Name(s):Punit K. Jha
NetID(s):punit2
Team name on Kaggle leaderboard:

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.

I tried the following set of the hyperparameter settings:

• Epoch: ○ 10 ○ 100 ○ 500 ○ 1000	• Learning rate: ○ 0.0005 ○ 0.005 ○ 0.05 ○ 0.5 ○ 1.0 ○ 1.5

Yes, implementation of the learning rate decay helped increase the accuracy of the predictions by descending into areas of the loss landscape that are “more optimal” in the later stages of gradient descent while initially a higher learning rate helped rapid exploration of the loss landscape. I implemented a linear constant decay learning rate which decreases the learning rate (if its magnitude is greater than 1) after 75% of the iterations (epoch) have taken place by a magnitude of 0.001. I have used the following code to achieve this objective:

```
        if (self.lr>1 and epo_counter>0.75*iti):
                        self.lr-=0.001
```

To train the Perceptron classifier on the Mushroom data set, I have used the update rules for Perceptron as a binary classifier where y={-1,1}. So, I have converted the y values using the following lines of codes:

```
y_train_MR = np.asarray([-1 if each_element == 0 else 1 for each_element in y_train_MR])
```

```
y_val_MR = np.asarray([-1 if each_element == 0 else 1 for each_element in y_val_MR])

y_test_MR = np.asarray([-1 if each_element == 0 else 1 for each_element in y_test_MR])
```

MUSHROOM DATASET

Optimal hyperparameters:	Learning rate = 0.005 Epoch= 100
Training accuracy:	93.22%
Validation accuracy:	92.30%
Test accuracy:	92.67%

CIFAR DATASET

Optimal hyperparameters:	Learning rate = 0.0005 Epoch= 100
Training accuracy:	29.58%
Validation accuracy:	29.24%
Test accuracy:	29.44%

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.

I tried the following set of the hyperparameter settings:

• Learning rate: ○ 0.005 ○ 0.05 ○ 0.5 ○ 1.0 ○ 10	• Regression Consts.: ○ 0.005 ○ 0.05 ○ 0.5 ○ 1.0 ○ 1.5	• Epoch : ○ 1 ○ 5 ○ 10 ○ 15 ○ 20

I have generalized the SVM to both binary and multiclass classification where y={0,1} for the binary case and y={0,1,2,..,9} for the multiclass case. Hence to train on and classify the Mushroom dataset, I have converted the labels for the Mushroom data set to y={0,1} using the following lines of codes:

```
y_train_MR = np.asarray([0 if each_element == -1 else 1 for each_element in y_train_MR])

y_val_MR = np.asarray([0 if each_element == -1 else 1 for each_element in y_val_MR])

y_test_MR = np.asarray([0 if each_element == -1 else 1 for each_element in y_test_MR])
```

MUSHROOM DATASET

Optimal hyperparameters:	Learning rate= 0.005 Epoch= 15 Regression Const.= 0.05
Training accuracy:	91.73%
Validation accuracy:	91.56%
Test accuracy:	91.20%

CIFAR DATASET

Optimal hyperparameters:	Learning rate= 0.005 Epoch= 15 Regression Const.= 0.0001
Training accuracy:	36.72%
Validation accuracy:	32.92%
Test accuracy:	32.83%

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.

I tried the following set of the hyperparameter settings:

• Learning rate:	• Regression Consts.:	• Epoch :
○ 0.005	○ 0.005	○ 1
○ 0.05	○ 0.05	○ 5
○ 0.5	○ 0.5	○ 10

○ 1.0 ○ 10	○ 1.5	○ 15

MUSHROOM DATASET

Optimal hyperparameters:	Learning rate = 0.05 Epoch =15 Regression Const. =0.0005
Training accuracy:	90.82%
Validation accuracy:	89.23%
Test accuracy:	88.24%

CIFAR DATASET

Optimal hyperparameters:	Learning rate = 0.005 Epoch=15 Regression Const. = 0.0005
Training accuracy:	34.45%
Validation accuracy:	30.58%
Test accuracy:	30.70%

Logistic

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.

I tried the following set of the hyperparameter settings:

• Learning rates: ○ 0.0005 ○ 0.005 ○ 0.05 ○ 0.5 ○ 1 ○ 1.5	• Epochs: ○ 10 ○ 100 ○ 500 ○ 1000

As per the lecture slides, the gradient equations for Logistic regression assumed that the response y={-1,1} and since my implementation of the algorithms is based on the lectures, I have converted the responses of the Mushroom dataset back to y={-1,1} using the following lines of code:

```
y_train_MR = np.asarray([-1 if each_element == 0 else 1 for each_element in y_train_MR])

y_val_MR = np.asarray([-1 if each_element == 0 else 1 for each_element in y_val_MR])

y_test_MR = np.asarray([-1 if each_element == 0 else 1 for each_element in y_test_MR])
```

MUSHROOM DATASET

Optimal hyperparameters:	Learning rate = 0.05 Epoch= 10
Training accuracy:	87.85%
Validation accuracy:	93.29%
Test accuracy:	93.29%