and Weston 2008). It has also been widely applied in the fields of bioinformatics and clinical informatics. In fact, one of earliest applications of multitask learning was in Caruana, Baluja, and Mitchell (1995) wherein results of related clinical tests were used as targets to improve prediction of a subset of tests that could be used to automate and improve risk assessment of pneumonia patients. Mordelet and Vert (2011) devised ProDiGe to prioritize causal gene candidates of related diseases by formulating a multi-task variant of the PU learning problem (learning from positive and unlabelled examples, X.-L. Li and B. Liu 2005). They use a pairwise correlation matrix produced from Driel et al. (2006) to as a kernel to define a similarity measure between phenotypes, and the inner product of gene feature vectors to define a kernel for genes. Causal gene candidates are picked based on their association to known causal genes of similar phenotypes. Y. Li et al. (2016) formulate survival time prediction (predicting whether an event of interest occurs at a given instant) as a multitask prediction problem by converting the original task into a series of related binary classification tasks. The primary motivation being to learn a shared representation that can be leveraged to improve prediction at each classification task. They demonstrate the competitiveness of their method against single-task formulations on several gene expression cancer survival benchmark datasets. Puniyani, S. Kim, and Xing (2010) propose multiple trait prediction and association from separate populations in order to detect variants with relatively weak effects, overall improving the power of the association analysis compared to traditional single-trait GWAS. Perhaps most similar to the work in this

chapter is the work in D. He, Kuhn, and Parida (2016), which demonstrates that multiple output regression (Breiman and Friedman 1997) increases prediction accuracy of related genetic traits compared to single-trait regression. All of the above methods model their targets using linear models. As far as we know, there has been no study on interpreting deep neural networks trained via multi-trait prediction from genetic data in order to detect potentially causal variants.

4.2 Multi-task prediction of simulated traits

In the previous chapter, we predicted categorical and continuous traits separately, with no relation between the two traits. We observed a tradeoff between the Top-Kaccuracy and both rank correlation metrics τ and τ_β, with the latter two being higher for the regression task and the former for the classification task. Can we improve the ranking of causal SNPs in a regression task if we simultaneously predict related binary traits? Can we improve the Top-Kaccuracy of a classification task if we simultaneously predict a related quantitative trait? This chapter explores whether we can use the feature importance methods tested in Chapter 3 to identify shared causal SNPs with homogeneous effects on related traits.

Genotype and trait model with shared causal SNPs More concretely, let's consider the same probabilistic genotype model for a population of N samples with M SNPs from Section 3.2, with the genotype matrix denoted by $X_{N\times M}$. We will denote the binary trait by $y_j^{cat}\ \forall\ j \in \{1, 2, \ldots, N\}$ and the quantitative trait by $y_j^{cont}\ \forall\ j \in \{1, 2, \ldots, N\}$, with both traits collectively denoted by $Y_{N\times 1}^{cat}$ and $Y_{N\times 1}^{cont}$ respectively. As mentioned previously, we will assume that both traits are a linear function of the genotype matrix X.

In order for the traits to be related, we will assume that they have been measured from the same population (i.e. they are generated from the same genotype matrix X) and that a portion of the true causal SNPs are common to both traits, with shared effects. More precisely, if vectors $\beta^{cat}_{M\times 1}$ and $\beta^{cont}_{M\times 1}$ hold the effects of each SNP on the binary and quantitative phenotype respectively (i.e. they are effect vectors of each phenotype), then the assumption of shared causes with homogeneous effects implies that there exists a non-empty set of SNP indices $\mathcal{M} = \{j \mid \beta^{cat}_j = \beta^{cont}_j\}$.

Multi-task prediction and attribution In contrast to Chapter 3, we cannot use two separate models if we are to leverage information shared between the two traits. In order for our prediction model to leverage information shared between both traits, we will need to train a model that shares parameters for the two prediction tasks. For the 2-layer networks considered in the pipeline in Chapter 3, we can easily modify the model used for the single prediction task by using two output layers instead of one as depicted in the figure below:

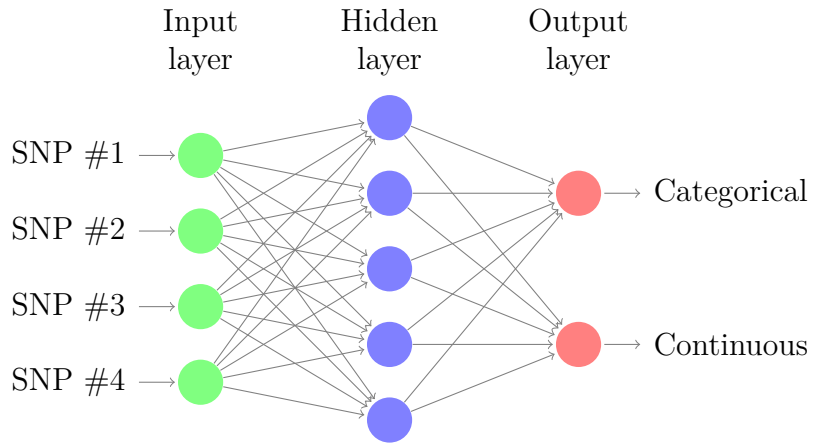

Figure 4.1: A feedforward neural network with two output heads that has shared input ***and*** shared hidden layers.

In other words, we feed a single, shared representation to two output heads by sharing all parameters until the output layer. We can also reduce the information shared between the two tasks by reducing the number of shared parameters by only

sharing the weights of the input layer like in Figure 4.2 below:

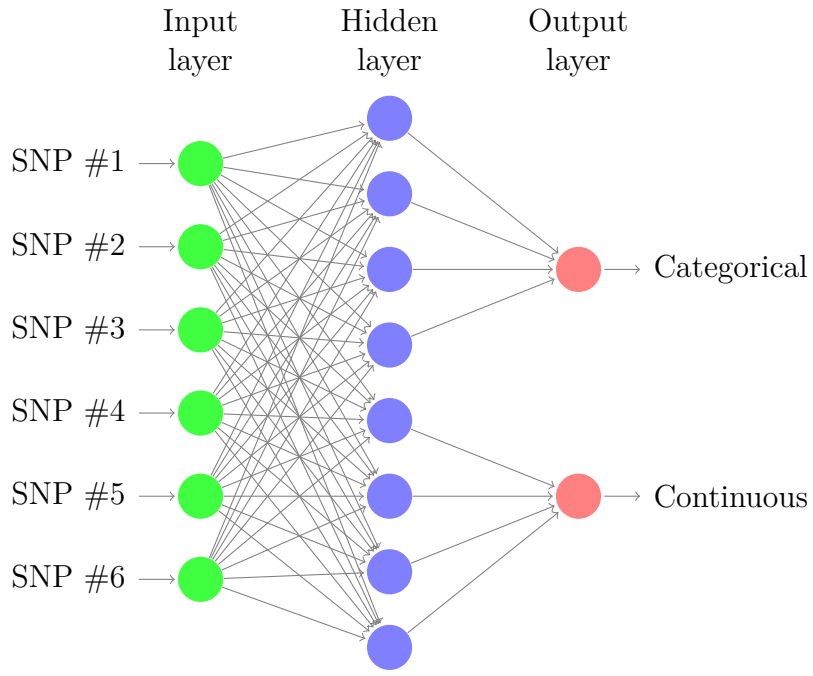

Figure 4.2: A feedforward neural network with two output heads that has shared input layers but separate hidden layers

Both architectures will learn to predict two traits simultaneously but the latter has the advantage of using more parameters specific to each prediction task, which could result in better attribution performance. We empirically compare their analysis in Section 4.3 but first we proceed to describe the training and attribution steps of the multitask pipeline.

In order to train the model to predict both targets, we must formulate an objective function that includes the prediction losses for both tasks. We simply add both loss functions together after weighing them using parameters λ_1 and λ_2, which gives us the flexbility to give importance to one task over the other. This gives us the equation below:

$$\mathcal{L}^{mult} = \lambda_1 \mathcal{L}^{cat} + \lambda_2 \mathcal{L}^{cont}, \tag{4.1}$$

but as before, we assume that a sparse subset of the input SNPs will be causal, and so we add the L1 norm of the first layer to give Equation 4.2 as the objective function:

$$J^{mult} = \mathcal{L}^{mult} + \lambda|\theta_1| \tag{4.2}$$

We run the attribution routine from Algorithm 2 (Chapter 3) on each output head of the model to get an attribution score that has been averaged over all samples in the test set. This gives us a pair of summarized attribution scores (one for each phenotype), that is also averaged over multiple seeds to improve robustness of the final results. This gives us Algorithm 3 which specifies a pipeline for GWAS using deep feedforward models that are trained to simultaneously predict a categorical and continuous trait. Since the final objective function J^{mult}, shares objectives of both the classification and regression tasks, it can get tricky to find the right validation metric h that suits both.

Evaluation Since we've explored the consistency and fidelity of our pipeline already in Chapter 3, we will evaluate the multitask pipeline by the accuracy of its attributions using the same metrics used in Chapter 3: Top-K, τ, and τ_β.

4.3 Experiments

We first describe the procedure used to generate the datasets used for our experiments. We then address the question of model validation and selection for our multi-trait prediction model.

4.3.1 Data

We generate 4 datasets of 10,000 individuals with 10,000 SNPs, including 10 causal SNPs, for both a categorical and a continuous trait. 5 of the 10 causal SNPs are

Algorithm 3: Our GWAS pipeline that uses feedfoward neural networks to predict two traits and feature importance methods to attribute a global importance score to each SNP for both traits

Input: A dataset $\mathcal{D} = (X, Y, Z)$ of N samples with M SNPs each, consisting of the genotype matrix X, phenotypes Y^{cat} and Y^{cont}, concatenated to form $Y_{N\times 2}$, and covariates Z. Feedforward Neural Network $f_\theta : \Re^M \times \Re^K \rightarrow \Re \times 2$, a feature importance method `Attribute()`, objective function J^{mult} and a validation metric $h : \Re^{N\times 2} \times \Re^{N\times 2} \rightarrow \Re$

Result: Optimized neural network model $f_{\hat{\theta}}$ and an attribution matrix $\hat{R}_{M\times 2} = (r_{ij})$ s.t. $r_{ij} \geq 0 \, \forall \, i, j$

begin
  Split the dataset tuple (X, Y, Z) into training $D_{train} = (X_{train}, Y_{train}, Z_{train})$, valid $D_{valid} = (X_{valid}, Y_{valid}, Z_{valid})$, and test $D_{test} = (X_{test}, Y_{test}, Z_{test})$
  for $t \leftarrow 1$ **to** T **do**
    Optimize f using gradient descent on J^{mult}, calculated on the training dataset D_{train}, to yield f_{θ^t}
    if $h(Y_{valid}, f_{\theta^t}(X_{valid}, Z_{valid})) \leq h(Y_{valid}, f_{\theta^{t-1}}(X_{valid}, Z_{valid}))$ **then**
      $f_{\hat{\theta}} \leftarrow f_{\theta^{t-1}}$
      break
    else
      $f_{\hat{\theta}} \leftarrow f_{\theta^t}$
    end
  end
  R $\leftarrow$ `Attribute`$(\mathrm{f}_{\hat{\theta}}, (X_{test}, Z_{test}), [\ldots])$
  $\hat{R} \leftarrow mean(|R|)$
  return $f_{\hat{\theta}}$, $\hat{R}$
end

shared between the two traits, with their corresponding causal effects.

We synthesize a multi-task dataset by generating a categorical and a continuous trait with a portion of the causal SNPs and their corresponding causal effects shared between the two traits. Let m^{cat} denote the set of indices corresponding to the SNPs that are uniquely causal for the binary trait, let m^{cont} denote the set of indices of the SNPs that are uniquely causal for the continuous trait, and let m^{shared} be the set of indices for the shared causal SNPs. Therefore, the total number of causal SNPs for the binary trait is $m^{cat} + m^{shared}$ and for the quantitative trait it is $m^{cont} + m^{shared}$.

The larger the cardinality of the set m^{shared}, the greater number of shared causal SNPs. For a genotype matrix with M SNPs, we define the 3 vectors $\overline{\beta}^{cat}_{M,1}$, $\overline{\beta}^{cont}_{M,1}$, and $\beta^{shared}_{M,1}$ as follows:

$$\bar{\beta}^{cat}_{m} \sim \begin{cases} \mathcal{N}(0, 0.5), & \text{if } m \in m^{cat} \\ 0, & \text{if } m \notin m^{cat} \end{cases} \tag{4.3}$$

$$\bar{\beta}^{cont}_{m} \sim \begin{cases} \mathcal{N}(0, 0.5), & \text{if } m \in m^{cont} \\ 0, & \text{if } m \notin m^{cont} \end{cases} \tag{4.4}$$

$$\beta^{shared}_{m} \sim \begin{cases} \mathcal{N}(0, 0.5), & \text{if } m \in m^{shared} \\ 0, & \text{if } m \notin m^{shared} \end{cases} \tag{4.5}$$

The final effect vector for each trait is constructed as follows:

$$\beta^{cat} = \beta^{shared} + \bar{\beta}^{cat} \tag{4.6}$$

$$\beta^{cont} = \beta^{shared} + \bar{\beta}^{cont} \tag{4.7}$$

The traits are simulated exactly as described in Section 3.3.1 but with effect vectors obtained from Equation 4.6. We set $m^{cat} = 2000, 2400, 2800, 3200, 3600$, $m^{cont} = 4500, 4900, 5300, 5700, 6100$, and $m^{shared} = 0, 300, 600, 900, 1200$, thus evenly separating elements of each causal set while mantaining a clear boundary between each set. Thus, we have a total of 10 causal SNPs per trait, with the first 5 causal SNPs being shared. The different dataset settings are characterized by the values of the spatial parameter $a \in \{0.01, 0.1, 0.5, 1\}$. Finally, all the classification datasets have an average case-control ratio of 0.3.

4.3.2 Model architecture

In order to be able to compare model prediction performance from the single-task experiments and multi-task experiments in this Section, we train 2-layer feedforward

NNs with the same combinations of hidden units as the models used in Table 3.1. However, since the models are being trained simultaneously on two related tasks, we also train and test larger 3-layer NNs with the first, second, and third layers having 512, 256, and 128 hidden units, respectively. The widths and depths of each model are listed in Table 4.1. We also train models that only share the input layer weights as depicted in Figure 4.2. The details and rationale behind the experiments using this architecture are explained in Section 4.4.2.

Table 4.1: Width of the 2-layer and 3-layer feedforward NNs trained on the simulated multitask data

	Layer #	Number of hidden units in layer
2-layer networks	First	{32,64,128}
	Second	{128,256,512}
3-layer networks	First	{512}
	Second	{256}
	Third	{64}

4.3.3 Training, Model selection, and Ranking

Similar to Section 3.3.2, each model was trained on each spatial configuration of the datasets, over 5 seeds, for both classification and regression tasks. We train each model on 50% of the samples, early stop on 25% of the samples, and validate the model architectures using the remaining 25%.

We consider the same set of values for λ as in Table 3.1. This resulted in experiments with $(3 * 3 + 1) * 4 = 40$ different models. All models share all weights, except the output layer weights, for both prediction tasks, as depicted in Figure 4.1. We use J^{mult} as the objective function for training and experiment with different combinations of λ_1 and λ_2. The different values of λ, λ_1, and λ_2 are listed in Table 4.2. Models are early-stopped as soon as the value of J^{mult} starts decreasing on the validation set.

Table 4.2: Values of the l1 regularization coefficient λ and the multi-task loss function hyperparameters λ_1 & λ_2 for the architectures listed in Table 4.1

λ	(λ_1,λ_2)
{0.01,0.1,1,10}	(1,1), (1,0.1), (0.1,1)

Model Selection Depending on the values of λ_1 and λ_2, we use different metrics h to select the best models. For example, for models trained on J^{mult} with $\lambda_1 > \lambda_2$, we use the validation set likelihood $\Pi_i P(y_i|x_i;\hat{\theta})$, whereas when $\lambda_2 > \lambda_1$, we use validation set EV, and finally when $\lambda_1 = \lambda_2$, we use $\Pi_i P(y_i|x_i;\hat{\theta}) + EV$, computed on the validation set. For each experiment we select models with the highest average value of h.

Ranking As in Subsection 3.2.2, we compute 3 different metrics; the number of causal SNPs in the top K highest ranked SNPs (Top-K), with $K = 10$, the kendall tau rank correlation for the causal SNPs (τ), and the weighted kendall tau rank correlation for the rankings of the causal SNPs (τ_β). We select 10 SNPs with the highest attribution score and count the number of causal SNPs in that set to compute Top-K. The rank correlations are computed between a ranking computed on the causal coefficient β and a ranking computed on the summarized attribution scores.

4.4 Results

We first report model performance and corresponding attribution accuracy of the top performing models on each dataset trained on both classification and regression tasks, while giving more importance to one over the other ($\lambda_1 > \lambda_2$ and vise-versa). Finally, we investigate the scenario where both prediction tasks are given equal importance ($\lambda_1 = \lambda_2 = 1$) at the end of Section 4.4.1 as well as all of Section 4.4.2. All reported results are on the test split of the dataset, and higher values are better for all reported attribution and model performance metrics.

4.4.1 Predictions using shared hidden layers

Table 4.3 lists the attribution metrics discussed in Section 3.2.2 for the top performing 2-layer NNs selected from Table 4.1, with more importance given to the binary trait over the quantitative trait ($\lambda_1 > \lambda_2$). At least half of the causal SNPs are always identified (Top-$K \geq 7$ for at least one attribution method) while consistently ranking most causal SNPs correctly ($\tau \geq 0.70$ for at least one attribution method per a). Compared to Table 3.3, Top-Kperformance has mostly improved for DL and IG, while decreasing for IXG. Rather surprisingly, IXG seems to rank on par or better than DL and IG for tasks on datasets with $a \in 0.01, 0.1, 0.5$. Furthermore, for DL and IG, values of both τ and τ_β have worsened, suggesting that predicting quantitative traits has had a detrimental effect to ranking of the causal SNPs as we first observed in Section 3.4.

Table 4.3: Prediction performance of the best 2-layer model and corresponding attribution accuracy on the classification tasks with $\lambda_1 = 1$ and $\lambda_2 = 0.1$ in loss equation 4.1

		Top-K			τ			τ_β		
a	$\Pi_i P(y_i \mid x_i; \hat{\theta})$	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.80	7.0	7.0	7.0	0.82	0.78	0.87	0.73	0.67	0.92
0.10	0.77	7.0	7.0	4.0	0.78	0.78	0.73	0.79	0.79	0.80
0.50	0.80	7.0	7.0	7.0	0.82	0.82	0.73	0.88	0.88	0.85
1.00	0.79	8.0	8.0	7.0	0.70	0.70	0.47	0.50	0.50	0.24

Table 4.4 lists the attribution metrics discussed in Section 3.2.2 for the top performing 2-layer NNs selected from Table 4.1, with more importance given to the quantitative trait over the binary trait ($\lambda_2 > \lambda_1$). All models identify at least 70% of the causal SNPs (Top-$K \geq 7$ for all attribution methods), while consistently ranking most causal SNPs correctly ($\tau \geq 0.78$ for at least one attribution method per a). Compared to Table 3.5, Top-Kperformance has slightly reduced for all attribution methods. However, the consistency in both rank correlation metrics has vastly improved across datasets, for all attribution methods. This is consistent with our

findings about the difference in performance on the single-task classification and regression tasks in Chapter 3; the added classification task has vastly improved the rankings of the causal SNPs but only slightly decreased the overall Top-K accuracy.

Table 4.4: Prediction performance of the best 2-layer NN and corresponding attribution accuracy on the regression tasks with $\lambda_2 = 1$ and $\lambda_1 = 0.1$ in loss equation 4.1

		Top-K			τ			τ_β		
a	EV	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.79	7.0	7.0	7.0	0.96	0.96	0.78	0.94	0.94	0.75
0.10	0.78	7.0	7.0	7.0	0.91	0.91	0.78	0.92	0.92	0.74
0.50	0.71	9.0	9.0	9.0	0.96	0.96	0.73	0.98	0.98	0.86
1.00	0.76	8.0	8.0	8.0	0.78	0.78	0.69	0.82	0.82	0.62

Table 4.5 lists the attribution metrics obtained from applying the top performing 2-layer NNs selected from Table 4.1, with both traits given equal importance ($\lambda_2 = \lambda_1 = 1$). In the interest of brevity, we present results using only IG and present the results for DL and IXG in Tables 7.4 and 7.5 of Section 7.2 of the Appendix. We observe that the attribution accuracy and correlation performance for the quantitative trait are drastically better than for the binary trait. In fact, attribution performance on the binary trait has drastically deteriorated, while on the quantitative trait it has remained similar to and at times even slightly better than in Table 4.4. This seems to suggest that attribution accuracy on the quantitative trait is far more robust than on the binary trait in the multi-task setting.

Table 4.5: Attribution performance of the best 2-layer model using IG for both traits (binary on the left and quantitative on the right), with $\lambda_2 = 1$ and $\lambda_1 = 1$ in loss equation 4.1

Top-K	τ	τ_β
4.0, 7.0	0.60, 0.96	0.65, 0.94
5.0, 7.0	0.60, 0.91	0.67, 0.92
5.0, 9.0	0.07, 0.96	0.11, 0.98
4.0, 8.0	0.47, 0.78	0.56, 0.82

In order to investigate the low attribution performance on the binary task, we plot

the mean attribution scores for the binary trait computed from applying IG to the models in Figure 4.3. These scores are for models trained on the dataset with $a = 0.5$ since the difference in the rank correlation metrics between the two traits is the largest for this dataset setting. The issue is immediately noticable; some of the causal SNPs of the quantitative trait have high attribution scores with respect to the binary trait. In other words, there is considerable "leakage" found in the attribution scores from the quantitative trait to the binary trait, decreasing the pipeline's performance on the binary trait in the multitask setting. This is expected since both output heads are sensitive to a common set of parameters. However, it is rather surprising that the leakage is only one-way: from the regression task to the classification task. A possible cause of this issue could be that given equal weights to the two prediction tasks, the model may find it easier to improve on the regression task to help optimize the overall objective. The focus on the regression task may have lead to the model's predictions of the binary trait becoming more sensitive to some SNPs that are exclusively causal to the quantitative trait. We attempt to ameliorate this issue in the next section.

4.4.2 Leakage of identified hits between traits

The output layers of the models used so far share all input and hidden layer weights (Figure 4.1). As presented earlier, this can lead to some signals leaking from one trait to the other (Figure 4.3). In this section, we present attribution performance results of multitask pipelines using models with the architecture from Figure 4.2. By sharing only the input layer weights, these models are forced to share the minimum set of weights for the two tasks. This should help improve attribution accuracy since there are more parameters available exclusively for either task.

Table 4.6 shows the attribution accuracies after applying IG on 2-layer models (with the aforementioned architecture) trained on both traits. Pipelines using models with this architecture have better attribution performance for the binary task as

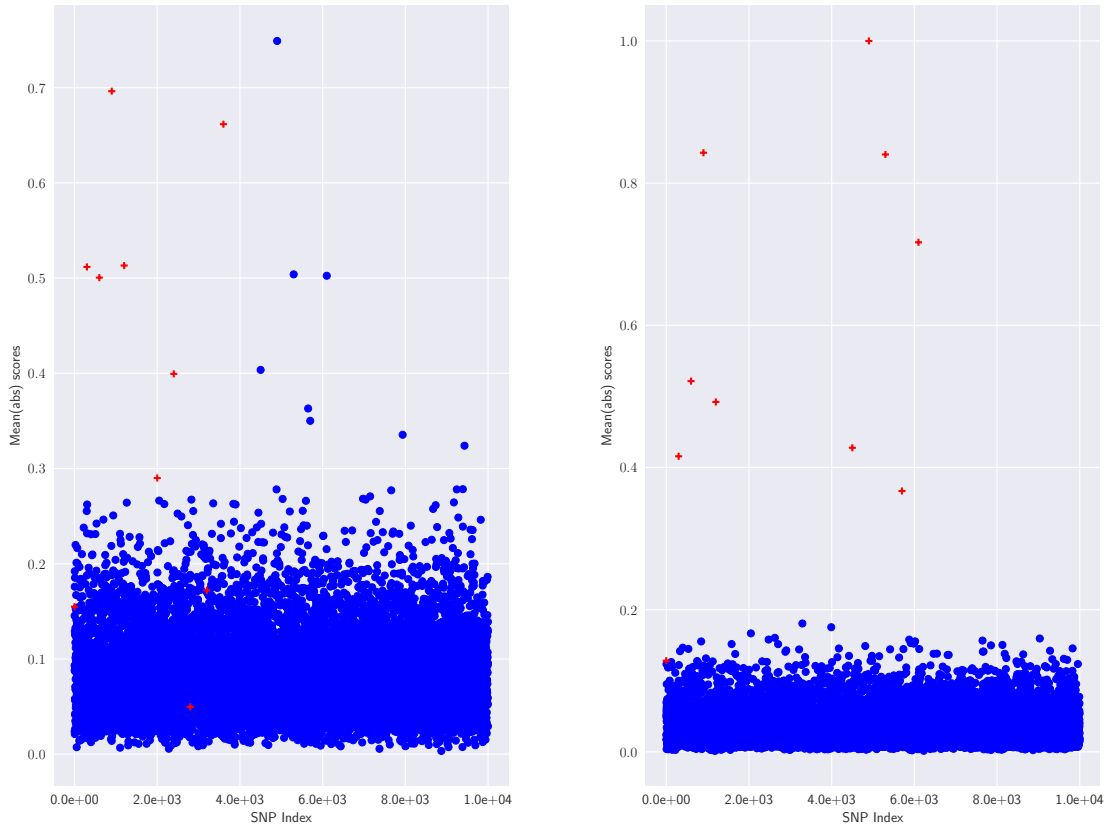

Figure 4.3: Mean absolute Integrated Gradients scores for the top performing model with architecture from Figure 4.1, with $\lambda_1 = \lambda_2 = 1$, and trained on both a categorical(left) and continuous (right) target with dataset $a = 0.5$, averaged across seeds

evidenced by the increase in Top-Kaccuracy and the drastically better rank correlation metrics for most dataset settings.

Table 4.6: Attribution performance of the best 2-layer model using the architecture in 4.2, using IG for both traits (binary on the left and quantitative on the right), with $\lambda_2 = 1$ and $\lambda_1 = 1$ in loss equation 4.1

a	Top-K	τ	τ_β
0.01	5.0, 7.0	0.60, 0.91	0.66, 0.92
0.10	6.0, 7.0	0.91, 0.91	0.96, 0.92
0.50	7.0, 9.0	0.69, 1.00	0.78, 1.00
1.00	6.0, 8.0	0.64, 0.64	0.66, 0.58

The scatter plot in Figure 4.2 shows that the leakage has certainly reduced but has not been completely eliminated. The SNP with the highest attribution score is now a correct causal hit for the binary trait but leakage from the regression task for

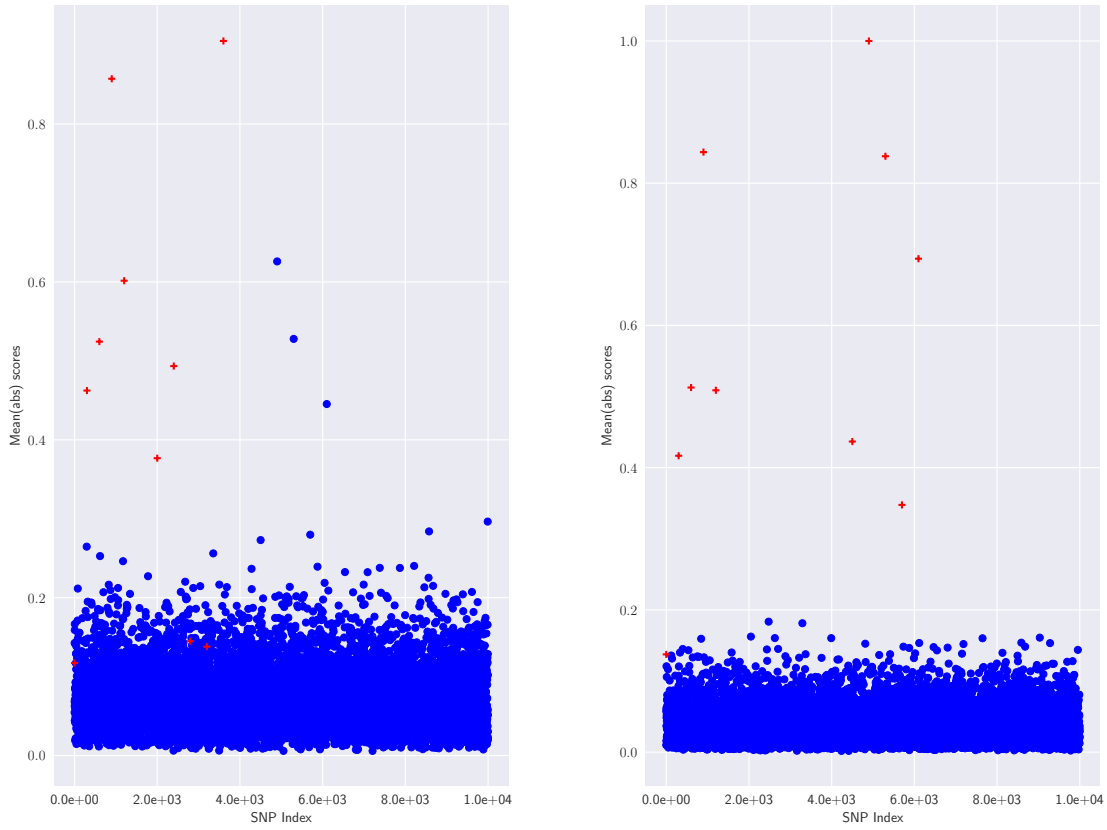

Figure 4.4: Mean absolute Integrated Gradients scores for the top performing model with architecture from Figure 4.2, with $\lambda_1 = \lambda_2 = 1$, and trained on both a categorical (left) and continuous (right) target with dataset $a = 0.5$, averaged across seeds

at least three SNPs is still present.

Since we cannot reduce the sharing of parameters between traits any further, we now train models with increased depth and perform the same comparison. Table 4.7 compares the attribution accuracies for classification between 3-layer models with all hidden layers shared (left) and with only the input layer weights shared (right). For brevity, we present results using IG but these results hold for DL and IXG as well. As expected, pipelines using models with only the input weights shared have better attribution accuracy (all three metrics are equal or higher). Furthermore, as expected, these pipelines have better performance compared to the two layer models in Table 4.6.

This is confirmed by Figure 4.5, which shows attribution scores obtained from using IG on the classification task of a 3-layer model trained on both tasks on the

Table 4.7: Attribution performance using IG on the best 3-layer model with both shared and unshared hidden layer (left and right below respectively) architectures for the binary trait, with $\lambda_2 = 1$ and $\lambda_1 = 1$ in loss equation 4.1

a	Top-K	τ	τ_β
0.01	6.0, 6.0	0.60, 0.69	0.70, 0.71
0.10	6.0, 7.0	0.47, 0.96	0.59, 0.98
0.50	6.0, 7.0	0.64, 0.96	0.69, 0.98
1.00	6.0, 8.0	0.60, 0.78	0.59, 0.65

dataset with $a = 0.5$. After switching from the architecture in Figure 4.1 (left subplot) to the one in Figure 4.2, the leakage is completely eliminated.

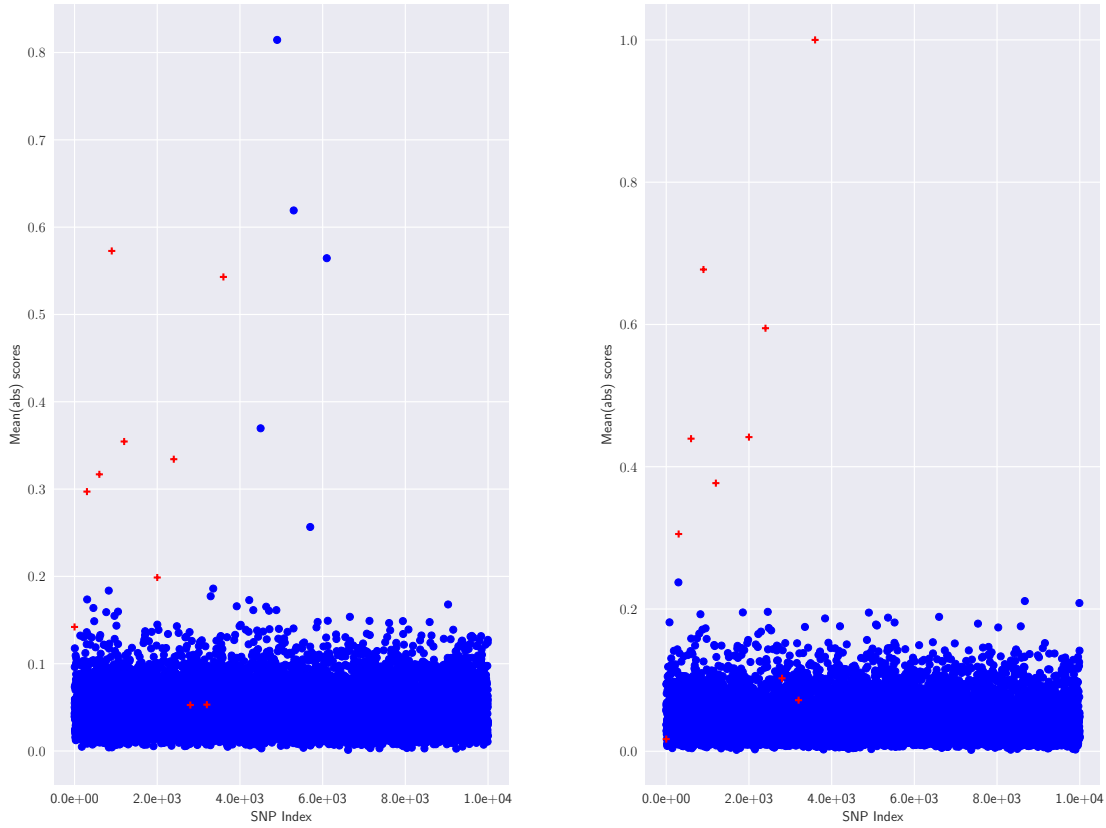

Figure 4.5: Mean absolute IG scores from binary trait attribution by the best 3-layer models with both shared and unshared hidden layer architectures (left and right above respectively), with $\lambda_2 = 1$ and $\lambda_1 = 1$ in loss equation 4.1. The models were trained on the dataset with $a = 0.5$.

4.5 Conclusion

In this chapter, we hypothesized that interpretation of models that jointly predict multiple related traits can help increse the number of identified causal SNPs. We show that in a simulated setting, interpretation of deep neural networks trained on multiple related phenotypes can increase the number of causal SNPs that are identified for each individual phenotype as compared to training and attribution of a single phenotype at a time (Chapter 3). We modified the pipeline in Chapter 3 for the joint prediction of a simulated binary and quantitative trait by training a single model that uses shared hidden and input layer weights to predict both traits simultaneously. We conducted a series of experiments to test the attribution accuracy of three pipelines that are all distinguished by the feature importance method used; DeepLIFT, GradInput, and Integrated Gradients. Therefore, we train several two-layer models that are trained to predict a binary and quantitative trait that share a portion of their causal SNPs. We show that the Top-Kaccuracy can be improved for causal SNPs of the binary trait if the classification task is prioritized over the regression task, while it slightly decreases for the quantitative task if the regression task is given more importance. However, in both scenarios, the ability of each pipeline to correctly rank the causal SNPs vastly improved compared to the single-task case.

We also show that the attribution accuracy for the quantitative trait remains unchanged if we weigh prediction losses for each trait equally, on the other hand it is significantly reduced for the binary trait. We identify the source of this issue as some leakage in sensitivities of the causal SNPs of the quantitative trait to the binary trait due to the presence of too many shared weights from the hidden layers of the models. We ameliorate this issue by reducing the shared parameters to only the input layer and see an improvement in attribution performance on the classification task. We show that by using more powerful models we can almost eliminate this issue.

Overall, we showed that deep neural networks can be trained and interpreted in

simulated multi-task settings to identify and correctly rank the majority of causal SNPs for two traits that share half of their causal SNPs and corresponding causal effects (i.e. homogenous effects of shared causal SNPs). But multi-trait GWAS studies deal with scores of related traits simultaneously, with more complex relationships between the traits and the SNPs. For example, some SNPs may be causal for only a subset of the investigated traits, or traits may share a few causal SNPs but not the corresponding causal effects (heterogeneous effects). These scenarios are significantly more complex than the one we've investigated in this Chapter, and can exacerbate the leakage issue we identified in Section 4.4. We leave this for further work and proceed to applying pipelines from both Chapters 3 and 4 to a real-world dataset in the next Chapter.

5

Applications to the UK Biobank

We've spent the previous chapters analyzing the accuracy and robustness of GWAS pipelines that interpret Neural Networks trained on simulated datasets containing a small number of causal SNPs. In this Chapter, we will test these techniques on a large, real-world genomics dataset, namely the UK Biobank (Sudlow et al. 2015), a large population cohort including more than 500,000 genotyped participants.[1].

The UK Biobank is widely used for genomic research, with over 2000 researchers already having access to its data (Manolio 2018). It contains a variety of phenotypic information available over hundreds of thousands of participants who have been extensively genotyped, allowing us to test the pipelines that we've developed so far on various clinically relevant traits, on a practically significant scale. This makes the UK Biobank a challenging and practical benchmark for the pipelines that we've proposed in Chapters 3 and 4.

5.1 Related Work

The UK Biobank contains a rich variety of phenotypic measurements on each participant, all of whom are aged 40 to 69, which is when complex chronic diseases are likely to manifest (Manolio 2018). This gives researchers ample opportunities to dis-

[1]This research has been conducted using the UK Biobank Resource under Application Number 20168.

cover novel genetic associations. It has enabled several genetic association studies and meta-analysis studies spanning an expansive range of diseases such as Coronary Artery Disease (CAD, Harst and Verweij 2018), Type-2 Diabetes (Xue et al. 2018), Osteoporosis (Morris et al. 2019), or neurological diseases such as Parkinson's disease (Blauwendraat, Nalls, and Singleton 2020), Alzheimer's disease (Marioni et al. 2018), or Depression (D. M. Howard et al. 2019), as well as several other genetic, molecular, and physiological biomarkers (Elliott et al. 2018; Sinnott-Armstrong, Tanigawa, et al. 2021; Sinnott-Armstrong, Naqvi, et al. 2021) that can help decipher biological mechnisms behind complex traits and diseases. Furthermore, the large sample size and dense genotyping in the UK Biobank matches the scale of many recent genetic association studies that investigate common traits and complex diseases (Ahlqvist et al. 2015; Yengo et al. 2018; Tam et al. 2019; Pulit et al. 2019).

However, the application of machine learning techniques to the UK Biobank data has been limited. A notable category is using machine learning methods to improve phenotype prediction performance. For example, Oster et al. (2020) compare the detection of atrial fibrillation (AF) using UK Biobank electrocardiogram (ECG) data, done automatically by a combination of classical machine learning (Support Vector Machines) and deep learning models versus manually by experts. They find that the agreement between the proposed approach and another expert is similar to the inter-observer agreement between two experts, leading them to conclude that automated detection of AF in large datasets similar to the UK Biobank is possible. Schulz et al. (2020) investigate whether complex non-linear machine learning models (such as deep learning models) can use brain imaging data in the UK Biobank to improve the classification of individuals into their respect subgroups of age and gender. Finally, Bellot, Campos, and Pérez-Enciso (2018) compared deep MLPs and CNNs against Bayesian linear regression on the prediction of five phenotypes: height, bone heel mineral density, body mass index, systolic blood pressure, and waist–hip ratio, on a sample set of 100,000 individuals and 500,000 SNPs from the UK Biobank. They found that CNNs

were competitive to the linear models but did not outperform them by a wide margin.

Slightly distinct but related is the work of Alaa et al. (2019), who investigate the use of machine learning algorithms for cardiovascular (CVD) risk prediction. They trained an ensemble of statistical and machine learning algorithms (ranging from logistic regression to deep neural networks) using Bayesian optimization to prioritize and select algorithms that are optimal for CVD risk prediction and find that it outperformed established methods at risk prediction for relevant subpopulations such as individuals with a history of Diabetes.

Finally, we were unable to find published work that uses deep learning techniques on the UK Biobank genomics data for SNP-trait association.

5.2 Predicting Diabetes and HbA1c using the UK Biobank

In order to test the scalability and robustness of our approach on large real-world genetic datasets, we apply the pipelines from Chapters 3 and 4 to predict traits obtained from the UK Biobank (Section 5.2). The importance scores obtained from the pipelines are then compared to results obtained from a conventional GWAS conducted on the same traits.

In this Section, we first present and justify the selection of Diabetes and HbA1c as the target traits from the UK Biobank. We then briefly discuss the results of a conventional GWAS targetting both traits, separately. Finally, we describe the procedure used to collect and process the genotype and phenotype data before it is used as input for the experiments in this Chapter.

Type 2 Diabetes (T2D) is a common chronic health condition that is amongst the top 10 causes of death globally (Forouzanfar et al. 2016). Its presence is associated with increased cause-specific mortality with causes such as infections, cardiovascular

disease, stroke, chronic kidney disease, chronic liver disease, and cancer (Yang et al. 2019; Policardo et al. 2014). Diabetes has the second largest net negative effect on reducing health adjusted life expectancy at birth ($HALE_0$, C. Chen et al. 2019). According to the International Diabetes Federation, 451 million people worldwide were living with diabetes in 2017, with the number expected to rise to 700 million by 2045 (N. Cho et al. 2018). It is primarily caused by a failure in the body's normal response to insulin or by insufficient production of insulin by the body's cells (Taylor 2013). Genome-Wide Association studies helped discover an initial list of T2D-associated loci (e.g. the genes PPARG and TCF7L2, Altshuler et al. 2000; Grant et al. 2006). However, the etiology of this disease remains largely unknown and subclassification could improve patient management (Udler et al. 2018).

The level of HbA1c, also known as glycated haemoglobin, is an important risk factor for Diabetes. It is made when glucose (sugar) sticks to the body's red blood cells. Increase in HbA1c increases the likelihood of developing Diabetes related complications. In fact, it is commonly used as a test to diagnose Diabetes (Sarnowski et al. 2019). Thus, its clinical relevance and relationship to Diabetes makes HbA1c a potential quantitative target for analysis in this Chapter. Furthermore, if we observe the scatter plot of the signals obtained from a conventional GWAS conducted on both Diabetes (left) and HbA1c (right) on the data in the UK Biobank[2] (Figure 5.1), we find that these traits share at least three signals, with the presence of strong signals unique to each trait. Therefore, selecting Diabetes and HbA1c , allows us to stay close to the settings that we investigated on simulated data in previous chapters: in Chapter 3, we evaluated our pipeline on individual classification and regression tasks, while in Chapter 4, we evaluated our pipeline after training the model simultaneously on a pair of classification and regression tasks with shared causal SNPs.

We now describe how we build two datasets for the phenotypes Diabetes and HbA1c :

[2]the preparation and preprocessing of the data is described later in this section

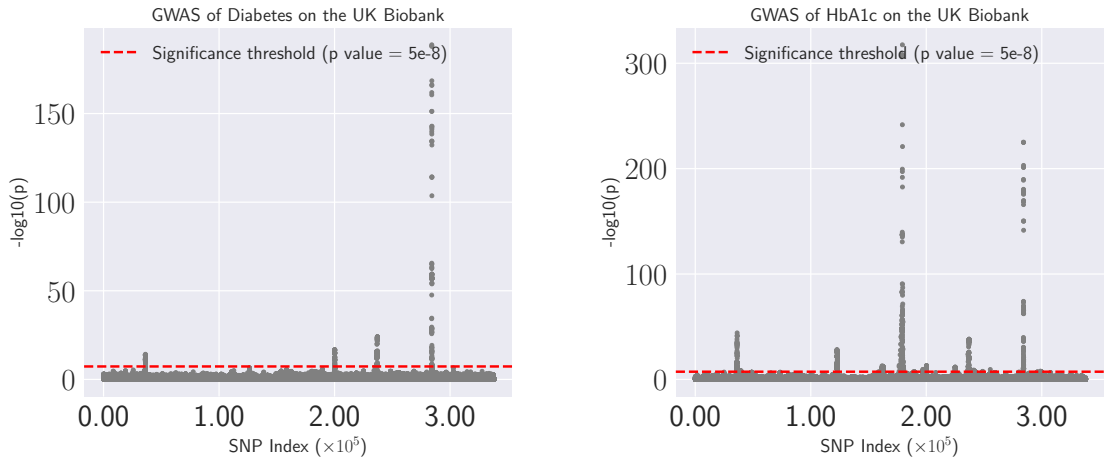

Figure 5.1: Scatter plot of the negative p-values from a conventional GWAS conducted on Diabetes (left) and HbA1c (right)

Filtering individuals We consider the imputed genotypes, where we filter out individuals with more than 2% missing genotypes, in addition to individuals with sexual chromosome aneuploidies or with genetically inferred sex different from the self-reported sex. We then select a subset of the cohort of European ancestry to avoid population stratification (*i.e.* confounding bias due to ethnicity) by using the UK Biobank provided principal components and keeping individuals near the cluster of individuals self-reporting as of *white British ancestry*. To avoid including related individuals, we randomly select one individual from pairs with a kinship coefficient above 0.0884 (corresponding to a 2nd degree relationship). This results in the selection of $N = 413,173$ individuals (samples).

Filtering variants After selecting individuals, we filter genetic variants to be used as features. Starting from all variants on chromosome 10, we filter out variants with minor allele frequency under 1%, variants with a call rate (Reed et al. 2015) under 99% and set genotypes with a probability under 90% to missing. This results in the selection of $M = 336,814$ variants (SNPs) per individual.

Phenotype extraction The Diabetes phenotype is defined based on a combination of hospitalization codes and the self-reported verbal interview data. Specifically, we

code as cases any participant with data coding for Diabetes (field #20002, coded as 1220) as cases, or with the '249' or '250' ICD9 codes, or E10, E11, E12, E13, or E14 ICD10 codes as the primary or secondary reason for hospitalization. The remaining individuals are used as controls. This results in a dataset with 24,717 diabetes cases and 388,456 controls.

The HbA1c phenotype (field #30750) is extracted for 395,042 participants. If multiple measurements are available for an individual, the arithmetic mean is used. The values are also log-transformed to ensure an approximately normal distribution as typical in continuous trait GWAS.

5.3 Experiments

We run two sets of experiments, one to predict Diabetes and HbA1c in the traditional single-task setting (Chapter 3), wherein a single target is predicted, and another in the multi-task setting (Chapter 4), wherein the two traits are predicted together.

Training We always partition the data into 80% for training, 5% for early stopping and validation, and report test metrics on the final 15%. Since repeat our experiments on 5 seeds, we use 5 different partitions of the data. We perform grid-search over the set of model architectures listed in Table 5.1, and with the L1-regularization penalty $\lambda \in \{0.1, 1, 10\}$. The same architectures are used for both single-task and multi-task settings. For the multi-task setting, we use the model with unshared hidden weights as depicted in Figure 4.2. As for the values of λ_1 and λ_2 in the multi-task setting, we train a set of models with $(\lambda_1, \lambda_2) = (1, 0.1)$ and another with $(\lambda_1, \lambda_2) = (0.1, 1)$. All experiments are repeated with 5 seeds. This results in 9 different architecures, times 3 different values of λ, 2 phenotypes, and finally $2 + 1$ different settings (1 for single-task and 2 for multi-task), bringing the total number of experiments to 108 * 5 (seeds) = 540.

Table 5.1: Widths of the 2-layer feedforward NNs trained for either classification, regression, or both.

		Number of hidden units in layer
Layer #	First	{32,64,128}
	Second	{128,256,512}

As we mentioned earlier in Step 1 of Section 3.2.1, we force the models to pick a handful of SNPs by adding an L1-penalty to the first layer weights. We use J^{cat} as the objective function for the Diabetes prediction task, and J^{cont} for the HbA1c prediction task.

Model selection After training, we select model architectures based on h, which is set as the validation set likelihood $\Pi_i P(y_i|x_i;\hat{\theta})$ for Diabetes classification, and validation set EV for regression. Model performance is compared on the average value of h over 5 seeds. For both single-task and multi-task settings, we select models with the highest average value of h.

Attribution Similar to the procedure in Section 3.3.2, we compute the attribution values of the selected models on the test set. Due to computational constraints of IG, and the instability of IXG, we compute attributions using DL (see Section 3.5). We use the mean genotype over all samples as the reference input for DL, and then compute the mean of the absolute value of the test set attribution scores (see Algorithm 1 or 3).

Evaluation We evaluate the proposed approach by comparing with GWAS analyses adjusted for age (field #21022), sex (field #31), and the first 10 ethnicity principal components as provided by the UK Biobank. We compare the negative p-values obtained from this GWAS against the mean absolute value of the test set attribution scores computed using DL.

5.4 Results

We first describe the Single-task attribution and model performance results for both traits before moving on to the Multi-task setting.

5.4.1 Single-task

Table 5.2 and 5.3 report the architectures of the best performing models on the Single-task prediction of Diabetes and HbA1c respectively. Although the likelihood of the target trait on the test set is 76%, the PR AUC is extremely low, likely owing to the high imbalance in cases vs controls (20 times more controls versus cases). Curiously, as observed in Table 5.3, a similar disconnect between the test prediction error (MSE, Equation 3.4) and EV is observed for the HbA1c prediction task. Here, both the test set MSE and EV are very low. Figure 5.3 compares the histogram of true and predicted values of HbA1c on the test set for this model. The figure shows that the model makes predictions around the mean value accurately but quickly fails to generalize outside of this region.

Table 5.2: Model performance and architecture of the best model on the Diabetes classification task.

Architecture	λ	$\Pi_i P(y_i \mid x_i; \hat{\theta})$	PR AUC	ROC AUC
64,128	0.1	0.76	0.127	0.598

Table 5.3: Model performance and architecture of the best model on the HbA1c regression task.

Architecture	λ	MSE	EV
64,128	0.1	0.020	0.056

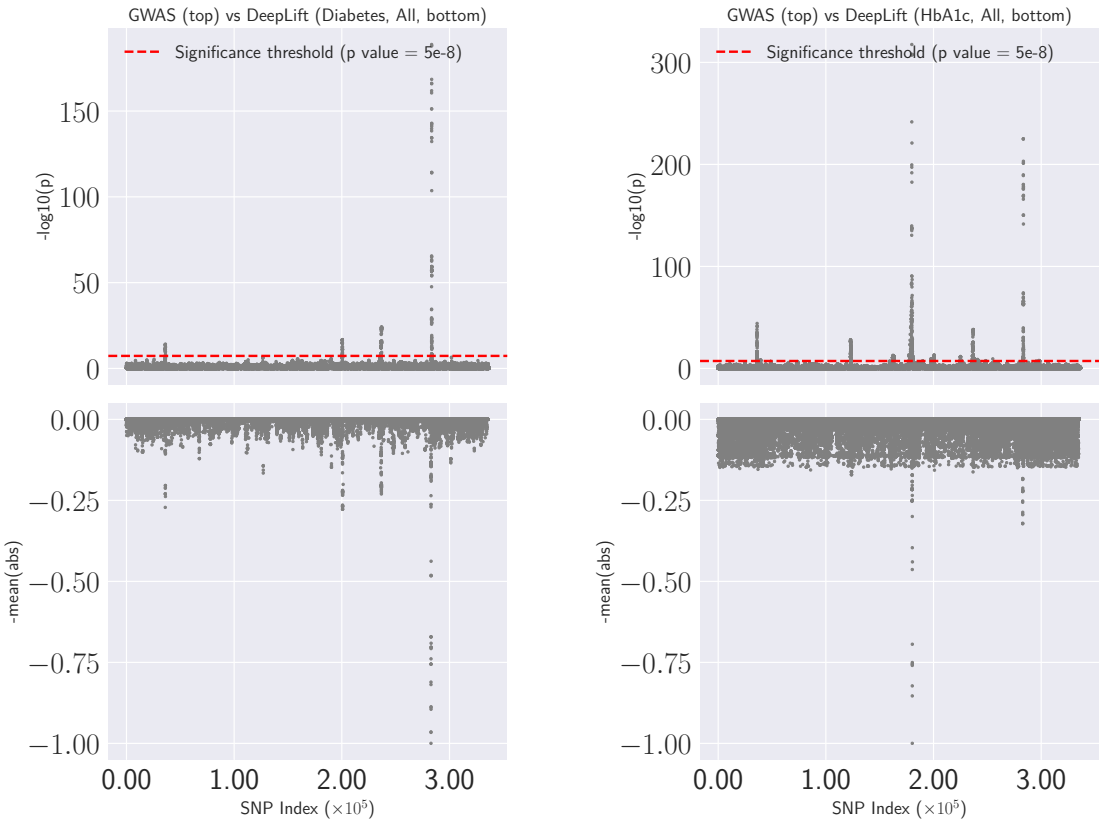

Figure 5.2: Miami plot of a conventional GWAS against the mean absolute DeepLIFT scores on Diabetes (left) and HbA1c (right). The best performing Diabates model had 64 by 128 hidden units, and the best HbA1C model had 64 by 128 units. The L1 regularization parameter λ was 0.1 for both models. The attribution scores were averaged over scores from the 5 seeds for each selected model.

Despite this, the Miami plot in Figure 5.2 shows that for both traits, the pattern of signal obtained via our pipeline closely matches the pattern obtained by a conventional GWAS. For example, the topmost peaks in both halves of both Miami plots occur at the same index. In fact, for Diabetes, all significant peaks can be matched with the peaks obtained via the summarized attribution scores. For HbA1c , it appears that the 3 smaller peaks at indices 50000, 100000, and at 260000 are missing. This is in contrast to our findings about the attribution accuracy of the single-task pipeline on the regression task from Chapter 3. However, in this case, the prediction performance of the model is quite low.

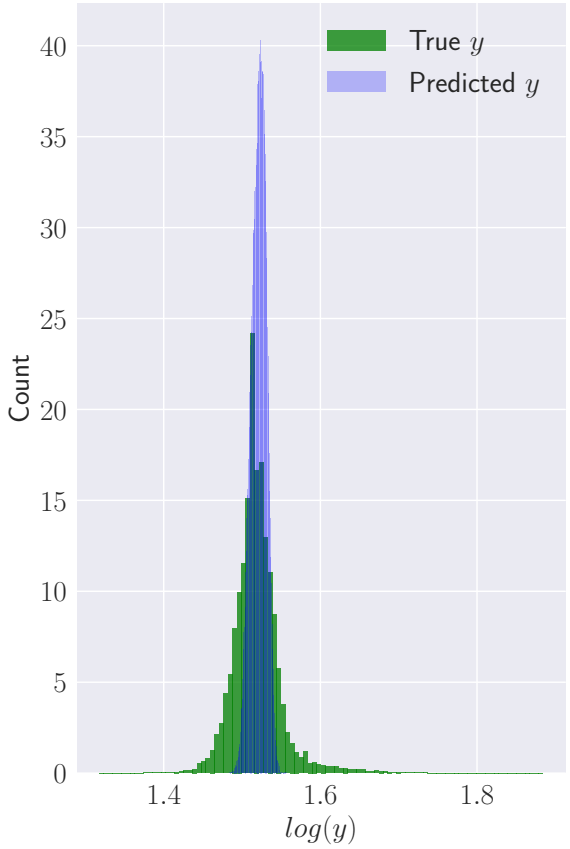

Figure 5.3: A histogram of the predictions made by the best HbA1c model (light blue) from Table 5.3 on the test set and overlayed on the histogram of the true output values (dark green)

5.4.2 Multitask

Tables 5.4 and 5.5 report the architectures of the best performing models on the Multi-task prediction of Diabetes (with $\lambda_1 = 1$ and $\lambda_2 = 0.1$) and HbA1c (with $\lambda_1 = 0.1$ and $\lambda_2 = 1$) respectively. The model performance on both tasks is very similar to the single-task case, with the likelihood of Diabetes on the test set being 80%, and the test set mean prediction error of HbA1c being the same at 0.020. The PR, ROC AUC, or *EV* prediction metrics have not changed either. Thus, in contrast to the improvement that we observed going from the Single-task to the Multi-task case for simulated datasets, we do not see a benefit in prediction performance here. However, the Miami plot in Figure 5.4 shows that again for both traits, the pattern

of signals closely match their conventional GWAS counterparts. There are a few changes compared to the pattern in Figure 5.2: for Diabetes, the signal close to index 250000 is missing, while for HbA1c , the signal near index 125000 is slightly more prominent. Additionally, there seems to be a signal at approximately index 100,000 in the right half of the Miami plot of Figure 5.4 that is missing in the corresponding half of Figure 5.1. This is probably spurious. The presence of spurious peaks and the absence of known peaks can potentailly be fixed by taking the intersection (to reduce the false positive rate of loci captured by conventional GWAS) or union (to increase true positive rate of loci captured by conventional GWAS) of the peaks of top performing models.

Table 5.4: Model performance and architecture of the best multitask model trained with $\lambda_1 = 1$ and $\lambda_2 = 0.1$ on the Diabetes prediction task

Architecture	λ	$\Pi_i P(y_i \mid x_i; \hat{\theta})$	PR AUC	ROC AUC
64,128	0.1	0.80	0.115	0.596

Table 5.5: Model performance and architecture of the best multitask model trained with $\lambda_1 = 0.1$ and $\lambda_2 = 1$ on the HbA1c prediction task

Architecture	λ	MSE	EV
32,128	0.1	0.020	0.059

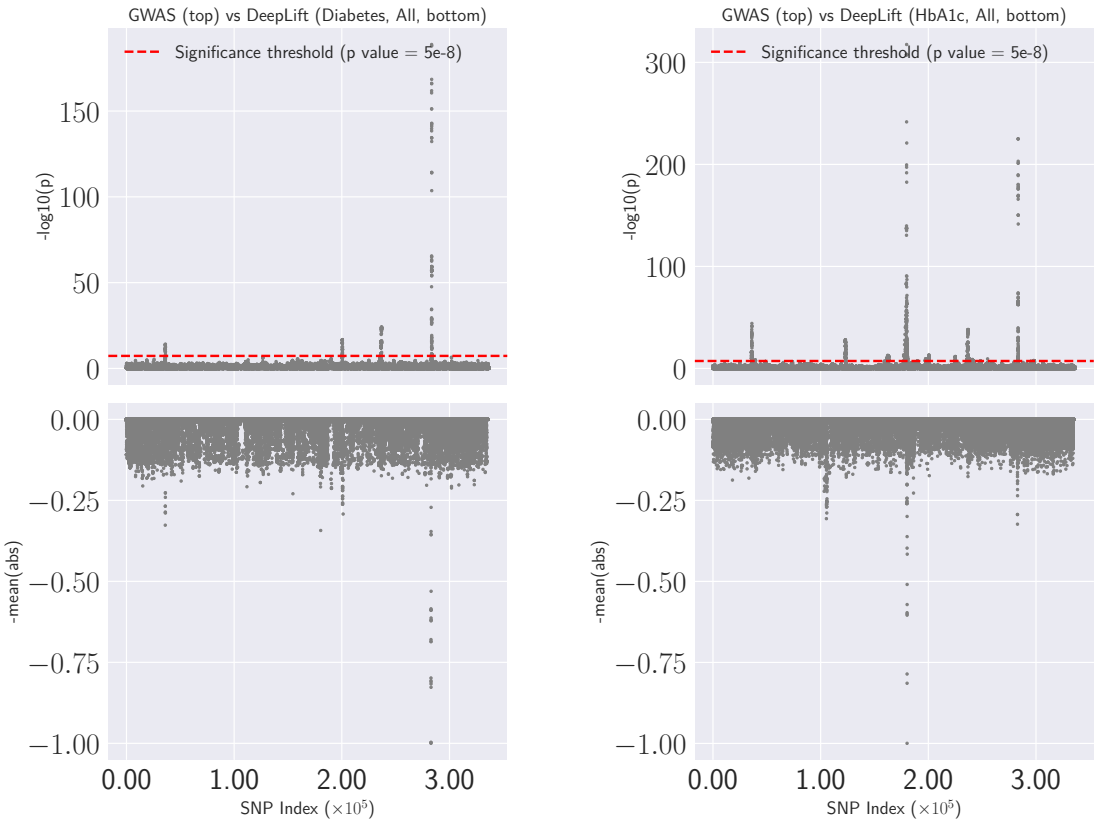

Figure 5.4: Miami plot of a conventional GWAS against the mean absolute DeepLIFT scores on Diabetes (left) and HbA1c (right). The best performing Diabates model had 64 by 128 hidden units, and the best HbA1C model had 32 by 128 units. The L1 regularization parameter λ was 0.1 for both models. The attribution scores were averaged over scores from the 5 seeds for each selected model.

5.5 Conclusion

In this chapter, we sought to compare the single-task and multi-task GWAS pipelines from Chapters 3 and 4 on the large genetic dataset contained in the UK Biobank. We ran experiments to predict Diabetes and HbA1c in both single and multi-task settings, and compared feature importances computed using DeepLIFT against the negative p-values of a conventional GWAS conducted on the same dataset.

In the single-task setting, we found that the prediction performance in each setting was very low, especially in terms of PR AUC for Diabetes prediction, and *EV* for HbA1c . Probably as a result, the prediction performance in the multi-task setting

did not improve much at all. Despite this, we found that the best performing model for Diabetes prediction has attribution scores that closely align with all of the major peaks obtained from conventional GWAS. In contrast, for HbA1c, the two topmost peaks were clearly identifiable but a few minor peaks were not. For Diabetes, we found that one out of the three smaller peaks identified in the single-task setting disappeared in the multi-task setting. Furthermore, there was no clear increase in the number of GWAS significant loci that were identified for HbA1c. This is in contrast to the improvement in results observed in the simulated multi-task setting (Chapter 4) over the single-task setting (Chapter 3). However, given the low prediction performance of all models trained on real-world data, it is difficult to compare the utility of the multi-task setting to the single-task setting. Regardless of their comparative performance, our results clearly show that the models were looking at the right loci for their predictions. Furthermore, their results can be combined by either taking their union to help increase coverage or by taking their intersection, which should reduce the chance for false positives. This leads us to conclude that deep learning models trained on large, real-world genomic datasets can be interpreted to confirm known genomic signals but more work needs to be done in improving their prediction accuracy. Thus, future work on the interpretability of deep models for large-scale GWAS should rely on models with high prediction performance before different techniques are analyzed.

Key points to consider when designing a machine learning pipeline for this purpose are the difference in dimensions of the input features compared to the output, the imbalance between cases and controls, and the long range interactions between features that may be crucial to identify SNP-SNP interactions that could improve prediction accuracy. In this chapter, we considered the first two by incorporating an L_1 penalty term to encourage sparsity and oversampling the cases over the controls. However our choice of architecture does not take into account long-range interactions between inputs. This can be better modelled by self-attention based architectures such as

Transformers (Vaswani et al. 2017) which have recently become immensely popular for Natural Language Processing (NLP) tasks such as text generation or machine translation, where performance relies on the ability of the model to model long-range interactions.

6

Conclusion

The most common approach to discover SNP to phenotype associations is to model phenotypes using a single SNP at a time. Furthermore, common approaches that try to capture any interaction use linear models and thus completely ignore any complex (non-linear) interactions between them. Modelling these interactions may aid the discovery of variants that are indirectly causal for a disease or trait. With the broader goal of using Deep Neural Networks to model such interactions in mind, this thesis focussed on incorporating Deep Neural Networks into a GWAS pipeline that is accurate, reliable, and as scalable as conventional GWAS on modern genomic datasets. In this Chapter, we'll first discuss how we've addressed the objectives that we listed in the Introduction (Chapter 1), including the conclusions that we can reasonably make with regards to each objective. We then summarize the key findings and contributions of this study. Finally, we note key limitations and topics for future work.

Our first objective (Objective 1) was to investigate how we could interpret a neural network to measure the importance of its inputs. We started by covering related literature on the interpretability of deep neural networks, and found that feature-importance techniques could be used to assess the relative importance of different parts of the input, to a trained model's predictions. In Chapter 3, we proposed a pipeline that applies and interprets deep feedforward models to calculate importance scores for each input SNP. We found that on a simulated dataset, if a model can

accurately predict the target trait, our approach correctly identifies and ranks at least half of all causal SNPs. Thus, importance scores generated by gradient-based feature importance techniques applied on deep neural networks can be used to accurately identify and rank causal SNPs in order of their known causal coefficients on simulated datasets. Thus, in Chapter 3, we accomplished our first objective by showing that we could accurately and reliably assess the relative importance of input SNPs to a model's prediction by using certain feature importance techniques that could be applied directly to a trained model.

In Chapters 3 and 4, we also accomplished our second objective (Objective 2), which was to devise a GWAS pipeline that incorporated Deep Neural Networks. Our investigation of feature importance techniques used to interpret deep models for GWAS went hand in hand with the design of our pipeline in Algorithm 1. We proposed a methodology to quantitatively evaluate deep network based pipelines for GWAS on a simulated dataset, which allowed us to compare different versions of the proposed pipeline. We concluded that pipelines using Integrated Gradients and DeepLIFT should be preferred over GradInput since their results were more consistent for both classification and regression tasks, over a wide range of prediction models. Furthermore, in Chapter 4, we modified our pipeline to Algorithm 3, which uses multi-task learning on related traits to improve coverage and ranking of known causal SNPs. We showed that by favouring classification over regression, we could improve the number of correctly identified causal SNPs of the binary trait, or by favouring regression over classification we could improve the ranking accuracy for the causal SNPs of the quantitative trait.

Our final objective (Objective 3) was to compare our pipline's accuracy, reliability and scalability to a real-world GWAS. In Chapter 5, we applied our pipelines from Chapters 3 and 4 to a large and complex genomic dataset, the genomic data in the UK Biobank. We predicted two related and clinically important traits, the chronic disease Type-2 Diabetes and the quantitative trait HbA1c. Since we didn't know the true

causal effects of variants on traits in the real-world, we first conducted an association study using linear models in order to have a few "true" SNP-trait associations to compare to. We then applied the pipelines developed in Chapters 3 and 4 on the same dataset and showed that the most significant peaks found using the conventional analysis are clearly distinguishable using our pipelines as well. In order to ensure the robustness of our results, all experiments were repeated over five differrent seeds and averaged to produce the final result. Although we did see some reduction in coverage while using the multi-task pipeline, it was hard to conclude about its utility relative to the single-task pipeline due to the low prediction performance of our models on both Diabetes and HbA1c prediction. We concluded that our pipelines can be relied upon to clearly distinguish most important genomic signals of a target trait from a large genomic dataset. Furthermore, we could improve coverage by taking the union of the single and multi-task pipelines if they consistently showed different signals.

6.1 Summary of contributions

Overall, this thesis conducted the first empirical analysis of the strengths and weaknesses of prevailing deep learning feature importance techniques in the context of GWAS. We proposed a GWAS pipeline using these techniques on deep feedforward neural networks, and demonstrated its applicability to modern single-trait GWAS, and multi-trait GWAS. We showed that on simulated data, multi-task learning could improve the identification accuracy and ranking of causal SNPs compared to the single-task setting. However, we also found that the importance scores generated from gradient-based techniques are prone to leakage but this leakage can be mitigated by reducing the number of layers that are shared between the tasks and by using deeper models. Finally, our approach is potentially faster and computationally lighter since only a single model needs to be trained and analyzed per trait, and in the multi-task setting, it can model several related traits together, which is biologically

more realistic for multi-trait analysis, while also being faster since only one model needs to be interpreted for each trait in the study.

6.2 Limitations

While we're excited by the possibility of applying deep neural networks at scale for GWAS, some limitations would still need to be addressed for our approach to be practical:

Model prediction performance Although this work focusses mainly on model interpretability, the low model performance on each task of Chapter 5 is an important limitation of our study. Trust in the model's predictions is necessary to build trust in our approach. After all, the model has to be "right" for practictioners to use it for scientific discovery. The low prediction accuracy of the model on the UK Biobank data exhibits the challenges of training deep models on large, unbalanced datasets. These include the class imbalance between cases and controls, the difference in dimensions of the single-dimensional output trait and the several thousand dimensional input SNPs, as well as the relatively few number of samples compared to the dimensionality of the input. There are several techniques that could be useful at tackling some of these challenges. We cover them as part of the Future Work section since some approaches might be particularly effective when combined with recently novel methods of model training (e.g. self-supervised representation learning).

Simulation data for pipeline benchmarking This study adopted a model-based probabilistic genotype and phenotype simulation procedure from Hao, Song, and Storey 2016, without a focus on modelling realistic patterns of Linkage Disequlibrium found in real-world genomic datasets. This can be solved by resampling based approaches (Wright et al. 2007; Su, Marchini, and Donnelly 2011), where samples

of genotypes from real-world biobanks are combined to generate the samples of a simulation study, all the while retaining the original LD patterns.

Scaling multi-trait prediction using multi-task learning Multi-trait GWAS studies deal with scores of related traits simultaneously, with more complex relationships between the traits and the SNPs. This could exacerbate the leakage issue we identified in Chapter 4.

Additionally, our simulations of the binary and quantitative phenotypes in the multi-task setting also assume a homogenous effect of the shared causal SNPs for both traits. For e.g. it is possible for the variants of a portion of the shared causal SNPs to increase the likelihood of the binary trait, while having the opposite effect on the quantitative trait. This effect heterogeneity could also be purely in terms of the effect magnitude (Porter and O'Reilly 2017). Future analysis of the multi-task pipeline should explore performance across a variety of genetic architectures, including varying the number of shared SNPs and adding effect heterogeneity, or adding indirect effects by simulating a trait as a function of another trait.

True genome-wide Although our results certainly show that our approach is scalable to sizes comparable to modern genomic biobanks like the UK Biobank, they are not truly genome-wide because the variants included in the study are all from Chromosome 10. This reduced the number of variants that could be incorporated in our real-world experiments. However, our approach can be easily scaled to variants from multiple chromosomes by training a separate copy of the same model on each chromosome. This parallelizes our procedure across each chromosome.

Group sparsity prior We added an L1 penalty to the first layer to encourage sparsity and force the model to pick a subset of the SNPs. However, since the L1 penalty is applied without grouping together the weights of a given SNP, it will likely fail to force all input weights of a non-causal SNP to 0. This means that the model will

use all the SNPs to some degree. This group specific sparsity penalty can be applied using the group lasso (Yuan and Lin 2006), with a group for each SNP containing the set of weights connecting all hidden units to a single SNP.

6.3 Future Directions

Our results motivate lots of interesting directions for future study:

Perturbation-based interpretability techniques We focussed mostly on post-hoc, gradient-based feature importance techniques to interpret deep neural networks. We did not include perturbation based interpretability techniques due to the computational cost of perturbing such a large number of inputs (S. M. Lundberg and S.-I. Lee 2017). But theoretically, such techniques should be more accurate since they force the model to make predictions without a certain input. However, care needs to be in making sure that the perturbations remain inside the distribution of inputs that the model has been trained on. The only way to be sure of this is to retrain the model on the dataset with the target feature removed from each input (Hooker et al. 2019).

Novel architectures for prediction The field of natural language processing has given rise to architectures that are good at contending with interactions between elements spanning several words or sentences. Most popular today are Transformers (Vaswani et al. 2017), which have proven to be particularly effective at Natural Language Processing (NLP) tasks like text generation or machine translation. Compared to purely Feedforward models, Transformer based architectures are inherently better suited for prediction tasks that take genetic sequences as input.

Furthermore, a key component of the Transformer-based model architectures is attention (Vaswani et al. 2017; Bahdanau, K. Cho, and Bengio 2015), while allows the model to learn how to route information from various parts of the input to make pre-

dictions. Incorporating self-attention could help in modelling long range interactions that are particularly predictive of the target.

Pre-training and self-supervision Performance of many NLP tasks have been shown to be improved by pre-training the model (Dai and Le 2015; Devlin et al. 2019; J. Howard and Ruder 2018). Pre-training via self-supervised representation learning, wherein the objective is to reproduce parts of a genetic sequence given other parts, could be effective at learning representations that are distinctive of the combinations of variants that comprise them. A classifier might have an easier time making predictions from a representation that is more distinctive of the set of variants that produced it.

Evaluating causal SNPs While the true causal SNPs are not known in a real-world GWAS, our approach can be used to reduce the set of SNPs that need to be tested using a conventional GWAS as well as to generate hypothesis of combinations of SNPs that are causal together. These combinations can be tested using a joint linear model.

7

Appendix

7.1 Single Task Prediction

Table 7.1: Negative controls: Model performance and attribution accuracy of untrained models on the classification tasks

		Top-K			τ			τ_β		
a	$\Pi_i P(y_i\|x_i;\hat{\theta})$	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.00	0.00	0.00	0.00	0.01	0.01	-0.02	0.00	0.01	-0.03
0.10	0.00	0.00	0.00	0.00	0.10	0.10	0.16	0.12	0.12	0.16
0.50	0.00	0.00	0.00	0.00	-0.20	-0.21	-0.13	-0.21	-0.22	-0.14
1.00	0.00	0.00	0.00	0.00	-0.27	-0.28	-0.38	-0.20	-0.21	-0.27

Table 7.2: Standard deviation in model performance and attribution accuracy of the top 10% models on the classification datasets.

		Top-K			τ			τ_β		
a	$\Pi_i P(y_i\|x_i;\hat{\theta})$	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.06	0.80	0.80	0.95	0.16	0.16	0.10	0.11	0.11	0.07
0.10	0.06	0.88	0.89	1.05	0.12	0.12	0.07	0.10	0.09	0.06
0.50	0.07	1.39	1.39	1.52	0.11	0.12	0.07	0.10	0.11	0.07
1.00	0.08	1.19	1.17	1.13	0.11	0.11	0.09	0.09	0.09	0.07

Table 7.3: Standard deviation in model performance and attribution accuracy of the top 10% models on the regression datasets.

		Top-K			τ			τ_β		
a	EV	DL	IG	IXG	DL	IG	IXG	DL	IG	IXG
0.01	0.028	0.0	0.0	0.000	0.073	0.078	0.054	0.064	0.066	0.038
0.10	0.031	0.0	0.0	0.000	0.018	0.013	0.042	0.017	0.006	0.099
0.50	0.038	0.0	0.0	0.000	0.055	0.053	0.026	0.032	0.025	0.083
1.00	0.033	0.0	0.0	0.149	0.024	0.023	0.028	0.104	0.040	0.023

7.2 Multi-task Prediction

Table 7.4: Attribution performance of the best 2-layer model using DeepLIFT for both traits (binary on the left and quantitative on the right), with $\lambda_2 = 1$ and $\lambda_1 = 1$ in loss equation 4.1

a	Top-K	τ	τ_β
0.01	4.0, 7.0	0.60, 0.96	0.65, 0.94
0.10	5.0, 7.0	0.60, 0.91	0.67, 0.92
0.50	5.0, 9.0	0.07, 0.96	0.11, 0.98
1.00	4.0, 8.0	0.47, 0.73	0.56, 0.77

Table 7.5: Attribution performance of the best 2-layer model using GradInput for both traits (binary on the left and quantitative on the right), with $\lambda_2 = 1$ and $\lambda_1 = 1$ in loss equation 4.1

a	Top-K	τ	τ_β
0.01	5.0, 7.0	0.56, 0.82	0.63, 0.80
0.10	5.0, 7.0	0.60, 0.78	0.68, 0.74
0.50	5.0, 9.0	0.11, 0.73	0.20, 0.86
1.00	4.0, 8.0	0.47, 0.69	0.56, 0.62

Bibliography

Adadi, A. and M. Berrada (2018). "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160.

Ahlqvist, Emma et al. (Mar. 2015). "The genetics of diabetic complications". In: *Nature Reviews Nephrology* 11.5, pp. 277–287.

Alaa, A. et al. (2019). "Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 UK Biobank participants". In: *PloS one*.

Altshuler, David et al. (Sept. 2000). "The common PPAR Pro12Ala polymorphism is associated with decreased risk of type 2 diabetes". In: *Nature Genetics* 26.1, pp. 76–80.

Alvarez-Melis, David and T. Jaakkola (2018). "On the Robustness of Interpretability Methods". In: *ArXiv* abs/1806.08049.

Ancona, Marco et al. (2018). "Towards better understanding of gradient-based attribution methods for Deep Neural Networks". In: *International Conference on Learning Representations*.

Arora, Sanjeev et al. (2019). "On Exact Computation with an Infinitely Wide Neural Net". In: *NeurIPS*.

Auton (Sept. 2015). "A global reference for human genetic variation". In: *Nature* 526.7571, pp. 68–74.

Bach, Sebastian et al. (July 2015). "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10.7. Ed. by Oscar Deniz Suarez, e0130140.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Bahdanau, Dzmitry, J. Chorowski, et al. (2016). "End-to-end attention-based large vocabulary speech recognition". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4945–4949.

Barredo Arrieta, Alejandro et al. (June 2020). "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58, pp. 82–115.

Bellot, Pau, Gustavo de los Campos, and Miguel Pérez-Enciso (Aug. 2018). "Can Deep Learning Improve Genomic Prediction of Complex Human Traits?" In: *Genetics* 210.3, pp. 809–819.

Ben-Tal, Aharon et al. (2013). "Robust Solutions of Optimization Problems Affected by Uncertain Probabilities". In: *Management Science* 59.2, pp. 341–357.

Bengio, Yoshua et al. (2000). "A Neural Probabilistic Language Model". In: *J. Mach. Learn. Res.*

Bhattacharjee, Samsiddhi et al. (May 2012). "A Subset-Based Approach Improves Power and Interpretation for the Combined Analysis of Genetic Association Studies of Heterogeneous Traits". In: *The American Journal of Human Genetics* 90.5, pp. 821–835.

Bianconi, Eva et al. (July 2013). "An estimation of the number of cells in the human body". In: *Annals of Human Biology* 40.6, pp. 463–471.

Bien, Jacob and Robert Tibshirani (Dec. 2011). "Prototype selection for interpretable classification". In: *The Annals of Applied Statistics* 5.4, pp. 2403–2424.

Blauwendraat, C., M. Nalls, and A. Singleton (2020). “The genetic architecture of Parkinson’s disease”. In: *The Lancet Neurology*.

Bolormaa, Sunduimijid et al. (Mar. 2014). “A Multi-Trait, Meta-analysis for Detecting Pleiotropic Polymorphisms for Stature, Fatness and Reproduction in Beef Cattle”. In: *PLoS Genetics* 10.3. Ed. by Jonathan Flint, e1004198.

Breiman, L. and J. Friedman (1997). “Predicting Multivariate Responses in Multiple Linear Regression”. In: *Journal of The Royal Statistical Society Series B-statistical Methodology* 59, pp. 3–54.

Brown, Tom B. et al. (2020). *Language Models are Few-Shot Learners*.

Buniello, Annalisa et al. (Jan. 2019). “The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019”. In: *Nucleic Acids Research* 47.D1, pp. D1005–D1012.

Camburu, Oana-Maria et al. (2018). “e-SNLI: Natural Language Inference with Natural Language Explanations”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc.

Caruana, R. (1993). “Multitask Learning: A Knowledge-Based Source of Inductive Bias”. In: *ICML*.

Caruana, R., H. Kangarloo, et al. (1999). “Case-based explanation of non-case-based learning methods”. In: *Proc AMIA Symp*, pp. 212–215.

Caruana, Rich, Shumeet Baluja, and Tom Mitchell (1995). “Using the Future to "Sort out" the Present: Rankprop and Multitask Learning for Medical Risk Evaluation”. In: *Proceedings of the 8th International Conference on Neural Information Processing Systems*. NIPS’95. Denver, Colorado: MIT Press, pp. 959–965.

Chavali, Sreenivas et al. (June 2010). “Network properties of human disease genes with pleiotropic effects”. In: *BMC Systems Biology* 4.1.

Chen, Chaofan et al. (2019). “This Looks Like That: Deep Learning for Interpretable Image Recognition”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc.

Chen, Xiangli et al. (May 2016). "Robust Covariate Shift Regression". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Vol. 51. Proceedings of Machine Learning Research. Cadiz, Spain: PMLR, pp. 1270–1279.

Ching, Travers et al. (Apr. 2018). "Opportunities and obstacles for deep learning in biology and medicine". In: *Journal of The Royal Society Interface* 15.141, p. 20170387.

Cho, N.H. et al. (Apr. 2018). "IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045". In: *Diabetes Research and Clinical Practice* 138, pp. 271–281.

Chorowski, Jan and Jacek M. Zurada (Jan. 2015). "Learning Understandable Neural Networks With Nonnegative Weight Constraints". In: *IEEE Transactions on Neural Networks and Learning Systems* 26.1, pp. 62–69.

Chouldechova, Alexandra (June 2017). "Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments". In: *Big Data* 5.2, pp. 153–163.

Chu, Lingyang et al. (2018). "Exact and Consistent Interpretation for Piecewise Linear Neural Networks: A Closed Form Solution". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*.

Clarke, Geraldine M et al. (Feb. 2011). "Basic statistical analysis in genetic case-control studies". In: *Nature Protocols* 6.2, pp. 121–133.

Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing". In: *Proceedings of the 25th international conference on Machine learning - ICML '08*. ACM Press.

Dai, Andrew M and Quoc V Le (2015). "Semi-supervised Sequence Learning". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc.

Deloukas, Panos et al. (Dec. 2012). "Large-scale association analysis identifies new risk loci for coronary artery disease". In: *Nature Genetics* 45.1, pp. 25–33.

Deng, J. et al. (2009). "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*.

Deng, Li, Geoffrey Hinton, and Brian Kingsbury (2013). "New types of deep neural network learning for speech recognition and related applications: an overview". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8599–8603.

Devlin, J. et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *NAACL-HLT*.

Doshi-Velez, Finale and Been Kim (2017). "Towards A Rigorous Science of Interpretable Machine Learning". In: *arXiv: Machine Learning*.

Driel, Marc A van et al. (Feb. 2006). "A text-mining analysis of the human phenome". In: *European Journal of Human Genetics* 14.5, pp. 535–542.

Du, Simon S. et al. (2019). "Gradient Descent Provably Optimizes Over-parameterized Neural Networks". In: *International Conference on Learning Representations*.

Eichler, Evan E. et al. (June 2010). "Missing heritability and strategies for finding the underlying causes of complex disease". In: *Nature Reviews Genetics* 11.6, pp. 446–450.

Elliott, Lloyd T. et al. (Oct. 2018). "Genome-wide association studies of brain imaging phenotypes in UK Biobank". In: *Nature* 562.7726, pp. 210–216.

Escalante, Hugo Jair et al., eds. (2018). *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Springer International Publishing.

Fahrmeir, L. et al. (2013). *Regression: Models, Methods and Applications*. Springer Berlin Heidelberg.

Fan, Fenglei, Jinjun Xiong, and Ge Wang (2020). "On Interpretability of Artificial Neural Networks". In: *CoRR* abs/2001.02522.

Ferreira, Manuel A. R. and Shaun M. Purcell (Nov. 2008). "A multivariate test of association". In: *Bioinformatics* 25.1, pp. 132–133.

Fiebrink, Rebecca Anne (2011). "Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance". AAI3445567. PhD thesis. USA.

Forouzanfar, Mohammad H et al. (Oct. 2016). "Global, regional, and national comparative risk assessment of 79 behavioural, environmental and occupational, and metabolic risks or clusters of risks, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015". In: *The Lancet* 388.10053, pp. 1659–1724.

Galesloot, Tessel E. et al. (Apr. 2014). "A Comparison of Multivariate Genome-Wide Association Methods". In: *PLoS ONE* 9.4. Ed. by Yurii S. Aulchenko, e95923.

Galtung, Johan (1969). "Violence, Peace, and Peace Research". In: *Journal of Peace Research* 6.3, pp. 167–191.

Gilpin, Leilani H. et al. (2019). *Explaining Explanations: An Overview of Interpretability of Machine Learning*.

Girshick, Ross (2015). "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Grant, Struan F. A. et al. (Mar. 2006). "Variant of transcription factor 7-like 2 (TCF7L2) gene confers risk of type 2 diabetes". In: *Nature Genetics* 38.3, pp. 320–323.

Graves, A., Abdel-rahman Mohamed, and Geoffrey E. Hinton (2013). "Speech recognition with deep recurrent neural networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649.

Gunning, David et al. (Dec. 2019). "XAI—Explainable artificial intelligence". In: *Science Robotics* 4.37, eaay7120.

Hao, W., M. Song, and J. D. Storey (Mar. 2016). "Probabilistic models of genetic variation in structured populations applied to global human studies". In: *Bioinformatics* 32.5, pp. 713–721.

Harst, P. van der and N. Verweij (2018). "Identification of 64 Novel Genetic Loci Provides an Expanded View on the Genetic Architecture of Coronary Artery Disease". In: *Circulation research*.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning*. Springer New York.

He, Dan, David Kuhn, and Laxmi Parida (June 2016). "Novel applications of multitask learning and multiple output regression to multiple genetic trait prediction". In: *Bioinformatics* 32.12, pp. i37–i43.

He, Kaiming et al. (2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean (2015). *Distilling the Knowledge in a Neural Network*.

Hooker, Sara et al. (2019). "A Benchmark for Interpretability Methods in Deep Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc.

Hotelling, H. (Dec. 1936). "RELATIONS BETWEEN TWO SETS OF VARIATES". In: *Biometrika* 28.3-4, pp. 321–377.

Howard, David Mark et al. (2019). "Genome-wide meta-analysis of depression identifies 102 independent variants and highlights the importance of the prefrontal brain regions". In: *Nature Neuroscience*.

Howard, Jeremy and Sebastian Ruder (2018). "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

"Initial sequencing and analysis of the human genome" (Feb. 2001). In: *Nature* 409.6822, pp. 860–921.

Jacot, Arthur, Franck Gabriel, and Clement Hongler (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc.

Jansen, Iris E. et al. (Jan. 2019). "Genome-wide meta-analysis identifies new loci and functional pathways influencing Alzheimer's disease risk". In: *Nature Genetics* 51.3, pp. 404–413.

Jolliffe, I. T. (1986). *Principal Component Analysis*. Springer New York.

Kaplan, Jared et al. (2020). *Scaling Laws for Neural Language Models*.

Kendall, M. G. (June 1938). "A NEW MEASURE OF RANK CORRELATION". In: *Biometrika* 30.1-2, pp. 81–93.

Kim, Been (2015a). "Interactive and Interpretable Machine Learning Models for Human Machine Collaboration". Ph.D. Thesis. Cambridge, MA: MIT.

— (Jan. 2015b). "Interactive and interpretable machine learning models for human machine collaboration". In:

Kim, Been et al. (July 2018). "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2668–2677.

Kolodner, Janet L. (1992). "An introduction to case-based reasoning". In: *Artificial Intelligence Review* 6.1, pp. 3–34.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., pp. 1097–1105.

LeCun, Y. et al. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4, pp. 541–551.

Lee, Jaehoon et al. (2018). "Deep Neural Networks as Gaussian Processes". In: *International Conference on Learning Representations*.

Li, Bing and G. Jogesh Babu (2019). *A Graduate Course on Statistical Inference*. Springer New York.

Li, Xiao-Li and Bing Liu (2005). "Learning from Positive and Unlabeled Examples with Different Data Distributions". In: *Machine Learning: ECML 2005*. Springer Berlin Heidelberg, pp. 218–229.

Li, Yan et al. (Aug. 2016). "A Multi-Task Learning Formulation for Survival Analysis". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Libbrecht, Maxwell W. and William Stafford Noble (May 2015). "Machine learning applications in genetics and genomics". In: *Nature Reviews Genetics* 16.6, pp. 321–332.

Lipton, Zachary C. (June 2018). "The Mythos of Model Interpretability". In: *Queue* 16.3, pp. 31–57.

Liu, Hui, Qingyu Yin, and William Yang Wang (July 2019). "Towards Explainable NLP: A Generative Explanation Framework for Text Classification". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 5570–5581.

Locatello, Francesco et al. (June 2019). "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 4114–4124.

Lundberg, Scott M. and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural In-*

formation Processing Systems. NIPS'17. Long Beach, California, USA: Curran Associates Inc., pp. 4768–4777.

Manolio, Teri A. (Dec. 2018). "UK Biobank debuts as a powerful resource for genomic research". In: *Nature Medicine* 24.12, pp. 1792–1794.

Marees, Andries T. et al. (Feb. 2018). "A tutorial on conducting genome-wide association studies: Quality control and statistical analysis". In: *International Journal of Methods in Psychiatric Research* 27.2, e1608.

Marioni, Riccardo E. et al. (May 2018). "GWAS on family history of Alzheimer's disease". In: *Translational Psychiatry* 8.1, pp. 1–7.

Mikolov, Tomas et al. (2010). "Recurrent neural network based language model". In: *INTERSPEECH*.

Miller, Rupert G. (1981). *Simultaneous Statistical Inference*. Springer New York.

Miller, Tim (2017). "Explanation in Artificial Intelligence: Insights from the Social Sciences". In: *CoRR* abs/1706.07269.

Montaez, C. A. C. et al. (2018). "Deep Learning Classification of Polygenic Obesity using Genome Wide Association Study SNPs". In: *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8.

Mordelet, Fantine and Jean-Philippe Vert (Oct. 2011). "ProDiGe: Prioritization Of Disease Genes with multitask machine learning from positive and unlabeled examples". In: *BMC Bioinformatics* 12.1.

Morris, John A. et al. (Feb. 2019). "An atlas of genetic influences on osteoporosis in humans and mice". In: *Nature Genetics* 51.2, pp. 258–266.

Nair, Vinod and Geoffrey E. Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, pp. 807–814.

Nicholls, Hannah L. et al. (Apr. 2020). "Reaching the End-Game for GWAS: Machine Learning Approaches for the Prioritization of Complex Disease Loci". In: *Frontiers in Genetics* 11.

O'Reilly, Paul F. et al. (May 2012). "MultiPhen: Joint Model of Multiple Phenotypes Can Increase Discovery in GWAS". In: *PLoS ONE* 7.5. Ed. by Stacey Cherny, e34861.

Oster, Julien et al. (Mar. 2020). "Identification of patients with atrial fibrillation: a big data exploratory analysis of the UK Biobank". In: *Physiological Measurement* 41.2, p. 025001.

Panagiotou, Orestis A. et al. (Aug. 2013). "The Power of Meta-Analysis in Genome-Wide Association Studies". In: *Annual Review of Genomics and Human Genetics* 14.1, pp. 441–465.

Papernot, Nicolas and Patrick D. McDaniel (2018). "Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning". In: *CoRR* abs/1803.04765.

Pe'er, Itsik et al. (May 2008). "Estimation of the multiple testing burden for genomewide association studies of nearly all common variants". In: *Genetic Epidemiology* 32.4, pp. 381–385.

Pearson, Thomas A. (Mar. 2008). "How to Interpret a Genome-wide Association Study". In: *JAMA* 299.11, p. 1335.

Peters, Jonas, Dominik Janzing, and Bernhard Schlkopf (2017). *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press.

Policardo, Laura et al. (Nov. 2014). "Effect of diabetes on hospitalization for ischemic stroke and related in-hospital mortality: a study in Tuscany, Italy, over years 2004–2011". In: *Diabetes/Metabolism Research and Reviews* 31.3, pp. 280–286.

Porter, Heather F. and Paul F. O'Reilly (Mar. 2017). "Multivariate simulation framework reveals performance of multi-trait GWAS methods". In: *Scientific Reports* 7.1.

Pulit, Sara L et al. (Jan. 2019). “Meta-analysis of genome-wide association studies for body fat distribution in 694 649 individuals of European ancestry”. In: *Human Molecular Genetics* 28.1, pp. 166–174.

Puniyani, K., S. Kim, and E. P. Xing (June 2010). “Multi-population GWA mapping via multi-task regularized regression”. In: *Bioinformatics* 26.12, pp. i208–i216.

Raji, Inioluwa Deborah et al. (2020). “Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*.

Rasmussen, Carl Edward and Christopher K. I. Williams (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Reed, Eric et al. (Sept. 2015). “A guide to genome-wide association analysis and post-analytic interrogation”. In: *Statistics in Medicine* 34.28, pp. 3769–3792.

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin (2016). “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *CoRR* abs/1602.04938.

Richardson, R., Jason Schultz, and K. Crawford (2019). “Dirty Data, Bad Predictions: How Civil Rights Violations Impact Police Data, Predictive Policing Systems, and Justice”. In:

Robnik-Šikonja, Marko and Marko Bohanec (2018). “Perturbation-Based Explanations of Prediction Models”. In: *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*. Ed. by Jianlong Zhou and Fang Chen. Cham: Springer International Publishing, pp. 159–175.

Romagnoni, Alberto et al. (July 2019). “Comparative performances of machine learning methods for classifying Crohn Disease patients using genome-wide genotyping data”. In: *Scientific Reports* 9.1.

Ronneberger, O., P. Fischer, and T. Brox (2015). “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *MICCAI*.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986a). "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, pp. 318–362.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (Oct. 1986b). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536.

Sandhu, Manjinder S et al. (Feb. 2008). "LDL-cholesterol concentrations: a genome-wide association study". In: *The Lancet* 371.9611, pp. 483–491.

Sarnowski, Chloé et al. (Oct. 2019). "Impact of Rare and Common Genetic Variants on Diabetes Diagnosis by Hemoglobin A1c in Multi-Ancestry Cohorts: The Trans-Omics for Precision Medicine Program". In: *American Journal of Human Genetics* 105.4, pp. 706–718.

Schulz, Marc-Andre et al. (Aug. 2020). "Different scaling of linear models and deep learning in UKBiobank brain images versus machine-learning datasets". In: *Nature Communications* 11.1, p. 4238.

Selvaraju, R. R. et al. (2017). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626.

Senior, Andrew W. et al. (Jan. 2020). "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792, pp. 706–710.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (Aug. 2017). "Learning Important Features Through Propagating Activation Differences". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 3145–3153.

Shrikumar, Avanti, Peyton Greenside, Anna Shcherbina, et al. (2016). "Not Just a Black Box: Learning Important Features Through Propagating Activation Differences". In: *CoRR* abs/1605.01713.

Sinnott-Armstrong, Nasa, Sahin Naqvi, et al. (Feb. 2021). "GWAS of three molecular traits highlights core genes and pathways alongside a highly polygenic background". In: *eLife* 10.

Sinnott-Armstrong, Nasa, Yosuke Tanigawa, et al. (Feb. 2021). "Genetics of 35 blood and urine biomarkers in the UK Biobank". In: *Nature Genetics* 53.2, pp. 185–194.

Sluis, Sophie van der, Danielle Posthuma, and Conor V. Dolan (Jan. 2013). "TATES: Efficient Multivariate Genotype-Phenotype Analysis for Genome-Wide Association Studies". In: *PLoS Genetics* 9.1. Ed. by Nicholas J. Schork, e1003235.

Solovieff, Nadia et al. (June 2013). "Pleiotropy in complex traits: challenges and strategies". In: *Nature Reviews Genetics* 14.7, pp. 483–495.

Sturmfels, Pascal, Scott Lundberg, and Su-In Lee (2020). "Visualizing the Impact of Feature Attribution Baselines". In: *Distill*.

Su, Zhan, Jonathan Marchini, and Peter Donnelly (June 2011). "HAPGEN2: simulation of multiple disease SNPs". In: *Bioinformatics* 27.16, pp. 2304–2305.

Sudlow, C. et al. (Mar. 2015). "UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age". In: *PLoS Med.* 12.3, e1001779.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 3319–3328.

Szegedy, Christian, Sergey Ioffe, and Vincent Vanhoucke (2016). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *CoRR* abs/1602.07261.

Tam, Vivian et al. (May 2019). "Benefits and limitations of genome-wide association studies". In: *Nature Reviews Genetics* 20.8, pp. 467–484.

Taylor, R. (Mar. 2013). "Type 2 Diabetes: Etiology and reversibility". In: *Diabetes Care* 36.4, pp. 1047–1055.

Thung, Kim-Han and Chong-Yaw Wee (Aug. 2018). "A brief review on multi-task learning". In: *Multimedia Tools and Applications* 77.22, pp. 29705–29725.

Tran, Dustin and David M. Blei (2018). "Implicit Causal Models for Genome-wide Association Studies". In: *International Conference on Learning Representations*.

Turley, Patrick et al. (Jan. 2018). "Multi-trait analysis of genome-wide association summary statistics using MTAG". In: *Nature Genetics* 50.2, pp. 229–237.

Udler, Miriam S. et al. (Sept. 2018). "Type 2 diabetes genetic loci informed by multi-trait associations point to disease mechanisms and subtypes: A soft clustering analysis". In: *PLOS Medicine* 15.9. Ed. by Claudia Langenberg, e1002654.

Vaswani, Ashish et al. (2017). "Attention is All You Need". In:

Vigna, Sebastiano (2015). "A Weighted Correlation Index for Rankings with Ties". In: *Proceedings of the 24th International Conference on World Wide Web*. WWW '15. Florence, Italy: International World Wide Web Conferences Steering Committee, pp. 1166–1176.

Visscher, Peter M., Matthew A. Brown, et al. (Jan. 2012). "Five Years of GWAS Discovery". In: *The American Journal of Human Genetics* 90.1, pp. 7–24.

Visscher, Peter M., Naomi R. Wray, et al. (July 2017). "10 Years of GWAS Discovery: Biology, Function, and Translation". In: *The American Journal of Human Genetics* 101.1, pp. 5–22.

Waldmann, Patrik (Dec. 2018). "Approximate Bayesian neural networks in genomic prediction". In: *Genetics Selection Evolution* 50.1.

Wallace, Eric, Shi Feng, and Jordan L. Boyd-Graber (2018). "Interpreting Neural Networks with Nearest Neighbors". In: *BlackboxNLP@EMNLP*, pp. 136–144.

Wang, Y. et al. (2018). "Interpret Neural Networks by Identifying Critical Data Routing Paths". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8906–8914.

Warne, Russell (2014). "A Primer on Multivariate Analysis of Variance (MANOVA) for Behavioral Scientists". In:

Williams, David R. et al. (Feb. 2010). "Race, socioeconomic status, and health: Complexities, ongoing challenges, and research opportunities". In: *Annals of the New York Academy of Sciences* 1186.1, pp. 69–101.

Wood, Andrew R et al. (Oct. 2014). "Defining the role of common variation in the genomic and biological architecture of adult human height". In: *Nature Genetics* 46.11, pp. 1173–1186.

Wright, F. A. et al. (Sept. 2007). "Simulating association studies: a data-based resampling method for candidate regions or whole genome scans". In: *Bioinformatics* 23.19, pp. 2581–2588.

Xu, Bing et al. (2015). "Empirical Evaluation of Rectified Activations in Convolutional Network". In: *CoRR* abs/1505.00853.

Xue, Angli et al. (July 2018). "Genome-wide association analyses identify 143 risk variants and putative regulatory mechanisms for type 2 diabetes". In: *Nature Communications* 9.1.

Yang, Jae Jeong et al. (Apr. 2019). "Association of Diabetes With All-Cause and Cause-Specific Mortality in Asia". In: *JAMA Network Open* 2.4, e192696.

Yengo, Loic et al. (Oct. 2018). "Meta-analysis of genome-wide association studies for height and body mass index in 700000 individuals of European ancestry". In: *Human Molecular Genetics* 27.20, pp. 3641–3649.

Yuan, Ming and Yi Lin (Feb. 2006). "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 49–67.

Zeiler, Matthew D. and Rob Fergus (2014). "Visualizing and Understanding Convolutional Networks". In: *Computer Vision – ECCV 2014*. Springer International Publishing, pp. 818–833.

Zhou, B., A. Khosla, A. Lapedriza, et al. (2016). "Learning Deep Features for Discriminative Localization". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929.

Zhou, Bolei, Aditya Khosla, Àgata Lapedriza, et al. (2015). "Object Detectors Emerge in Deep Scene CNNs". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun.

Zhu, Xiaofeng et al. (Jan. 2015). "Meta-analysis of Correlated Traits via Summary Statistics from GWASs with an Application in Hypertension". In: *The American Journal of Human Genetics* 96.1, pp. 21–36.