Interpreting Deep Neural Networks Through Variable Importance

Jonathan Ish-Horowicz [*] [1] **Dana Udwin** [*] [2] **Seth Flaxman** [1] **Sarah Filippi** [1] **Lorin Crawford** [2]

Abstract

While the success of deep neural networks (DNNs) is well-established across a variety of domains, our ability to explain and interpret these methods is limited. Unlike previously proposed local methods which try to explain particular classification decisions, we focus on global interpretability and ask a universally applicable question: given a trained model, which features are the most important? In the context of neural networks, a feature is rarely important on its own, so our strategy is specifically designed to leverage partial covariance structures and incorporate variable dependence into feature ranking. Our methodological contributions in this paper are two-fold. First, we propose an effect size analogue for DNNs that is appropriate for applications with highly collinear predictors (ubiquitous in computer vision). Second, we extend the recently proposed "RelATive cEntrality" (RATE) measure (Crawford et al., 2019) to the Bayesian deep learning setting. RATE applies an information theoretic criterion to the posterior distribution of effect sizes to assess feature significance. We apply our framework to three broad application areas: computer vision, natural language processing, and social science.

1. Introduction

Due to their high predictive performances, deep neural networks (DNNs) have become increasingly used in many fields including computer vision and natural language processing. Unfortunately, DNNs operate as "black boxes": users often do not have access to the internal workings of the network. As a result, their adoption in many scientific and high-risk decision-making fields has been relatively limited. In the former case, variable selection tasks are often as important as prediction — one particular example being the identification of biomarkers related to the development of a disease. In the latter scenario, it is crucial to ensure that methods deployed for high-risk decision making (e.g. automated medical diagnostics, self-driving cars) do not make predictions based on artifacts or biases in the training data. Therefore, there is both a strong theoretical and practical motivation to increase the interpretability of DNNs and to better characterize the types of relationships they exploit to achieve improved predictive performance.

Despite being an increasingly important concept in machine learning, interpretability lacks a well-established definition in the literature. Such inconsistencies have lead to a lack of consensus on how interpretability should be achieved or evaluated. Variable importance is one possible approach to achieve global interpretability, where the goal is to rank each input feature based on its contributions to predictive accuracy. This is in contrast to local interpretability, which aims to simply provide an explanation behind a specific prediction or group of predictions. In this paper, we follow a more recently proposed definition which refers to interpretability as "the ability to explain or to present in understandable terms to a human" (Doshi-Velez & Kim, 2017). To this end, our main contribution is to address the problem of being given a trained neural network with the desire to identify the subset of predictor variables that are best associated with the modeled response.

Here, we describe an approach to achieve global interpretability for deep neural networks using "RelATive cEntrality" (RATE) (Crawford et al., 2019), a recently-proposed variable importance criterion for (Bayesian) nonlinear regression models. This flexible approach can be used with any network architecture where some notion of uncertainty can be computed over the predictions. The rest of the paper is structured as follows. Section 2 outlines related work on the interpretation of deep neural networks. Section 3 describes the RATE computation within the context for which it was originally proposed (Gaussian process regression). Section 4 contains the main methodological innovations of this paper. Here, we describe a unified framework under which RATE can be applied to deep neural networks and propose novel extensions for variable selection-based tasks. We also derive closed-form expressions for posterior inference. In Section 5, we demonstrate the utility of our method

[*]Equal contribution [1]Imperial College London [2]Brown University School of Public Health. Correspondence to: Seth Flaxman <s.flaxman@imperial.ac.uk>, Sarah Filippi <s.filippi@imperial.ac.uk>, Lorin Crawford <lorin_crawford@brown.edu>.

on three datasets: optical character recognition using the MNIST dataset (LeCun, 1998), sentiment analysis using a large movie review dataset (Maas et al., 2011), and the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) recidivism risk scores (Larson et al., 2016). The first two are popular datasets where deep neural networks have been shown to have high predictive accuracy and, therefore, we illustrate the utility of our approach by identifying the predictor variables that effectively explain these significant performance gains. The third dataset has been used to investigate the fairness of COMPAS, a commercial tool developed to predict the risk of recidivism (or re-offending) of criminal defendants (Angwin et al., 2016). In this application, we use deep learning and RATE to offer insight into the social implications underlying these scores.

2. Related Work

In the absence of a robustly defined metric for interpretability, a large proportion of the current research for DNNs has been limited to applications where methods can be evaluated visually. Advancements in global interpretability have been focused on identifying predictor variables that maximize the activation of each layer within the network (Erhan et al., 2009). These type of approaches can be used to highlight what a particular architecture learns during training. In the context of local interpretability, it can be useful to study the gradient of a network output with respect to its input features. In imaging applications, saliency and sensitivity may be represented as a heat map where each cell depicts this gradient at a corresponding pixel (Simonyan et al., 2013). Recently, improvements to such local interpretability methods have been proposed by: (i) systematically adding noise to the input image (Smilkov et al., 2017) or (ii) using some linear interpolation on the input image according to some user-specified baseline and then averaging the gradient over the newly interpolated set of data (Sundararajan et al., 2017). Alternatively, more sophisticated DNN-specific techniques quantify variable significance by first forwardly measuring how each input variable impacts the accuracy of the predictive function, and then reversely summing these contributions back through the network layers using the estimated weights, observed activations, or approximations based on Taylor series expansions (Bach et al., 2015; Montavon et al., 2017; Shrikumar et al., 2017).

One viable approach for achieving global interpretability is to train more conventional statistical methods to mimic the predictive behavior of a DNN. This imitation model is then retrospectively used to explain the predictions that a DNN would make. For example, using a decision tree (Frosst & Hinton, 2017) or falling rule list (Wang & Rudin, 2015) can yield straightforward characterizations of predictive outcomes. Unfortunately, these simple models can struggle to mimic the accuracy of DNNs effectively. A random forest, on the other hand, is much more capable of matching the predictive power of neural networks. Here, measures of feature selection can be computed by permuting information within the input variables and examining this null effect on test accuracy or Gini impurity (Breiman, 2001). The ability establish variable importance in random forests is a significant reason for their popularity in fields such as the life sciences (Chen et al., 2007) — thus, providing motivation for developing analogous approaches for DNNs.

3. Relevant Background

In this section, we give a brief review on previous results that are relevant to our main methodological innovations. We first describe the concept of the "RelATive cEntrality" (RATE) measure: a method for variable importance proposed within the context of Gaussian process regression. We also detail a direct parallel with Bayesian neural networks. Throughout this section, we assume that we have access to some trained model and that we are able to draw samples from its predictive posterior. Again, this reflects the *post-hoc* objective of finding important and explanatory subsets of variables.

3.1. Effect Size Analogues for Nonlinear Models

The presentation in this section follows a previous description of the effect size analogues which generalizes the concept of coefficients within linear models to include nonlinear regression settings (Crawford et al., 2019). Assume that we have an n-dimensional response vector $\mathbf{y}$ and an $n \times p$ design matrix $\mathbf{X}$ with p covariates. Recall that, for linear models, an effect size is defined as the projection of the response onto the column space of the data:

$$\widehat{\boldsymbol{\beta}} = \text{Proj}(\mathbf{X}, \mathbf{y}). \tag{1}$$

One standard projection operation is $\text{Proj}(\mathbf{X}, \mathbf{y}) = \mathbf{X}^{\dagger}\mathbf{y}$, with $\mathbf{X}^{\dagger}$ being the Moore-Penrose pseudo-inverse. In the case of a full rank design matrix, $\mathbf{X}^{\dagger} = (\mathbf{X}^{\intercal}\mathbf{X})^{-1}\mathbf{X}^{\intercal}$ and leads to the classic ordinary least squares (OLS) regression coefficient estimates.

In the Bayesian nonparametric setting (e.g. GP regression), we consider a learned nonlinear function that has been evaluated on the n-observed samples, where $\mathbb{E}(\mathbf{y} \,|\, \mathbf{X}) = \boldsymbol{f}$. The effect size analogue can then be defined as the result of projecting the vector $\boldsymbol{f}$ onto the original design matrix $\mathbf{X}$,

$$\widetilde{\boldsymbol{\beta}} = \text{Proj}(\mathbf{X}, \boldsymbol{f}). \tag{2}$$

Here, the Moore-Penrose pseudo-inverse may be used once again as a linear projection operator, yielding

$$\widetilde{\boldsymbol{\beta}} = \mathbf{X}^{\dagger}\boldsymbol{f}. \tag{3}$$

The argument for why the p-dimensional vector $\widetilde{\boldsymbol{\beta}}$ is an effect size analogue for nonparametric regression models is because, on the n-observations, $\boldsymbol{f} \approx \mathbf{X}\widetilde{\boldsymbol{\beta}}$. In the case of kernel machines, theoretical results for identifiability and sparsity conditions have been previously developed (Crawford et al., 2018). However, more general intuition can be derived as follows. After having fit a model, we consider the fitted values $\boldsymbol{f}$ and regress these predictions onto the input variables so as to see how much variance these features explain. This is a simple way of understanding the relationships that the model has learned. The coefficients produced by this linear projection have their normal interpretation — they provide a summary of the relationship between the covariates in $\mathbf{X}$ and $\boldsymbol{f}$. For example, while holding everything else constant, increasing some feature $\mathbf{x}_j$ by 1 will increase $\boldsymbol{f}$ by $\widetilde{\beta}_j$.

3.2. Relative Centrality Measures

Similar to regression coefficients in linear models, the effect size analogues are not used to solely determine variable significance. Indeed, there are many approaches to infer associations based on the magnitude of effect size estimates (Stephens & Balding, 2009), but many of these techniques rely on arbitrary thresholding and fail to account for key covarying relationships that exist within the data. The "RelATive cEntrality" measure (or RATE) was developed as a *post-hoc* approach for variable selection that mitigates these concerns (Crawford et al., 2019).

Consider a collection of deterministic samples from the predictive distribution of $\widetilde{\boldsymbol{\beta}}$ (which is obtained by iteratively transforming draws from the posterior of $\boldsymbol{f}$ via Equation (3)). The RATE criterion summarizes how much any one variable contributes to what the model has learned. Effectively, this is done by taking the Kullback-Leibler divergence (KLD) between (i) the conditional posterior predictive distribution $p(\widetilde{\boldsymbol{\beta}}_{-j} \,|\, \widetilde{\beta}_j = 0)$ with the effect of the j-th predictor being set to zero, and (ii) the marginal distribution $p(\widetilde{\boldsymbol{\beta}}_{-j})$ with the effects of the j-th predictor being integrated out. Namely, $\text{RATE}(\widetilde{\beta}_j) := \text{KLD}(\widetilde{\beta}_j)/\sum_{\ell}\text{KLD}(\widetilde{\beta}_\ell)$ where

$$\begin{aligned}\text{KLD}(\widetilde{\beta}_j) &:= \text{KL}\left(p(\widetilde{\boldsymbol{\beta}}_{-j}) \,\|\, p(\widetilde{\boldsymbol{\beta}}_{-j} \,|\, \widetilde{\beta}_j = 0)\right)\\ &= \int_{\widetilde{\boldsymbol{\beta}}_{-j}} \log\left(\frac{p(\widetilde{\boldsymbol{\beta}}_{-j})}{p(\widetilde{\boldsymbol{\beta}}_{-j} \,|\, \widetilde{\beta}_j = 0)}\right) p(\widetilde{\boldsymbol{\beta}}_{-j})\, \mathrm{d}\widetilde{\boldsymbol{\beta}}_{-j}.\end{aligned} \tag{4}$$

There are two main takeaways from the RATE formulation. First, the KLD is a non-negative quantity, and equals zero if and only if variable j is of little importance, since removing its effect has no influence on the other variables. The second key takeaway is that the RATE criterion is bounded within the range $[0, 1]$, with the natural interpretation of measuring a variable's relative entropy — with a higher value equating to more importance. To this end, $1/p$ is a practical threshold for characterizing a predictor as "significant" since it represents the value at which the influence of all variables is uniform and indistinguishable.

3.3. Scalable Bayesian Neural Networks

In order to make RATE amenable for deep learning, we are required to take a more probabilistic view on prediction. This is possible using a Bayesian neural network. In contrast to a "standard" neural network, which uses maximum likelihood point-estimates for its parameters, a Bayesian neural network assumes a prior distribution over its weights. The posterior probability over the weights, learned during the training phase, can then be used to compute the posterior predictive distribution. This provides a notion of uncertainty about the prediction, which is particularly valuable in high-risk settings that also demand interpretable models.

As the size of datasets in many application areas continues to grow, it has become less feasible to implement traditional Markov Chain Monte Carlo (MCMC) algorithms for inference. This has motivated approaches for supervised learning that are based on variational Bayes and the stochastic optimization of a variational lower bound (Hinton & Van Camp, 1993; Barber & Bishop, 1998; Graves, 2011). Some recent works have focused on reducing the variance of stochastic gradients by using different reparameterization techniques (Kingma & Welling, 2013; Rezende et al., 2014), while others have worked to expand this "trick" to non-Gaussian distributions (Blundell et al., 2015; Kingma et al., 2015; Ruiz et al., 2016). Finally, the popular neural network regularization technique known as dropout has been shown to be theoretically equivalent to computing variational approximations for the posterior distribution of network weights (Blundell et al., 2015; Gal & Ghahramani, 2016).

4. Methodological Contributions

We now detail the main methodological contributions of this paper. First, we describe our motivating deep neural network architecture. Next, we propose a new effect size analogue projection that is more robust to collinear input data. Lastly, we derive a closed-form solution for the RATE methodology under this new framework.

4.1. Motivating Neural Network Architecture

Consider a binary classification problem with n observations. We have an n-dimensional set of labels $\mathbf{y} \in \{0, 1\}^n$ and an $n \times p$ design matrix $\mathbf{X}$ with p covariates. We assume the following hierarchical network architecture to learn the predicted label for each observation in the data

$$\widehat{\mathbf{y}} = \sigma(\boldsymbol{f}), \quad \boldsymbol{f} = \mathbf{H}(\boldsymbol{\theta})\mathbf{w} + \mathbf{b}, \quad \mathbf{w} \sim \pi(\bullet)\,, \tag{5}$$

where $\sigma(\bullet)$ is a sigmoid function, and $\boldsymbol{f}$ is an n-dimensional vector of smooth latent values or "logits" that need to be estimated. Here, we use an $n \times k$ matrix $\mathbf{H}(\boldsymbol{\theta})$ to denote the activations from the penultimate layer (which are fixed given a set of inputs and point estimates $\boldsymbol{\theta}$), $\mathbf{w}$ is a k-dimensional vector of weights at the output layer assumed to follow prior distribution $\pi(\bullet)$, and $\mathbf{b}$ is an n-dimensional vector of the deterministic bias that is produced during the training phase.

The structure of Equation (5) is motivated by the fact that we are most interested in the posterior distribution of the latent variables when computing RATE measures and, subsequently, the effect size analogue. To this end, we may logically split the network into three components: (i) an input layer of the original predictor variables, (ii) hidden layers where parameters are deterministically computed, and (iii) the logit layer where the parameters and activations are treated as random variables. Since the resulting logits are a linear combination of these components, their joint distribution will be closed-form if the posterior distribution of the weight parameters is also of closed-form.

There are also two important features of this network setup. First, we may easily generalize this architecture to the multi-class problem by simply increasing the number of output nodes to match the number of categories, and redefining $\sigma(\bullet)$ to be the softmax function. Second, the structure of the hidden layers can be of any size or type, provided that the additional parameters are given by point estimates. Ultimately, this flexibility means that a wide range of existing architectures can be easily modified to be used with RATE. The simplest example of such an architecture is illustrated in Figure 1.

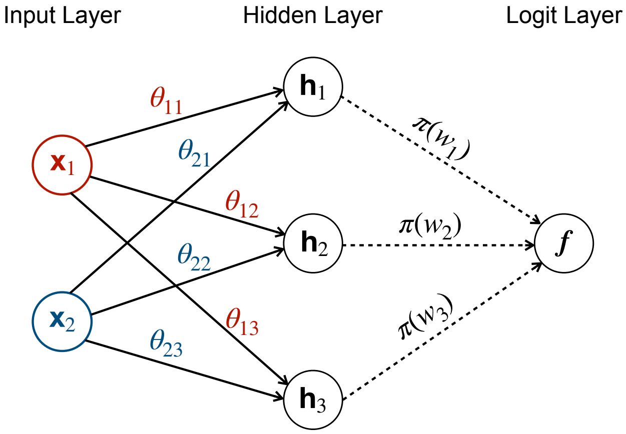

Figure 1. An example of the network architecture used in this work. Here, the first layer parameters $\boldsymbol{\theta}$ are computed deterministically, while the logit layer weights $\mathbf{w}$ are assumed to be distributed under the prior $\pi(\bullet)$. The input variables $\mathbf{x}_1$ and $\mathbf{x}_2$ are fed through the hidden layers $(\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3)$. Estimates of the predicted logits $\boldsymbol{f}$ are obtained via a linear combination of these components and samples from the posterior distribution of (w_1, w_2, w_3). Note that this figure does not include the bias terms.

4.2. Posterior Inference with Variational Bayes

As previously discussed in Section 3.3, using classic MCMC algorithms to obtain estimates of the logit layer weights is not always a scalable option. We therefore turn to variational Bayes, which has the additional benefit of providing closed-form expressions for the posterior distribution of $\mathbf{w}$ — and, subsequently, the logits $\boldsymbol{f}$.

To use variational Bayes to train a neural network, we first specify a prior $\pi(\mathbf{w})$ over the weights and replace the intractable true posterior $p(\mathbf{w} \,|\, \mathbf{y}) \propto p(\mathbf{y} \,|\, \mathbf{w})\pi(\mathbf{w})$ with an approximating family of distributions $q_{\boldsymbol{\phi}}(\mathbf{w})$. The variational parameters $\boldsymbol{\phi}$ are selected by minimizing $\mathrm{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{w}) \,\|\, p(\mathbf{w} \,|\, \mathbf{y})\right)$, with respect to $\boldsymbol{\phi}$, with the goal of selecting the member of the approximating family that is closest to the true posterior. This is equivalent to maximizing the following variational lower bound

$$\underset{\boldsymbol{\phi}}{\arg\max} - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{w}) \,\|\, \pi(\mathbf{w})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{w})}\left[\log p(\mathbf{y} \,|\, \mathbf{w})\right].$$

However, since the architecture we specified in Equation (5) contains point estimates at the hidden layers, training the network cannot simply involve maximizing the lower bound with respect to the variational parameters. Instead, all parameters must be optimized jointly:

$$\underset{\boldsymbol{\phi},\boldsymbol{\theta}}{\arg\max} - \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{w}) \,\|\, \pi(\mathbf{w})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{w})}\left[\log p(\mathbf{y} \,|\, \mathbf{w}, \boldsymbol{\theta})\right].$$

We will use stochastic optimization to train the network. Depending on the chosen variational family, the gradients of the minimized $\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{w}) \,\|\, \pi(\mathbf{w}))$ may be available in closed-form, while gradients of the log-likelihood $\log p(\mathbf{y} \,|\, \mathbf{w}, \boldsymbol{\theta})$ are evaluated using Markov chain samples and the local reparameterization trick (Kingma et al., 2015).

Following this procedure, we have an optimal set of parameters for $q_{\boldsymbol{\phi}}(\mathbf{w})$, with which we can sample posterior draws for the logit layer. Hereafter, we will refer to this optimal set as $\{\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\phi}}\}$.

4.3. Assuming Gaussian Variational Posteriors

In this work, we conveniently choose the diagonal Gaussian as the family for $q_{\widehat{\boldsymbol{\phi}}}(\mathbf{w})$, which assumes that the variational posterior fully factorizes over the elements of $\mathbf{w}$. Indeed, this simple choice does not consider correlations between the logit layer weights *a priori* and can be outperformed by other variational approximations in terms of mean squared error (Gal & Ghahramani, 2016). However, given our motivations, a fully interpretable model is preferable to a seemingly black box approach with a higher test accuracy. Furthermore, the key advantage to choosing the diagonal Gaussian is that, when combined with an independent normal prior for $\pi(\mathbf{w})$, it ensures that the predicted logits $\boldsymbol{f}$ will follow a multivariate Gaussian as well. We begin by writing the variational posterior as

$$q_{\widehat{\boldsymbol{\phi}}}(\mathbf{w}) = \mathcal{N}(\mathbf{m}, \operatorname{diag}(\mathbf{v})), \quad \widehat{\boldsymbol{\phi}} = \{\mathbf{m}, \mathbf{v}\}, \tag{6}$$

with mean vector $\mathbf{m}$ and a covariance matrix with diagonal elements $\mathbf{v}$. Using Equations (5) and (6), we may then derive the implied distribution over the logits as

$$\boldsymbol{f} \mid \mathbf{X}, \mathbf{y} \sim \mathcal{N}(\mathbf{H}(\widehat{\boldsymbol{\theta}})\mathbf{m} + \mathbf{b}, \mathbf{H}(\widehat{\boldsymbol{\theta}})\operatorname{diag}(\mathbf{v})\mathbf{H}(\widehat{\boldsymbol{\theta}})^{\intercal}). \tag{7}$$

While the elements of $\mathbf{w}$ are independent, dependencies in the input data (via the activations $\mathbf{H}(\widehat{\boldsymbol{\theta}})$) induce non-diagonal covariance between the elements of $\boldsymbol{f}$. The resulting means and covariances from Equation (7) can then be used for variable prioritization and selection.

4.4. Covariance Projection Operator

After having conducted (variational) Bayesian inference, we use posterior draws from Equation (7) to define an effect size analogue for neural networks. We could use the linear projection operator in Equation (3); but, in the case of highly correlated inputs, the Moore-Penrose pseudo-inverse suffers from instability (see a small simulation study in Appendix A). This is part of the underlying reason why linear regression models suffer in the presence of collinearity. While regularization poses a viable solution to this problem, the selection of an optimal penalty parameter is not always a straightforward task. As a result, we propose a much simpler projection operator that can be very effective, particularly in application areas where data measurements can be perfectly collinear (e.g. pixels in an image). Our solution is to use a linear measure of dependence separately for each predictor based on the sample covariance. Namely, for each of the p input variables

$$\widetilde{\boldsymbol{\beta}} := \operatorname{cov}(\mathbf{X}, \boldsymbol{f}) = [\operatorname{cov}(\mathbf{x}_1, \boldsymbol{f}), \ldots, \operatorname{cov}(\mathbf{x}_p, \boldsymbol{f})]. \tag{8}$$

Since the proposed operator is based on the sample covariance, the effect size analogues have the following form

$$\widetilde{\boldsymbol{\beta}} = \frac{1}{n-1}\mathbf{X}^{\intercal}\mathbf{C}\boldsymbol{f}, \tag{9}$$

where $\mathbf{C} = \mathbf{I} - \mathbf{1}\mathbf{1}^{\intercal}/n$ denotes the centering matrix, $\mathbf{I}$ is an n-dimensional identity matrix and $\mathbf{1}$ is an n-dimensional vector of ones. Probabilistically, since we assume the posterior of the logits to be normally distributed, the above is equivalent to assuming that $\widetilde{\boldsymbol{\beta}} \mid \mathbf{X}, \mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Omega})$ where

$$\boldsymbol{\mu} = \frac{1}{n-1}\mathbf{X}^{\intercal}\mathbf{C}\mathbf{H}(\widehat{\boldsymbol{\theta}})\mathbf{m} \tag{10}$$

$$\boldsymbol{\Omega} = \frac{1}{(n-1)^2}\mathbf{X}^{\intercal}\mathbf{C}\mathbf{H}(\widehat{\boldsymbol{\theta}})\operatorname{diag}(\mathbf{v})\mathbf{H}(\widehat{\boldsymbol{\theta}})^{\intercal}\mathbf{C}^{\intercal}\mathbf{X}. \tag{11}$$

Intuitively, each element in $\widetilde{\boldsymbol{\beta}}$ represents some measure of how well the original data at the input layer explains the variation between observation classes. Moreover, under this approach, if two predictors $\mathbf{x}_r$ and $\mathbf{x}_s$ are almost perfectly collinear, then the corresponding effect sizes will also be very similar since $\operatorname{cov}(\mathbf{x}_r, \boldsymbol{f}) \approx \operatorname{cov}(\mathbf{x}_s, \boldsymbol{f})$. To build a better intuition for identifiability under this covariance projection, recall simple linear regression where ordinary least squares (OLS) estimates are unique modulo the span of the data (Wold et al., 1984). A slightly different issue will arise for the effect size analogues computed via Equation (8), where now two estimates are unique modulo the span of a vector of ones, or $span\{\mathbf{1}\}$. We now make the following formal statement.

Claim 4.1. *Two effect size analogues computed via the covariance projection operators, $\widetilde{\boldsymbol{\beta}}_1 = cov(\mathbf{X}, \boldsymbol{f}_1)$ and $\widetilde{\boldsymbol{\beta}}_2 = cov(\mathbf{X}, \boldsymbol{f}_2)$, are equivalent if and only if the corresponding functions are related by $\boldsymbol{f}_1 = \boldsymbol{f}_2 + c\mathbf{1}$, where $\mathbf{1}$ is a vector of ones and c is some arbitrary constant.*

The proof of this claim is trivial and follows directly from the covariance being invariant with respect to changes in location. Other proofs connecting this effect size to classic statistical measures can be found in the Appendix B.

4.5. Closed-Form for Centrality Measures

Under our modeling assumptions, the posterior distribution of $\widetilde{\boldsymbol{\beta}}$ is (approximately) multivariate normal with an empirical mean vector $\boldsymbol{\mu}$ and positive semi-definite covariance/precision matrix $\boldsymbol{\Omega} = \boldsymbol{\Lambda}^{-1}$. Given these values, we may partition such that, for the j-th input variable, $\boldsymbol{\mu} = (\mu_j; \boldsymbol{\mu}_{-j})$ and

$$\boldsymbol{\Omega} = \begin{pmatrix} \omega_j & \boldsymbol{\omega}_{-j}^{\intercal} \\ \boldsymbol{\omega}_{-j} & \boldsymbol{\Omega}_{-j} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_j & \boldsymbol{\lambda}_{-j}^{\intercal} \\ \boldsymbol{\lambda}_{-j} & \boldsymbol{\Lambda}_{-j} \end{pmatrix}.$$

Now we may compute RATE values using Equation (4), which in the case of Gaussian distributions, has the following closed form (Crawford et al., 2019)

$$\begin{aligned}\operatorname{KLD}(\widetilde{\beta}_j) = &\frac{1}{2}\Big[\operatorname{tr}(\boldsymbol{\Omega}_{-j}\boldsymbol{\Lambda}_{-j\,|\,j}) - \log|\boldsymbol{\Omega}_{-j}\boldsymbol{\Lambda}_{-j}| \\ &- (p-1) + \delta_j(\widetilde{\beta}_j - \mu_j)^2\Big]\end{aligned} \tag{12}$$

where $\delta_j = \boldsymbol{\lambda}_{-j}^{\intercal}\boldsymbol{\Lambda}_{-j}^{-1}\boldsymbol{\lambda}_{-j}$ and characterizes the implied linear rate of change of information when the effect of any predictor is absent — thus, providing a natural (non-negative) numerical summary of the role of each $\widetilde{\beta}_j$ in the multivariate distribution.

5. Results

In this section, we apply our interpretable Bayesian DNN framework to three broad application areas: computer vision, natural language processing, and public policy relevance.

5.1. Image Classification using MNIST

We construct binary classification tasks from MNIST (LeCun, 1998) by selecting two digits and training a network to distinguish between them. Results in the main text are based on comparing (i) ones and zeros and (ii) ones and eights. This Bayesian network contains a single convolutional layer, followed by two fully-connected layers, and satisfies the architectural requirements outlined in Section 4.1. Extensions to the multi-class problem are included in Appendix D.

Following training, we compute the RATE values for each pixel, where a high value indicates that a pixel is important for the network when differentiating between the two classes. Results for the two comparisons are shown in Figure 2. Note that RATE values do not provide information about the class-specific associations, but we can assign a direction to each pixel using the sign of $\mathbb{E}[p(\widetilde{\beta}_j \mid \mathbf{X}, \mathbf{y})]$. The pixels identified by RATE are consistent with human intuition. When distinguishing between a zero and a one (see Figure 2A), the most important pixels are in the center (where the vertical line of a one would appear) and in a ring (corresponding to the shape of zero). Similarly, for ones and eights (see Figure 2B), the shape of an eight is clearly visible.

While our results show a plausible set of important pixels under visual inspection, a natural followup analysis is to assess if these pixels are important for the network when it makes an out-of-sample prediction. One way to establish this is to tabulate the prediction accuracy as certain pixels in the test images are shuffled, thus uncorrelating the observations and their labels. Figure 3 shows the test set accuracy when subsets of pixels are shuffled. Subsets are chosen according to the rank of their RATE values (both in ascending and descending order) and at completely random. The test set accuracy decreases much more steeply when pixels with the highest RATE values are shuffled (blue line) versus when pixels are selected at random (green line). The converse is also true — when pixels with the lowest RATE values are shuffled, the test accuracy is far more robust (orange line). This indicates that pixels with high RATE values are indeed the features used by the network when making a classification.

5.2. Large Movie Review Sentiment Analysis

The Large Movie Review dataset (Maas et al., 2011) contains 50,000 reviews labeled as having either positive or negative sentiment. These are also split equally into test and training sets. The reviews are encoded using bag-of-words such that each observation is a D-dimensional vector for dictionary size D, with the j-th element denoting the number of times that word j appears in the review. In this analysis, the dictionary consisted of the $D = 1000$ most frequent words in the entire corpus.

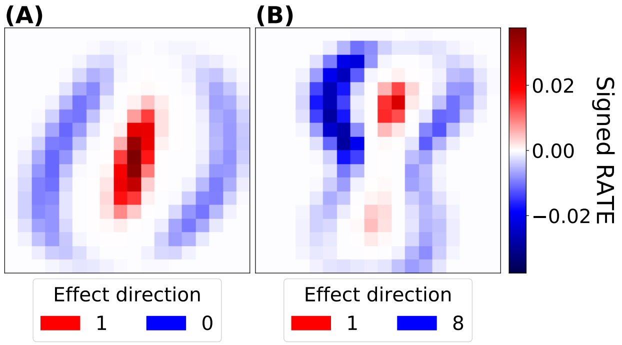

Figure 2. "Signed" RATE values associated with each pixel in binary classification tasks using (A) zeros and ones and (B) ones and eights from the MNIST dataset. Pixels with a higher RATE value are more important for the DNN to distinguish between classes. Here, the RATE values have been multiplied by the sign of the effect size analogue posterior mean in order to illustrate the direction of the pixel's effect. In (A), the RATE values in a straight red line at the center correspond to the one, while the other important pixels forming the outside blue ring associate with the zero. In (B), the shape of the eight is clearly visible in blue, while the red regions in the center correspond to the one.

A Bayesian neural network with three fully-connected layers and ReLU activations was trained to predict sentiment from the encoded reviews. For comparison, we also trained a random forest (with hyperparameters selected using random grid search) and a logistic regression model. Random forest is a popular nonparametric, nonlinear model that has an established variable importance methodology based on Gini impurity, while classic logistic regression is conventionally interpretable as it provides odds ratios and associated p-values.

Figure 4 shows the top 15 most important words according to (A) a Bayesian DNN with RATE, (B) the random forest model and (C) logistic regression. For RATE, the directions of the effect (positive versus negative) are assigned via the sign of $\mathbb{E}[p(\widetilde{\beta}_j \mid \mathbf{X}, \mathbf{y})]$ for that word. The words identified by our approach are all associated with positive or negative sentiment, with 12 of the 15 most highly ranked words having negative sentiment. This reflects an established phenomenon from psychology which poses that negative sentiments tend to outweigh positive ones (Baumeister et al., 2001). Furthermore, the results of our framework match many of the words that the other two methods deem as important. Out of the top 15 words ranked by RATE, 6 of them also appear in the equivalently ranked list for the ran-

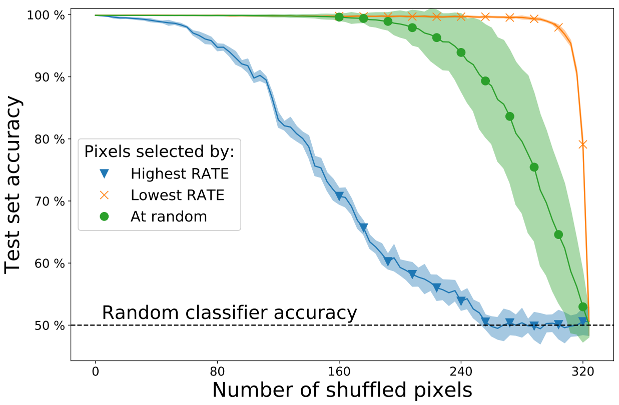

Figure 3. Test set accuracies on the MNIST binary classification task for zeros and ones when shuffling pixels selected according to their RATE values or at random. Shuffling pixels with the highest RATE values (blue) decreases the test set accuracy more quickly than shuffling pixels with the lowest RATE values (orange) or completely selected at random (green). This indicates that pixels identified as significant by RATE are used by the network when making a classification. Shuffling was repeated 10 times and predictions were made using 20 MC samples. Lines indicate the mean accuracy of these repeats and the shaded area indicates ± two standard deviations.

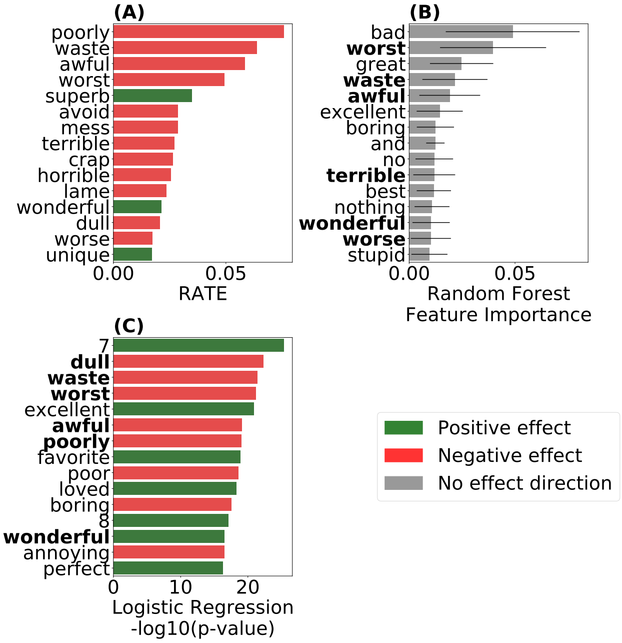

Figure 4. The 15 most important words for (A) a Bayesian neural network according to RATE, (B) a random forest classifier, and (C) logistic regression. The color of the bars indicates the direction of the variable's effect, which is taken from the sign of (A) the effect size analogue posterior mean, or (C) the sign of the regression coefficient. While this information is not available in (B), this plot does show ± the standard deviation of Gini importance across the trees of the random forest. Words in bold also appear in the top ranked list by the neural network and RATE.

dom forest and 6 appear for logistic regression. For the 112 words with RATE values greater than the $1/p$ cutoff, 38 and 74 words appear in the equivalently ranked list for random forest and logistic regression, respectively. The Bayesian neural network utilizes rankings that make intuitive sense and produces results that are supported by established interpretable models.

5.3. COMPAS Risk Score Study

The criminal risk assessment tool, Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), has been widely used to predict offender recidivism since its release in 1998. Recently, writers for *ProPublica* analyzed the algorithm on approximately 10000 individuals arrested in Broward County, Florida between 2013 and 2014 and found its predictions to be racially biased (Angwin et al., 2016; Dressel & Farid, 2018). Here, logistic regression revealed that being African American was significant in predicting COMPAS-assigned risk scores. Our aim in this section is to analyze this same data with a Bayesian DNN and RATE to examine the relationship between COMPAS-assigned risk scores and race.

To begin, we filtered the data to only include individuals who had either recidivated in two years, or had at least two years outside of a correctional facility. This resulted in a final dataset with 6172 observations with 5 covariate types: number of prior offenses, race (labeled as African American, Asian, Hispanic, Native American, or Other), gender (labeled as 1 for being female, 0 for otherwise), age group (labeled as older than 45 years old or less than 25), and the severity of charge. The COMPAS system classifies people into high, medium, and low risk categories. In the main text, we focus on the binary classification problem between the high and low risk categories. In Appendix E, we turn to the multi-class problem and jointly examine all three levels together.

In the case of the binary classification analysis, we follow the methodology described in Section 4. Here, we again fit a Bayesian neural network with three fully-connected layers and ReLU activations. We then estimated the variational distribution for the weights on the final layer. Next, we inferred the implied posterior distribution of the covariance effect size analogue and compute RATE measures for each covariate. The number of prior offenses was identified as the only significant variable in the model (see Figure 5A).

There is a difference between our results using a deep learning approach and *ProPublica*'s simple logistic model which identified racial factors (specifically an individual be identified as being African American) as being a large factor

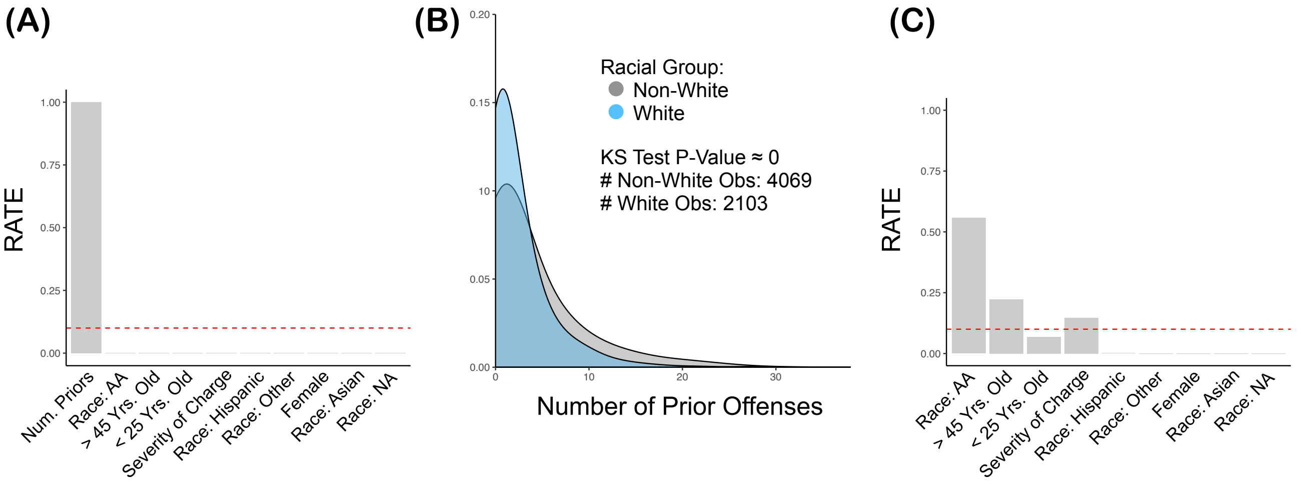

Figure 5. (A) First order RATE values from the Bayesian DNN. (B) Observed distribution of number of prior offenses by racial group. (C) First order RATE values from Bayesian DNN when number of prior offenses is omitted from the analysis. The dashed line is drawn at the level of relatively equal importance (i.e. $1/p$ and $1/(p-1)$ for panels (A) and (B), respectively).

in predicting COMPAS risk scores. This motivated us to compare the distribution of prior offense counts across racial groups (see Figure 5B). Approximately two-thirds of the modeling sample is non-white, of whom 31.06% have no prior offenses and some individuals were recorded as having more than 30 previous accounts. In the white cohort only 39.04% were without prior offenses, but the tails of that distribution did not exceed far passed 15. A Kolmogorov-Smirnov (KS) test between the two groups yielded a p-value near zero — confirming that this statistic was inherently racially biased. This also explains why the DNN did not bother to place any significant weight/prioritization on the other predictor variables that were included in the model.

To this end, we omitted the number of prior offenses from the neural network and subsequent analyses led to RATE identifying the African-American racial factor as being the most significant predictor in classifying COMPAS risk scores (see Figure 5C). The number of prior offenses essentially masked this effect. A DNN trained on the full dataset had a predictive accuracy of 75.5%. Fitting the same model with the number of prior offenses as its only input yielded an accuracy of 68.2% — but this is not a sizable improvement over the 67.49% predictive accuracy we observed when this variable was omitted.

6. Discussion

We developed an approach to achieve global interpretability for deep neural networks through variable importance. In particular, we first proposed a hierarchical network architecture that allows for efficient estimation of posterior distribution for network parameters. Here, we placed conjugate priors on the weights in the last layer of the DNN and estimated the joint posterior of the network using variational Bayes with a diagonal Gaussian family. Next, we proposed a sample covariance operator as an effect size (or coefficient) analogue for the input variables of DNNs. This new operator mitigates the well-known concern that linear estimators of regression coefficients are more sensitive in applications with highly collinear predictors. We then extended the recently limited RelATive cEntrality measure to a Bayesian deep learning framework and provided closed-form solutions for its implementation. Lastly, we illustrated the performance of our framework in three broad applications including computer vision, natural language processing, and public policy.

Motivated by these results, several interesting future directions remain. For example, in the current study, we strictly focus on interpreting the significance of variables at the input layer of DNNs. However, given the network architecture that we consider, it is also possible to examine the importance of hidden layers using RATE. Essentially, if we impose interpretations onto these layers (in the context of some application), then we may use the centrality measure to assess how the corresponding nodes (i.e. specific groups of input variables) contribute to predictive accuracies.

7. Software Availability

Software for implementing the interpretable Bayesian DNN framework with RATE significance measures is carried out in R and Python code, which is freely available at `https://github.com/lorinanthony/RATE`.

8. Acknowledgments

Financial support was provided by start up funds from Brown University. Sarah Filippi is also partially supported by the EPSRC (grant EP/R013519/1) and Jonathan Ish-Horowicz gratefully acknowledges funding from the Wellcome Trust (PhD studentship 108908/Z/15/Z). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of the funders.

References

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica*, 2016.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS One*, 10(7):e0130140, 2015.

Barber, D. and Bishop, C. M. Ensemble learning in Bayesian neural networks. *NATO ASI Series F Computer and Systems Sciences*, 168:215–238, 1998.

Baumeister, R. F., Bratslavsky, E., Finkenauer, C., and Vohs, K. D. Bad is Stronger than Good. *Review of General Psychology*, 5(4):323, 2001.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1613–1622. JMLR. org, 2015.

Breiman, L. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

Chen, X., Liu, C.-T., Zhang, M., and Zhang, H. A forest-based approach to identifying gene and gene–gene interactions. *Proceedings of the National Academy of Sciences*, 104(49):19199–19203, 2007.

Crawford, L., Wood, K. C., Zhou, X., and Mukherjee, S. Bayesian approximate kernel regression with variable selection. *Journal of the American Statistical Association*, 113(524):1710–1721, 2018.

Crawford, L., Flaxman, S., Runcie, D., and West, M. Variable prioritization in nonlinear black box methods: A genetic association case study. *Annals of Applied Statistics*, 2019.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608*, 2017.

Dressel, J. and Farid, H. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1):eaao5580, 2018. doi: 10.1126/sciadv.aao5580. URL `http://advances.sciencemag.org/content/4/1/eaao5580.abstract`.

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Frosst, N. and Hinton, G. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

Gal, Y. and Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1050–1059, 2016.

Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, pp. 2348–2356, 2011.

Hinton, G. E. and Van Camp, D. Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13. ACM, 1993.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pp. 2575–2583, 2015.

Larson, J., Mattu, S., Kirchner, L., and Angwin, J. How we Analyzed the COMPAS Recidivism Algorithm. *ProPublica (5 2016)*, 9, 2016.

LeCun, Y. The MNIST Database of Handwritten Digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pp. 142–150. Association for Computational Linguistics, 2011.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

Ruiz, F. R., AUEB, M. T. R., and Blei, D. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, pp. 460–468, 2016.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3145–3153, 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv:1312.6034*, 2013.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: Removing noise by adding noise. *arXiv:1706.03825*, 2017.

Stephens, M. and Balding, D. J. Bayesian statistical methods for genetic association studies. *Nature Reviews Genetics*, 10:681–690, 2009.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Proceeding of the 34th International Conference on Machine Learning*, pp. 3319–3328, 2017.

Wang, F. and Rudin, C. Falling rule lists. In *Artificial Intelligence and Statistics*, pp. 1013–1022, 2015.

Wold, S., Ruhe, A., Wold, H., and Dunn, III, W. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743, 1984.

A. Robustness of the Covariance Projection Operator in the Presence of Collinearity

In this section, our goal is to motivate the use of the covariance projection operator for the effect size analogue in Bayesian neural networks. We do this via a small simulation study which shows that the conventional linear estimation of regression coefficients is unstable in applications with highly collinear predictors. Here, we generate a synthetic design matrix with $n = 5000$ individuals and $p = 2$ covariates ($\mathbf{x}_1$ and $\mathbf{x}_2$) randomly drawn from standard normal distributions. We then assess two simulation scenarios with continuous outcomes created under the following linear model

$$\mathbf{y} = 2\mathbf{x}_1 - 2\mathbf{x}_2 + \boldsymbol{\varepsilon}, \quad \boldsymbol{\beta} = [2, -2], \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

In the first simulation scenario, $\mathbf{x}_1$ and $\mathbf{x}_2$ are uncorrelated; while, in the second scenario, the two covariates are set to share a Pearson correlation coefficient of $\rho = 0.999$. In each case, we compare the classic ordinary least squares (OLS) estimate for regression coefficients $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^{\intercal}\mathbf{X})^{-1}\mathbf{X}^{\intercal}\mathbf{y}$ and the proposed covariance effect size analogue $\widetilde{\boldsymbol{\beta}} = [\text{cov}(\mathbf{x}_1, \mathbf{y}), \text{cov}(\mathbf{x}_2, \mathbf{y})]$. Figure A1 depicts the results for both cases repeated 100 different times. In Figure A1A, we see that both types of estimators are able to properly capture the true effects when the predictors are uncorrelated. This finding is expected. However, in the extremely collinear scenario with $\mathbf{x}_1 \approx \mathbf{x}_2$, the total true effect size in the simulation is effectively equal to $\beta = 2 - 2 = 0$. The OLS estimators are unstable under this condition, while the covariance effect size analogues accurately and robustly estimate this value (see Figure A1B).

B. Connection to Marginal Association Tests

In this section, we prove a connection between the covariance projection operator and the conventional hypothesis testing strategies for marginal feature associations. Assume that we have an n-dimensional outcome variable $\mathbf{y}$ that is to be modeled by an $n \times p$ design matrix $\mathbf{X}$. In linear regression, a simple (yet effective) approach is to take each covariate $\mathbf{x}_j$ in turn and assess associations based upon a two-tailed alternative hypothesis. The significance of this test is then summarized via p-values (e.g. $\widehat{p}_j$ for feature j), which may then be ranked in the order of importance from smallest to largest. Here, we show that the effect size analogues $\widetilde{\boldsymbol{\beta}}$ in Equation (8) correspond exactly to the test statistics for this frequented univariate approach.

Begin by recalling that the covariance projection operator simply produces the sample covariance between a given predictor variable $\mathbf{x}_j$ and the model predictions $\boldsymbol{f}$ — where both largely positive or negative covariances are informative. Next, recall that the sample covariance between two random variables is equal to their Pearson correlation coefficient (ρ) multiplied by their respective standard errors σ_X and σ_Y,

$$\text{cov}(X, Y) = \rho\, \sigma_X \sigma_Y. \tag{A1}$$

The standard formula for p-values starts by calculating a t-statistic of the following form

$$T_j = \rho_j \sqrt{\frac{n-2}{1-\rho_j^2}}, \quad j = 1, \ldots, p. \tag{A2}$$

Corresponding p-values are then computed by comparing these values to a Student's t-distribution function under the null hypothesis — with the intuition being that larger test statistics will result in smaller p-values.

We now verify that these transformations are all monotonic — thus, our proposed covariance effect size analogue will result in the same ranking of variable importance as the classical t-test.

Theorem 1. *If two predictor variables have covariance effect size analogues such that* $\widetilde{\beta}_1 = cov(\mathbf{x}_1, \boldsymbol{f}) >$

$cov(\mathbf{x}_2, \boldsymbol{f}) = \widetilde{\beta}_2$, then the resulting p-values from a t-test with these features will have the relationship $\widehat{p}_1 < \widehat{p}_2$.

Proof. Consider the covariance projection operation on two different predictor variables, $\text{cov}(\mathbf{x}_1, \boldsymbol{f}) > \text{cov}(\mathbf{x}_1, \boldsymbol{f})$. Since standard deviations are positive

$$\text{cov}(\mathbf{x}_1, \boldsymbol{f})\sigma_{\mathbf{x}_1}\sigma_{\boldsymbol{f}} > \text{cov}(\mathbf{x}_1, \boldsymbol{f})\sigma_{\mathbf{x}_2}\sigma_{\boldsymbol{f}} \quad \Longleftrightarrow \quad \rho_1 > \rho_2.$$

The same applies when multiplying both sides by $\sqrt{n-2}$. Also note that since we are concerned with the magnitude of covariances (and subsequently correlations),

$$\rho_1 > \rho_2 \quad \Longleftrightarrow \quad \sqrt{1-\rho_1} \leq \sqrt{1-\rho_2}.$$

Therefore we conclude that

$$\rho_1\sqrt{\frac{n-2}{1-\rho_1^2}} > \rho_2\sqrt{\frac{n-2}{1-\rho_2^2}} \quad \Longleftrightarrow \quad T_1 > T_2.$$

Since the distribution function is monotonic, $\widehat{p}_1 < \widehat{p}_2$. □

C. Real Data Quality Control Procedures

We use three real datasets in the present study. The first comes from the MNIST database of handwritten images (http://yann.lecun.com/exdb/mnist/) (LeCun, 1998). The downloaded digits had already been size-normalized and centered in a fixed-size image with 28×28 dimensions. It has been noted that the error rate of classification methods can improve when the digits are centered by bounding box rather than center of mass. To this end, we further cropped the images with a 5-pixel border — resulting in a final dataset with digits of size 18×18. We note that this border region contained only zeros in the vast majority of the images, and so the pixels in these regions were not informative. The binary classification task involving zeros and ones had training and test set sizes of 12,665 and 2115 pixels respectively.

The Large Movie Review dataset included the 1,000 most frequently used words (excluding padding, unknown, and start characters) for 50,0000 reviews (http://ai.stanford.edu/~amaas/data/sentiment/) (Maas et al., 2011). These reviews were split into equally sized test and training sets. The unformatted data consists of sequences of integers, where each integer corresponds to a word. These were encoded using bag-of-words, which resulted in each review being represented by a 1000-dimensional vector whose j-th element denotes the relative frequency of the j-th most frequent word in the corpus. The relative frequency is the number of times a word appears in a document divided its total number of appearances in the training or test set.

For the COMPAS analysis, we downloaded the same dataset that *ProPublica* used on criminal defendants from Broward County, Florida and follow the recommended data cleaning procedures (https://github.com/propublica/compas-analysis) (Angwin et al., 2016). Individuals with charge dates more than thirty days before or after arrest were dropped because the arrest was likely associated with a different crime than the one inciting a COMPAS score. The dataset is also pruned for individuals who either recidivated in two years or have two years outside a correctional facility. We also pruned for people whose COMPAS-scored crime was not an ordinary traffic offense. Like *ProPublica*, we binned the response into low risk versus medium and high risk (the multinomial case is reviewed in Appendix E), but deviate from *ProPublica* by omitting the “two-year recidivism” covariate from the considered feature set.

D. Multi-class Analysis in MNIST Dataset

In the multi-class case, the we set up the neural network to contain 10 output nodes (corresponding to the 10 digits of MNIST). We compute effect size analogues and RATE values for each node — meaning that each pixel has a set of 10 RATE values that indicate its importance in identifying each of the 10 digits. These RATE values are shown for each node in Figure A2, which illustrates how each feature associates with a particular digit. Results for nodes 0, 3, 6, 8 and 9 are particularly easy to understand visually.

In order to evaluate the pixel rankings produced by RATE, we perform two experiments. In the first, any pixel with a RATE value greater than $1/p$ for any of the 10 classes is shuffled (221 pixels in total). In the second experiment, the remaining pixels are shuffled — this is the set of pixels that are not important for distinguishing between digit classes according to RATE (103 pixels). In both experiments, the same number of randomly-selected pixels are selected and shuffled. The results for both experiments are shown in Figure A3. Figure A3A shows that removing high-RATE pixels causes the test accuracy to effectively resemble a random classifier (i.e. the red dotted line). Removing pixels at random at least retains some information. Figure A3B shows that the reverse to be true. After shuffling all the low-RATE pixels, the test accuracy is insignificantly reduces to 93.4% (i.e. reduction of 5.0%). However, shuffling the same number of pixels at random reduces the test accuracy to 68.6% (i.e. reduction of 29.8%).

E. Multi-class Analysis in COMPAS Study

Instead of binning the COMPAS responses to create a binary classification problem, it is natural to consider the original three risk scores categories: low, medium, and high. Feeding the data matrix through a neural network to predict the multinomial case gives RATE values for each of the three possible class labels (see Figure A4A). This simply requires redefining $\sigma(\bullet)$ in Equation (5) to be the softmax function.

The number of prior offenses accounts for almost all of the relative significance for each class label, similar to the binary case. No other variable rises in importance to distinguish either the risk scores. Removing the number of prior offenses from the feature set and retraining the neural network gives RATE values for each of the three class labels (see Figure A4B) — which again mimic the result seen in the binary case.

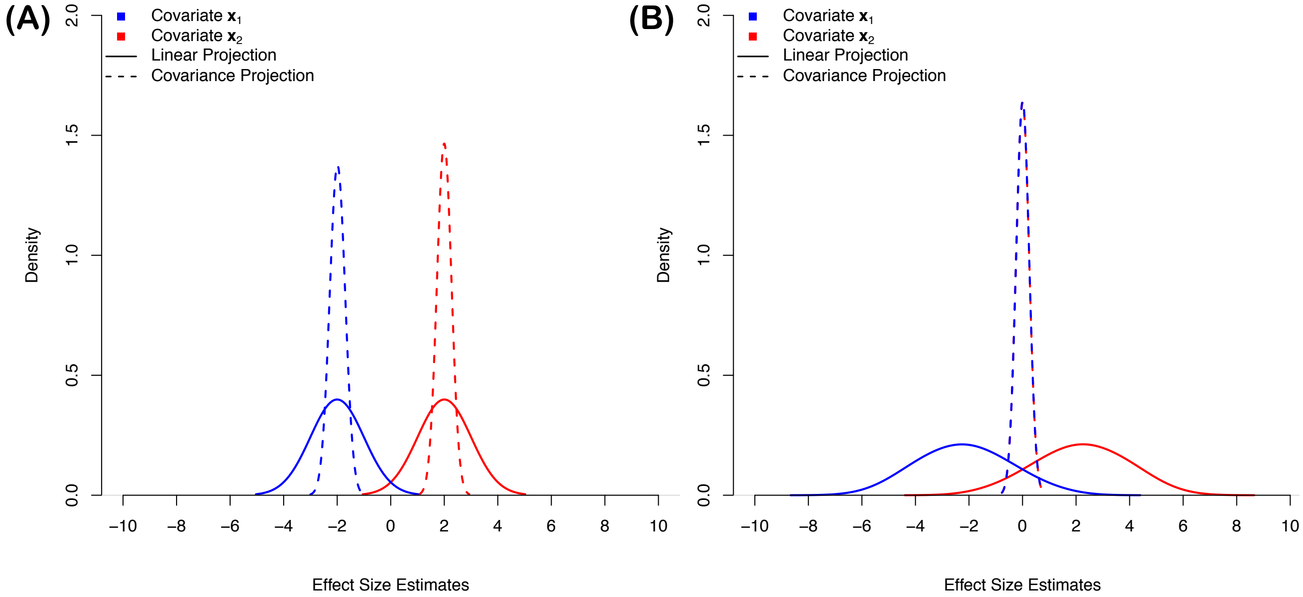

Figure A1. Results from a small simulation study showing the robustness of the covariance projection operator in the presence of collinear predictor variables. Synthetic data is generated as $\mathbf{y} = 2\mathbf{x}_1 - 2\mathbf{x}_2 + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. OLS effect sizes are compared as a baseline. In panel (A), outcomes variables are generated with uncorrelated predictors; while in panel (B), the two covariates have a Pearson correlation coefficient of $\rho = 0.999$.

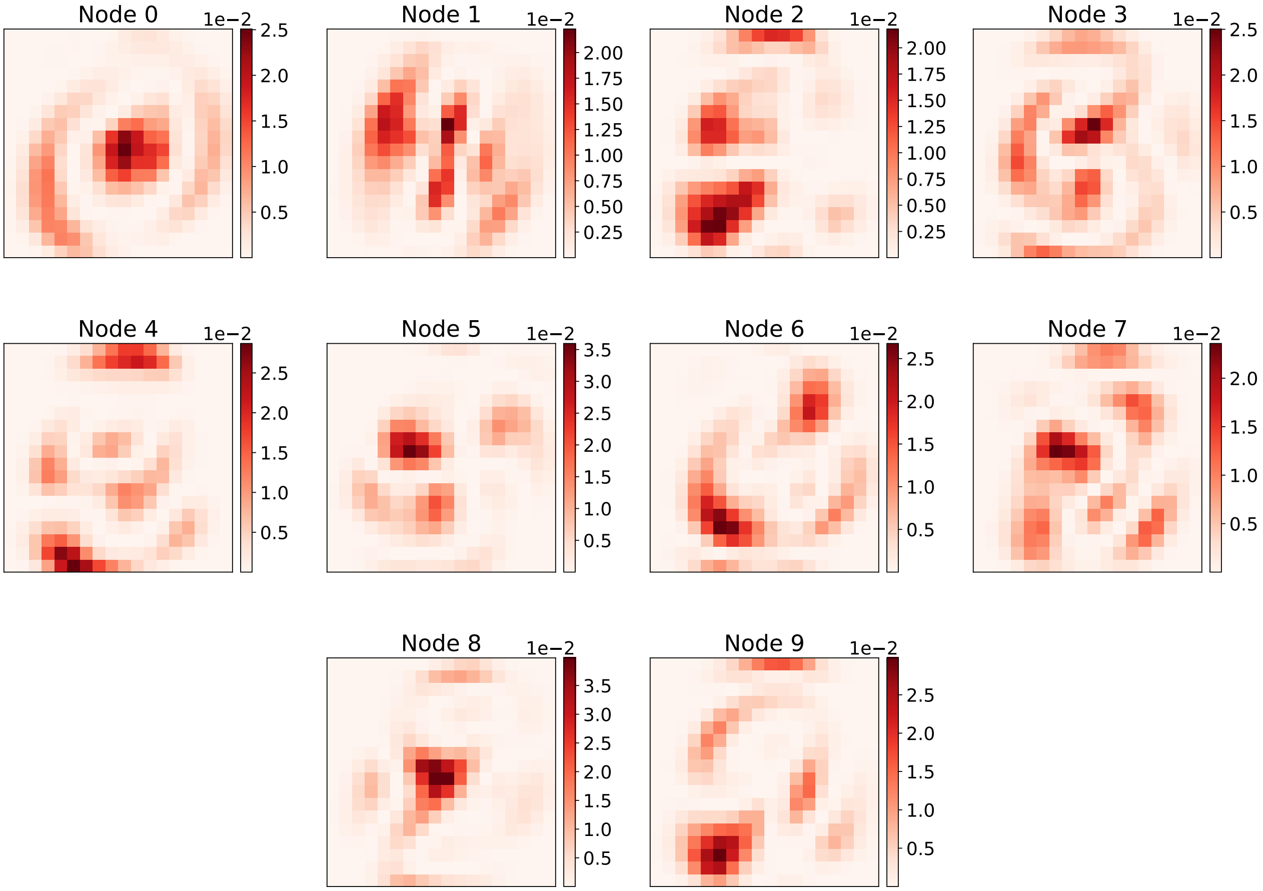

Figure A2. RATE values for a Bayesian neural network trained on the whole MNIST dataset. For the 10-class problem there are 10 output nodes and 10 corresponding RATE values for each pixel. The digit corresponding to each node can be seen clearly in several examples (e.g. 0, 3, 6, 8, and 9).

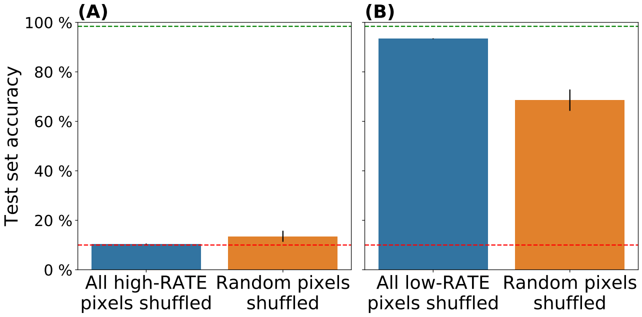

Figure A3. (A) Test set accuracies for MNIST when the 221 pixels with a RATE value greater than $1/p$ for any class are shuffled (blue), versus when the same number of any randomly-selected set of pixels are shuffled (orange). Shuffling pixels with high RATE values reduces the test accuracy to that of a random classifier (red dotted line), while shuffling the same number of any randomly-selected set of pixels does not. (B) Test set accuracies for MNIST when the 103 pixels with a RATE value less than $1/p$ for every class are shuffled, versus when the same number of any randomly-selected set of pixels are shuffled. Compared to the accuracy on the unshuffled test set (green dotted line), shuffling the low-RATE pixels only reduces accuracy by 5.0%. Shuffling the randomly-selected pixels reduces the accuracy by 19.8%.

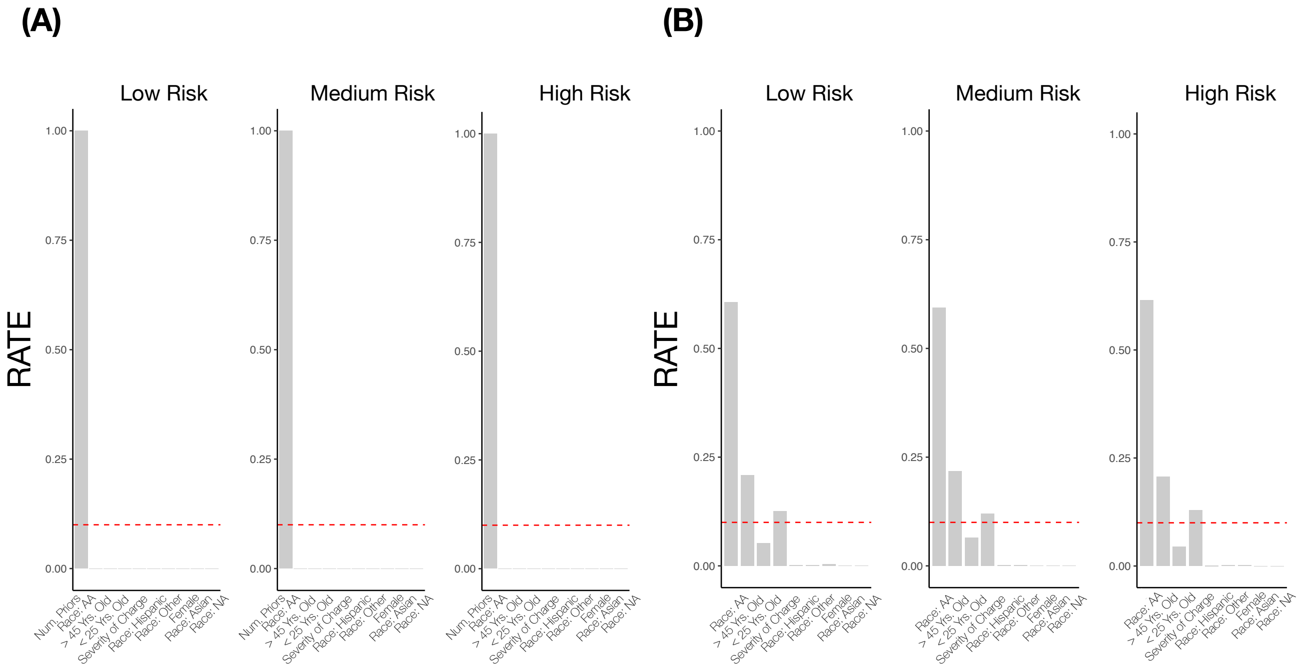

Figure A4. (A) First order RATE values from the Bayesian DNN. (B): First order RATE values from Bayesian DNN when number of prior offenses is omitted from the analysis. The dashed line is drawn at the level of relatively equal importance (i.e. $1/p$ and $1/(p-1)$ for panels (A) and (B), respectively).