Dropout Feature Ranking for Deep Learning Models

Chun-Hao Chang, Ladislav Rampasek, and Anna Goldenberg

University of Toronto, Department of Computer Science, Toronto, ON, Canada
The Hospital for Sick Children, Toronto, ON, Canada
Vector Institute, Toronto, ON, Canada
{kingsley,rampasek}@cs.toronto.edu, anna.goldenberg@utoronto.ca

Abstract. Deep neural networks are a promising technology achieving state-of-the-art results in biological and healthcare domains. Unfortunately, DNNs are notorious for their non-interpretability. Clinicians are averse to black boxes and thus interpretability is paramount to broadly adopting this technology. We aim to close this gap by proposing a new general feature ranking method for deep learning. We show that our method outperforms LASSO, Elastic Net, Deep Feature Selection and various heuristics on a simulated dataset. We also compare our method in a multivariate clinical time-series dataset and demonstrate our ranking rivals or outperforms other methods in Recurrent Neural Network setting. Finally, we apply our feature ranking to the Variational Autoencoder recently proposed to predict drug response in cell lines and show that it identifies meaningful genes corresponding to the drug response.

Keywords: Interpreting black-box, Feature ranking, Deep learning, Variational dropout

1 Introduction

Deep neural networks (DNNs) have started to come out as top performers in biology and healthcare including genomics [1], medical imaging [2], EEG[3] and EHR [4]. However, DNNs are black-box models and notorious for their non-interpretability. In the fields of biology and healthcare, to derive hypotheses that could be experimentally verified, it is paramount to provide information about which biological or clinical features are driving the prediction. Often, data is very expensive to collect, thus it is important to generate experimental designs that will collect the right data leading to the highest accuracy within reasonable budget. Therefore, there is a strong need for feature ranking for deep learning methods to advance their use in biology and healthcare. We aim to close this gap by proposing a new general feature ranking method for deep learning.

We propose to rank features by variational dropout [5]. Dropout is an effective technique to regularize neural networks by randomly removing a subset of hidden node values and setting them to 0. In this work we use the Dropout concept on the input feature layer and optimize the corresponding feature-wise dropout rate. Since each feature is removed stochastically, our method creates a similar effect to feature bagging [6] and manages to rank correlated features better than other non-bagging methods such as LASSO. We compare our method to another feature bagging method Random Forest (RF), and other methods such as LASSO, ElasticNet, and several heuristics and show that our approach reaches state-of-the-art result for a variety of dataset sizes and levels of feature correlation in a simulated setting. Then we test it on a multivariate clinical time-series dataset to interpret Recurrent Neural Network (RNN). We show that our method outperforms other methods in several classifier's setting. Finally, we test our method on a real-world drug response prediction problem using a previously proposed Variational Autoencoder (VAE). In this proof-of-concept application, we show that our method identifies genes relevant to the drug-response.

2 Related Work

Many previously proposed approaches to interpret DNNs focus on interpreting a decision (such as assigning a particular classification label in an image) for a specific example at hand (e.g. [7–15]). In this case, a method would aim to figure out which parts of a given image make the classifier think that this particular image

should be classified as a dog. These methods are unfortunately not easy to use for the purpose of feature selection or ranking, where the importance of the feature should be gleaned across the whole dataset.

There are also some attempts [16, 17] to use autoencoders to learn a meaningful low-dimensional latent space. Autoencoder is a neural network that goes through a bottleneck layer and reconstructs its input. These approaches select features in the latent space by the smallest reconstruction error [16] or using a random forest [17]. These approaches have to train an additional autoencoder, and their selection is not directly related to the classifier.

Several works have mentioned using variational dropout to achieve better performance [5] [18], have a Bayes interpretation of dropout [19], or compress the model architecture [20]. These works focus on tuning the dropout rate to automatically get the best performance, but do not consider applying it to the feature ranking problems.

Li et al. [21] proposed Deep Feature Selection (Deep FS). Deep FS adds another hidden layer to the network with one connection per input node to this hidden layer (of the same size as input) and uses an ℓ_1 penalty on this layer. The weights between these layers are initialized to 1 but since they are not constrained to $[0, 1]$, they can become large positive and negative values. Thus, this additional layer can amplify a particular input and will need to be balanced within the original network architecture. Additionally, using ℓ_1 penalty prevents Deep FS from selecting correlated features, important in many biological and health applications.

Finally, several works also targeted interpreting features in a clinical setting. Che et al. [22] uses Gradient-Boosted Trees to mimic a recurrent neural network on a healthcare dataset and achieves comparable performance. Nezhad et al. [17] interprets the clinical features by autoencoder and random forest. Suresh et al. [23] uses Recurrent Neural Network to predict the clinical dataset and use the ranking heuristics called 'Zero' in our settings. These approaches rely on additional decision trees architecture to learn the features, or use heuristics which have a weaker ranking performance in our experiments.

3 Methods

3.1 Variational Dropout

Dropout [24] is one of the most effective and widely used regularization techniques for neural networks. The mechanism is to inject a multiplicative Bernoulli noise for each hidden unit within a neural network. Specifically, during forward pass, for each hidden unit k in layer j a dropout mask $z_{jk} \sim \text{Bern}(z|\theta_{jk})$ is sampled. The original hidden node value h_{jk} is then multiplied by this mask $h'_{jk} = h_{jk}z_{jk}$, which stochastically sets the hidden node value to h_{jk} or 0.

Variational dropout [19] optimizes the dropout rate θ as a parameter instead of it being a fixed hyperparameter. For a neural network $f(\boldsymbol{x})$, given a mini-batch of size M (sampled from training set of N samples) and a dropout mask $\boldsymbol{z}$, the loss objective function that follows from the variational interpretation of dropout can be written as:

$$L(\theta) = -\frac{1}{M}\sum_{i=1}^{M} \log p(\boldsymbol{y}_i|f(\boldsymbol{x}_i, \boldsymbol{z}_i)) + \frac{1}{N} KL(q_\theta(\boldsymbol{z})||p(\boldsymbol{z})). \tag{1}$$

Here, $\boldsymbol{z}_i \sim q_\theta(\boldsymbol{z})$, where $q_\theta(\boldsymbol{z})$ is the variational mask distribution and $p(\boldsymbol{z})$ is a prior distribution.

3.2 Feature Ranking Using Variational Dropout

Figure 1 shows our approach. To analyze which features are important for a given pre-trained model $\mathcal{M}$ to correctly predict its target variable y, we introduce Dropout Feature Ranking (Dropout FR) method. In our method we add variational dropout regularization to the input layer of $\mathcal{M}$. To achieve minimum loss, the Dropout FR model should learn small dropout rate for features that are important for correct target prediction by the analyzed model $\mathcal{M}$, while increasing the dropout rate for the rest of unimportant features. Specifically, given D features, we set a variational mask distribution $q_{\boldsymbol{\theta}}(\boldsymbol{z}) = \prod_{j=1}^{D} q(z_j|\theta_j) = \prod_{j=1}^{D} \text{Bern}(z_j|\theta_j)$ as a fully factorized distribution. This gives us a feature-wise dropout rate θ_j where magnitude indicates the importance of feature j.

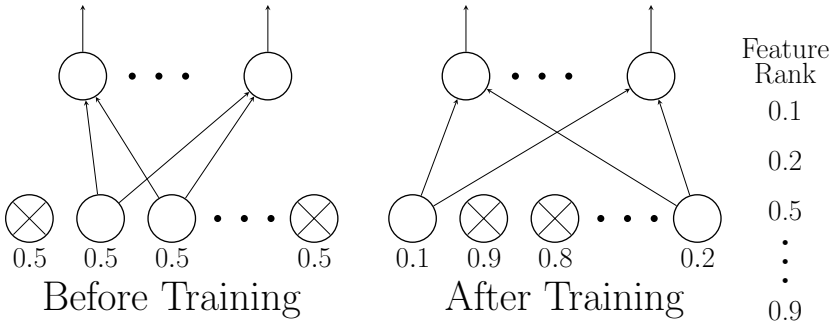


Fig. 1: Dropout feature ranking diagram. Before training (Left), the dropout rate for each feature is initialized to 0.5. After training (Right), each feature gets a different dropout rate. We then rank all features based on the magnitude of the dropout rate - the lower the magnitude, the higher the rank.

Instead of having a $KL(q_\theta(\boldsymbol{z})||p(\boldsymbol{z}))$ in the equation 1 to regularize the dropout distribution $q_\theta(\boldsymbol{z})$, we directly penalize the number of existing features (features not dropped-out). This avoids the need to set the prior dropout rate $p(\boldsymbol{z})$ and is aligned with the ℓ_0 penalty for linear regression [25]. Our loss function can thus be written as:

$$L(\theta) = -\frac{1}{M}\sum_{i=1}^{M}\log p_{\mathcal{M}}(\boldsymbol{y_i}|f(\boldsymbol{x}_i \odot \boldsymbol{z_i})) + \frac{\lambda}{M}\sum_{i=1}^{M}\sum_{j=1}^{D} z_{ij} \tag{2}$$

where $\boldsymbol{z}_i \sim q_\theta(\boldsymbol{z})$ and λ is determined by cross validation. To optimize $L(\theta)$ w.r.t. the parameters θ, we need to backpropagate through $\boldsymbol{z}$. Since $\boldsymbol{z}$ is discrete, we could use REINFORCE [26] estimator, however, it suffers from high variance and has poor convergence. Recently, Gal et al. [5] have shown how to optimize the dropout rate θ by 'Concrete relaxations' [27]. This estimator has lower variance and is easier to train, thus we utilize it in all our experiments.

4 Results

First, we simulate a binary classification dataset with known feature importance ranking and compare our results with other baselines to show the advantages of our method in a setting where the ground truth is known. We then compare our approach on PhysioNet, a clinical time series dataset. Finally, we apply our approach to a drug-response task to understand which genes contribute to drug response.

4.1 Compared methods

We compare our approach to LASSO, Random Forest, Deep Feature Selection (Deep FS) and other heuristics. LASSO uses an ℓ_1 penalty while Elastic Net uses a mix of ℓ_1 and ℓ_2 penalty, in which the feature importance is derived from the order each feature goes to 0 as the penalty increases. For Elastic Net, we choose $\alpha = 0.5$ to balance between ℓ_1 and ℓ_2 penalties. For Random Forest, feature importance depends on the decrease of impurity for each feature across different trees to determine the importance.

Deep FS [21] uses ℓ_1 penalty to select features. Following optimization, we use the magnitude of the connection weight as a proxy of the importance of each variable. Note that to correctly evaluate importance of each feature and to ultimately rank features, we should examine the order with which weights drop to 0 as the ℓ_1 penalty increases. This would require hundreds of manual settings of the ℓ_1 penalty hyperparameter, which is not scalable, so we use the connection weight instead.

Finally, we use two heuristics to rank features in a DNN. We call the first approach 'Zero' method: we zero out one feature at a time and rank feature importance based on the corresponding increase in the training

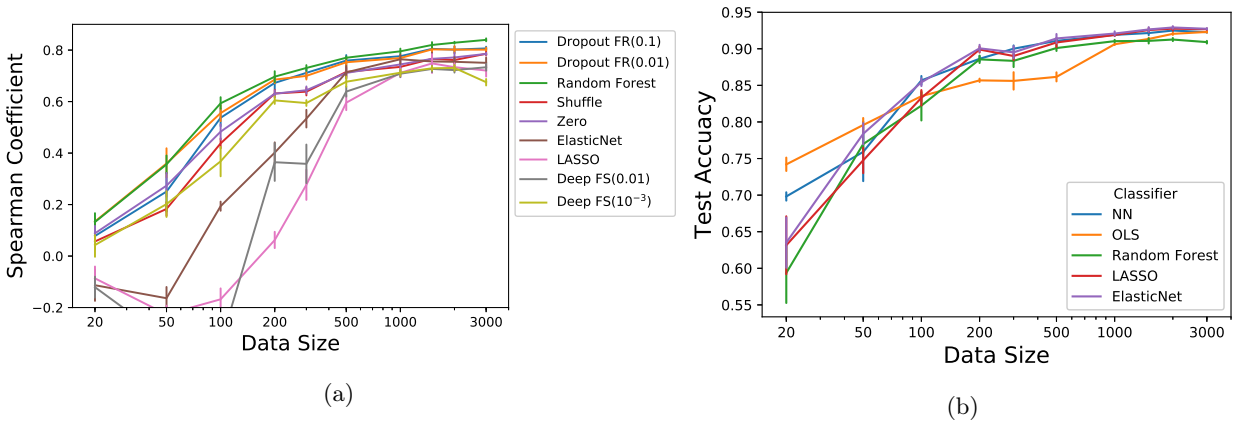
loss. Our second method is called 'Shuffle': for each feature we permute its values across the samples and evaluate importance by the increase of the training loss.

All training is done on a training set (as described below) and the ultimate performance is reported on an independent test set.

4.2 Simulation

To compare the feature ranking methods, we generated a dataset with 40 weakly correlated features and 60 purely noisy features. For each instance $y \in \{-1, 1\}$ and each feature $d \in \{1, \ldots, 40\}$, x_d was sampled from a Gaussian distribution $N(0.01 \cdot d \cdot y, 1)$. The noisy features were simulated as $x_{d \in \{41,\ldots,100\}} \sim N(0, 1)$. Thus, the most important feature is the 40-th and the least important feature is the first. We normalized each feature to have mean 0 and variance 1. We split the generated dataset of size N (see below) into training (90%) and validation (10%) and generated a separate test set with 1000 samples to evaluate each performance. We trained a 2-layer neural network with hidden size 50 and ReLU activation function and compare the classifiers using this framework.

We use grid search in 5-fold cross validation to determine the penalty hyperparameter λ for both Deep FS and our method when dataset size is 1000. We find that the λ that is best for feature selection is usually bigger than the best one for the accuracy [25]. Consequently, we show the performance for two values of λ, one for the best validation accuracy, and the other one for the next higher λ in our grid search. We search the value for $\lambda \in \{10^{-5}, \ldots, 1\}$, then fix these hyperparameters in all of our experiments. We compare 2 ranking coefficient: Spearman coefficient and Kendal Tau coefficient. Since they have similar trends, we show Kendall coefficient result in the appendix A.



(a) (b)

Fig. 2: Simulation varying the size of the dataset. (a) Spearman coefficient to ground truth importance for different data size. Higher is better. FS: feature selection, FR: Feature Ranking. (b) Test set accuracy for different data size. The error bar is the confidence interval with 68% probability sampled for 10000 times.

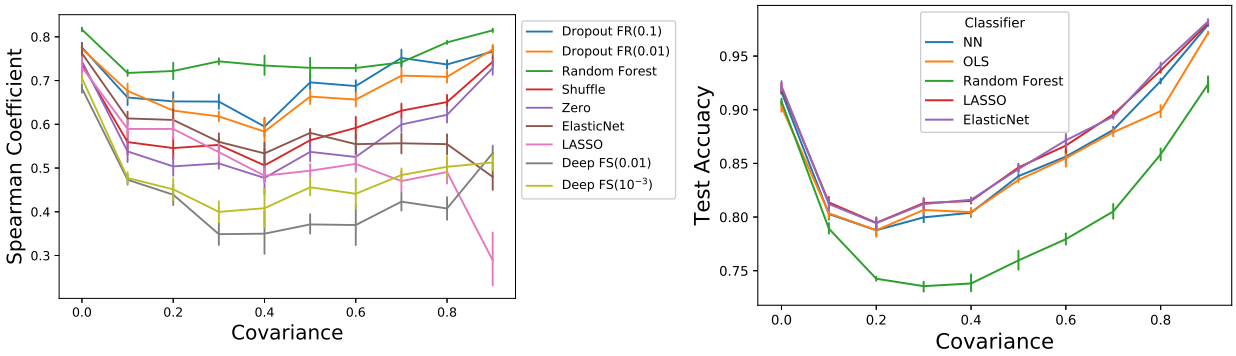
Varying the size of the dataset We varied the size of the simulated dataset N from 20 to 3000 samples (holding 10% for validation), simulating 5 datasets per sample size. Figure 2a shows Spearman coefficient between identified and true feature ranking for each dataset size. We report $\lambda = 0.01, 0.1$ for our method and $\lambda = 10^{-3}, 10^{-2}$ for Deep FS. Our method (Dropout FR) performs similarly to Random Forest. These are the top 2 best performers. The two heuristics, Zero and Shuffle, perform worse especially when the data size is small. The widely used ElasticNet, LASSO and Deep FS perform substantially worse.

Figure 2b shows the test set accuracy for each of the classifiers. We compare the neural network (NN), Ordinary Least Squares (OLS), LASSO, Elastic Net and Random Forest (RF). As the data size increases,

NN starts outperforming OLS and RF, performing similarly to LASSO and Elastic Net. Note that although RF does well on feature ranking, it suffers from lower accuracy for all dataset sizes we tested.

Varying correlation between features To test how correlation among features could confound feature ranking, we fixed the data size to 1000 and increased the correlation between the 40 relevant features. Specifically, we sample the features from a multivariate Gaussian, with the diagonal values of the co-variance set to 1 and off-diagonal entries varying from 0 to 0.9. Note that when the off-diagonal entries are all 0, it corresponds to the previous experiment.

Figure 3a demonstrates Spearman coefficient as feature covariance increases. As expected, LASSO and Deep FS perform poorly since ℓ_1 penalty tends to arbitrarily choose only one of the correlated features. We again see that Random Forest and our method (both settings) are the top performers across different correlations. Our method significantly outperforms all methods except RF. Figure 3b shows that RF is again the worst performer, the rest of the methods performing comparably.



(a) (b)

Fig. 3: Simulation varying the covariance of the dataset. (a) Spearman coefficient to ground truth importance for different covariance. Higher is better. FS: feature selection, FR: Feature Ranking. (b) Test set accuracy for different covariance.

4.3 Predicting in-hospital mortality

In this experiment, we evaluate the performance of our method using a multivariate time-series clinical dataset to determine the importance of clinical covariates in predicting in-hospital mortality. This dataset, from PhysioNet 2012 Challenge [28], is a publicly available collection of multivariate clinical time series with 8000 intensive care unit (ICU) patients. It contains 37 patient measurements within the first 48 hours in the ICU. The goal is to predict the in-hospital mortality as a binary classification problem. We use the only publicly available *Training Set A* subset which contains $4,000$ patient measurements with 554 patients having the positive mortality labels.

We following the preprocessing of Lipton et al. [29] work. First, we use binary features indicating whether or not a feature was measured at a given time point. If a feature was not measured, we set the binary variable to 1 and if it was measured, we set it to 0. These reverse-indicator variable concatenated with the original 37 features results in 74 features in total. Second, we bin the input features into 1-hour intervals, take average of multiple measurements within 1-hour time window, and impute missing values with 0. Finally, we normalize each feature to zero mean and unit variance except for the binary features. These lead to a time-series with 48 time points and 74 features. We split the dataset randomly into 80%, 10%, and 10% as training, validation and test set, respectively, and repeat the procedure 40 times.

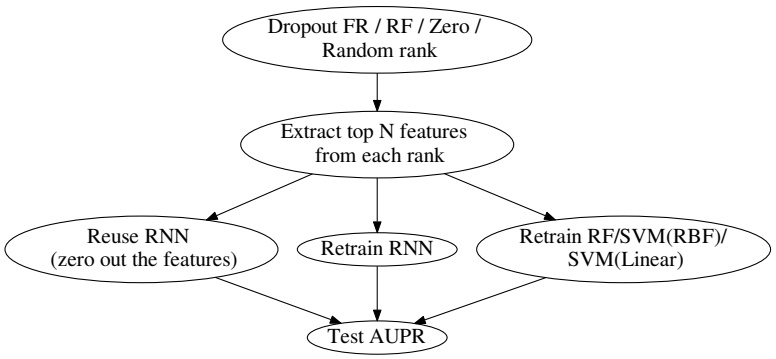


Fig. 4: Our procedure to compare different ranking. We move the results of reusing RNN by zeroing out features to appendix B.

Table 1: Test set performance of different methods

Method	AUPR	AUROC
SVM-linear	0.317 ± 0.063	0.709 ± 0.030
SVM-RBF	0.428 ± 0.070	0.790 ± 0.024
RF	0.415 ± 0.062	0.796 ± 0.023
RNN	$\mathbf{0.448 \pm 0.063}$	$\mathbf{0.808 \pm 0.026}$

We follow the architecture used in Che et al. [30]. Our RNN consists of a Gated Recurrent Unit (GRU) [31] with hidden node size 64, dropout 0.3 on input features, and dropout 0.5 as well as batch normalization on RNN's final output. We apply early stopping on validation set during RNN training, and apply Dropout FR and Zero method on the validation set to avoid overfitting. We use 10-fold cross validation to select λ. For random forest, we use $1,000$ trees and sum the feature importance across all the time points (since in RF each timepoint is considered independently for each of the features), including original feature and its corresponding reverse-indicator features.

Our procedure is described in Figure 4. We compare Dropout FR, RF, Zero ranking method, and random ranking serving as a baseline. To understand which feature ranking is the best, we subset the features from each ranking and evaluate the test set AUPR as a function of the number of features. We evaluate each of the rankings using 4 different classifiers: RNN, RF, SVM with RBF kernel (SVM-RBF) and SVM with linear kernel (SVM-Linear). We evaluate RNN with two settings. We call the first setting 'zero-out': after taking top N features, we set the rest of the features to 0 and evaluate the test AUPR using the already trained RNN. The second setting is 'retrain': we retrain RNN using only the top N features, scaling down hidden node size to avoid overfitting. The size of the hidden layer is set to be proportional to the number of input features, with the minimum size set to 5 to avoid underfitting. In our experiments we have found that 'zero-out' and 'retrain' settings perform similarly and henceforth report 'retrain' results in the main paper as all the other compared methods are retrained using a smaller set of features. The results for the 'zero-out' setting are in the Appendix B.

First, we compare each classifier's performance on the PhysioNet dataset. Table 1 shows comparative performance of the Recurrent Neural Network (RNN), Random Forest (RF), SVM-linear and SVM-RBF. RNN outperforms all other methods in test set AUPR and AUROC.

In Figure 5, we compare 4 feature rankings described above using RNN, RF, SVM-RBF and SVM-Linear. In RNN classifier (Fig. 5a), Dropout FR slightly outperforms Random Forest with significant difference when

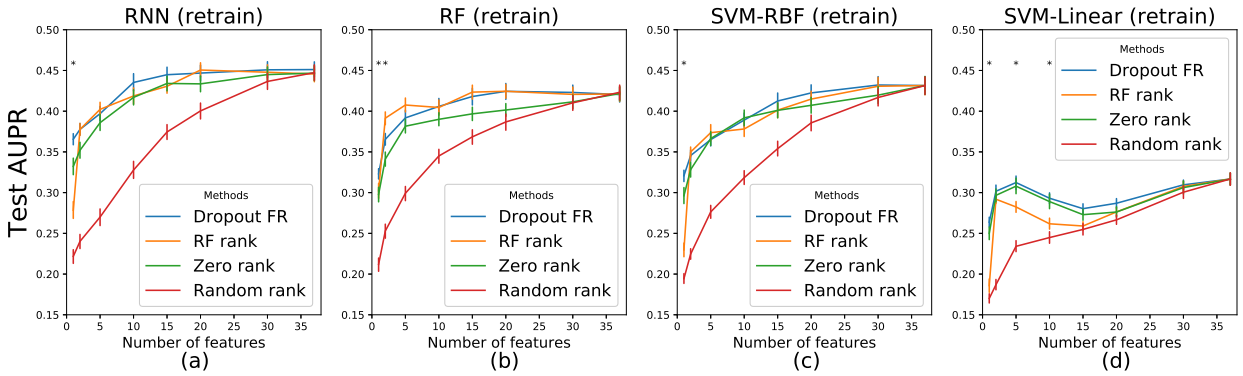


Fig. 5: Test set AUPR for varying number of features for different ranking method in RNN, RF, SVM-RBF and SVM-linear classifiers. Symbol $*$ represents the significant difference between Dropout FR and RF calculated under one-tailed Wilcoxn Rank Sum Test.

Table 2: Top 10 features selected from RF rank and Dropout FR

	RF rank	Present rate (%)		RNN rank	Present rate (%)
1	Urine	1.87	1	GCS	0.86
2	GCS	0.86	2	Urine	1.87
3	HR	2.43	3	BUN	0.20
4	SysABP	1.46	4	MechVent	0.41
5	Temp	1.00	5	Temp	1.00
6	NISysABP	1.14	6	HR	2.43
7	NIMAP	1.12	7	Lactate	0.11
8	Weight	1.43	8	Weight	1.43
9	NIDiasABP	1.14	9	NIDiasABP	1.14
10	MAP	1.45	10	SysABP	1.46

using 1 feature, and clearly outperforms the Zero method and the random baseline. For the rest of the 3 classifiers, Dropout FR has slightly worse performance than RF in RF classifier, (Fig. 5b), perform similarly in SVM-RBF (Fig. 5c) and outperform RF in SVM-Linear classifier (Fig. 5d). Zero method is worse than Dropout FR across all settings.

Overall, we find that RF rank is better when using RF classifier, while Dropout FR performs slightly better in RNN and SVM-linear classifiers, having similar performance in SVM-RBF. Interestingly, we find that the top feature selected by RF yields significantly worse performance than Dropout FR in all 4 classifiers.

In Table 2, we show the top 10 features in RF and Dropout FR. Overall these two approaches select similar features. We find that the difference in ranking of 'Urine' and 'GCS' features (RF selects 'GCS' as second) is the reason for the inferior RF performance observed in Figure 5. The table also demonstrates that feature importance does not simply follow the frequency of the features in the dataset for either of the methods.

4.4 Drug Response Prediction

We apply our method to a real-world drug response dataset to find which genes determine drug response using the semi-supervised variational autoencoder (SSVAE) [32] model applied to this task by Rampasek et al. [33]. The SSVAE takes gene expression of 903 preselected genes as input and performs a binary classification to find whether the given cell line responds to the drug.

We examined genes contributing to the response of bortezomib, a drug commonly used in multiple myeloma patients. The gene that was ranked the highest by our algorithm (with lowest dropout probability), NR1H2, was previously found to be indicative of Multiple Myeloma (MM) non-response to anti- agents such as bortezomib [34]. The second ranked gene, BLVRA, is known to be amplified in cells sensitive to anti-MM treatment, such as bortezomib [35]. Interestingly, BVLRA was also ranked second by RF (and not ranked highly by t-test). The gene ranked first by both RF and t-test is FOSL1 which was not directly found to be linked to response by bortezomib, but is tangentially related through osteoclass process (FOSL1 helps with differentiation into bone cells and there is a secondary effect of bortezomib to prevent bone loss during inflammation processes). Overall, we found that ranking of RF follows rather closely ranking by t-test. Dropout FR ranking was significantly different, capturing the importance of the ranking for the SSVAE classification.

5 Discussion

In this work we proposed a new general approach for understanding the importance of features in deep learning. We believe that variational dropout works well because it acts similarly to feature bagging [6] by subsetting the features during training. It allows to decouple correlated variables in certain instances, and optimizes the corresponding feature-wise dropout rate. This may also be the reason for the excellent performance by random forest which we have observed in our experiments and also the reason for poor performance of ℓ_1 used in LASSO and Deep FS.

Random Forest certainly seems to be a great method for feature ranking in most of our experiments for the reasons described above. The goal of our study, though, was to create a general method for feature ranking in a deep learning framework. Indeed, we tested our approach in a simple feed-forward network, a recurrent neural network and a semi-supervised variational autoencoder showing that Dropout FR is at least as good and in some cases better than the best ranking method, Random Forest. As the clinical and biological datasets continue to grow, the power of deep learning will likely supersede non deep methods, as has been shown in many research areas. Our method is essential to provide interpretability to these methods, to ensure acceptance in the biological and clinical domains.

6 Conclusion

We propose a new way of feature ranking for deep neural networks by variational dropout. Our approach compares favorably in the simulation, and uncover important genes in drug response datasets. Although Random Forest performs similarly in feature ranking, our proposed Droupout Feature Ranking can be applied to a great variety of deep learning architectures. Compared to LASSO, our method is applicable to settings with nonlinear relationships between features, which is crucial in highly complicated biological systems and medical domain.

Bibliography

[1] Hui Y Xiong, Babak Alipanahi, Leo J Lee, Hannes Bretschneider, Daniele Merico, Ryan KC Yuen, Yimin Hua, Serge Gueroussov, Hamed S Najafabadi, Timothy R Hughes, et al. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218):1254806, 2015.

[2] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639): 115–118, 2017.

[3] Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*, 2017.

[4] Joseph Futoma, Sanjay Hariharan, Mark Sendak, Nathan Brajer, Meredith Clement, Armando Bedoya, Cara O'Brien, and Katherine Heller. An improved multi-output gaussian process rnn with real-time validation for early sepsis detection. *arXiv preprint arXiv:1708.05894*, 2017.

[5] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete Dropout. *arXiv:1705.07832 [stat]*, 2017.

[6] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[7] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[8] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. February 2016.

[10] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[11] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[12] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

[13] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

[14] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*, 2017.

[15] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *arXiv preprint arXiv:1705.07857*, 2017.

[16] Qin Zou, Lihao Ni, Tong Zhang, and Qian Wang. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 12(11):2321–2325, 2015.

[17] Milad Zafar Nezhad, Dongxiao Zhu, Xiangrui Li, Kai Yang, and Phillip Levy. Safs: A deep feature selection approach for precision medicine. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 501–506. IEEE, 2016.

[18] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583, 2015.

[19] Shin-ichi Maeda. A bayesian encourages dropout. *arXiv preprint arXiv:1412.7003*, 2014.

[20] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. *arXiv preprint arXiv:1701.05369*, 2017.

[21] Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

[22] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*, 2015.

[23] Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical intervention prediction and understanding using deep networks. *arXiv preprint arXiv:1705.08498*, 2017.
[24] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.
[25] Kevin P Murphy. *Machine learning: a probabilistic perspective*. 2012.
[26] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
[27] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.
[28] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220, 2000.
[29] Zachary C Lipton, David C Kale, and Randall Wetzel. Modeling missing data in clinical time series with rnns.
[30] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.
[31] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
[32] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.
[33] Ladislav Rampasek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr.VAE: Drug Response Variational Autoencoder. *arXiv:1706.08203 [stat]*, 2017.
[34] Jasmin R Agarwal, Qiuju Wang, Toshihiko Tanno, Zeshaan Rasheed, Akil Merchant, Nilanjan Ghosh, Ivan Borrello, Carol Ann Huff, Farhad Parhami, and William Matsui. Activation of liver x receptors inhibits hedgehog signaling, clonogenic growth, and self-renewal in multiple myeloma. *Molecular cancer therapeutics*, 13(7):1873–1881, 2014.
[35] GP Soriano, L Besse, N Li, M Kraus, A Besse, N Meeuwenoord, J Bader, B Everts, H den Dulk, HS Overkleeft, et al. Proteasome inhibitor-adapted myeloma cells are largely independent from proteasome activity and show complex proteomic changes, in particular in redox and energy metabolism. *Leukemia*, 30(11):2198, 2016.

Appendix

A Comparison in Kendall Tau correlation

We show our additional comparison in Kendall Tau correlation. Figure A.1 shows Kendall Tau correlation for varying number of data size. Random forest and Dropout FR are still the top 2 best methods, with Deep FS, Shuffle and Zero method in the middle. The worst 2 methods are ElasitcNet and LASSO.

Figure A.2 shows Kendall Tau correlation for varying number of correlation in 1000 datasize. Random forest and Dropout FR are still the top 2 best methods. Elastic Net and two heuristics, Zero and Shuffle, perform in the middle with ell_1 based methods such as LASSO and Deep FS perform the worst.

B Zero-out setting

Figure B.1 shows the comparison of the zero-out method compared to retraining. In the left panel (zero-out setting), Dropout FR significantly outperforms RF when using 1 and 10 features (Wilcoxon rank-sum one-sided test), also beating the Zero heuristic and Random baseline.

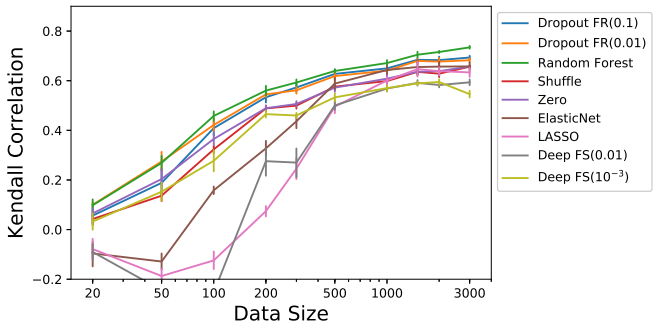


Fig. A.1: Kendall Tau coefficient to ground truth importance for different data size. Higher is better. FS: feature selection, FR: Feature Ranking.

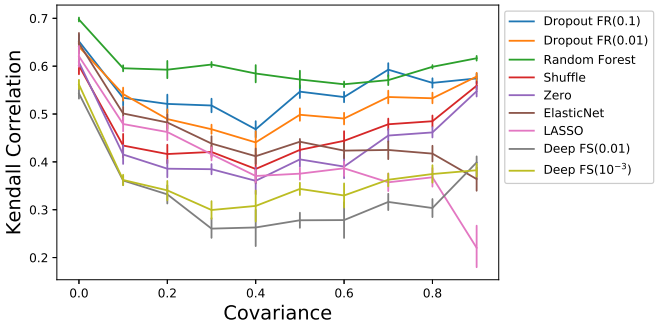


Fig. A.2: Kendall Tau coefficient to ground truth importance for different correlation. Higher is better. FS: feature selection, FR: Feature Ranking.

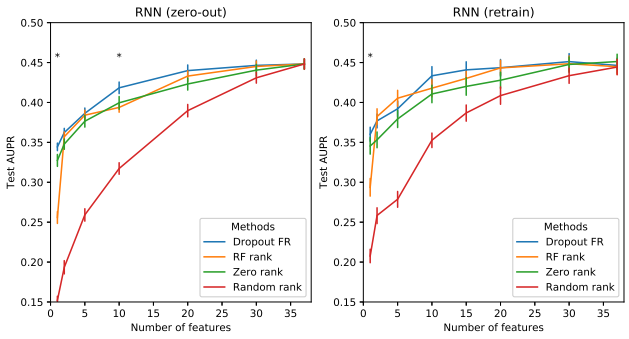


Fig. B.1: Test set AUPR for varying number of features for different ranking method in RNN with zeroing out the features (Left) and retraining (Right). Symbol $*$ represents the significant difference between Dropout FR and RF calculated under one-tailed Wilcoxon Rank Sum Test.