Extensive experimentation for Hyper-Parameters selection of Convolutional Neural Networks with Fashion MNIST data for image classification applications

Keerthiraj Nagaraj
Department of Electrical and Computer Engineering
University of Florida
k.nagaraj@ufl.edu

Abstract— **In this project, data classification problem is addressed using the concepts of supervised learning. Fashion MNIST dataset is used to train a Convolutional Neural Network (MLPNN) model to classify the images of fashion products into their respective classes. Hyper-parameters such as optimal number of units in the hidden layers, learning rates, stopping criteria etc were experimentally decided based on mean square error and classification accuracy for test data. CNNs have more hyperparameters compared to MLPNN, hence needs more data to be properly trained. We created a dataset containing 210000 images from the original data by applying some transformations. We trained the CNN model with this new dataset and obtained much higher training and test accuracy. The model with best hyper-parameters was chosen at the end from all these experiments and a confusion matrix is shown to the explain the performance of CNN model.**

I. Introduction

In this section, we will briefly introduce various technical concepts required to understand the analysis carried out in our project.

A. Supervised Learning

Supervised learning [?] deals with the class of problems in which we have a target dataset guiding the model on what decisions to make while it is being trained. In these problems, learning or the adaptation of model is supervised by the desired response. Supervised problems mainly deal with Polynomial Regression or also called as function finding and Logistic Regression or also called as Classification problems.

Polynomial regression tries to find a polynomial function for the output variable in terms of different combinations of input variables by assigning weights to them. The obtained polynomial equation could be used to predict the values of output for any values of the input. This problem can also be formulated as fitting a polynomial data curve through the data points which results in the least sum of normal distances from the curve to the points. In regression problems, the output value obtained is a continuous real valued number.

Logistic Regression or classification problems are the type of supervised learning methods where the outputs are non-continuous/discrete real values, where each value represents a different class. Each data point in the input is mapped into a class and a model is trained with this data. The objective of the trained model is to classify the test dataset into proper classes with least number of wrong classifications.

B. Convolutional Neural Networks

Convolutional Neural Networks are a special class of MLP which are designed for performing computations on image data and efficiently solve computer vision problems. A typical CNN comprises of convolutional layers, pooling layers and fully connected layers on top of a linear classification output layer.

II. Dataset and Experimental Setup

In this section, we will briefly discuss about the dataset used and the experimental tools used to carry out this process.

A. Dataset

The dataset used in this project is called Fashion MNIST data [1] which is taken from a larger family of datasets from NIST. This dataset contains thousands of gray scale images of fashion products. There are images containing T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot, totally images belonging to 10 different classes.

For this project, we have used 46,667 images for training of MLPNN models and 23,333 separate images for testing the trained models as mentioned in the project guidelines. Initially each image is converted into a 28x28 data matrix (784 input features).

B. Experimental setup and procedure

We have conducted the experiments in Python programming language in the Anaconda environment mainly using the libraries such as Keras, Scikit-learn, Pandas, Matplotlib, NumPy and SciPy. Fashion MNIST dataset was given to us in a .mat format, this file had both inputs and labels for training and testing.

We combined these training and test sets, shuffled it to form a new dataset and also augmented the original set with slightly transformed images for better generalization of the model.

III. Experimental Results

In this section, we provide a detailed discussion about various experiments conducted in choosing the hyper parameters such as optimal number of units in the hidden layers, learning rates, number of epochs and training test split. We provide learning curves and performance metrics namely Mean Square Error percentage (MSE) and Accuracy (in percentage) for each of the experiments along with a note on the significant observations.

When varying a certain hyper-parameter all other are kept same to understand the impact of varying parameter on the performance of the model. The default parameters are

- 1st Conv layer: Number of PEs: 32
- 1st Conv layer: kernel size: 3x3
- 1st Conv layer: strides: (1,1)
- 1st Maxpooling layer: pool size: 2x2
- 1st Maxpooling layer: strides: 2x2
- 2nd Conv layer: Number of PEs: 64
- 2nd Conv layer: kernel size: 3x3
- 2nd Maxpooling layer: pool size: 2x2
- Fully connected layer: Number of PEs: 1000
- Fully connected layer: Activation function: Softmax
- Learning rate or Step size: 0.1
- Activation function: Rectified Linear Unit (ReLU)
- Optimizer: Stochastic Gradient Descent
- Cost function: Categorical cross-entropy
- Validation split: 0.1 for original data
- Validation split: 0.2 for augmented data

A. Topology - PEs in Hidden layers

One of the most important hyper parameters that should be selected for CNN is the number of inputs in last fully connected hidden layer. We experimented with the varied number of units in this hidden layer. We observed the values of accuracy and MSE for test dataset for making the decisions. We varied the number of units in the this hidden layer from 250 to 1500.

Fig. 1 shows the Accuracy of CNN model for different number of PEs in last hidden layer.

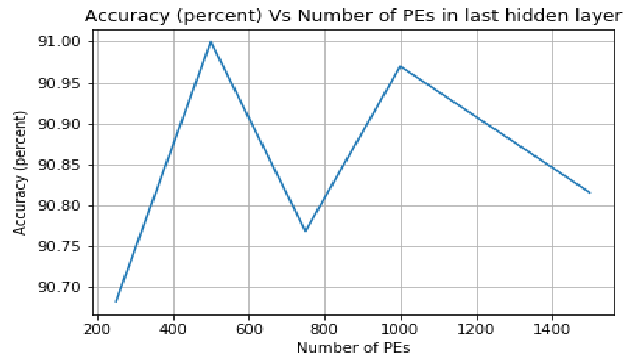

Fig. 1. Accuracy for number of PEs in last hidden layer

Fig. 2 shows the MSE of CNN model for different number of PEs in last hidden layer.

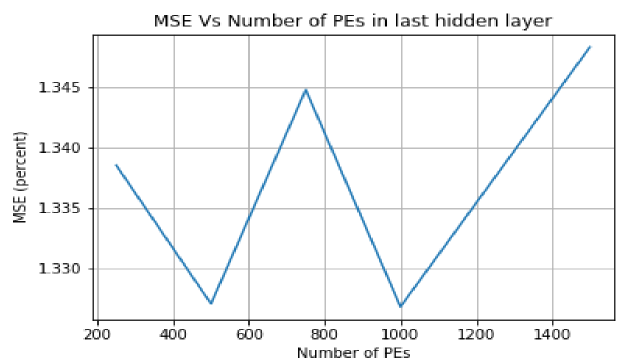

Fig. 2. MSE for number of PEs in last hidden layer

Fig. 3 shows the learning curves of MLPNN model for different number of PEs in last hidden layer.

B. Learning Rate

As we are using stochastic gradient descent, learning rate or step size is a very important hyper parameter. It can affect whether our model can miss in a global optima or not. It is important to vary step size in a large range and test its impact on the model performance. Here, we have experimented with different step sizes and show the performance of the CNN model.

Fig. 4 shows the Accuracy of CNN model for different step sizes.

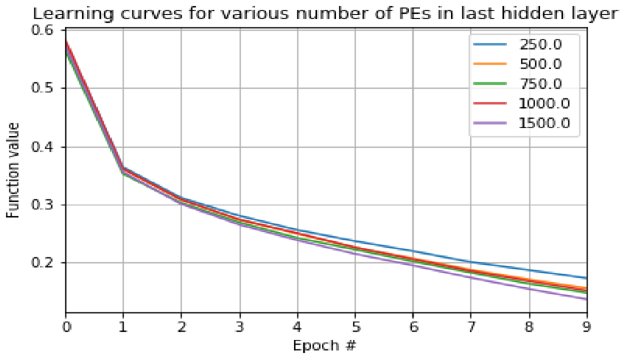

Fig. 3. Learning Curves for different numnber of PEs in last hidden layer

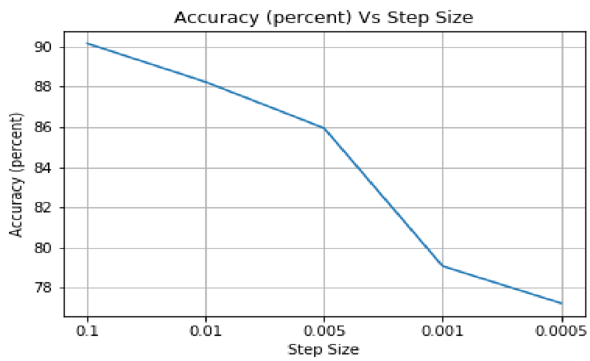

Fig. 4. Accuracy Vs step size

Fig. 5 shows the MSE of CNN model for different step sizes.

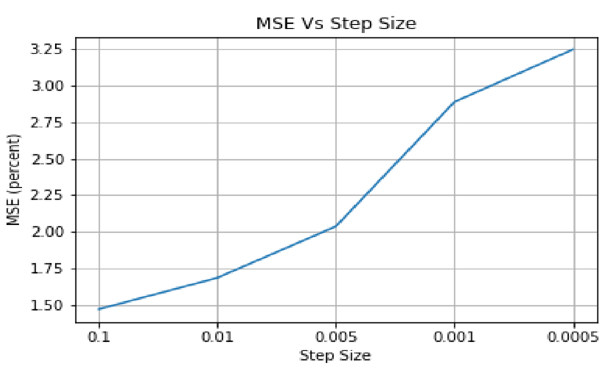

Fig. 5. MSE Vs step size

Fig. 6 shows the learning curves of CNN model for different step sizes.

C. Training sizes

In this subsection, we have experimented with different number of training samples and we show the per-

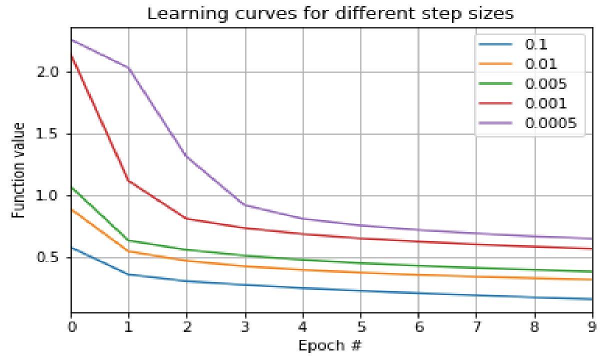

Fig. 6. Learning Curves Vs step size

formance of the CNN model for these different training testing split. Fig. **??** shows the performance metrics of MLPNN model for different activation functions.

Fig. 7 shows the Accuracy of CNN model for different training sizes.

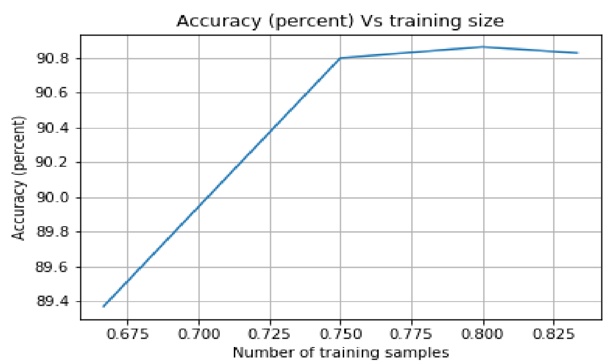

Fig. 7. Accuracy Vs training size

Fig. 8 shows the MSE of CNN model for different training sizes.

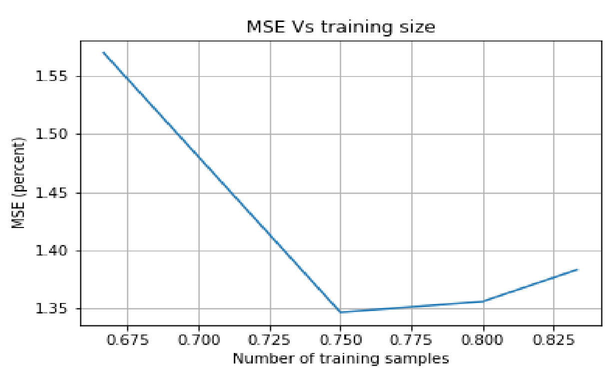

Fig. 8. MSE Vs training size

Fig. 9 shows the learning curves of CNN model for different training sizes.

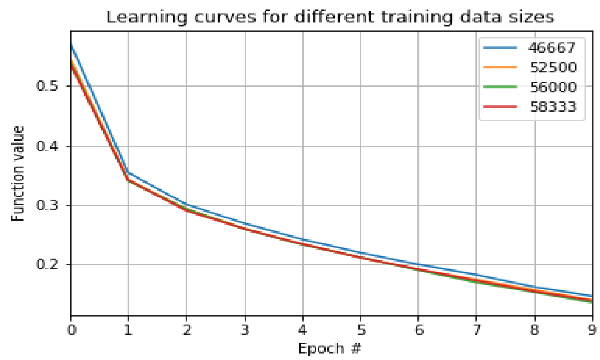

Fig. 9. Learning Curves Vs training size

D. Stopping Condition

In this subsection, we have experimented with different number of epochs as the stopping condition and we show the performance of the CNN model for 10, 20, 30, 40, 50 epochs of model training.

Fig. 10 shows the Accuracy of CNN model for different epochs.

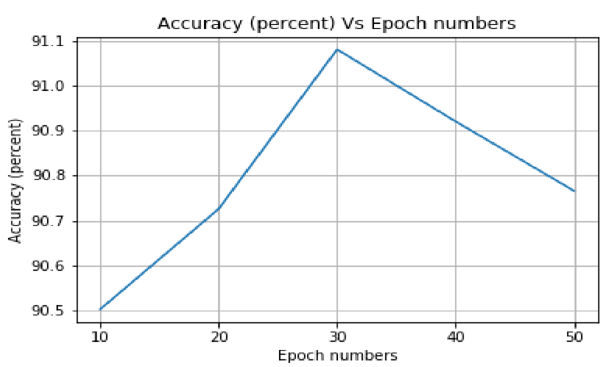

Fig. 10. Accuracy Vs epochs

Fig. 11 shows the MSE of CNN model for different epochs

Fig. 12 shows the learning curves of CNN model for different epochs.

Fig. **??** shows the learning curves of CNN model for different number of training epochs. We can notice that performance initially increases as the number of epochs increases and starts to decrease after 30 epochs because of overfitting. Hence, we decided to use 30 epochs for all the other experiments. In Fig. **??**, the legend bar shows the number of training epochs.

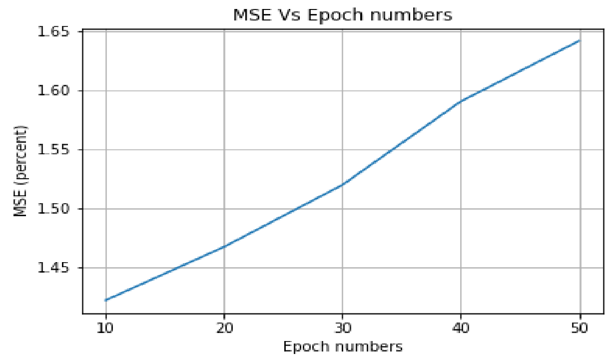

Fig. 11. MSE Vs Vs epochs

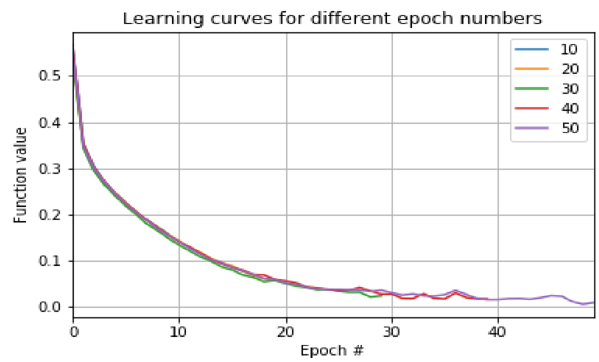

Fig. 12. Learning Curves Vs Vs epochs

E. CNN model with data augmentation

In this subsection, we talk about the data augmentation method. We created a dataset with thrice the size (210000) of original dataset (70,000) by applying transformations such as tilt, zoom and shift. For each image we would tilt the image with an angle randomly selected from the range of -30 to +30 degrees, and shift the image in both x and y directions randomly with a shift of 0.1 and zoom the image in both x and y directions randomly in range of 0.9 to 1.2 times the original image.

We created 2 transformed versions for each image with random transform parameters to create a large dataset of slightly different images for better generalization of the model. We have provided confusion matrix which shows the classification labels for each of the 10 classes in the Fashion MNIST dataset. We have also provided classification report in terms of precision, recall and F1-score for each of the classes. Fig. 13 shows the confusion matrix for the fashion MNIST dataset with the optimal hyper-parameters for the augmented data.

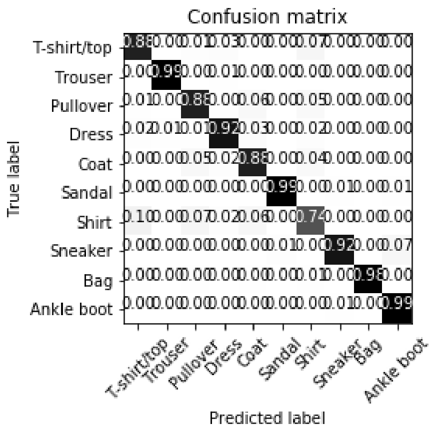

Fig. 13. Confusion Matrix

Fig. 14 shows the classification report for the fashion MNIST dataset with the optimal hyper-parameters and augmented data. From Fig. 13, we can say that for majority of classes, the true prediction rate is more than 90%.

	precision	recall	f1-score	support
T-shirt/top	0.87	0.88	0.87	4231
Trouser	0.99	0.99	0.99	4255
Pullover	0.86	0.88	0.87	4103
Dress	0.93	0.92	0.92	4205
Coat	0.85	0.88	0.87	4203
Sandal	0.98	0.99	0.99	4269
Shirt	0.80	0.74	0.77	4143
Sneaker	0.98	0.92	0.95	4284
Bag	0.99	0.98	0.99	4165
Ankle boot	0.93	0.99	0.96	4142
micro avg	0.92	0.92	0.92	42000
macro avg	0.92	0.92	0.92	42000
weighted avg	0.92	0.92	0.92	42000

Fig. 14. classification report

Confusion matrices and Classification reports for MLPNN model are shown in Fig. 15 and Fig. 16 respectively. We can observe that the test accuracy has increased from 88% to 92% overall. If we do not consider the bad performance of shirt class, the overall test accuracy would be around 93%.

IV. CONCLUSIONS

In this project, we trained a CNN model with Fashion MNIST dataset to solve a classification problem. We chose the hyper parameters such as optimal number of units in the hidden layers, learning rates, number of epochs by considering performance metrics such as accuracy and MSE for test dataset. We provided plots of accuracy and MSE for each of the experiment

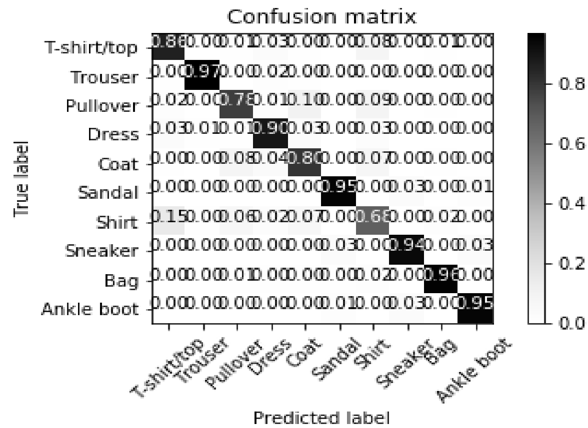

Fig. 15. Confusion Matrix

	precision	recall	f1-score	support
T-shirt/top	0.81	0.86	0.83	2345
Trouser	0.98	0.97	0.97	2329
Pullover	0.82	0.78	0.80	2350
Dress	0.88	0.90	0.89	2328
Coat	0.79	0.80	0.80	2343
Sandal	0.95	0.95	0.95	2337
Shirt	0.71	0.68	0.69	2327
Sneaker	0.93	0.94	0.94	2355
Bag	0.96	0.96	0.96	2316
Ankle boot	0.95	0.95	0.95	2303
avg / total	0.88	0.88	0.88	23333

Fig. 16. classification report

conducted. This project report serves as a guide to understand how a classification problem can be solved on popular dataset such as Fashion MNIST dataset and how to select various hyper parameters and what kind of plots are important and how they are helpful to understand the performance of CNN model. We also created a dataset containing 210000 images from the original data by applying transformations such as zoom, shift and tilt for each image. We trained the CNN model with this new augmented dataset and obtained much higher training and test accuracy. Test accuracy before data augmentation was 91% and after data augmentation the test accuracy increased to 92%. We also show the comparison between performance of CNN and MLPNN models.

REFERENCES

[1] Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Han Xiao, Kashif Rasul, Roland Vollgraf.