# Table of Contents

# 1. Introduction

# 2. Background

# 2.1 Mixture of experts

# 2.2 Ensemble method

# 3. Problem Statement

# 4. Methodology

# 5. Results and analysis

# 5.1 Challenges and solutions

# 6. Conclusion

# 7. References

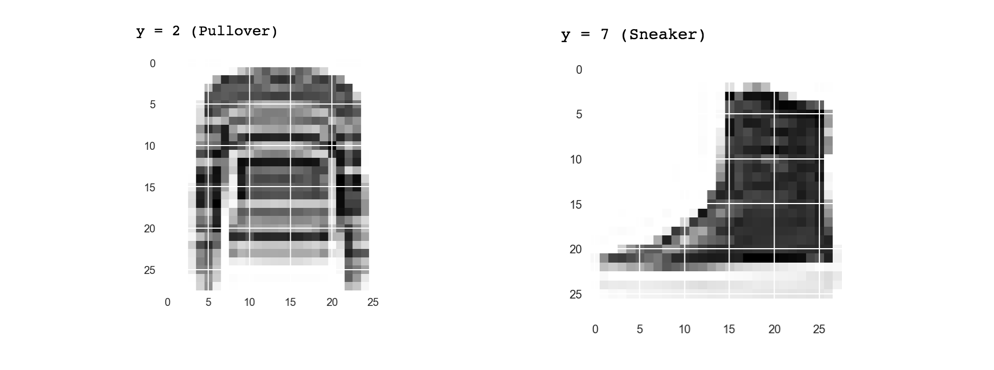
# 1. Introduction

For the following project, we have compared the performance and training times of two types of machine learning methods on the Fashion MNIST data set. Both the mixture of experts and the ensemble method acted on the multilayer perceptron model of a neural network. We trained the neural networks for 1 and 3 epochs and compared the accuracies and time to train for both methods. As a control we also included the basic M.L.P network to compare training times and performance.

# 2. Background

The experiment was done on the Fashion MNIST dataset. This dataset consists of 70 000 black and white images which are 28 by 28 pixels in size. Each image is labeled as 1 of 10 different types of clothing such as trousers, dresses, or coats. Using k-fold cross validation, the set can be divided in to 63 000 images for the training set and 7 000 images for the test set. The Fashion MNIST data was used due to our observation that the regular MNIST is relatively easy for basic feed forward networks to classify, and many neural network exercises use the regular MNIST dataset.

Figure 1. Two images from the fashion MNIST dataset, with labels “Pullover” and “Sneaker” .[1]



# 2.1. Mixture of experts

The Mixture of Experts technique was developed in the early 1990s to deal with extremely large datasets. In machine learning this technique involves having multiple neural networks all trained on different portions of the data. This is called specialization, where each neural network has good performance on one subset of the whole data set. This requires the data to be separated in to regimes that are given to different models during training. During training, the idea is to identify the model which is already doing better than the others on a particular subset of the data, and have it focus on predicting the right answers on the subset while ignoring the others, which leads to specialization. In order to match each model to a regime of the dataset we need to cluster the data using a gating function. We use a simpler method by computing a k-means clustering on the data and associate a neural network with a specific cluster. When new input is to be given to the whole model, the k-means cluster closest to the input is computed and the associated model is responsible for the classification. Ultimately the mixture of experts technique uses the dataset in a unique way, in that if the dataset is not large enough, it will not make good use of the data due to the partitioning that occurs over multiple models. However as the dataset grows, the mixture of experts method allows efficient use of all the data.

# 2.2. Ensemble method

Ensemble methods for machine learning employ the idea that multiple neural networks can be combined to give a result that is better than any single one. We have multiple predictors that may be the same type of learner or different types of learner. In our experiment we used the same type of multilayer perceptron with the sum rule used for the final classification from all the predictors in our ensemble. The dataset was partitioned into disjoint subsets to create variance among the predictors and increasing training efficiency. When the ensemble neural network receives input, it runs every predictor on data. The output of all the neural networks in the ensemble are combined and the summed-up classification whose value is greatest is used to make the final classification of the ensemble network.

# 3. Problem statement

Will one of the ensemble method, mixture of experts or a single neural network machine learning techniques far surpass the others in accurately classifying a large data set while having similar training times?

# 4. Methodology

Each neural network across all experiments were composed of 784 neurons for the input layer, 625 for the first hidden layer, 300 for the second hidden layer and 10 for the output. Each hidden layer used the sigmoid activation function. Softmax was used as the cost function along with backpropagation with a learning rate of 0.5 for training. Training consisted of mini-batching batches of size 128. If the data for a model did not divide evenly by 128, a batch less than 128 in size with the remaining data was also used to train to ensure each model trained on all its data.

Keeping all parameters the same for mixture of experts and the ensemble method, the amount of models used for each method was experimented on for all the integer values in the interval [2, 10]. The goal was to see if there were different optimal values of the number of models trained for each method for the problem of classifying fashion MNIST images. The time to train and accuracy for each methods were recorded, both with 1 epoch to train and 3 epochs to train. Time to train was calculated by measuring the call to the full backpropagation algorithm, summed up for all batches across all epochs and k-folds for all models for a single method. A k-fold cross validation with a value of 10 for k was used for all experimentation so that a confidence interval could be calculated using the results from each fold. The time to train for preprocessing k-means clustering for the mixture of experts was recorded separately, since it only needs to be computed once given a training dataset. On top of this, the time to train and accuracy for a single multi layer perceptron on all the training data was also recorded.

5. Results and analysis

Figure 2. Confidence interval showing the classification accuracy of each method after 1 epoch using k-fold cross validation

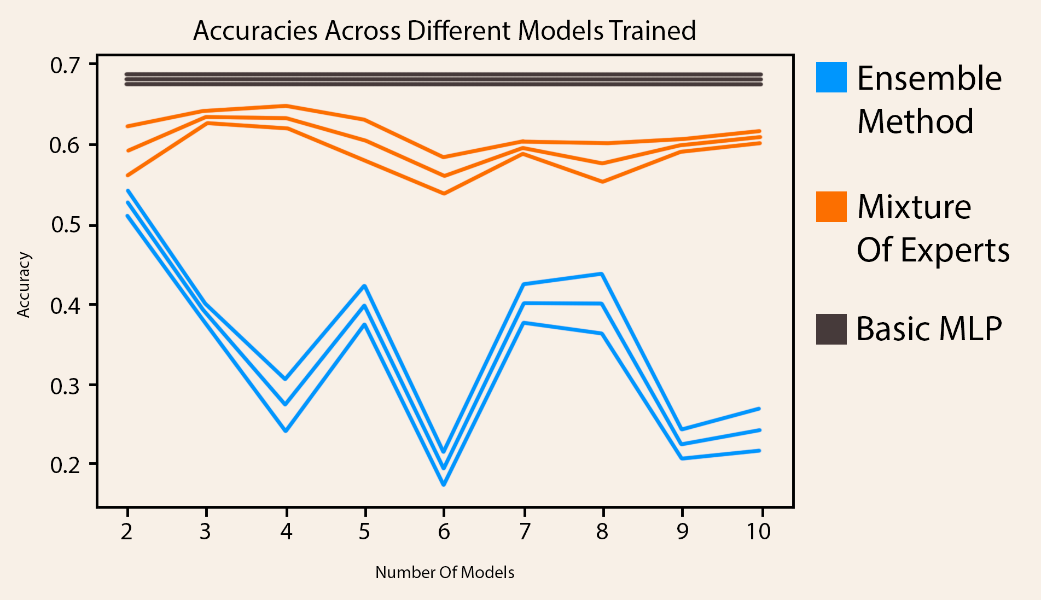


Figure 3. Confidence interval showing the time to train of each method after 1 epoch using k-fold cross validation

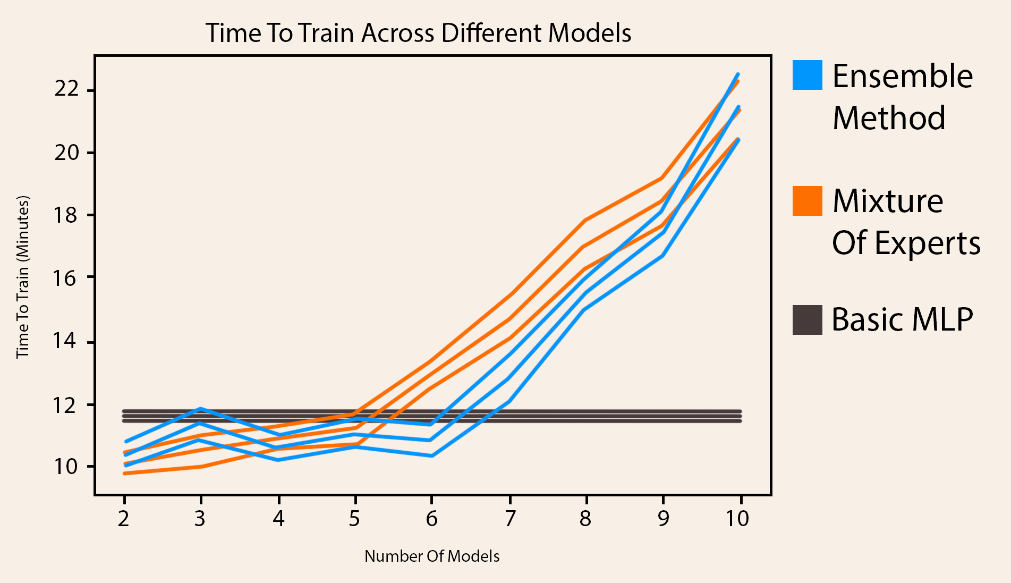
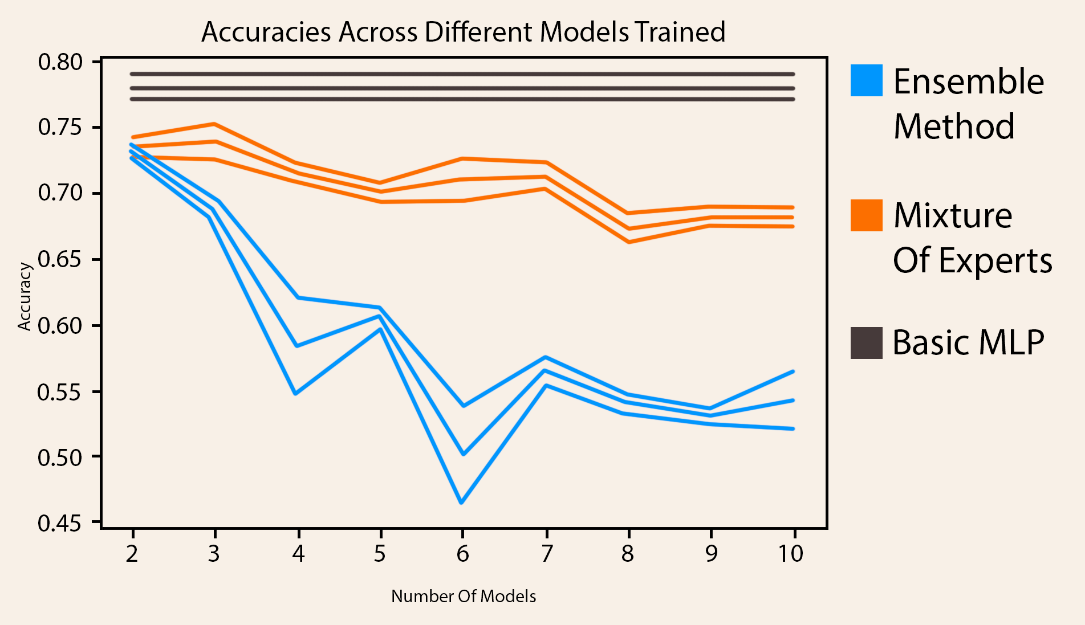


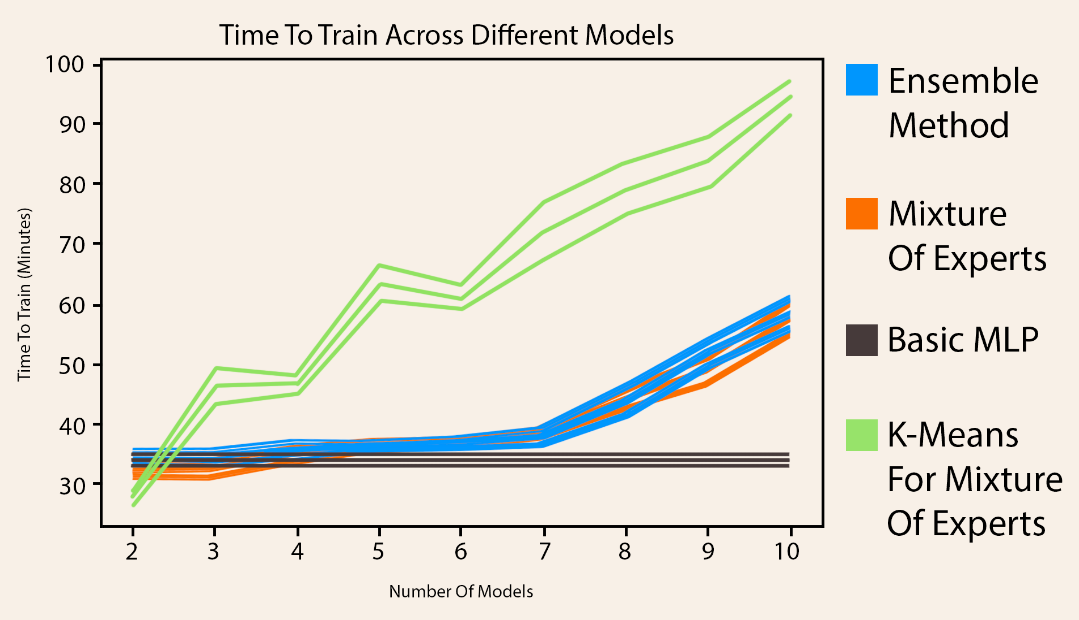
Figure 3. Confidence interval showing the classification accuracy of each method after 3 epochs using k-fold cross validation



The performance of the networks was initially tested for only 1 epoch. After seeing a single MLP performing better than all other methods, the performance for after 3 epochs was tested to see if some of the methods converged more slowly than the single MLP. The single MLP continued to improve while the other 2 techniques did not improve enough to show that training past 3 epochs would yield one technique outperforming a single MLP. The fashion MNIST data set was most likely not large enough to take advantage of combining variance amongst models or having specialized models. The resulting disjoint data sets for even only 2 models for the mixture of experts and ensemble method made the network’s performance drop severely due to not having as much training data to fine tune the parameters of each neural network model.

Figure 3. Confidence interval showing the time to train of each method after 3 epochs using k-fold cross validation. The time to compute k-means clustering is included in this graph only.

]



Mixture of experts with three models trained is shown to be similar in accuracy to the basic MLP. The time to train increases largely as more models are trained for both the mixture of experts and ensemble method, which is not what was initially expected. Since k-means only needs to be computed once for a data set, it was included separately in the training times.

# 5.1. Challenges and solutions

The time to train results were not what were expected. The goal was initially to have the time to train essentially the same across all the methods described. No matter the number of models used to train, the same amount of data is used to train in each epoch for the mixture of experts, ensemble method and standard method described since the data partitions across the models is disjoint and adds together to equal the entire data set. Tensorflow was used for the design of all the models. It is speculated that some sort of optimization which Tensorflow performs is being punished when multiple models are created and deleted frequently, since we only had one model in main memory at a time while performing experiments. A valid reason for why time to train was increased is since with more models partitioning data, there will be more partial batches less than 128 in size used for a single epoch, thus increasing the total calls to the backpropagation algorithm. This alone was not a significant enough factor to account for the increase in time to train. More investigation is required.

Another challenge in testing was partitioning data properly. A library for training multiple models easily would be very beneficial for further testing and applications in the tech industry.

# 6. Conclusion

The mixture of experts and ensemble method machine learning techniques were tested on the fashion MNIST data set for both 1 and 3 epochs recording time to train and accuracy. The results show that with both the methods described, a single multi layer perceptron trained clearly surpassed in performing accurate classifications of the fashion MNIST data set. Further investigation on the time required to train is required.

More experimentation is required to show whether the simple mixture of experts or simple ensemble method is preferable in a general case with very large data sets. It is possible that the mixture of experts, ensemble method and a single neural network require different hyperparameters (such as learning rate) to show optimal performance for each technique. A larger data set is recommended for further experimentation. Simulating a real-life situation where large amounts of more training data is rapidly available to the model after it has already been trained and assessed may show that training an additional model when new data is presented could be far superior in terms of training time required with similar accuracy than retraining a single neural network.

For further experimentation a combination of the mixture of experts and the ensemble method could prove to be more effective. One idea is to have multiple networks trained via the ensemble method but for each given input data multiply one network’s predictions by some scalar if it is assessed as been trained on a data set which is the most similar out of all the models to the given data to classify during testing.

# 7. References

<http://www.scholarpedia.org/article/Ensemble_learning>

^^^ sum rule vs majority vote