

Name(s): Chiranjeevi Konduru

NetID(s): konduru4

Team name on Kaggle leaderboard: C_K

For each of the sections below, your reported test accuracy should approximately match the accuracy reported on Kaggle.

Perceptron

Briefly describe the hyperparameter settings you tried. In particular, you should list the different values for learning rate and number of epochs you tried. You should also mention whether adding a learning rate decay helped and how you implemented this decay. Report the optimal hyperparameter setting you found in the table below. Report your training, validation, and testing accuracy with your optimal hyperparameter setting.

Following values for learning rate [0.01, 0.005, 0.001] and the epochs = [20, 50, 100, 1000] were experimented on the datasets. The combination given below for each of the datasets gave an optimal accuracy. Initially, added a linear lr_decay for each epoch, however this didn't yield good accuracy for MNIST dataset. Then tried exponential decay, this helped to cross the 81% accuracy on the training dataset.

RICE DATASET

Optimal hyperparameters:	Learning rate = 0.001, epochs = 1000
Training accuracy:	99.184
Validation accuracy:	99.147
Test accuracy:	99.037

Fashion-MNIST DATASET

Optimal hyperparameters:	Learning rate = 0.001, epochs = 100
Training accuracy:	84.29
Validation accuracy:	83.16
Test accuracy:	81.95

SVM

Describe the hyperparameter tuning you tried for learning rate, number of epochs, and regularization constant. Report the optimal hyperparameter setting you found in the table below. Also report your training, validation, and testing accuracy with your optimal hyperparameter setting.

Tried reducing the learning by 10 folds $lr = [0.1, 0.01, 0.001, 0.001]$ and epochs from 5 to 25. There wasn't much improvement in terms of accuracy for the MNIST-fashion dataset. Even after using the mini batch technique for processing and tried adjusting the learning rate with exponential decay too, still got a low accuracy on Rice. The below are the hyperparameters that gave optimal results on both the datasets

RICE DATASET

Optimal hyperparameters:	Learning rate = 0.001, epochs = 20, reg_const = 0.005
Training accuracy:	79.45
Validation accuracy:	79.15
Test accuracy:	79.81

Fashion-MNIST DATASET

Optimal hyperparameters:	Learning rate = 0.001, epochs = 20, reg_const = 0.005
Training accuracy:	84.4
Validation accuracy:	83.48
Test accuracy:	82.05

Softmax

Once again, describe the hyperparameter tuning you tried for learning rate, number of epochs, and regularization constant. Report the optimal hyperparameter setting you found in the table below. Also report your training, validation, and testing accuracy with your optimal hyperparameter setting.

Didn't get great results with hyperparam tuning for the Rice dataset. Initially went with lower epochs, but accuracy was way worse, so increased the number of epochs, with a hope that it would converge. But the

results were not that great. However, reducing the batch size with the use of Exponential learning rate decay, it did help me in getting a reasonable accuracy.

RICE DATASET

Optimal hyperparameters:	Learning rate = 0.01, epochs = 1000, reg_const = 0.001
Training accuracy:	83.118
Validation accuracy:	83.172
Test accuracy:	83.200

Fashion-MNIST DATASET

Optimal hyperparameters:	Learning rate = 0.001, epochs = 100, reg_const = 0.001
Training accuracy:	84.750
Validation accuracy:	83.77
Test accuracy:	82.42

Logistic

Once again, describe the hyperparameter tuning you tried for learning rate, number of epochs, and threshold. Report the optimal hyperparameter setting you found in the table below. Also report your training, validation, and testing accuracy with your optimal hyperparameter setting.

Learning rates[0.1, 0.5, 0.01] and epochs [1000, 10000, 100000] were tried on this dataset for logistic regression. Got around 98% accuracy with epoch = 1000, lr = 0.01, then increased it by 10 fold each time, which helped to get the optimal accuracy of 99.91 %

RICE DATASET

Optimal hyperparameters:	Learning rate = 0.01, epochs = 100000
Training accuracy:	99.926679
Validation accuracy:	99.917514
Test accuracy:	99.91714

