

**Name(s): Alice Huang, Matthew Tang**

**NetID(s): aliceh4, mhtang2**

**Team name on Kaggle leaderboard: AliceMatthew**

**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.*

For each epoch we iterated over every sample and updated the parameters after each sample. Learning rate decay was not tried or needed. We noticed that as we increased the number of epochs and decreased our learning rate, our model performed better. We did not add a learning rate decay since we assumed our test accuracies without it were sufficient.

We tried epochs = 20, 50, 100 and lr = 0.05, 0.005, 0.0005 as well but found that the hyperparameters listed below performed the best. As epochs increased and learning rate decreased, our accuracy was the highest. If our learning rate was too low and our number of epochs was low, we found that our accuracy was quite poor. This goes for both the mushroom and fashion datasets.

#### **MUSHROOM DATASET**

Optimal hyperparameters:	LR: 1.5E-5; Epochs:250
Training accuracy:	92.39%
Validation accuracy:	91.45%
Test accuracy:	93.42%

We tried epochs = 20, 50, 100, 250 and lr = 0.05, 0.005, 0.0005, 0.00005 as well but none of those seemed to beat the benchmark on Kaggle.

#### **Fashion-MNIST DATASET**

Optimal hyperparameters:	LR: 5E-7; Epochs: 250
--------------------------	-----------------------

Training accuracy:	80.52%
Validation accuracy:	78.88%
Test accuracy:	78.51%

## 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.*

We implemented learning rate decay as exponential decay towards 0 on every epoch. The model performs significantly better (~6% accuracy gain) with learning rate decay. We also used mini batches with size 512 for both datasets.

We tried  $lr = 0.05, 0.005, 0.2$  and  $epochs = 5, 10, 20$  and  $reg\_const = 0.05, 0.1$ . We found that increasing our  $reg\_const$  seemed to increase our accuracy and that the final learning rates were not as low as the final learning rates for our perceptron model. Because we had exponential decay, we found that we did not need to have that many epochs (because the learning rate converged to 0 at a certain point).

## MUSHROOM DATASET

Optimal hyperparameters:	LR: 0.1; Epochs: 10, Regularization: 0.1; Batch size: 512
Training accuracy:	83.03%
Validation accuracy:	81.35%
Test accuracy:	81.78%

We tried  $lr = 0.05, 0.005, 0.1, 0.7$  and  $epochs = 5, 10, 20$  and  $reg\_const = 0.05, 0.1$ . Initially, our model was not performing well but adding an exponential decay improved our performance to beat the benchmarks.

## Fashion-MNIST DATASET

Optimal hyperparameters:	LR: 0.08; Epochs: 10; Regularization: 0.1; Batch size: 512
Training accuracy:	84.3%
Validation accuracy:	83.2%

Test accuracy:	82.75%
----------------	--------

## 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.*

We used no learning rate decay since the model performed well without it, and the decay did not improve performance. We used mini batches of size 250 for both datasets.

We tried  $lr = 0.05, 0.005, 0.5$  and  $epochs = 10, 12, 100$  and  $r\_const = 0.05, 0.1$  for both our mushroom and fashion datasets. We found that softmax did not need a low learning rate and high number of epochs in order to perform well. Increasing the number of epochs/lowering our learning rate did not perform as well and also took a long time to run.

### MUSHROOM DATASET

Optimal hyperparameters:	LR 0.5; Epochs 10; Regularization 0.5; Batch size 250
Training accuracy:	88.45%
Validation accuracy:	86.89%
Test accuracy:	87.02%

### Fashion-MNIST DATASET

Optimal hyperparameters:	LR 0.5; Epochs 10; Regularization: 0.5; Batch size: 250
Training accuracy:	84.11%
Validation accuracy:	82.16%
Test accuracy:	81.65%

## 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.*

No learning rate decay was used. We updated parameters after every sample, and iterated over every sample during each epoch.

We tried  $lr = 0.05, 0.005$  and  $epochs = 10, 20, 50$ . We kept the threshold at 0.5 the whole time since our model seemed to be performing fine without us changing it. Our logistic model ran quite quickly so we decided to use a high number of epochs and a low learning rate to achieve the optimal accuracies listed below.

#### MUSHROOM DATASET

Optimal hyperparameters:	LR 5E-4; Epochs 100; Threshold 0.5
Training accuracy:	93.66%
Validation accuracy:	92.80%
Test accuracy:	92.86%