Report for Assignment-1

Introduction:

For this assignment we had to implement Deep Learning models for the data set Fashion MNIST. We had to create at least 16 models with varying numbers of hidden layers, units present in each layer and activation function for each layer. We had to comment about the error generated by training each model and how it could be improved in the next further models by changing any of the above mentioned parameters. Training dataset consisted of 60000 images and the testing dataset consisted of 10000 images. Each image consisted of 784 pixels. We used frameworks like Keras and Tensorflow to aid in the implementation of our models. Following is the summary about the architecture and the performance of our Deep Learning models.

Models implemented:

Model 1:

- Architecture consists of 2 hidden layers having 32 and 16 units.
- Both these layers have ReLU as their activation function.
- The output layer has Softmax as its activation function.
- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 53.17% and test accuracy of 51.32%.

Model 2:

- Architecture consists of 1 hidden layer having 1024 units.
- Activation function for the hidden layer is ReLU.
- The output layer has Softmax as its activation function.
- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 90.19% and test accuracy of 85.77%. Thus there is overfitting in this model.

Model 3:

- Next we try models having more units in hidden layers.
- This model has 2 hidden layers having 128 and 64 hidden units.
- Both of them have ReLU as their activation function.
- The output layer has Softmax as its activation function.
- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 89.24% and test accuracy of 84.18%. Thus there is overfitting in this model.
- This model has more hidden units than model 1.

Model 4:

- This model has 3 hidden layers consisting of 128, 64 and 32 units.
- All 3 have ReLU as their activation function.
- The output layer has Softmax as its activation function.
- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 92.32% and test accuracy of 85.37%. Thus there is overfitting in this model.

Model 5:

- This model has 2 hidden layers consisting of 256 and 128 units.
- Both of them have ReLU as their activation function.
- The output layer has Softmax as its activation function.
- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 94.32% and test accuracy of 84.88%. Thus there is overfitting in this model.

Model 6:

- In this model we make use of Sigmoid as an activation function.
- This model has 3 hidden layers consisting of 128, 64 and 32 units.
- First layer has ReLU as activation function and next 2 have Sigmoid as activation function.
- The output layer has Softmax as its activation function.

- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 91.21% and test accuracy of 87.60%.

Model 7:

- Previous models seemed to work well so we tried to increase the number of nodes in hidden layers.
- To counter overfitting we added regularisation terms in the hidden layers.
- This model has 3 hidden layers consisting of 128, 64 and 64 units.
- First layer has ReLU as activation function and next 2 have Sigmoid as activation function.
- The output layer has Softmax as its activation function.
- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 93.44% and test accuracy of 88.94%.

Model 8:

- In this model we increased units in each layer and maintained regularisation terms as the previous model.
- This model has 3 hidden layers consisting of 512, 64 and 32 units.
- First layer has ReLU as activation function and next 2 have Sigmoid as activation function.
- The output layer has Softmax as its activation function.
- The loss function is sparse_categorical_crossentropy.
- We achieved a training accuracy of 97.17% and test accuracy of 90.02%. Thus there is overfitting in this model.
- Because of overfitting, dropout should be used in next models.

Model 9:

- Architecture consists of 3 hidden layers with 512, 64, 32 nodes in each layer.
- We used KL divergence as the loss function.
- We used sigmoid as the activation function.
- We achieved a training accuracy of 10.18% and a test accuracy of 11.69%.
- This signifies poor performance as compared to the cross entropy loss function.

Model 10:

- Architecture consists of 4 hidden layers with 32 nodes each in each layer.
- We used cross entropy as the loss function.
- We used ReLU as the activation function.
- We achieved a training accuracy of 85.92% and a test accuracy of 84.02%.
- This shows that even after keeping a lesser number of nodes in each layer, increasing the number of layers gives decent performance.

Model 11:

- We observe that models 5-8 have given us very similar results.
- Hence we may attempt to use dropout regularization after the first 2 hidden layers in an attempt to reduce overfitting.
- Dropout randomly sets some of the outgoing edges from neurons to zero so as to avoid overfitting
- *Success!*
- Dropout 0.2 after 1st 2 hidden layers.
- Architecture consists of 3 hidden layers with 512, 64, 32 nodes in each layer.
- We used cross entropy as the loss function.
- We used ReLU as the activation function for the first layer and sigmoid as the activation function for the other layers.
- We achieved a training accuracy of 94.3% and a test accuracy of 89.74%.

Model 12:

- We constructed a deeper neural network with 5 hidden layers having 512,256,256,128,64 nodes in the hidden layers with dropout on the first 4 layers
- We used cross entropy as the loss function.
- We used ReLU as the activation function.
- We achieved a training accuracy of 94.3% and a test accuracy of 88.59%.

Model 13:

- We constructed a model with the same architecture as Model 12 but increased the number of epochs to 400 (instead of 100), as we thought that the model may take more iterations as it is a deep network.
- We achieved a training accuracy of 97.75% and a test accuracy of 89.61%.
- Thus, we observe that compared to model 12, training accuracy has gone up by a significant margin, the difference in test accuracy is not so big.
- Also, the test accuracy barely goes up after the 100th epoch, this shows that we might be overfitting the data.

Model 14:

- Architecture consists of 3 hidden layers with 512, 256, 64 nodes in each layer.
- We have also used Dropout layers after the first 2 hidden layers.
- We used tanh as the activation function and cross entropy as the loss function.
- We achieved a training accuracy of 84.07% and a test accuracy of 85.25%.
- This accuracy is lower than the accuracy that we achieved as compared to ReLU and sigmoid activation functions.

Model 15:

- Architecture consists of 5 hidden layers with 64 nodes at each layer
- We ran this model for 400 epochs, used sigmoid as the activation function and cross entropy as the loss function.
- We achieved a training accuracy of 84.07% and a test accuracy of 85.39%.
- We observe that the model is not able to converge even after 400 epochs due to so many hidden layers.

Model 16:

- Previously KL divergence loss functions and tanh activation function gave us poor results so we decided to combine both of them and observe the output.
- Architecture consists of 128,64,32 nodes in the 3 hidden layers.
- We achieved a Training accuracy of 10.13% Test accuracy of 5.57%

- Thus, these functions gave us very poor performance with a lot of spikes in the graph.

Conclusion:

Model 11,12 and 13 gave us the highest test accuracies of 89.74% , 88.59% and 89.61% respectively. In all the 3 models, we used cross entropy as our loss function and put Dropout layers. In model 11, we used a mix of ReLU and sigmoid as the activation functions and was run for 100 epochs. Model 12 consisted of a deeper network as compared to Model 11 and ReLU was used as the activation function. We observed that Model 12 gave slightly lesser accuracy as compared to Model 11, which might be due to lower number of epochs. Thus, in Model 12, we increased the number of epochs to 400 resulting in a slightly better test accuracy almost similar to Model 11. This shows that deeper models may not always give us better results. Also, dropout gave us really good results by eliminating overfitting to some extent and removing unnecessary complexity.

Primarily, we used 2 loss functions namely cross entropy and KL divergence. We observed that KL divergence performed extremely poorly with both the activation functions. Also tanh, although provided decent performance, was still subpar as compared to ReLU and Sigmoid which performed really well.

A model which consists of 5 layers took a very long time to run and even after 400 epochs was not able to converge properly. We also attempted to run a model with 10 hidden layers for 400 epochs but this gave us very low accuracy. We decided to increase the number of epochs to 4000 as it was a very deep model but unfortunately this crashed our system.

DONE BY:
Navam Shrivastav 2019A7PS0092H
Aryan Saraf 2019A7PS0138H
Arjun Prasad 2019A7PS0183H
Rishabh Jain 2018A7PS0275H