Probing Decision Boundaries in Cancer Data Using Noise Injection and Counterfactual Analysis

Rajeev Jain*, Ashka Shah¶, Jamaludin Mohd-Yusof†, Justin M. Wozniak‡, Thomas S. Brettin§, Fangfang Xia§, Rick L. Stevens§

* Mathematics and Computer Science, Argonne National Laboratory, Lemont, IL, USA
† Computer, Computational and Statistical Sciences, Los Alamos National Laboratory, Los Alamos, NM, USA
‡ Data Science and Learning, Argonne National Laboratory, Lemont, IL, USA
§ Computing, Environment, and Life Sciences, Argonne National Laboratory, Lemont, IL, USA
¶ Department of Computer Science, University of Chicago, Chicago, IL, USA

I. Overview

Advanced analyses and computations based on gene expressions are prone to errors as they depend on experimental design, chemical operations/measurements and data analysis. The assembly and aggregation of such data for creating deep neural network models may further influence the accuracy of these analyses. For example, the CANDLE [1] NT3 Benchmark uses a table of laboratory-obtained data mapping RNA expression data to a normal or tumor designation, and is used to make predictions about given expression samples. In this work, we use the NT3 Benchmark to study the effects of injecting bad data at different rates to study the impacts on the resulting predictions. Our data manipulations include flipping classification labels (label noise) and introducing noise in gene expressions (feature noise).

The introduction of either correlated or uncorrelated label noise results in increased validation loss and reduced validation accuracy in the base model. When a random Gaussian noise with varying standard deviation to gene expressions for all the samples we see validation accuracy declines sharply indicating a poor fit compared to model with the original data. We introduce an abstaining [2] version of the model, which adds an extra abstention class, allowing the model to abstain when it is not confident of the prediction. This model retains accuracy while abstaining on progressively higher fractions of the data as more noise is injected. To identify which features might be the most susceptible to noise, we perform analysis with counterfactual examples which selects a subset of x indices out of the original 60,483 feature set. We find that using this method as a form of feature attribution identifies genes that are correlated well with several cancer types in the existing literature.

The contributions of this work include 1) a methodology to study model performance on incremental noise injection to input data and, 2) use of abstention classifiers to combat noisy data in the NT3 dataset, 3) a technique to highlight the decision boundary of the NT3 model and identify key genes for cancer research with counterfactual analysis.

II. Background

The NT3 benchmark attempts to separate tumor tissue from normal tissue using gene-expression-level sample signatures. The associated deep neural network (DNN) has an input layer for RNA sequence gene expression. It is a 1D convolutional network for classifying RNA-seq gene expression profiles into normal or tumor tissue categories.

The model is trained/tested on the matched normal-tumor gene expression profile pairs available from the NCI genomic data commons [3] respectively. The full set of expression features contains 60,483 float columns transformed from RNA-seq FPKM-UQ [4] values. Before our modifications, this model achieved around 98% classification accuracy. It is also useful for studying the difference and transformation of latent representation between normal and tumor tissues. The model also acts as a quality control check for synthetically generated gene expression profiles. Many researchers have focused on the detectability of some differentially expressed genes in RNA-seq expression results [5]. In a recent study [6] match lung cancer patients to appropriate treatments. They use DNN with RNA-seq dataset and "put the burden of learning non-invertible aspects of noise due to the measurement process on the neural network model." They also discuss the ability of their model to learn and make useful prediction despite non-random noise. In this work, we focus on studying and developing a framework for measuring the impacts of varying degrees of training data errors on the accuracy of the model.

The training data is a simple data table (in CSV format) of RNA expression values (floats). The prediction target, normal or tumor tissue, was encoded using an integer value of 0 or 1. The number of RNA values is 60,483 and the number of training samples was 1120, with 280 samples reserved for validation.

III. Basic noise injection

This study provides an insight into the amount of error in the training data (RNA sequence expressions) that is allowable without any significant loss of accuracy of the model.

A. Experiment configuration

The benchmark runs in this paper used the CANDLE hyperparameters [7] shown in Table I.

The error analysis during training was configured as follows. The NT3 output float probability (p) is compared against the

Network architecture		**Training limits**	
conv	[(64, 20, 1), (64, 10, 1)]	epochs	100
pool	[1, 10]	timeout	3600.0
dense	[200, 20]	**Noise injection**	
classes	2	noise_add	true
out_act	'softmax'	noise_gaussian	false
activation	'relu'	noise_level	0.2
Training settings		noise_correlated	true
optimizer	'sgd'	noise_labels	0.2
loss	'categorical_crossentropy'	feature_threshold	0.01
metrics	'accuracy'	feature_col	11180
batch_size	16		
learning_rate	0.002		
drop	0.0		

TABLE I
HYPERPARAMETERS USED FOR NT3 BENCHMARK IN THIS PAPER.

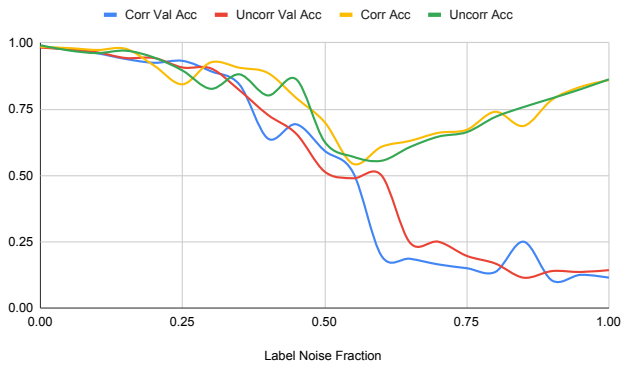

Fig. 1. Performance of the base network on correlated and uncorrelated label noise

ground truth Boolean normal/tumor value and assigned a loss value using categorical cross-entropy.

$$\mathcal{L}_{standard} = -\sum_{i=1}^{k} t_i \log p_i \tag{1}$$

where t_i is the target for the current sample, and p_i is the probability of i-th class. In our simple system, this is simply $\log(|\text{true\_value} - p|)$ for each row, and the row errors are summed over the whole validation set. This method is noted for its accommodation of imperfect predictions, while highly penalizing predictions that are far from the ground truth.

B. Resulting accuracy under noise

We introduce noise into the data in two ways, on the labels and on the features. For the labels, we consider both uncorrelated and correlated noise, set by **noise_correlated** . For uncorrelated noise, we randomly flip the normal/tumor labels on a fraction of the samples corresponding to the **noise_level**. In correlated label noise, we perform the same type of label flips, but only on samples where the expression on a certain gene (set by **feature_col**) is above a certain threshold (**feature_threshold**), so that the prevalence of noisy labels is correlated with the expression of that gene.

For correlated label noise, there are some factors to consider. The first is the number of samples in which the gene is expressed; if the gene is rarely above the threshold, then the total number of sample eligible for noise injection is small, and the effective noise added will likewise be small. Second, we must consider whether the gene is itself correlated with the normal/tumor label value; if not, then injection of label noise correlated with that gene is unlikely to confuse the training process. In this study we choose to inject correlated label noise on a feature which is highly correlated with the labels and sufficiently prevalent in the samples.

For feature noise, we again consider two types of noise injection: increasing the feature values by a fixed percentage, and introducing Gaussian random noise across the features.

Figure 1 shows the performance of the base model in the presence of correlated and uncorrelated label noise in the training set. In both cases, the training accuracy reaches a minimum at 50% noise but then increases for higher noise rates. The performance on the validation set (indicative of the models ability to correctly classify new data) degrades significantly above this threshold, as the training is essentially learning the incorrect pattern. We therefore choose to use an abstaining classifier, introduced below, which can recognize the presence of unreliable training data and abstain from classifying rather than making unreliable predictions.

C. Gaussian feature noise injection with Abstention

For analysis of sensitivity of RNA-sequence to DNN models, a simple Gaussian noise [8] is added to all the 60,483 normalized RNA sequence gene expression. Standard deviation of noise is increased from 0 to 0.5 with a step of 0.025 and a mean set to 0. The minimum and maximum values of gene sequence expression in the original training data are 0 and 1 respectively. The results in Figure 2 show that the model is able to achieve validation accuracy and validation loss values similar to no-noise with a noise of 0.1 (standard deviation or scale) value. After standard deviation of 0.1 the performance deteriorates rapidly causing validation accuracy to decrease and validation loss to increase.

A deep abstaining classifier [2], or DAC, introduced first for combating label noise, adds an extra class, the abstention class, to the original DNN and uses a custom loss function that permits abstention during training. This allows the DAC to abstain on (or decline to classify) confusing samples while continuing to learn and improve classification performance on the non-abstained samples. The custom loss function for an abstaining classifier is a modified version of the standard cross-entropy and given by,

$$\mathcal{L}(x_j) = (1 - p_{k+1})(-\sum_{i=1}^{k} t_i \log \frac{p_i}{1 - p_{k+1}}) + \alpha \log \frac{1}{1 - p_{k+1}} \tag{2}$$

where p_{k+1} is the probability of the abstention class and α is the penalty term for abstention.

In Figure 2 we compare the performance of the base and abstention models. Although the base model is able to retain good accuracy on the validation set, the abstention model nevertheless improves on this performance, while again abstaining on a fraction of the data approximately twice that of the naive estimate. This is consistent with the notion that

with only two classes, the base model can perform better by a factor of two due to "lucky guesses."

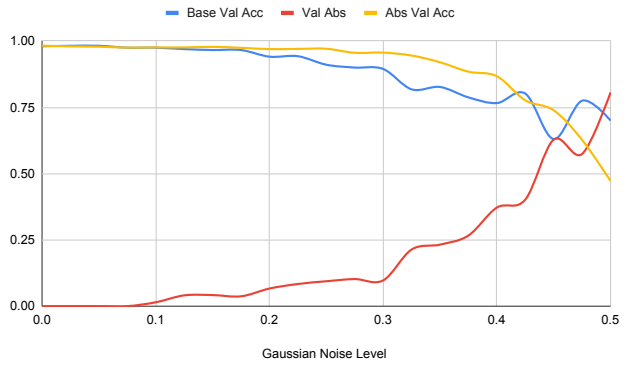

Fig. 2. Comparison of validation accuracy and abstention with the base model in the presence of Gaussian feature noise.

IV. Feature Attribution with Counterfactual Examples

While the previous results indicate the vulnerability of models subject to noisy data and the efficacy of an abstention model to combat noise, they fail to address the issue of how meaningful the noise injection is on the gene expression data. Especially in the 60k dimensional space, its unclear what it means to inject Gaussian noise across all features. Doing this does not illuminate which genes are most susceptible to noise or the model's decision boundary is weak – both of which would help quantify the error bounds for the NT3 model and dataset. To address this, we incorporate counterfactual example generation.

Counterfactual examples are an example-based interpretability technique used by the explainable AI community. The technique aims to mirror human counterfactual reasoning by finding a minimal subset of changes to an input example so that a machine learning model classifies the input into a different class [9]. The input representation can be text, tabular or image, and the computation of the counterfactual is simply an optimization problem — we want to minimize the distance between the generated example and the original original input, but maximize the misclassification error.

$$L(X'|X) = (f_t(X') - p_t)^2 + \lambda L_1(X', X) \tag{3}$$

Where X is the original input, X' is the generated counterfactual, t is the desired class for X', f_t is the model prediction on class t, p_t is the target probability of the desired class, and L_1 is the distance function. λ is a hyperparameter that tunes the contribution of the two competing terms[10].

Counterfactual examples can highlight decision boundaries and provide a picture of which inputs are sensitive to particular "perturbation vectors" (the difference between the generated example and the original input). From the perturbation vectors, we select features that surpass a certain threshold and label these genes as important for classification since they require a large change in the original value to flip the class for the counterfactual. We use this technique as a rough version of feature attribution that suits our use case. We note that many existing feature attribution techniques exist for neural networks that rank or provide a quantitative estimate for the importance of features in a model. This includes LIME, Shapley Values, Leave One Out, as well as others. However, for our purposes we are only interested in identifying a set of of important genes rather than the effect of each gene. Because our feature space is 60k dimensions, we opt for a simpler technique that runs more efficiently since it is simply an optimization of an objective function.

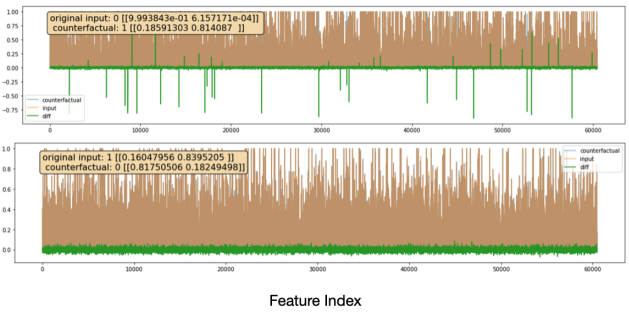

Fig. 3. Two examples of input samples, counterfactuals and perturbation vectors. Perturbation vectors (green) tend to be relatively uniform low random noise or sparse.

We generate counterfactual examples for each sample in the NT3 dataset using the Alibi Explain Python library [10]. We set $p_t = 0.9$ and $\lambda = 0.1$. An example of the generated perturbation is shown in Figure 3. Due to the sparsity of the perturbation, we hypothesize that the features correlated to spikes in the perturbation vector may have some biological meaning related to the sample's cancer type.

A. Experiment Configuration

Perturbation vectors were separated by class and clustered into groups using KMeans clustering with sklearn. The input samples corresponding to the two largest clusters became the targets for the noise injection. We used the centroid of each cluster as a representation of all perturbation vectors in the cluster, so that genes that have large spikes in the centroid are likely to have large spikes in most of the perturbation vectors in the cluster (Figure 5). For each centroid, we select genes that surpass t * the maximum magnitude in the vector, where t is a threshold fraction we varied from 0.5 to 0.9. For the two largest clusters, the fraction of targeted samples was varied from 10% to 90% of the whole cluster. Noise was injected by simply adding the indices of perturbation vector that exceeded the threshold to the corresponding sample. The label is kept the same for these targeted samples. We see that for a trained NT3 model without abstention, this does in fact lead to steeper degradation of accuracy compared a random set of 20 indices with Gaussian noise injection (Figure 4). Finally, we note that these genes also correspond to important markers for cancer in literature.

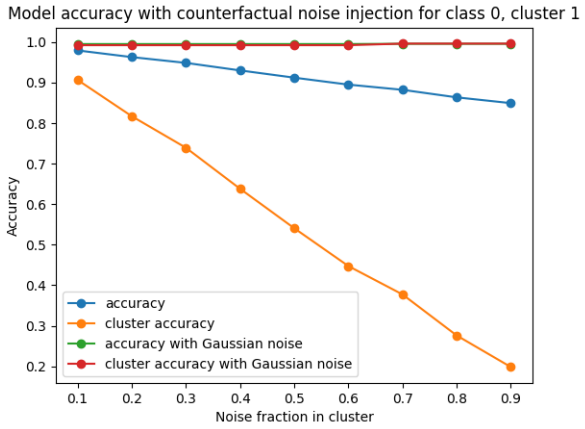

Fig. 4. Accuracy and cluster accuracy as a function of fraction of injected noise for counterfactual class 0, cluster 1. The threshold for the perturbation vector was set to 0.5, which resulted in 20 filtered indices. Random Gaussian noise is injected in 20 random indices for comparison.

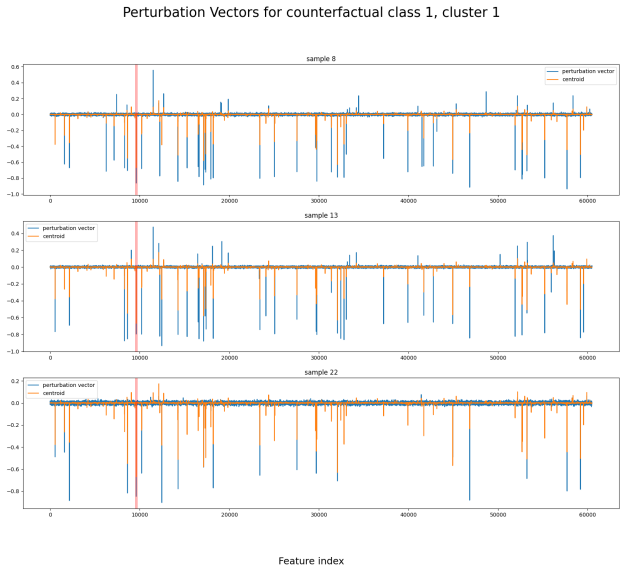

Fig. 5. Examples of perturbation vectors in cluster 0 for counterfactual class 0. Plotted with the centroid in orange, this shows consistency in spikes for the centroid and each perturbation vector. The red highlighted index corresponds to *PLOD2*.

B. Interpretability of Perturbation Vectors

The decision boundary of all the RNA expression data at which a sample changes from *having tumor* to *not having tumor* is identified by each of the perturbation vectors. Here, we try to find out some overexpressed RNA protein identified in clusters described in Section IV-A. With a threshold value of 75%, several gene symbols were identified as overexpressed – ten are listed in Table II. The *PLOD2* gene, found in multiple clusters is one major cancer identifiable gene, that is recently confirmed to be an unfavorable prognostic marker in renal, liver, lung, cervical and stomach cancer. *PLOD2* is considered to be the highway of cancer cell-migration as per a 2017 article: [11]. *LRTM1*, *RGS5*, *TP53I13*, *MAN1B1*, *TRRAP* and *TP53I13* have been found overexpressed and linked to studies

Symbol	Ensmbl	Cancer	Symbol	Ensmbl	Cancer
PLOD2	ENSG00000152952	Multiple	GP9	ENSG00000169704	Multiple
LRTM1	ENSG00000144771	Urothelial	TRRAP	ENSG00000196367	Ovarian
RGS5	ENSG00000232995	Lung	ZNF736P11Y	ENSG00000215537	Unknown
TP53I13	ENSG00000167543	Renal	SYN3-AS1	ENSG00000236054	Unknown
MAN1B1	ENSG00000177239	Bladder	TP53I13	ENSG00000167543	Bone

TABLE II
OVEREXPRESSED GENES USING A THRESHOLD OF 75% OF MAXIMUM AMONG ALL CLUSTERS DETAILED IN SECTION IV-A

of urothelial, lung, renal, bladder, ovarian and bone cancer respectively. The *TP53I13* gene inhibits tumor cell growth when overexpressed.

V. DISCUSSION

The notion of counterfactuals allows us to identify the minimal set of perturbations guaranteed to result in the migration of a sample across the local decision boundary of a trained classifier. Due to the relative sparsity of points ($\sim$ 1k points in a $\sim$ 60k dimensional space) it is difficult for such noise to affect training, since the resulting points can often be accommodated by a new decision boundary. That is, the same perturbation that will result in an incorrect prediction with a trained classifier will not result in an observable reduction in performance during training. Nevertheless, the correspondence between the genes which appear in the counterfactuals and those known to be significant in cancer supports the notion that the classifier is indeed discovering a relevant manifold within the high-dimensional space.

In contrast to perturbations on the data, perturbations on the labels are essentially guaranteed to generate interspersed data points, leading to blurring of the decision boundary and, in the case of the DAC, high abstention in the intermediate ranges of noise injection. In this case the abstaining classifier can be an indicator of poor label quality by flagging sets of samples for abstention rather than erroneously classifying them.

VI. SUMMARY

We presented results for the performance of both the base NT3 Benchmark and with the addition of the abstention class in the presence of various types of injected noise. For higher noise levels, the ability of the base network to correctly predict the normal/tumor classification (as measured by the validation accuracy) degrades significantly. Use of the abstaining classifier allows the model to learn when the labels have become unreliable and abstain from providing a prediction in that case. Such noise analysis studies would help set error tolerance for actual experimental measurement of RNA-seq gene expression profiles. We used counterfactual examples to understand the decision boundaries *"from normal to tumor"* and did further analysis to identify specific overexpressed genes. We showed that some of the genes identified by counterfactual analysis are known to correlate with various types of cancer, supporting the validity of the trained models, and believe that other genes found here might serve as a good starting point and even lead to new discoveries in the area cancer research.

References

[1] J. M. Wozniak, R. Jain, P. Balaprakash, J. Ozik, N. Collier, J. Bauer, F. Xia, T. Brettin, R. Stevens, J. Mohd-Yusof, C. G. Cardona, B. V. Essen, and M. Baughman, "CANDLE/Supervisor: A workflow framework for machine learning applied to cancer research," *BMC Bioinformatics*, vol. 19, no. 18, p. 491, 2018. [Online]. Available: https://doi.org/10.1186/s12859-018-2508-4

[2] S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, and J. Mohd-Yusof, "Combating label noise in deep learning using abstention," in *Proceedings of the 36-th International Conference on Machine Learning*, 2019.

[3] "NCI web site," https://gdc.cancer.gov/.

[4] "GDC Documentation HTSeq-FPKM-UQ web site," https://docs.gdc.cancer.gov/Encyclopedia/pages/HTSeq-FPKM-UQ/.

[5] L. Wang, Y. Xi, S. Sung, and H. Qiao, "RNA-seq assistant: machine learning based methods to identify more transcriptional regulated genes," *BMC Genomics*, vol. 19, 2018.

[6] K. Wnuk, J. Sudol, K. B. Givechian, P. Soon-Shiong, S. Rabizadeh, C. Szeto, and C. Vaske, "Deep learning with implicit handling of tissue-specific phenomena predicts tumor dna accessibility and immune activity," *bioRxiv*, 2019. [Online]. Available: https://www.biorxiv.org/content/early/2019/04/18/229385

[7] "Candle pilot1 nt3 hyperparameters."

[8] "Numpy Documentation web site," https://numpy.org/doc/stable/reference /random/generated/numpy.random.normal.html.

[9] A. Van Looveren and J. Klaise, "Interpretable counterfactual explanations guided by prototypes," *arXiv preprint arXiv:1907.02584*, 2019.

[10] J. Klaise, A. V. Looveren, G. Vacanti, and A. Coca, "Alibi explain: Algorithms for explaining machine learning models," *Journal of Machine Learning Research*, vol. 22, no. 181, pp. 1–7, 2021. [Online]. Available: http://jmlr.org/papers/v22/21-0017.html

[11] H. Du, M. Pang, X. Hou, S. Yuan, and L. Sun, "Plod2 in cancer research," *Biomedicine & Pharmacotherapy*, vol. 90, pp. 670–676, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0753332217310636

VII. Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

(The following paragraph will be removed from the final version.)

This manuscript was created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.