Reducing Variance in a Text-Based Binary Classifier With Mechanistic Interpretability and Ensemble Methods

Dean Goldman

November 2023

Abstract

This research is an analysis of and experiment on network activations derived from inputs into trained neural networks, in order to focus in on specific neurons that contribute to a networks decision making in the various prediction contingencies (FP, TP, FN, TN). In doing so, this work sets out the goal of optimizing performance in a binary classification task on a text dataset, Systematic Review, using a simple bag-of-word text input model.

1. Introduction

Mechanistic Interpretability (mech interp) has been described as the study of reverse engineering neural networks in order to put them into human understandable terms [1]. Many of the mech interp strategies involve adjusting model weights or activations, in order to derive a signal relevant to a task, as that activation adjustment flows through the network. While much of the mech interp research has been focused on language models, this research is an experiment in mech interp for a simple neural network trained in a binary text classification task. The important difference here is that the mechanics are arguably simpler, their interpretation only matters with respect to a model activation's effect on the probability of a single binary output, rather than a chain of outputs. In ROME [3], Meng et al. showed that factual associations in language models may be identified among neuron activations and successfully manipulated in order to directly edit a model's beliefs, and its downstream behavior based on those modified beliefs. Similarly, this work attempts to use mech interp to directly edit the knowledge incorporated via training toward modifying the network output with the goal of informing an error reduction strategy.

1.1 Dataset

This research focuses on an imbalanced binary classification problem in the Systematic Review dataset. This dataset consists of studies or articles on the topic of surveying or improving health care, usually in countries without advanced health care. The goal is to correctly identify around a hundred relevant documents out of several thousand by referencing the document title, abstract keywords, authors, and various metadata about the document's publication.

One of the limitations made in this experiment is that in training and inference environments, the dataset was transformed into input for a bag-of-words (BOW) model. Bag-of-words models for document classification tend to be relatively simple in that they are composed of probabilistic functions that take as input a vector of dimension V, where V is the length of the vocabulary in a corpus, consisting of the indicator of the presence of a word, its frequency, or some other statistic. They do not generally incorporate the order of words in a sequence, but one experiment in this research does incorporate sequential word order, and will be discussed.

2. Method

2.1 Network Architecture and Training

The experiments made in this research build upon a single layer perceptron with a hidden layer of size 32, and an output layer of size 2 to represent the two classes in the prediction space: (-1, 1). The network was trained by gradient descent with the objective of minimizing the cross entropy loss between the predicted target distribution and the label distribution. The input dimension is dependent on the size of the vocabulary, generally on the order of 10,000 unique tokens. Ngram tokens derived from the training set variables Title, Abstract, Keywords, and Author. Higher ngrams such as 5, 7, 9 were included in the experiment, and certain vocabulary terms were excluded if their overall term frequency was less than a given threshold.

2.2 Baseline Results

A baseline model performance is described in Table 1. In addition to a neural network baseline, a set of baselines for two Naive Bayes classifiers (Bernoulli and Complement) are also included. Since these are among the simplest ways of fitting a BOW model, they are included to add a point of reference in certain analysis. From inspecting the baseline results, we can note that there is a tradeoff between precision and recall, a model that is too sensitive may yield too many false positives, whereas a model may fail to generate enough true positives otherwise. A metric that balances precision and recall is the F1-score.

Table 1: Baseline performance (MLP and NB classifier) sorted by F1-Score (Top-5)

tn	fp	fn	tp	auc	experiment	specificity	precision	recall	f1
4644	56	85	65	0.921058	model-mlp-author-h-32-ngrams-1-w-100	0.988085	0.537190	0.433333	0.479705
4521	179	49	101	0.922516	model-mlp-32-sz-2500-ngrams-1-3-tf-15	0.961915	0.360714	0.673333	0.469767
4532	168	59	91	0.924294	search-complement-sz-1000-up-500-ngrams-1-tf-1	0.964255	0.351351	0.606667	0.444988
4487	213	51	99	0.939088	model-mlp-32-sz-2500-ngrams-1-3-5-tf-15	0.954681	0.317308	0.660000	0.428571
4608	92	84	66	0.837955	model-seq-attn-pos-cf-80-20-seq-4-emb-256-h-32-ngrams-1-w-100	0.980426	0.417722	0.440000	0.428571
4458	242	44	106	0.922670	model-mlp-32-sz-2500-lr-decay-0.1	0.948511	0.304598	0.706667	0.425703
4513	187	59	91	0.924634	model-mlp-32-sz-2500-ngrams-1-3-5-9-tf-15	0.960213	0.327338	0.606667	0.425234
4664	36	101	49	0.897968	baseline-bernoulli	0.992340	0.576471	0.326667	0.417021
4621	79	90	60	0.903765	model-seq-attn-author-bin-256-seq-4-emb-256-h-32-ngrams-1-w-100	0.983191	0.431655	0.400000	0.415225
4684	16	107	43	0.931284	model-seq-attn-seq-4-emb-256-h-32-ngrams-1-w-1000	0.996596	0.728814	0.286667	0.411483

2.3 Analysis of Errors

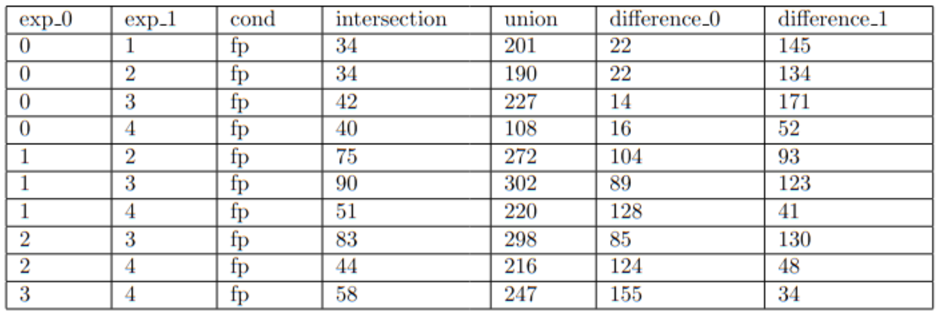
2.3.1 Errors

It can be noted that most of these models have a relatively high false positive rate, or false negative rate. In even the best models, either roughly half of positive predictions are false, or roughly half of positive samples are missed. This is not totally suprising, since the number of true positives is small in general, so these lower rates of precision/recall are in part a product of the dataset imbalance. Furthermore, some models tend to perform better as far as precision is concerned, and others have a better recall score. This indicates that the various models are fitting the data differently, depending on the data fold used in training, or other initial conditions. One way to inspect this is an analysis of set operations between contingencies over various models. If the sets of errors vary widely over models, this is a good indicator that the models are overfitting to the training data they are exposed to, and an ensemble may be useful in reducing that variance [4].

Table 2. Set Arithmetic on True Positive Predictions Among Multiple Networks

exp_0	exp_1	cond	intersection	union	difference_0	difference_1
0	1	tp	58	108	7	43
0	2	tp	53	103	12	38
0	3	tp	56	108	9	43
0	4	tp	57	74	8	9
1	2	tp	76	116	25	15
1	3	tp	80	120	21	19
1	4	tp	58	109	43	8
2	3	tp	71	119	20	28
2	4	tp	52	105	39	14
3	4	tp	54	111	45	12

Table 3. Set Arithmetic on False Positive Predictions Among Multiple Networks

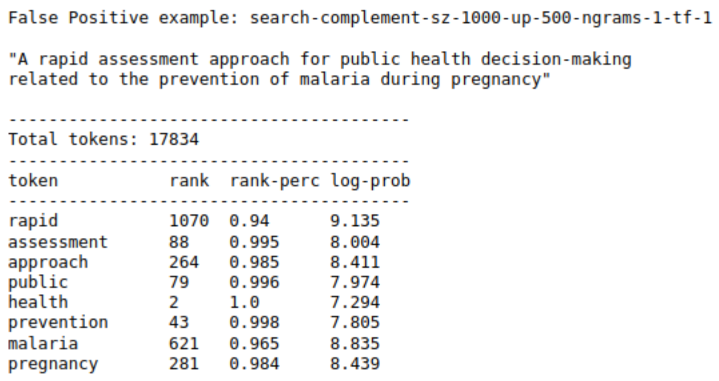


exp_0	exp_1	cond	intersection	union	difference_0	difference_1
0	1	fp	34	201	22	145
0	2	fp	34	190	22	134
0	3	fp	42	227	14	171
0	4	fp	40	108	16	52
1	2	fp	75	272	104	93
1	3	fp	90	302	89	123
1	4	fp	51	220	128	41
2	3	fp	83	298	85	130
2	4	fp	44	216	124	48
3	4	fp	58	247	155	34

The variation in choice of positive predictions across models is notably wide. Some models have a high level of agreement, while others do not. The fact that these sets are derived from competively performing models indicates that an ensembling technique may be effective [4].

2.3.2 Token Importance

In the Naive Bayes models, it is quite clear to interpret the effect a token weight has on the model output, as the classifier decision is fundamentally a product of log-probabilities. Below is a breakdown of tokens and their log probabilities learned by the classifier (along with the rankings of log-probabilities) for a false positive example. It is clear to see that the prevalence of high ranking log-probabilities is influencing the model to make a positive classification.

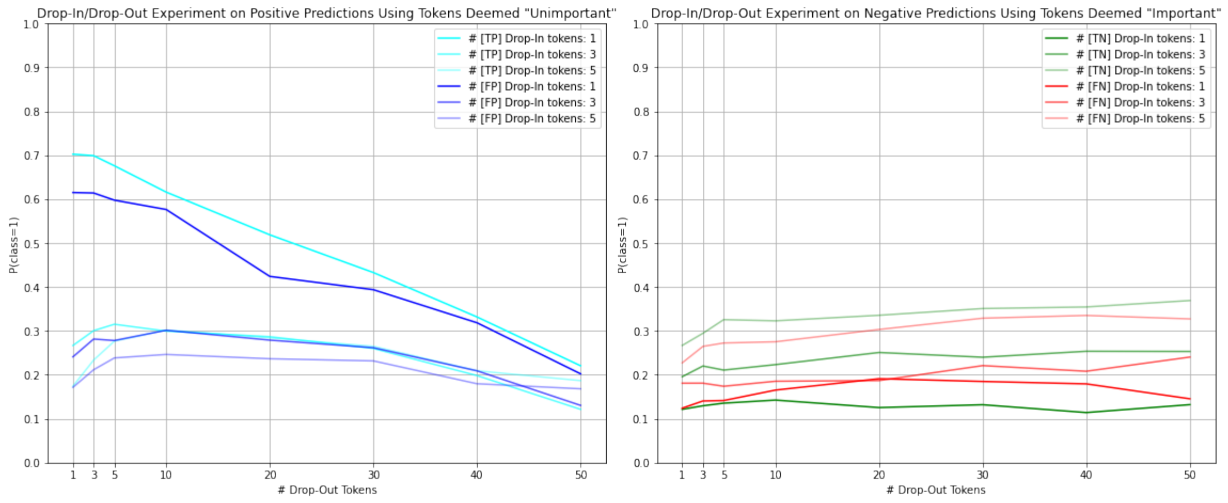


```
False Positive example: search-complement-sz-1000-up-500-ngrams-1-tf-1

"A rapid assessment approach for public health decision-making
related to the prevention of malaria during pregnancy"

----------------------------------------
Total tokens: 17834
----------------------------------------
token           rank  rank-perc log-prob
----------------------------------------
rapid           1070  0.94      9.135
assessment      88    0.995     8.004
approach        264   0.985     8.411
public          79    0.996     7.974
health          2     1.0       7.294
prevention      43    0.998     7.805
malaria         621   0.965     8.835
pregnancy       281   0.984     8.439
```

2.3.3 Token Ablation and Token Patching

The kind of token importance interpretation in 2.3.2 is somewhat less straightfoward for neural networks, as there are multiple layers of densely connected neurons. However, one available method is to iteratively drop out tokens and examine their effect on model behavior. Additionally, tokens can be "dropped-in" (or patched in) to see if they too have an effect. In the analysis shown in Figure 2, model inference was performed on test set samples where dropout was applied to input tokens, and replaced with tokens deemed "important" or "unimportant" according to Naive Bayes classifier log-probability weights for that token. Negative predictions were given "important" tokens with the expectation this would increase the network output probability, while positive predictions were given "unimportant" tokens.

Figure 2: Change in P(class=1) Over Drop-In/Drop-Out Conditions



It can be observed that the predicted samples behave fairly similarly whether they are a true or false positive, or whether they are a true or false negative. It also appears likely that manipulating positive predictions into negative ones can be expected to be easier than manipulating negative predictions into positive ones.

2.4 Ablation Study

A small model such as the kind used in this experiment lends itself well to simple plotting of activations. Partitioning all predictions by their class contingency, and computing statistics such as averages of activations, and frequency of activations among the top-k highest can shed some details on the behavior of the hidden layer of the network.

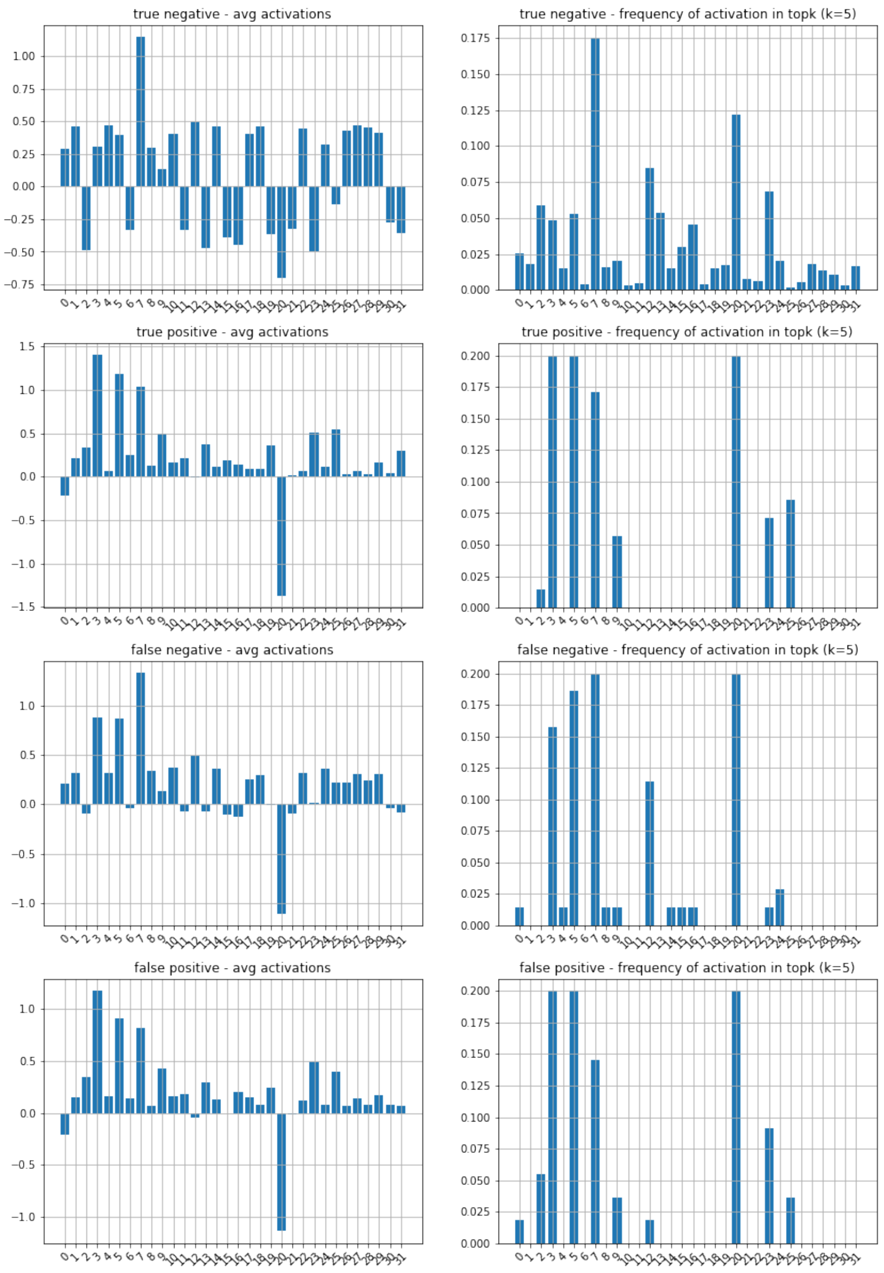


Figure 3. Expected Values of Hidden Layer Activations

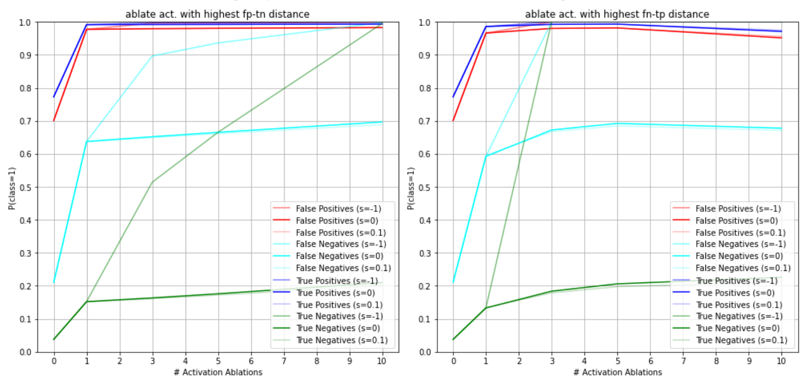
It can be observed that true negative predictions have a higher number of activations expected to be negatively signed, whereas the subset of false positives don't exhibit that trend. In fact, they look quite similar to true positives. This is sensible, since that similarity is part of the reason they are predicted positive. Interestingly, the false negatives appear to have a much more similar distribution of top-k activation rankings to true positives than they do to true negatives.

3. Experimental Results

3.1 Activation Patching and Ablation

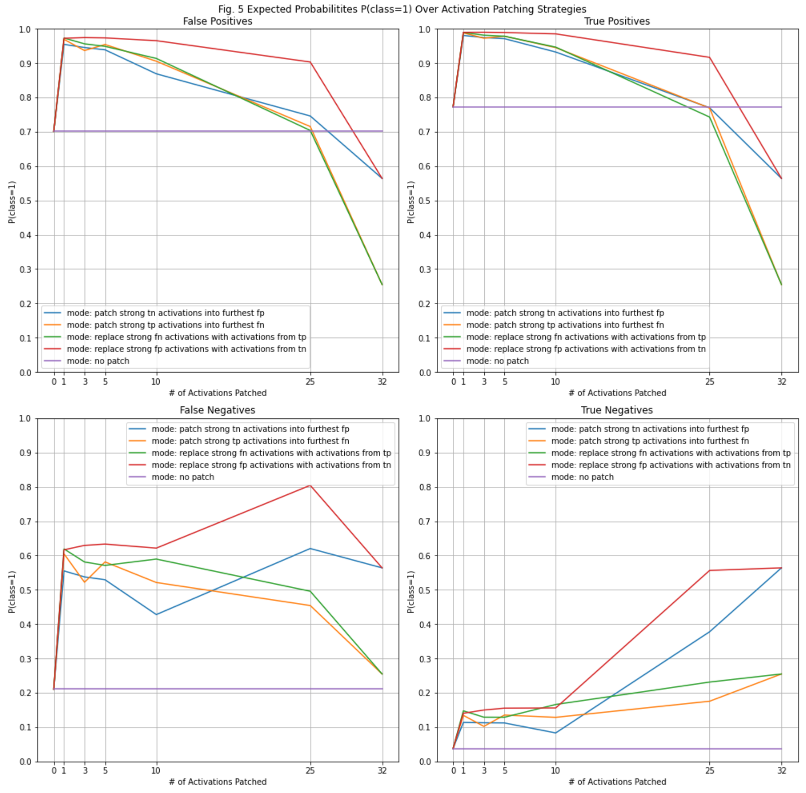
One of the primary methods used in mech interp is Activation Patching [3]. The basic idea is to take a set of activations from the network where the input into the model produced the intended output, and patch that into a known corrupted input, where the model produces some incorrect output. The exercise determines if the activation patch is sufficient to correct the output. This work is inspired by Activation Patching and attempts to transfer this technique to the task of correcting false predictions. Network weights whose activations are most correlated with false positive or false negative predictions will be identified and patched over with some "corrected" activation. In this case an aggregate of activations from correctly classified inputs is used. Similar to Patching, Activation Ablation is the process of removing network activations.

Fig. 4 Expected Probabilitites P(class=1) Over Several Ablation Strategies

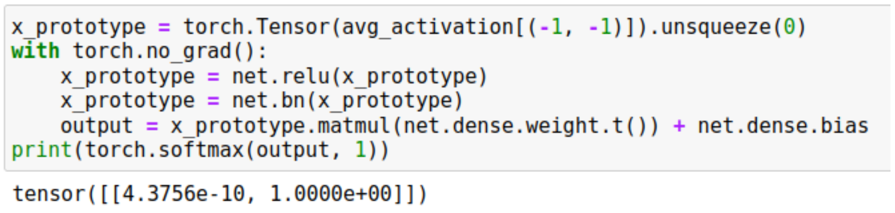


In the ablation study described by Fig. 4, activations with the greatest difference between FP and TN, and FN and TP were chosen to be scaled by (0, 0.1, or -1). The ablated activations were then sent into the output layer to produce a class probability. It can been seen from Figure 4 that either strategy results in an expected decision flip for false negatives. However, this does come at the price of an increase in likelihood of a decision flip for a true negative. Additionally, this method appears not to make any impact on identifying false positives.

Fig. 5 Expected Probabilitites P(class=1) Over Activation Patching Strategies



In Figure 5, a similar result to the ablation experiment can be seen, however it can be observed that patching with true negative activations actually increases the likelihood of a positive prediction. Upon manually inspecting the model, it can be seen that simply forcing in an average activation from the true negative class results in a positive prediction. The likely explanation for this is that the computed average of activations for true negatives is probably not a very good latent space prototype for true negative samples. This can be validated by manually computing the output of a prototype activation submitted into a trained neural network forward pass:



```
x_prototype = torch.Tensor(avg_activation[(-1, -1)]).unsqueeze(0)
with torch.no_grad():
    x_prototype = net.relu(x_prototype)
    x_prototype = net.bn(x_prototype)
    output = x_prototype.matmul(net.dense.weight.t()) + net.dense.bias
print(torch.softmax(output, 1))
```

```
tensor([[4.3756e-10, 1.0000e+00]])
```

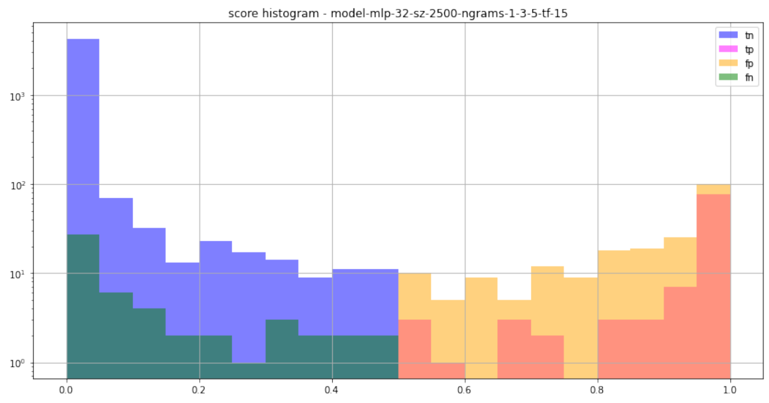
3.2 Ensembling

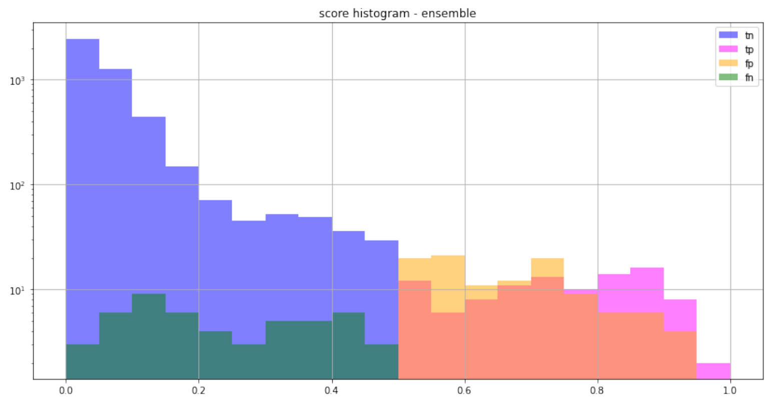
A large performance boost can be gained by averaging the scores over all of the listed networks. The resulting model is able to retain the best of both precision and recall performance.

Table 4. Ensemble Results (Top-10)

tn	fp	fn	tp	auc	exp	specificity	precision	recall	f1
4630	70	62	88	0.951688	top-10	0.985106	0.556962	0.586667	0.571429

One of the clearest illustrations of the effect of ensembling is in a score histogram plot. Ensembling has the effect of biasing model outputs. In many cases, this bias is precisely what is needed to overcome a specific model's propensity to overfit the data.





As shown in the above two plots, the ensemble biases the network outputs towards uncertainty, as multiple predictions are aggregated. This allows for many "voters" to discount a confident false positive prediction, as is relatively common in the individual mlp model.

5. Conclusion

While this implementation of methods inspired from the mech interp literature do fall short in producing useful error correction strategies for models, the exercise does shed light on the fact that neural networks tend to be highly fitted to their data, as their projection of the input data into high dimensional latent space allows for a highly variant decision boundary. It may also be likely that modifying network inputs or their activations during inference time can produce unexpected results. This is evidenced by the fact that submitting statistical "prototypes" of activations for a class is not guaranteed to produce a prediction for that corresponding class from the network output. One method that does prove out to work well in this problem is ensembling, which has the effect of reducing the overfitness of any individual model.
[1]

References

[1] Nanda, N. A Comprehensive Mechanistic Interpretability Explainer & Glossary. 2022.

[2] Olah, et al., "Zoom In: An Introduction to Circuits", Distill, 2020.

[3] Meng, K., A., Andonian, A., Belinkov, Y., and Bau, D. Locating and Editing Factual Associations in GPT. arXiv preprint arXiv:2202.05262, 2023.

[4] Goodfelllow, I., Bengio, Y., Courville, A. Deep Learning. The MIT Press. 2016. ISBN: 9780262035613

[1]The code used for training and experimentation can be found at https://github.com/deanjgoldman/EN.605.744.FA23-course-project