# Tuning Accuracy-Diversity Trade-off in Neural Network Ensemble via Novel Entropy Loss Function

Muhammad Ammar Ali[1], Yusuf Sahin[2], Süreyya Özöğür-Akyüz[3], Gozde Unal[2]

[1]Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Bahcesehir University, Istanbul, Turkey
muhammadammar.ali@bahcesehir.edu.tr
[2]Faculty of Computer and Informatics Engineering, Istanbul Technical University, Istanbul, Turkey
[3]Department of Mathematics, Faculty of Engineering and Natural Sciences, Bahcesehir University, Istanbul, Turkey

***Abstract*—An ensemble prediction is performed by combining the individual predictions from all members of the committee via a consensus method. For this technique it is best to reduce error dependence and ensure that all members of the ensemble are exploring different parts of the solution space. In order to ensure this we present a novel entropy based loss function to optimize the ensemble and improve accuracy and diversity. An ensemble of 300 Deep Neural Networks is tested on the CIFAR-10 dataset and results show an increase in accuracy while maintaining a level of relative entropy.**

## I. Introduction

Neural network ensemble learning is a technique, which uses multiple (say M>1) individual deep neural networks (DNNs) to formulate an ensemble learning team to work together by utilizing the strength of each member for accomplishing a particular task, supervised or unsupervised.

The ensemble DNN prediction is performed by combining the individual predictions from all members of the committee via a consensus method. The best scenario is when all members of an ensemble team of size M can learn and predict with independent errors.The worst scenario represents another end of the spectrum: namely, all M member models are M perfect duplicates such that they are identical in positive and negative predictions. However, when the errors are correlated to some extent, which is typical in practice, the overall error reduction will be smaller respectively and yet the expected ensemble committee error will not exceed the expected error of its member models [1].

In this study we present a novel equation to increase error independence via entropy measurement.

Suppose that in a system of M binary classifiers, classifier number $i$ calculates the probability that a selected sample belongs to Class 1 as $f_i^{\{1\}}$. In this system, to find a general result over all classifiers.

$$f_{ens} = \sum_{i=0}^{M-1} w_i f_i^{\{1\}} \tag{1}$$

A weighted majority voting method can be presented [2]. Here $w_i$ weights show the success of the classifier, but not only on this. These values should be calculated considering both the classification successes and the variation between the classifier results. At this point, to show the output distributions of $p_i$ classifiers, $f_{gt} \in 0, 1$ correct classifications.

$$L_{ens} = \alpha(f_{ens} - f_{gt})^2 + ... \\ ...(1-\alpha)(1 - (H(\sum_{i=0}^{M-1} w_i p_i) - \sum_{i=0}^{M-1} w_i H(p_i)) \tag{2}$$

It has been suggested that values should be found to minimize the loss function. Here, the first term is intended to increase classifier success and the second term to increase diversity. The second term is the Jensen Shannon term divergence, which mathematically expresses diversity within the community.

In order for the method to be used in multi-class classification problems, the loss function uses the class number $C$ and $f_{gt} \in 0, 1^C$ correct classification vector and $p_{i,j}$ the probability distributions from $f_{i,j}$ to show:

$$L_{ens} = \alpha \sum_{j=0}^{C-1} \frac{(f_{ens} - f_{gt})^2}{C} + ... \\ (1-\alpha)(1-\frac{\sum_{j=0}^{C-1}(H(\sum_{i=0}^{M-1} w_i p_{i,j})) - (\sum_{i=0}^{M-1} w_i H(p_{i,j}))}{C}) \tag{3}$$

In Equation 3 proposed above, the error is minimized with the first term, while the H function in the second term is maximized by using the Jensen Shannon entropy function $(H(x) = -x \log x))$. The $\alpha$ coefficients at the beginning of both terms correspond to the coefficients that will provide the trade-off between them. In this study we will test these equations on our ensemble while pruning.

## II. Methods

### A. Generating the Ensemble

Ensemble learning is an approach that can solve a machine learning problem with trained multiple learners. Conclusion of

this approach is made after compounding each output of single learners in accordance with some criteria. No Free Lunch theorem indicates that there is no single model that operates best for every problem [3]. For this reason, the purpose of the ensemble learning is to enhance the accuracy of the single classifiers. On the other hand, owing to the potential noise in the data, overlapping data dispersion and outliers generally single classifiers cannot acquire a determined classification accuracy. These have boosted the necessities to generate ensemble techniques. To generate a diverse ensemble using our deep neural networks we applied the following techniques:

- Bootstrap Aggregation (Bagging)
- Variation in number of hidden layers
- Varied placement of Dropout layer

*1) Bootstrap Aggregation (Bagging):* Bagging method is very useful for high dimensional data set problems and powerful for ensemble method to improve the performance of the model. It is a method of retraining the basic learner by deriving new training sets from an original training set. The training set is produced by random selection by putting a sample set consisting of n samples in bagging. Each selected sample is put back into the training set. In this case, some examples are not included in the new training set while others may take place more than once. Outputs of these randomly selected sub-data sets are aggregated with voting or averaging [4].

*2) Variation in depth of the Neural Networks:* In this technique we make multiple models with varying numbers of hidden layers. That means the models will have 10 to 15 hidden layers.

*3) Varied placement of Dropout layer:* Dropout (also called Dilution) is a regularization technique for reducing overfitting in artificial neural networks by preventing complex co-adaptations on training data. It is an efficient way of performing model averaging with neural networks. The term dropout refers to randomly ”dropping out”, or omitting, units (both hidden and visible) during the training process of a neural network [5].

To increase diversity in our ensemble we vary the placement of the dropout layer in the neural network. This approach not only would help create diversity but will fundamentally change learning rates and accuracies of the Neural Networks, making them unique [6].

To measure this diversity in the ensemble we use relative entropy or Kullback–Leibler divergence (KL-divergence). This a measure of how different two probability distributions are, conventionally taking one as a reference [7]. A relative entropy of 0 indicates that the two distributions in question are identical. This is calculated by considering two probability distributions, P and Q, where P represents the target probability distribution and Q represents a model output: a description or an approximation of P. The KL-divergence is interpreted as the average difference of the number of bits required for encoding samples of P using a code optimized for Q rather than one optimized for P.

The relative entropy from Q to P, in probability space X is defined to be [8]:

$$D_{KL}(P||Q) = \sum_{x \in X} (P(x) log(\frac{P(x)}{Q(x)}), \tag{4}$$

Which is equivalent to:

$$D_{KL}(P||Q) = \sum_{x \in X} (P(x) log(\frac{Q(x)}{P(x)}), \tag{5}$$

Relative entropy is defined only if for all $x$, $Q(x) = 0$ implies $P(x) = 0$. Whenever $P(x)$ is zero the contribution of the corresponding term is interpreted as zero because:

$$\lim_{x \to 0^+} x log(x) = 0 \tag{6}$$

A common way to refer to $D_{KL}(P||Q)$ is as the relative entropy of $P$ with respect to $Q$.

Relative entropy has been used to select statistical and machine learning models by treating it as a diversity metric [9] and reducing divergence between different probability distributions.

## III. Experiment

We use the concepts defined above (in MethodsII) on the CIFAR-10 dataset [10]. This dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. We perform a 50/20/30 data split manually. Leading to 30000, 12000, and 18000 images as training, validation and test set respectively. Class balance was maintained during the split.

To generate the ensemble, the training set was divided into 10 randomly generated and class balanced bags and tested on 6 Convolution Neural Networks with varying number of hidden layers (10 -15) and varying placement of dropout layer in the network. The placement of the dropout layer was selected in quartile ranges, that is if our network has 12 layers we can place a dropout layer after 3 hidden layers (D1), 6 hidden layers (D2), 9 hidden layer (D3), before the output (D4) or no dropout layer at all (D0). Therefore, this leads us to generate 300 (10*6*5) unique CNNs for our ensemble. A 5-fold cross validation was conducted on all the models. A ReLU activation function along with ADAM optimization method. The implementations were run using PyTorch 10.2, Sci-kit Learn 0.22 on Ubuntu 20.04.

## IV. Results

### A. Testing the Ensemble

After training and testing the ensemble, their predictions were aggregated via majority voting. The result of majority voting is compared with the average accuracy of each classifier and the diversity was evaluated using the KL-divergence. This highlights the contribution of majority voting in an ensembling process. The results are shown in Table IV-A, and the accuracy and diversity losses are shown in Figures 1 and 2 respectively.

TABLE I
TESTING RESULT OF ENSEMBLE

| | **Full Ensemble** |
|---|---|
| Average Accuracy | 83.07% |
| Voting Accuracy | 89.03% |
| KL-divergence | 7.52 |
| Ensemble Size | 300 |

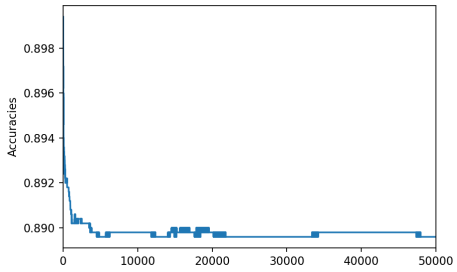


Fig. 1. Accuracy of Ensemble

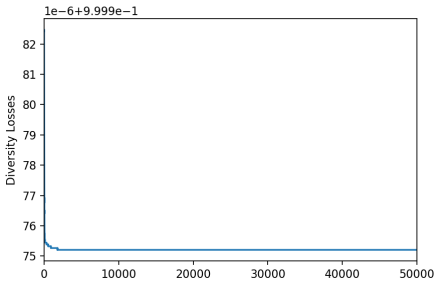


Fig. 2. Diversity Losses of Ensemble

### *B. Optimizing the Ensemble*

In this step we try to select the most optimal models using Shannon Entropy. While testing for alpha as 0.1, 0.3, 0.5, 0.7 and 0.9. We test the models on the validation set and select those with the highest accuracy to generate our newer ensemble. This was repeated 5 times to ensure reliability of the results. When the best value for alpha is selected (best alpha = 0.7) we test those models on the test set. Results are given in Table IV-B whereas the accuracy and diversity losses are shown in Figures 3 and 4 respectively:

TABLE II
TESTING RESULT OF ENSEMBLE

| | **Optimized Ensemble** |
|---|---|
| Average Accuracy | 86.80% |
| Voting Accuracy | 91.03% |
| KL-divergence | 7.52 |
| Ensemble Size | 186 |

## V. CONCLUSION

Looking at our results above we can see that our equations show an increase in accuracy without sacrificing diversity in the ensemble.

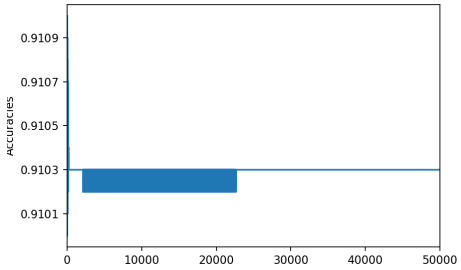


Fig. 3. Accuracy of Optimized Ensemble

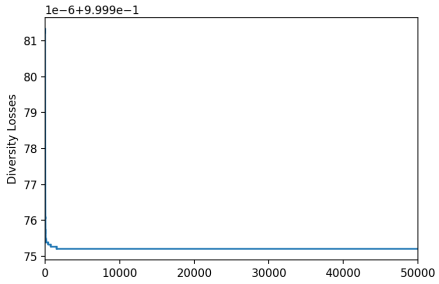


Fig. 4. Diversity Losses of Optimized Ensemble

## REFERENCES


[1] L. Liu, W. Wei, K.-H. Chow, M. Loper, E. Gursoy, S. Truex, and Y. Wu, "Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2019, pp. 274–282.

[2] R. Polikar, "Ensemble learning," in *Ensemble machine learning*. Springer, 2012, pp. 1–34.

[3] H. Xu, C. Caramanis, and S. Mannor, "Sparse algorithms are not stable: A no-free-lunch theorem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 1, pp. 187–193, 2011.

[4] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, and M. K. Rasel, "A review of ensemble learning based feature selection," *IETE Technical Review*, vol. 31, no. 3, pp. 190–198, 2014.

[5] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[6] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with lstms." in *Interspeech*, 2017, pp. 1586–1590.

[7] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[8] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[9] K. P. Burnham and D. R. Anderson, "A practical information-theoretic approach," *Model selection and multimodel inference*, vol. 2, 2002.

[10] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.