

Name(s): Wan Yoo

NetID(s): wanyoo2

Team name on Kaggle leaderboard: Wan Yoo

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

Perceptron

For the learning rate, I found that overly increasing them turned out high training accuracy but the test accuracy was rather low, which was also the case for increasing the number of epochs. I am assuming this is due to overfitting. Lowering these parameters caused the accuracy to drop, perhaps lowering the learning rate and increasing the number of epochs will make the model converge eventually. For me, I was unable to find a sufficient learning rate decay that significantly affects the accuracy. As of the initialization of weights, I found that setting them to zero simply converges the same as when the weights are initialized to random numbers but weights of zeros seem to require more epochs to gain similar accuracy.

MUSHROOM DATASET

Optimal hyperparameters:	Learning Rate: 0.5 Epochs: 10 Initial weights: random numbers of (#class, #D)
Training accuracy:	94.562987
Validation accuracy:	94.338462
Test accuracy:	94.030769

CIFAR DATASET

Optimal hyperparameters:	Learning Rate: 0.5 Epochs: 10 Initial weights: random numbers of (#class, #D)
Training accuracy:	37.4825
Validation accuracy:	32.4
Test accuracy:	32.4

SVM

Similar to the Perceptron, I found that the model convergence was most stable at the learning rate of 0.5. Also, the number of epochs only increased training accuracy. Thus I decided to stick with the hyperparameters above. When tuning the regression rate, setting them to 0.005 sufficed and lowering them to 0.0005 did not dramatically increase the accuracy compared to the SoftMax model. For the learning rate decay, I was unable to find a sufficient learning rate decay that significantly affects the accuracy. As of the initialization of weights, I found that setting them to zero simply converges the same as when the weights are initialized to random numbers but weights of zeros seem to require more epochs to gain similar accuracy.

MUSHROOM DATASET

Optimal hyperparameters:	Learning rate: 0.5 Epochs: 10 Regression const: 0.0005 Initial weight: random numbers(#class, #D)
Training accuracy:	82.498974
Validation accuracy:	80.369231
Test accuracy:	83.076923

CIFAR DATASET

Optimal hyperparameters:	Learning rate: 0.5 Epochs: 10 Regression const: 0.0005 Initial weight: random numbers(#class, #D)
Training accuracy:	34.3275
Validation accuracy:	30.74
Test accuracy:	30.83

Softmax

For the SoftMax, I attempted the gradient calculation thus batch sizes was one of the important hyperparameters. Similar to the above models, the learning rate, the number of epochs, and the initial weight did not really cause a significant difference for the accuracies but the size of batch and the regression rate really made the difference. With batch sizes > 400, I was getting 22% accuracies and batches sizes < 100 gave me 18% accuracies. I found that between 100 and 200 to be the optimal batch sizes. With regression rate of 0.005, the model converged on average 25% accuracies; however, when I lowered the regression rate to 0.005, I found that the model was much stable and accuracies increased significantly.

MUSHROOM DATASET

Optimal hyperparameters:	Learning rate: 0.5 Epochs: 10 Batch size: 200 Regression const: 0.0005 Initial weight: random numbers(#class, #D)
Training accuracy:	88.120640
Validation accuracy:	87.2
Test accuracy:	86.769231

CIFAR DATASET

Optimal hyperparameters:	Learning rate: 0.5 Epochs: 10 Batch size: 200 Regression const: 0.0005 Initial weight: random numbers(#class, #D)
Training accuracy:	33.385
Validation accuracy:	30.9
Test accuracy:	30.99

Logistic

Very similar to the above models, changing the learning rate, the number of epochs, and the initial weights was unable to affect the test accuracies significantly. However, the threshold was important because that was the determinant of the label for my model. Before, the threshold changed to 0 and the model was giving me 50% accuracies and I realized that the mushroom dataset had 0/1 label and that caused significant improvement of the model.

MUSHROOM DATASET

Optimal hyperparameters:	Learning rate: 0.5 Epochs: 10 Threshold: 0.5 Initial weight: random numbers(#class ,#D)
Training accuracy:	87.238408
Validation accuracy:	85.661538
Test accuracy:	87.323077