ECEN 689: Applied Information Science - Project Report

Samyuktha Sankaran, Harish Kumar, PRS Harinath

Abstract

Ensemble methods can take a large number of weak learners and aggregate them into a strong classifier. The outputs of the various weak learners are often aggregated either by voting in case of classification or by averaging in case of regression. In this project, we first examine whether strong learners such as neural networks can be aggregated in a similar manner to produce an ensemble classifier with higher performance than a single strong learner with equivalent complexity.

1 Introduction

Ensembling is a technique to combine the outputs of multiple predictors to answer a classification or regression problem. Usually, the outputs of a large aggregation of weak learners such as decision trees are combined to produce a strong learner, as in the case of a random forest.

Theoretically, a strong learner such as a neural network is capable of approximating any continuous function, and thus, any learner that we build with an ensemble of neural networks can be replaced by an equivalent larger neural network with the exact same performance. However, in practical scenarios, training deep and wide neural networks is a very difficult process and requires careful selection of network architecture, expensive computational power and millions of training data points.

If hypothetically we are able to match the performance of a deep neural network with an ensemble of shallow neural networks, then we have a model that can perform very well on datasets that have a size on the order of a few thousands datapoints. We choose some such datasets in this experiment and demonstrate that ensembles of shallow neural networks do manage to perform as well as complex deep neural networks.

2 Theory

We show a short theoretical proof that an ensemble will be at least as good as the individual strong learners. It might be intuitively obvious, but we do want to assert that we have sufficient mathematical basis behind our approach.

Consider an ensemble of strong learners which are from a PAC-Learnable Hypothesis class H. Say their parameters are ϵ and δ. This implies that the probability that one of our classifiers has worse than ϵ error is

$$P(L_D(H(s)) > \epsilon) \leq 1 - \delta \tag{1}$$

Without loss of generality, assume that we are working on a binary classification task. This is valid since we can always model a multi-class classification task as a large number of one-vs-rest binary class classification tasks. If we use a simple voting procedure to make ensemble predictions, we will classify to the wrong class only if at least half our predictors give the wrong output. If our ensemble has N models, the probability of the ensemble predictor's($E(S)$) error being bad is,

$$P(L_D(E(s)) > \epsilon) \leq \binom{N}{\frac{N}{2}} (1-\delta)^{\frac{N}{2}} \delta^{\frac{N}{2}} \tag{2}$$

This is since the correctness of the individual classifiers are i.i.d Bernoulli random variables. We use Stirling's approximation for $N!$ in the combinatorial expression and we get

$$P(L_D(E(s)) > \epsilon) \leq \frac{4^N}{\sqrt{\pi N}} (1-\delta)^{\frac{N}{2}} \delta^{\frac{N}{2}} \tag{3}$$

The quantity $(1-\delta)\delta$ is upper bounded by 0.25 and has its peak value at $\delta = \frac{1}{2}$. We know that our learner is strong, and thus, for at least one $\epsilon < \frac{1}{2}$, $\delta < \frac{1}{2}$. Thus, for $c = 4\delta(1-\delta) < 1$,

$$P(L_D(E(s)) > \epsilon) < \frac{c^N}{\sqrt{\pi N}} \tag{4}$$

Note the strict inequality for c since δ is strictly less than $\frac{1}{2}$. Thus, as N increases, for any ϵ, the probability that our ensemble classifier performs with an error rate worse than ϵ is strictly upper bounded by an function that decreases faster than an exponential function of N. The rate of decay of this function depends on the quality of the individual ensemble units, since c is a function of δ.

3 Breast Cancer detection using Ensemble of 7 CNNs

3.1 Dataset

The data for this study were a subset of digital mammography images (7100) of 1755 patients publicly available at Digital Database for Screening Mammography. 53% of the data were benign cases and rest malignant. Benign images represent screenings that caused enough concern to require investigation by another method (ultrasound, pathology, etc.), but ultimately end up being non-cancerous. Cancer cases were those confirmed to be malignant cases of breast cancer. Each patient had two separate images per breast, the standard mediolateral oblique (MLO) and craniocaudal (CC) views. Importantly, MLO and CC views are two different projections that cannot be co-registered, and thus represent different pieces of information about the subject. This dataset is of relatively small size in the context of deep learning and computer vision, but remains widely used in the mammography literature [10], thus representing a useful benchmark for comparison to the methods presented here. For the experiments in the following sections, we divided the dataset scans into 80% train, 10% validation, and 10% test partitions. All images belonging to a unique patient are in the same split, to prevent training and testing on different views of the same breast.

3.2 Image Preprocessing and storing

Given that images are often of slightly different size, each image is resized to 224 * 224 in order to align with the standard input dimensionality requirements of most classic deep learning architectures

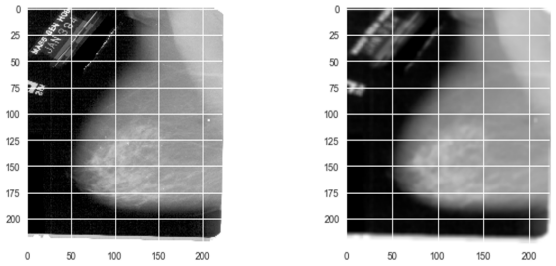

Figure 1: Example of DoG filter: Left, original grayscale image resized to (224 x 224). Right, after applying Difference of Gaussian filter (sigma=2).

(AlexNet, ResNet). Further modification we make is a pre-processing layer: 2 scales of Difference of Gaussian filters (Eq. 1) at sigma = 2. We set these 2 filters to be constant and attach them to the computational graph such that the original image passes through the filter bank of 2. The result (now dimension 224 * 224 * 2) is concatenated with the original grayscale image and fed as an input to our convolutional neural network (takes an input of 224 * 224 * 3). DoG filter reduces in filtering out high frequency noice from the image. In Figure 2, image on left has some random noise in top left corner which is blurred out after application of DoG filter.

We observed from first few runs that its not efficient to load every single image from the hard separately and apply image pre-processing and then pass it to the network to train, validate, or test. Moreover, pre-processing happens in CPU while training of Neural network happens in GPU and hence pre-processing becomes a bottleneck when number of iterations are high. So, after processing each image, we convert it into form of tensor and store them all in a single HDF5 file. Then we load them from the file in a batch-wise manner during training.

3.3 Network Architecture

We investigate CNN that are based on architectures that have been successful in general computer vision tasks, with few architectural changes: a single network that has both MLO and CC images of two breasts vote on the final prediction scores, but share all weights in between. During test time, we average the probability scores produced by the CC branch and the MLO branch to make sure we incorporate information from both the CC and MLO views.

Figure 2: Breast Cancer Detection Methodology

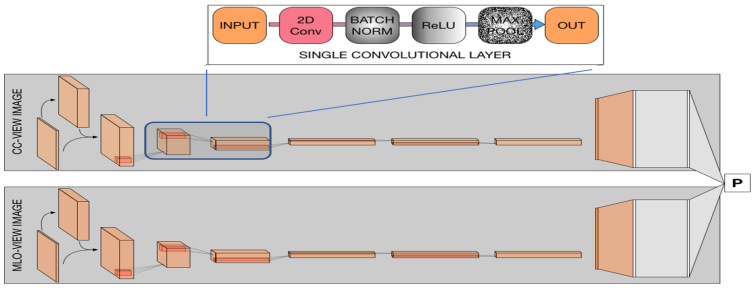

Figure 3: The figure shows Alexnet Neural Network layers and a single convolutional layer is shown with transformation it performs on the input.

The core algorithm shown in Figure 3 above is the famous AlexNet. We also replace this model with several successful architectures (see results), for comparison. AlexNet has multiple convolution layers and each layer performs 2D convolution, ReLu activation, max pooling etc. Output from these layers are then passed through a Global Average Pooling layer, whose output is feeded into a fully connected Neural network which classifies the image.

Figure 2 above shows the implementation workflow. After picking the top network structure based on test accuracy, we concluded that InceptionV3 structure gave us the best results. Now fixing this network architecture we tried combinations of different settings of Optimizer('Adam','RMSProp'), Dropout rate (0.1,0.2), Batch size(32,64) to generate different networks trained on same data. Average of predictions of these networks on same test data were used to make final prediction in 'InceptionV3-EnsembleModel'. This Ensemble technique used here is called 'Stacking'.

3.4 Results of best CNN model

We use the process described above to perform the classification task for the DDSM mammography data. Key hyper-parameters for the final model are a dropout probability of 0.1, Adam optimizer with a learning rate of 0.001 and decay rate of 0.99 per epoch, and a regularization coefficient of 0.000001. Model training was performed on GPUs in Google Colab, and took 6 hours for training over 190 epochs (111s per epoch) with a batch size of 64. The best model reported here was and Inception V3-based architecture that was able to achieve a held-out test accuracy of 72%, on all images.

Now the ensemble of InceptionV3 based network across 7 different settings voting on four images per patient (based on confidence metric) we achieve an accuracy of 74.7% on test set of patients. Thresholding didn't give much difference to results and result mentioned here corresponds to 0.5 value. The right panel of Fig. 4 illustrates the ROC curve for the model at one particular point during the training process.

3.5 Comparison of different CNN models with Ensemble network

From the results obtained after comparing different CNN Models such as Alexnet, Resnet, Inception-resnet, VGG-19, and Inception-V3, we analyzed that the Inception V3 model is the one giving maximum accuracy. Figure5 illustrates the respective prediction accuracy for each of the CNN Model. By running more epochs, we observed that the test accuracy starts saturating but the training accuracy is improving (Figure 4). This results in model over fitting. Also, with CNN Model we need to analyze large data sets /images for better results. Further CNN has multi-million

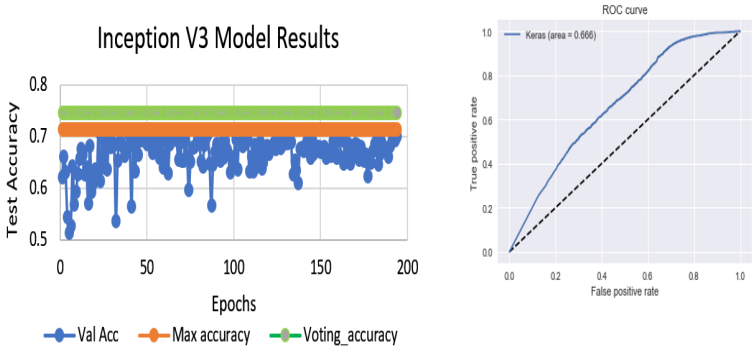

Figure 4: Training Results. (a) A curve showing the test-set results for Inception V3 model. This one achieves a held-out test set accuracy of 0.72. Ensemble of 7 similar models accuracy:0.75. (b) A ROC curve for one of the 7 models on the held-out test set. We achieve an ROC AUC of 0.67.

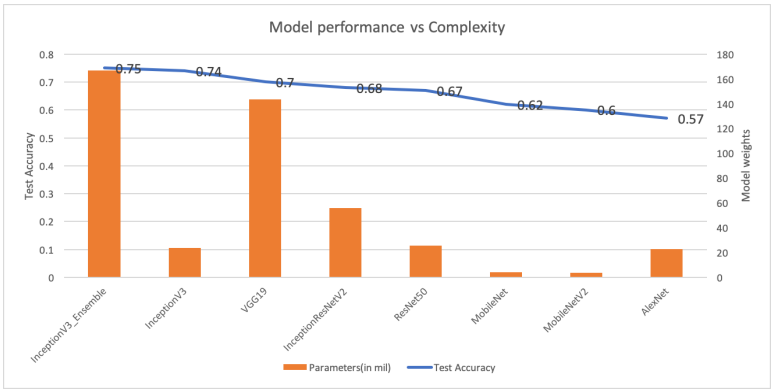

Figure 5: Comparison of different CNN Models such as Alexnet, Resnet, Inception-resnet, VGG-19, and Inception-V3. Best model was 'Ensemble of 7 models using InceptionV3 Network' with 0.75 accuracy on test set. Number of weights needs to be trained, shown by bar charts and values in secondary axis.

parameters to be learned âĂŞ requires large of number of data/images. For example InceptionV3 network we used had 23 mil parameters(weights) that needed to be trained and with dataset of just 7000 images we found it really difficult to avoid overfitting.

To overcome these issues we stopped the model training early (around 70 epochs) based on training and validation set accuracy values. Repeated these for different settings to get seven different models and created an ensemble of these networks to make final prediction. This Ensemble network as shown in Figure 5 gave 0.75 accuracy on test set. This ensemble network performed better than all the 7 individual models (best individual model 0.72 accuracy) and also better than equally strong network like VGG19 (143 mil parameters). In Figure 5 we have plotted Number of parameters and Test accuracy of each models used.

4 Ensemble of Neural Network on MNIST Fashion dataset

4.1 Dataset

The fashion MNIST data set is a lot similar to the MNIST digit data set, having 10 classes. The training set has about 60,000 images and testing data set has 10,000 images. This dataset is convenient to use as it does not require any preprocessing and can be directly loaded from the keras package.

4.2 Implementation

The ensemble network consists of 6 neural networks with different configuration.

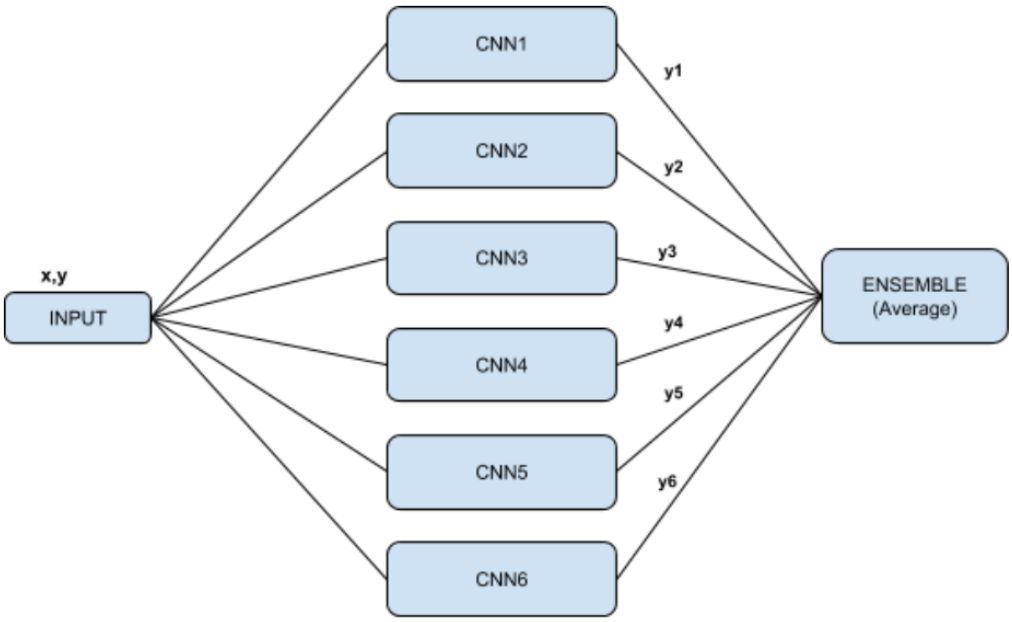

Figure 6: Ensemble model

- **CNN1**: This is a 2-layer CNN.
- **CNN2**: It is a single layer CNN.

- **CNN3**: 3 - layer CNN.
- **CNN4**: It features several convolution layer followed by a Max Pooling layer. It has a global average pooling layer in the end.
- **CNN5**: It has multiple convolution layers with stride 2 used instead of Max pooling.
- **CNN6**: It has three blocks of multi layer perceptron (MLP) convolution layers.

The train and test datasets are loaded to the variables. The data is pre-processed by reshaping it to the size the models are expecting, in this case 28x28x1. The training data is split as 80% training and 20% validation. The training data is separately passed to each of these six neural networks. Each model is trained for 100 epochs with a batch size of 128, categorical_crossentropy loss function and Adam optimizer. It is then compiled, evaluated and the accuracy and error is computed. The best set of weights are also stored as a .hdf5 file for each model. After training all six models, we proceed to the ensemble of these models. The weights for respective models are loaded and their corresponding predictions are passed as labels to the ensemble model. The prediction here is computed by taking the average of the predictions from the six neural networks. This ensemble model is then compiled and the accuracy and error are calculated.

4.3 Results

The accuracy and the error for the models are tabulated below.

Model	Accuracy	Error
cnn1	91.46%	9.029
cnn2	91.27%	8.989
cnn3	91.83%	8.935
cnn4	90.54%	8.523
cnn5	92.57%	7.675
cnn6	89.45%	6.789
Ensemble	92.56%	8.73

The classification of the MNIST fashion dataset was performed by the individual models as well as the ensemble. The accuracy of the ensemble model is slightly better than some of the individual models. The cnn6 model, also know as the network-in-network model, took the least time to run in comparison to others. The cnn4 model has the highest accuracy rate among the individual models owing to the usage of GlobalAveragePooling(). Here, an activation layer was not included contrary to using softmax activation at the output layer in the other models. A compromise on the accuracy was made. However, selectively choosing the networks will yield better results.

4.4 Inference

We attempted to make relationship between the individual performance of the neural network models and the ensemble of the models, and it is revealed that the latter performs better in a lot of cases. And in certain cases the ensemble of certain appropriate neural networks yields better results than the ensemble of all the neural networks. A threshold is chosen for a preferred metric, say the error or accuracy, and all the models that are above the threshold are chosen for the ensemble. This is called the **selective ensemble approach**.

5 Ensemble Neural Networks on CIFAR-10 Dataset

Here, we use CIFAR-10 to examine whether ensemble neural networks can lead to better performance than larger networks on restricted dataset. When we have a huge dataset, a deeper network will probably be superior, but the question we ask is, who wins in case of smaller datasets?

5.1 Dataset

The CIFAR-10 dataset contains 32×32 images from ten different classes such as airplanes, frogs, ships, vehicles, etc. It is generally a challenging classification problem for shallow neural networks and requires very deep and powerful nets to achieve accuracy above 90%. We restrict the size of the dataset to just 20000 to speed up ensemble training.

5.2 Network Architectures

We used a shallow neural network with three convolutional layers and three fully connected layers as the unit of our ensemble. The filter sizes were 32, 32, and 64 for the convolutional layers, and the dense layers had 128, 64 and 10 neurons respectively. Training was done using stochastic gradient descent, and the loss function used was categorical cross-entropy.

We augmented the size of the training set by generating additional images by translation, rotation, zooming, brightness variation, etc. We created an ensemble of 5 neural networks with the exact same architecture, but used different sub-samples from the training set to train them.

We also created a comparison network which had 5 times the number of parameters as an individual ensemble unit. We did this by adding more convolutional layers(along with intermediate dropout layers to reduce overfitting) and making the network deeper. The goal here is to compare the performance of ensembling against a single larger, more complex network.

5.3 Results

At saturation accuracies, the deeper network with a much larger number of parameters has only an incremental improvement of 2.85% upon the average performance of the shallow networks. This is probably since deeper networks become more powerful only with large datasets. The training accuracy of the deep network continues to increase, but this is only due to overfitting as the validation performance has already saturated.

In contrast, the ensemble shows an improvement of 4.82% upon the mean accuracy from the individual networks. The improvement is about 69.1% better than the deep network, thus demonstrating that ensembles are an effective way to train neural networks when we don't have millions of datapoints.

Fig. 7 shows plots of the above discussed results.

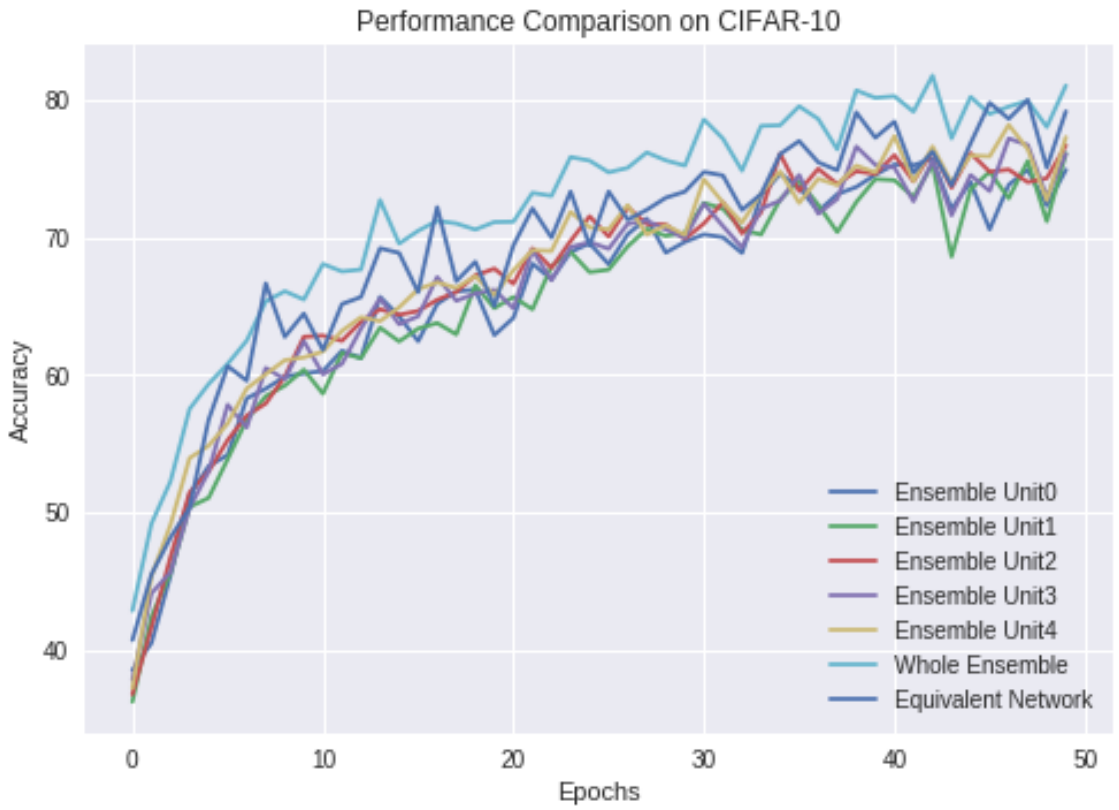

Figure 7: Performances on CIFAR-10. Note how the ensemble performs best, followed by the deep network.

6 Conclusion

In all the datasets that we tested on, we were able to demonstrate that an ensemble of small networks showed nearly as good, or sometimes slightly superior performance to networks that are much more complex. Complex networks require huge memory and computing power to train, and often overfit unless we have millions of datapoints. Smaller networks can easily be trained in parallel and in batches, and thus offer numerous advantages over larger networks.

A significant number of researchers have worked on ensemble of multiple classifiers to improve the performance of classification. Ensemble method in case of Neural Networks has several advantages:

- It efficiently uses all the networks of a population – none of the networks need to be discarded.
- It efficiently uses all of the available data for training without over-fitting.
- It inherently performs regularization by smoothing in functional space, which helps to avoid over-fitting.
- It utilizes local minima to construct improved estimates whereas other neural network algorithms are hindered by local minima.

Experimental results show that the ensemble method dramatically improves neural network performance on difficult real-world optical character recognition tasks.

However, it is true that when data size and computational power are not limiting factors, very deep neural networks will outperform any ensemble since the internal features that such a deep network would develop would be radically different from what any small network can develop. This is especially true in case of problems that require perceptual understanding, such as text, image and audio.

References

[Krizhevsky, Sutskever, and HintonKrizhevsky et al.2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12)*. Curran Associates Inc., USA, 1097–1105. http://dl.acm.org/citation.cfm?id=2999134.2999257

[Michael Heath and KegelmeyerMichael Heath and Kegelmeyer2001] Daniel Kopans Richard Moore Michael Heath, Kevin Bowyer and W. Philip Kegelmeyer. 2001. The Digital Database for Screening Mammography. In *Fifth International Workshop on Digital Mammography*. 212–218.