CS 444 Assignment-1

February 15, 2022

[15]:
```
import random
import numpy as np
from data_process import get_FASHION_data, get_MUSHROOM_data
from scipy.spatial import distance
from models import Perceptron, SVM, Softmax, Logistic
from kaggle_submission import output_submission_csv
%matplotlib inline

# For auto-reloading external modules
# See http://stackoverflow.com/questions/1907993/
 ↪autoreload-of-modules-in-ipython
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

1 Loading Fashion-MNIST

In the following cells we determine the number of images for each split and load the images. TRAIN_IMAGES + VAL_IMAGES = (0, 60000] , TEST_IMAGES = 10000

[16]:
```
# You can change these numbers for experimentation
# For submission we will use the default values
TRAIN_IMAGES = 50000
VAL_IMAGES = 10000
normalize = True
```

[17]:
```
data = get_FASHION_data(TRAIN_IMAGES, VAL_IMAGES, normalize=normalize)
X_train_fashion, y_train_fashion = data['X_train'], data['y_train']
X_val_fashion, y_val_fashion = data['X_val'], data['y_val']
X_test_fashion, y_test_fashion = data['X_test'], data['y_test']
n_class_fashion = len(np.unique(y_test_fashion))
```

2 Loading Mushroom

In the following cells we determine the splitting of the mushroom dataset. TRAINING + VALIDATION = 0.8, TESTING = 0.2

[18]:
```
# TRAINING = 0.6 indicates 60% of the data is used as the training dataset.
VALIDATION = 0.2
```

[19]:
```
data = get_MUSHROOM_data(VALIDATION)
X_train_MR, y_train_MR = data['X_train'], data['y_train']
X_val_MR, y_val_MR = data['X_val'], data['y_val']
X_test_MR, y_test_MR = data['X_test'], data['y_test']
n_class_MR = len(np.unique(y_test_MR))

print("Number of train samples: ", X_train_MR.shape[0])
print("Number of val samples: ", X_val_MR.shape[0])
print("Number of test samples: ", X_test_MR.shape[0])
```

```
Number of train samples:  4874
Number of val samples:  1625
Number of test samples:  1625
```

2.0.1 Get Accuracy

This function computes how well your model performs using accuracy as a metric.

[20]:
```
def get_acc(pred, y_test):
    return np.sum(y_test == pred) / len(y_test) * 100
```

3 Perceptron

Perceptron has 2 hyperparameters that you can experiment with: - **Learning rate** - controls how much we change the current weights of the classifier during each update. We set it at a default value of 0.5, but you should experiment with different values. We recommend changing the learning rate by factors of 10 and observing how the performance of the classifier changes. You should also try adding a **decay** which slowly reduces the learning rate over each epoch. - **Number of Epochs** - An epoch is a complete iterative pass over all of the data in the dataset. During an epoch we predict a label using the classifier and then update the weights of the classifier according to the perceptron update rule for each sample in the training set. You should try different values for the number of training epochs and report your results.

You will implement the Perceptron classifier in the **models/perceptron.py**

The following code: - Creates an instance of the Perceptron classifier class - The train function of the Perceptron class is trained on the training data - We use the predict function to find the training accuracy as well as the testing accuracy

3.1 Train Perceptron on Fashion-MNIST

[194]:
```
lr = 0.05
n_epochs = 80

percept_fashion = Perceptron(n_class_fashion, lr, n_epochs)
percept_fashion.train(X_train_fashion, y_train_fashion)
```

```
epoch 1
The training accuracy is: 77.574000
epoch 2
The training accuracy is: 79.788000
epoch 3
The training accuracy is: 76.232000
epoch 4
The training accuracy is: 78.882000
epoch 5
The training accuracy is: 79.030000
epoch 6
The training accuracy is: 80.246000
epoch 7
The training accuracy is: 78.408000
epoch 8
The training accuracy is: 79.098000
epoch 9
The training accuracy is: 81.442000
epoch 10
The training accuracy is: 79.300000
epoch 11
The training accuracy is: 80.472000
epoch 12
The training accuracy is: 79.686000
epoch 13
The training accuracy is: 79.042000
epoch 14
The training accuracy is: 79.440000
epoch 15
The training accuracy is: 80.756000
epoch 16
The training accuracy is: 81.080000
epoch 17
The training accuracy is: 79.534000
epoch 18
The training accuracy is: 81.334000
epoch 19
The training accuracy is: 80.572000
epoch 20
The training accuracy is: 81.312000
```

```
epoch 21
The training accuracy is: 81.034000
epoch 22
The training accuracy is: 81.052000
epoch 23
The training accuracy is: 80.856000
epoch 24
The training accuracy is: 80.868000
epoch 25
The training accuracy is: 81.374000
epoch 26
The training accuracy is: 81.834000
epoch 27
The training accuracy is: 81.754000
epoch 28
The training accuracy is: 82.776000
epoch 29
The training accuracy is: 82.160000
epoch 30
The training accuracy is: 82.466000
epoch 31
The training accuracy is: 82.402000
epoch 32
The training accuracy is: 82.498000
epoch 33
The training accuracy is: 82.504000
epoch 34
The training accuracy is: 83.228000
epoch 35
The training accuracy is: 83.164000
epoch 36
The training accuracy is: 83.548000
epoch 37
The training accuracy is: 83.482000
epoch 38
The training accuracy is: 83.316000
epoch 39
The training accuracy is: 83.762000
epoch 40
The training accuracy is: 83.678000
epoch 41
The training accuracy is: 83.852000
epoch 42
The training accuracy is: 83.622000
epoch 43
The training accuracy is: 83.884000
epoch 44
The training accuracy is: 83.858000
```

```
epoch 45
The training accuracy is: 84.030000
epoch 46
The training accuracy is: 84.140000
epoch 47
The training accuracy is: 84.156000
epoch 48
The training accuracy is: 84.246000
epoch 49
The training accuracy is: 84.218000
epoch 50
The training accuracy is: 84.182000
epoch 51
The training accuracy is: 84.266000
epoch 52
The training accuracy is: 84.322000
epoch 53
The training accuracy is: 84.432000
epoch 54
The training accuracy is: 84.456000
epoch 55
The training accuracy is: 84.438000
epoch 56
The training accuracy is: 84.476000
epoch 57
The training accuracy is: 84.456000
epoch 58
The training accuracy is: 84.464000
epoch 59
The training accuracy is: 84.466000
epoch 60
The training accuracy is: 84.476000
epoch 61
The training accuracy is: 84.476000
epoch 62
The training accuracy is: 84.484000
epoch 63
The training accuracy is: 84.458000
epoch 64
The training accuracy is: 84.442000
epoch 65
The training accuracy is: 84.454000
epoch 66
The training accuracy is: 84.510000
epoch 67
The training accuracy is: 84.506000
epoch 68
The training accuracy is: 84.504000
```

```
epoch 69
The training accuracy is: 84.488000
epoch 70
The training accuracy is: 84.458000
epoch 71
The training accuracy is: 84.454000
epoch 72
The training accuracy is: 84.440000
epoch 73
The training accuracy is: 84.406000
epoch 74
The training accuracy is: 84.414000
epoch 75
The training accuracy is: 84.420000
epoch 76
The training accuracy is: 84.440000
epoch 77
The training accuracy is: 84.420000
epoch 78
The training accuracy is: 84.462000
epoch 79
The training accuracy is: 84.464000
epoch 80
The training accuracy is: 84.460000
```

[195]:
```
pred_percept = percept_fashion.predict(X_train_fashion)
print('The training accuracy is given by: %f' % (get_acc(pred_percept,␣
 ↪y_train_fashion)))
```

```
The training accuracy is given by: 84.460000
```

3.1.1 Validate Perceptron on Fashion-MNIST

[196]:
```
pred_percept = percept_fashion.predict(X_val_fashion)
print('The validation accuracy is given by: %f' % (get_acc(pred_percept,␣
 ↪y_val_fashion)))
```

```
The validation accuracy is given by: 79.850000
```

3.1.2 Test Perceptron on Fashion-MNIST

[197]:
```
pred_percept = percept_fashion.predict(X_test_fashion)
print('The testing accuracy is given by: %f' % (get_acc(pred_percept,␣
 ↪y_test_fashion)))
```

```
The testing accuracy is given by: 79.520000
```

3.1.3 Perceptron_Fashion-MNIST Kaggle Submission

Once you are satisfied with your solution and test accuracy, output a file to submit your test set predictions to the Kaggle for Assignment 1 Fashion-MNIST. Use the following code to do so:

[198]:
```
output_submission_csv('kaggle/perceptron_submission_fashion.csv',␣
↪percept_fashion.predict(X_test_fashion))
```

3.2 Train Perceptron on Mushroom

[203]:
```
lr = 0.05
n_epochs = 50

percept_MR = Perceptron(n_class_MR, lr, n_epochs)
percept_MR.train(X_train_MR, y_train_MR)
```

```
epoch 1
The training accuracy is: 65.059499
epoch 2
The training accuracy is: 79.318835
epoch 3
The training accuracy is: 81.042265
epoch 4
The training accuracy is: 88.428396
epoch 5
The training accuracy is: 82.416906
epoch 6
The training accuracy is: 87.217891
epoch 7
The training accuracy is: 88.079606
epoch 8
The training accuracy is: 89.864588
epoch 9
The training accuracy is: 83.381206
epoch 10
The training accuracy is: 92.018876
epoch 11
The training accuracy is: 85.945835
epoch 12
The training accuracy is: 84.366024
epoch 13
The training accuracy is: 88.325810
epoch 14
The training accuracy is: 83.299138
epoch 15
The training accuracy is: 85.043086
epoch 16
The training accuracy is: 84.735330
```

```
epoch 17
The training accuracy is: 92.613870
epoch 18
The training accuracy is: 88.797702
epoch 19
The training accuracy is: 83.873615
epoch 20
The training accuracy is: 85.145671
epoch 21
The training accuracy is: 91.690603
epoch 22
The training accuracy is: 93.762823
epoch 23
The training accuracy is: 88.613049
epoch 24
The training accuracy is: 95.445220
epoch 25
The training accuracy is: 92.388182
epoch 26
The training accuracy is: 90.562167
epoch 27
The training accuracy is: 90.664752
epoch 28
The training accuracy is: 93.803857
epoch 29
The training accuracy is: 92.778006
epoch 30
The training accuracy is: 95.568322
epoch 31
The training accuracy is: 96.347969
epoch 32
The training accuracy is: 96.060730
epoch 33
The training accuracy is: 94.009027
epoch 34
The training accuracy is: 94.727124
epoch 35
The training accuracy is: 95.424703
epoch 36
The training accuracy is: 95.917111
epoch 37
The training accuracy is: 95.609356
epoch 38
The training accuracy is: 95.835043
epoch 39
The training accuracy is: 95.855560
epoch 40
The training accuracy is: 95.732458
```

```
epoch 41
The training accuracy is: 95.609356
epoch 42
The training accuracy is: 95.814526
epoch 43
The training accuracy is: 95.609356
epoch 44
The training accuracy is: 95.650390
epoch 45
The training accuracy is: 95.588839
epoch 46
The training accuracy is: 95.588839
epoch 47
The training accuracy is: 95.629873
epoch 48
The training accuracy is: 95.629873
epoch 49
The training accuracy is: 95.486254
epoch 50
The training accuracy is: 95.465737
```

[204]:
```
pred_percept = percept_MR.predict(X_train_MR)
print('The training accuracy is given by: %f' % (get_acc(pred_percept,␣
 ↪y_train_MR)))
```

```
The training accuracy is given by: 95.465737
```

3.2.1 Validate Perceptron on Mushroom

[205]:
```
pred_percept = percept_MR.predict(X_val_MR)
print('The validation accuracy is given by: %f' % (get_acc(pred_percept,␣
 ↪y_val_MR)))
```

```
The validation accuracy is given by: 94.584615
```

3.2.2 Test Perceptron on Mushroom

[206]:
```
pred_percept = percept_MR.predict(X_test_MR)
print('The testing accuracy is given by: %f' % (get_acc(pred_percept,␣
 ↪y_test_MR)))
```

```
The testing accuracy is given by: 94.892308
```

4 Support Vector Machines (with SGD)

Next, you will implement a "soft margin" SVM. In this formulation you will maximize the margin between positive and negative training examples and penalize margin violations using a hinge loss.

We will optimize the SVM loss using SGD. This means you must compute the loss function with respect to model weights. You will use this gradient to update the model weights.

SVM optimized with SGD has 3 hyperparameters that you can experiment with: - **Learning rate** - similar to as defined above in Perceptron, this parameter scales by how much the weights are changed according to the calculated gradient update. - **Epochs** - similar to as defined above in Perceptron. - **Regularization constant** - Hyperparameter to determine the strength of regularization. In this case it is a coefficient on the term which maximizes the margin. You could try different values. The default value is set to 0.05.

You will implement the SVM using SGD in the **models/svm.py**

The following code: - Creates an instance of the SVM classifier class - The train function of the SVM class is trained on the training data - We use the predict function to find the training accuracy as well as the testing accuracy

4.1 Train SVM on Fashion-MNIST

[179]:
```
lr = 0.0000001
n_epochs = 100
reg_const = 0.125

svm_fashion = SVM(n_class_fashion, lr, n_epochs, reg_const)
svm_fashion.train(X_train_fashion, y_train_fashion)
```

```
41
epoch 1
The training accuracy is: 80.410000
epoch 2
The training accuracy is: 81.756000
epoch 3
The training accuracy is: 82.726000
epoch 4
The training accuracy is: 83.152000
epoch 5
The training accuracy is: 83.626000
epoch 6
The training accuracy is: 83.988000
epoch 7
The training accuracy is: 84.312000
epoch 8
The training accuracy is: 84.676000
epoch 9
The training accuracy is: 84.582000
epoch 10
The training accuracy is: 84.876000
epoch 11
The training accuracy is: 84.698000
epoch 12
```

```
The training accuracy is: 85.112000
epoch 13
The training accuracy is: 85.382000
epoch 14
The training accuracy is: 85.398000
epoch 15
The training accuracy is: 85.522000
epoch 16
The training accuracy is: 85.744000
epoch 17
The training accuracy is: 85.670000
epoch 18
The training accuracy is: 85.674000
epoch 19
The training accuracy is: 85.986000
epoch 20
The training accuracy is: 85.962000
epoch 21
The training accuracy is: 86.012000
epoch 22
The training accuracy is: 86.130000
epoch 23
The training accuracy is: 86.140000
epoch 24
The training accuracy is: 86.076000
epoch 25
The training accuracy is: 86.182000
epoch 26
The training accuracy is: 86.158000
epoch 27
The training accuracy is: 86.228000
epoch 28
The training accuracy is: 86.224000
epoch 29
The training accuracy is: 86.288000
epoch 30
The training accuracy is: 86.356000
epoch 31
The training accuracy is: 86.412000
epoch 32
The training accuracy is: 86.368000
epoch 33
The training accuracy is: 86.406000
epoch 34
The training accuracy is: 86.504000
epoch 35
The training accuracy is: 86.540000
epoch 36
```

```
The training accuracy is: 86.578000
epoch 37
The training accuracy is: 86.592000
epoch 38
The training accuracy is: 86.610000
epoch 39
The training accuracy is: 86.680000
epoch 40
The training accuracy is: 86.688000
epoch 41
The training accuracy is: 86.698000
epoch 42
The training accuracy is: 86.750000
epoch 43
The training accuracy is: 86.784000
epoch 44
The training accuracy is: 86.790000
epoch 45
The training accuracy is: 86.804000
epoch 46
The training accuracy is: 86.868000
epoch 47
The training accuracy is: 86.842000
epoch 48
The training accuracy is: 86.880000
epoch 49
The training accuracy is: 86.908000
epoch 50
The training accuracy is: 86.948000
epoch 51
The training accuracy is: 86.962000
epoch 52
The training accuracy is: 86.984000
epoch 53
The training accuracy is: 87.006000
epoch 54
The training accuracy is: 87.022000
epoch 55
The training accuracy is: 87.062000
epoch 56
The training accuracy is: 87.086000
epoch 57
The training accuracy is: 87.090000
epoch 58
The training accuracy is: 87.102000
epoch 59
The training accuracy is: 87.086000
epoch 60
```

```
The training accuracy is: 87.124000
epoch 61
The training accuracy is: 87.128000
epoch 62
The training accuracy is: 87.142000
epoch 63
The training accuracy is: 87.154000
epoch 64
The training accuracy is: 87.164000
epoch 65
The training accuracy is: 87.142000
epoch 66
The training accuracy is: 87.140000
epoch 67
The training accuracy is: 87.128000
epoch 68
The training accuracy is: 87.136000
epoch 69
The training accuracy is: 87.128000
epoch 70
The training accuracy is: 87.118000
epoch 71
The training accuracy is: 87.106000
epoch 72
The training accuracy is: 87.124000
epoch 73
The training accuracy is: 87.138000
epoch 74
The training accuracy is: 87.136000
epoch 75
The training accuracy is: 87.140000
epoch 76
The training accuracy is: 87.140000
epoch 77
The training accuracy is: 87.138000
epoch 78
The training accuracy is: 87.120000
epoch 79
The training accuracy is: 87.124000
epoch 80
The training accuracy is: 87.144000
epoch 81
The training accuracy is: 87.156000
epoch 82
The training accuracy is: 87.168000
epoch 83
The training accuracy is: 87.160000
epoch 84
```

```
The training accuracy is: 87.158000
epoch 85
The training accuracy is: 87.158000
epoch 86
The training accuracy is: 87.148000
epoch 87
The training accuracy is: 87.166000
epoch 88
The training accuracy is: 87.152000
epoch 89
The training accuracy is: 87.162000
epoch 90
The training accuracy is: 87.174000
epoch 91
The training accuracy is: 87.184000
epoch 92
The training accuracy is: 87.186000
epoch 93
The training accuracy is: 87.202000
epoch 94
The training accuracy is: 87.196000
epoch 95
The training accuracy is: 87.202000
epoch 96
The training accuracy is: 87.200000
epoch 97
The training accuracy is: 87.206000
epoch 98
The training accuracy is: 87.200000
epoch 99
The training accuracy is: 87.210000
epoch 100
The training accuracy is: 87.210000
```

[180]:
```
pred_svm = svm_fashion.predict(X_train_fashion)
print('The training accuracy is given by: %f' % (get_acc(pred_svm,␣
 ↪y_train_fashion)))
```

```
The training accuracy is given by: 87.210000
```

4.1.1 Validate SVM on Fashion-MNIST

[181]:
```
pred_svm = svm_fashion.predict(X_val_fashion)
print('The validation accuracy is given by: %f' % (get_acc(pred_svm,␣
 ↪y_val_fashion)))
```

```
The validation accuracy is given by: 83.640000
```

4.1.2 Test SVM on Fashion-MNIST

[182]:
```
pred_svm = svm_fashion.predict(X_test_fashion)
print('The testing accuracy is given by: %f' % (get_acc(pred_svm,␣
 ↪y_test_fashion)))
```

```
The testing accuracy is given by: 82.750000
```

4.1.3 SVM_Fashion-MNIST Kaggle Submission

Once you are satisfied with your solution and test accuracy output a file to submit your test set predictions to the Kaggle for Assignment 1 Fashion-MNIST. Use the following code to do so:

[158]:
```
output_submission_csv('kaggle/svm_submission_fashion.csv', svm_fashion.
 ↪predict(X_test_fashion))
```

4.2 Train SVM on Mushroom

[209]:
```
lr = 0.01
n_epochs = 100
reg_const = 0.125

svm_MR = SVM(n_class_MR, lr, n_epochs, reg_const)
svm_MR.train(X_train_MR, y_train_MR)
```

```
4
epoch 1
The training accuracy is: 91.075092
epoch 2
The training accuracy is: 93.003693
epoch 3
The training accuracy is: 93.926959
epoch 4
The training accuracy is: 94.973328
epoch 5
The training accuracy is: 95.547805
epoch 6
The training accuracy is: 95.199015
epoch 7
The training accuracy is: 94.501436
epoch 8
The training accuracy is: 94.686089
epoch 9
The training accuracy is: 95.445220
epoch 10
The training accuracy is: 94.829709
epoch 11
The training accuracy is: 95.363151
```

```
epoch 12
The training accuracy is: 94.645055
epoch 13
The training accuracy is: 94.891260
epoch 14
The training accuracy is: 95.424703
epoch 15
The training accuracy is: 94.562987
epoch 16
The training accuracy is: 95.445220
epoch 17
The training accuracy is: 94.645055
epoch 18
The training accuracy is: 95.773492
epoch 19
The training accuracy is: 95.609356
epoch 20
The training accuracy is: 95.116947
epoch 21
The training accuracy is: 95.383668
epoch 22
The training accuracy is: 95.711941
epoch 23
The training accuracy is: 95.629873
epoch 24
The training accuracy is: 95.835043
epoch 25
The training accuracy is: 96.163316
epoch 26
The training accuracy is: 95.711941
epoch 27
The training accuracy is: 96.286418
epoch 28
The training accuracy is: 96.286418
epoch 29
The training accuracy is: 96.204350
epoch 30
The training accuracy is: 96.101764
epoch 31
The training accuracy is: 96.224867
epoch 32
The training accuracy is: 96.101764
epoch 33
The training accuracy is: 96.122281
epoch 34
The training accuracy is: 96.101764
epoch 35
The training accuracy is: 96.553139
```

```
epoch 36
The training accuracy is: 96.471071
epoch 37
The training accuracy is: 96.594173
epoch 38
The training accuracy is: 96.696758
epoch 39
The training accuracy is: 96.491588
epoch 40
The training accuracy is: 96.717275
epoch 41
The training accuracy is: 96.676241
epoch 42
The training accuracy is: 96.717275
epoch 43
The training accuracy is: 96.676241
epoch 44
The training accuracy is: 96.778826
epoch 45
The training accuracy is: 96.758309
epoch 46
The training accuracy is: 96.860895
epoch 47
The training accuracy is: 96.922446
epoch 48
The training accuracy is: 96.881412
epoch 49
The training accuracy is: 96.963480
epoch 50
The training accuracy is: 97.066065
epoch 51
The training accuracy is: 97.148133
epoch 52
The training accuracy is: 97.066065
epoch 53
The training accuracy is: 97.086582
epoch 54
The training accuracy is: 97.086582
epoch 55
The training accuracy is: 97.045548
epoch 56
The training accuracy is: 97.025031
epoch 57
The training accuracy is: 97.107099
epoch 58
The training accuracy is: 97.086582
epoch 59
The training accuracy is: 97.066065
```

```
epoch 60
The training accuracy is: 97.107099
epoch 61
The training accuracy is: 97.086582
epoch 62
The training accuracy is: 97.168650
epoch 63
The training accuracy is: 97.148133
epoch 64
The training accuracy is: 97.168650
epoch 65
The training accuracy is: 97.086582
epoch 66
The training accuracy is: 97.127616
epoch 67
The training accuracy is: 97.066065
epoch 68
The training accuracy is: 97.148133
epoch 69
The training accuracy is: 97.209684
epoch 70
The training accuracy is: 97.168650
epoch 71
The training accuracy is: 97.209684
epoch 72
The training accuracy is: 97.168650
epoch 73
The training accuracy is: 97.189167
epoch 74
The training accuracy is: 97.209684
epoch 75
The training accuracy is: 97.148133
epoch 76
The training accuracy is: 97.250718
epoch 77
The training accuracy is: 97.291752
epoch 78
The training accuracy is: 97.271235
epoch 79
The training accuracy is: 97.291752
epoch 80
The training accuracy is: 97.291752
epoch 81
The training accuracy is: 97.291752
epoch 82
The training accuracy is: 97.332786
epoch 83
The training accuracy is: 97.291752
```

```
epoch 84
The training accuracy is: 97.373820
epoch 85
The training accuracy is: 97.353303
epoch 86
The training accuracy is: 97.373820
epoch 87
The training accuracy is: 97.373820
epoch 88
The training accuracy is: 97.291752
epoch 89
The training accuracy is: 97.291752
epoch 90
The training accuracy is: 97.271235
epoch 91
The training accuracy is: 97.291752
epoch 92
The training accuracy is: 97.271235
epoch 93
The training accuracy is: 97.291752
epoch 94
The training accuracy is: 97.291752
epoch 95
The training accuracy is: 97.312269
epoch 96
The training accuracy is: 97.271235
epoch 97
The training accuracy is: 97.291752
epoch 98
The training accuracy is: 97.312269
epoch 99
The training accuracy is: 97.291752
epoch 100
The training accuracy is: 97.312269
```

[210]:
```
pred_svm = svm_MR.predict(X_train_MR)
print('The training accuracy is given by: %f' % (get_acc(pred_svm, y_train_MR)))
```

```
The training accuracy is given by: 97.312269
```

4.2.1 Validate SVM on Mushroom

[211]:
```
pred_svm = svm_MR.predict(X_val_MR)
print('The validation accuracy is given by: %f' % (get_acc(pred_svm, y_val_MR)))
```

```
The validation accuracy is given by: 97.046154
```

4.3 Test SVM on Mushroom

[212]:
```
pred_svm = svm_MR.predict(X_test_MR)
print('The testing accuracy is given by: %f' % (get_acc(pred_svm, y_test_MR)))
```

```
The testing accuracy is given by: 96.984615
```

5 Softmax Classifier (with SGD)

Next, you will train a Softmax classifier. This classifier consists of a linear function of the input data followed by a softmax function which outputs a vector of dimension C (number of classes) for each data point. Each entry of the softmax output vector corresponds to a confidence in one of the C classes, and like a probability distribution, the entries of the output vector sum to 1. We use a cross-entropy loss on this sotmax output to train the model.

Check the following link as an additional resource on softmax classification: http://cs231n.github.io/linear-classify/#softmax

Once again we will train the classifier with SGD. This means you need to compute the gradients of the softmax cross-entropy loss function according to the weights and update the weights using this gradient. Check the following link to help with implementing the gradient updates: https://deepnotes.io/softmax-crossentropy

The softmax classifier has 3 hyperparameters that you can experiment with: - **Learning rate** - As above, this controls how much the model weights are updated with respect to their gradient. - **Number of Epochs** - As described for perceptron. - **Regularization constant** - Hyperparameter to determine the strength of regularization. In this case, we minimize the L2 norm of the model weights as regularization, so the regularization constant is a coefficient on the L2 norm in the combined cross-entropy and regularization objective.

You will implement a softmax classifier using SGD in the **models/softmax.py**

The following code: - Creates an instance of the Softmax classifier class - The train function of the Softmax class is trained on the training data - We use the predict function to find the training accuracy as well as the testing accuracy

5.1 Train Softmax on Fashion-MNIST

[228]:
```
lr = 0.005
n_epochs = 60
reg_const = 0.1

softmax_fashion = Softmax(n_class_fashion, lr, n_epochs, reg_const)
softmax_fashion.train(X_train_fashion, y_train_fashion)
```

```
41
epoch: 1
The training accuracy is: 73.352000
epoch: 2
The training accuracy is: 78.686000
```

```
epoch: 3
The training accuracy is: 73.216000
epoch: 4
The training accuracy is: 76.814000
epoch: 5
The training accuracy is: 79.594000
epoch: 6
The training accuracy is: 83.442000
epoch: 7
The training accuracy is: 79.984000
epoch: 8
The training accuracy is: 80.340000
epoch: 9
The training accuracy is: 82.724000
epoch: 10
The training accuracy is: 84.210000
epoch: 11
The training accuracy is: 84.672000
epoch: 12
The training accuracy is: 84.356000
epoch: 13
The training accuracy is: 83.852000
epoch: 14
The training accuracy is: 84.908000
epoch: 15
The training accuracy is: 84.822000
epoch: 16
The training accuracy is: 85.008000
epoch: 17
The training accuracy is: 85.236000
epoch: 18
The training accuracy is: 85.270000
epoch: 19
The training accuracy is: 85.304000
epoch: 20
The training accuracy is: 85.320000
epoch: 21
The training accuracy is: 85.368000
epoch: 22
The training accuracy is: 85.398000
epoch: 23
The training accuracy is: 85.432000
epoch: 24
The training accuracy is: 85.472000
epoch: 25
The training accuracy is: 85.450000
epoch: 26
The training accuracy is: 85.436000
```

```
epoch: 27
The training accuracy is: 85.430000
epoch: 28
The training accuracy is: 85.410000
epoch: 29
The training accuracy is: 85.408000
epoch: 30
The training accuracy is: 85.442000
epoch: 31
The training accuracy is: 85.444000
epoch: 32
The training accuracy is: 85.456000
epoch: 33
The training accuracy is: 85.474000
epoch: 34
The training accuracy is: 85.480000
epoch: 35
The training accuracy is: 85.478000
epoch: 36
The training accuracy is: 85.484000
epoch: 37
The training accuracy is: 85.486000
epoch: 38
The training accuracy is: 85.456000
epoch: 39
The training accuracy is: 85.472000
epoch: 40
The training accuracy is: 85.474000
epoch: 41
The training accuracy is: 85.474000
epoch: 42
The training accuracy is: 85.460000
epoch: 43
The training accuracy is: 85.464000
epoch: 44
The training accuracy is: 85.458000
epoch: 45
The training accuracy is: 85.452000
epoch: 46
The training accuracy is: 85.444000
epoch: 47
The training accuracy is: 85.444000
epoch: 48
The training accuracy is: 85.446000
epoch: 49
The training accuracy is: 85.456000
epoch: 50
The training accuracy is: 85.454000
```

```
epoch: 51
The training accuracy is: 85.456000
epoch: 52
The training accuracy is: 85.456000
epoch: 53
The training accuracy is: 85.456000
epoch: 54
The training accuracy is: 85.452000
epoch: 55
The training accuracy is: 85.454000
epoch: 56
The training accuracy is: 85.450000
epoch: 57
The training accuracy is: 85.448000
epoch: 58
The training accuracy is: 85.448000
epoch: 59
The training accuracy is: 85.446000
epoch: 60
The training accuracy is: 85.448000
```

[229]:
```
pred_softmax = softmax_fashion.predict(X_train_fashion)
print('The training accuracy is given by: %f' % (get_acc(pred_softmax,␣
 ↪y_train_fashion)))
```

```
The training accuracy is given by: 85.448000
```

5.1.1 Validate Softmax on Fashion-MNIST

[230]:
```
pred_softmax = softmax_fashion.predict(X_val_fashion)
print('The validation accuracy is given by: %f' % (get_acc(pred_softmax,␣
 ↪y_val_fashion)))
```

```
The validation accuracy is given by: 83.150000
```

5.1.2 Testing Softmax on Fashion-MNIST

[231]:
```
pred_softmax = softmax_fashion.predict(X_test_fashion)
print('The testing accuracy is given by: %f' % (get_acc(pred_softmax,␣
 ↪y_test_fashion)))
```

```
The testing accuracy is given by: 81.860000
```

5.1.3 Softmax_Fashion-MNIST Kaggle Submission

Once you are satisfied with your solution and test accuracy output a file to submit your test set predictions to the Kaggle for Assignment 1 Fashion-MNIST. Use the following code to do so:

[232]:
```
output_submission_csv('kaggle/softmax_submission_fashion.csv', softmax_fashion.
↪predict(X_test_fashion))
```

5.2 Train Softmax on Mushroom

[243]:
```
lr = 0.25
n_epochs = 50
reg_const = 0.1

softmax_MR = Softmax(n_class_MR, lr, n_epochs, reg_const)
softmax_MR.train(X_train_MR, y_train_MR)
```

```
4
epoch: 1
The training accuracy is: 48.297087
epoch: 2
The training accuracy is: 54.329093
epoch: 3
The training accuracy is: 69.737382
epoch: 4
The training accuracy is: 75.030776
epoch: 5
The training accuracy is: 77.164547
epoch: 6
The training accuracy is: 78.764875
epoch: 7
The training accuracy is: 81.144850
epoch: 8
The training accuracy is: 83.401723
epoch: 9
The training accuracy is: 85.514977
epoch: 10
The training accuracy is: 86.356176
epoch: 11
The training accuracy is: 86.704965
epoch: 12
The training accuracy is: 87.012721
epoch: 13
The training accuracy is: 87.176857
epoch: 14
The training accuracy is: 87.320476
epoch: 15
The training accuracy is: 87.402544
epoch: 16
The training accuracy is: 87.464095
epoch: 17
The training accuracy is: 87.484612
```

```
epoch: 18
The training accuracy is: 87.525646
epoch: 19
The training accuracy is: 87.587197
epoch: 20
The training accuracy is: 87.628231
epoch: 21
The training accuracy is: 87.669265
epoch: 22
The training accuracy is: 87.628231
epoch: 23
The training accuracy is: 87.689783
epoch: 24
The training accuracy is: 87.689783
epoch: 25
The training accuracy is: 87.710300
epoch: 26
The training accuracy is: 87.730817
epoch: 27
The training accuracy is: 87.751334
epoch: 28
The training accuracy is: 87.771851
epoch: 29
The training accuracy is: 87.792368
epoch: 30
The training accuracy is: 87.792368
epoch: 31
The training accuracy is: 87.792368
epoch: 32
The training accuracy is: 87.792368
epoch: 33
The training accuracy is: 87.792368
epoch: 34
The training accuracy is: 87.771851
epoch: 35
The training accuracy is: 87.771851
epoch: 36
The training accuracy is: 87.771851
epoch: 37
The training accuracy is: 87.771851
epoch: 38
The training accuracy is: 87.771851
epoch: 39
The training accuracy is: 87.771851
epoch: 40
The training accuracy is: 87.771851
epoch: 41
The training accuracy is: 87.771851
```

```
epoch: 42
The training accuracy is: 87.771851
epoch: 43
The training accuracy is: 87.771851
epoch: 44
The training accuracy is: 87.771851
epoch: 45
The training accuracy is: 87.771851
epoch: 46
The training accuracy is: 87.771851
epoch: 47
The training accuracy is: 87.771851
epoch: 48
The training accuracy is: 87.771851
epoch: 49
The training accuracy is: 87.771851
epoch: 50
The training accuracy is: 87.771851
```

[244]:
```
pred_softmax = softmax_MR.predict(X_train_MR)
print('The training accuracy is given by: %f' % (get_acc(pred_softmax,␣
 ↪y_train_MR)))
```

```
The training accuracy is given by: 87.771851
```

5.2.1 Validate Softmax on Mushroom

[245]:
```
pred_softmax = softmax_MR.predict(X_val_MR)
print('The validation accuracy is given by: %f' % (get_acc(pred_softmax,␣
 ↪y_val_MR)))
```

```
The validation accuracy is given by: 87.876923
```

5.2.2 Testing Softmax on Mushroom

[246]:
```
pred_softmax = softmax_MR.predict(X_test_MR)
print('The testing accuracy is given by: %f' % (get_acc(pred_softmax,␣
 ↪y_test_MR)))
```

```
The testing accuracy is given by: 85.907692
```

6 Logistic Classifier

The Logistic Classifier has 2 hyperparameters that you can experiment with: - **Learning rate** - similar to as defined above in Perceptron, this parameter scales by how much the weights are changed according to the calculated gradient update. - **Number of Epochs** - As described for perceptron. - **Threshold** - The decision boundary of the classifier.

You will implement the Logistic Classifier in the **models/logistic.py**

The following code: - Creates an instance of the Logistic classifier class - The train function of the Logistic class is trained on the training data - We use the predict function to find the training accuracy as well as the testing accuracy

6.0.1 Training Logistic Classifer

[252]:
```
learning_rate = 0.25
n_epochs = 100
threshold = 0.5

lr = Logistic(learning_rate, n_epochs, threshold)
lr.train(X_train_MR, y_train_MR)
```

```
epoch: 1
The training accuracy is: 76.159212
epoch: 2
The training accuracy is: 90.459581
epoch: 3
The training accuracy is: 78.662290
epoch: 4
The training accuracy is: 81.575708
epoch: 5
The training accuracy is: 86.745999
epoch: 6
The training accuracy is: 83.278621
epoch: 7
The training accuracy is: 87.833402
epoch: 8
The training accuracy is: 85.986869
epoch: 9
The training accuracy is: 84.735330
epoch: 10
The training accuracy is: 84.878950
epoch: 11
The training accuracy is: 84.160854
epoch: 12
The training accuracy is: 87.320476
epoch: 13
The training accuracy is: 88.243742
epoch: 14
The training accuracy is: 89.762002
epoch: 15
The training accuracy is: 91.957325
epoch: 16
The training accuracy is: 90.274928
epoch: 17
```

```
The training accuracy is: 85.145671
epoch: 18
The training accuracy is: 85.535494
epoch: 19
The training accuracy is: 85.679114
epoch: 20
The training accuracy is: 86.192039
epoch: 21
The training accuracy is: 87.197374
epoch: 22
The training accuracy is: 88.715634
epoch: 23
The training accuracy is: 90.398030
epoch: 24
The training accuracy is: 91.895773
epoch: 25
The training accuracy is: 93.126795
epoch: 26
The training accuracy is: 94.009027
epoch: 27
The training accuracy is: 94.788675
epoch: 28
The training accuracy is: 95.281083
epoch: 29
The training accuracy is: 95.629873
epoch: 30
The training accuracy is: 95.896594
epoch: 31
The training accuracy is: 96.142799
epoch: 32
The training accuracy is: 96.183833
epoch: 33
The training accuracy is: 96.327452
epoch: 34
The training accuracy is: 96.327452
epoch: 35
The training accuracy is: 96.368486
epoch: 36
The training accuracy is: 96.306935
epoch: 37
The training accuracy is: 96.327452
epoch: 38
The training accuracy is: 96.327452
epoch: 39
The training accuracy is: 96.347969
epoch: 40
The training accuracy is: 96.306935
epoch: 41
```

```
The training accuracy is: 96.306935
epoch: 42
The training accuracy is: 96.265901
epoch: 43
The training accuracy is: 96.245384
epoch: 44
The training accuracy is: 96.245384
epoch: 45
The training accuracy is: 96.245384
epoch: 46
The training accuracy is: 96.183833
epoch: 47
The training accuracy is: 96.163316
epoch: 48
The training accuracy is: 96.142799
epoch: 49
The training accuracy is: 96.142799
epoch: 50
The training accuracy is: 96.163316
epoch: 51
The training accuracy is: 96.163316
epoch: 52
The training accuracy is: 96.101764
epoch: 53
The training accuracy is: 96.040213
epoch: 54
The training accuracy is: 96.081247
epoch: 55
The training accuracy is: 96.081247
epoch: 56
The training accuracy is: 96.081247
epoch: 57
The training accuracy is: 96.142799
epoch: 58
The training accuracy is: 96.101764
epoch: 59
The training accuracy is: 96.081247
epoch: 60
The training accuracy is: 96.101764
epoch: 61
The training accuracy is: 96.142799
epoch: 62
The training accuracy is: 96.163316
epoch: 63
The training accuracy is: 96.163316
epoch: 64
The training accuracy is: 96.142799
epoch: 65
```

```
The training accuracy is: 96.142799
epoch: 66
The training accuracy is: 96.163316
epoch: 67
The training accuracy is: 96.163316
epoch: 68
The training accuracy is: 96.163316
epoch: 69
The training accuracy is: 96.163316
epoch: 70
The training accuracy is: 96.183833
epoch: 71
The training accuracy is: 96.163316
epoch: 72
The training accuracy is: 96.163316
epoch: 73
The training accuracy is: 96.163316
epoch: 74
The training accuracy is: 96.163316
epoch: 75
The training accuracy is: 96.163316
epoch: 76
The training accuracy is: 96.163316
epoch: 77
The training accuracy is: 96.163316
epoch: 78
The training accuracy is: 96.142799
epoch: 79
The training accuracy is: 96.163316
epoch: 80
The training accuracy is: 96.163316
epoch: 81
The training accuracy is: 96.163316
epoch: 82
The training accuracy is: 96.163316
epoch: 83
The training accuracy is: 96.142799
epoch: 84
The training accuracy is: 96.142799
epoch: 85
The training accuracy is: 96.142799
epoch: 86
The training accuracy is: 96.142799
epoch: 87
The training accuracy is: 96.142799
epoch: 88
The training accuracy is: 96.142799
epoch: 89
```

```
The training accuracy is: 96.142799
epoch: 90
The training accuracy is: 96.163316
epoch: 91
The training accuracy is: 96.163316
epoch: 92
The training accuracy is: 96.163316
epoch: 93
The training accuracy is: 96.163316
epoch: 94
The training accuracy is: 96.163316
epoch: 95
The training accuracy is: 96.163316
epoch: 96
The training accuracy is: 96.163316
epoch: 97
The training accuracy is: 96.163316
epoch: 98
The training accuracy is: 96.163316
epoch: 99
The training accuracy is: 96.163316
epoch: 100
The training accuracy is: 96.163316
```

[253]:
```
pred_lr = lr.predict(X_train_MR)
print('The training accuracy is given by: %f' % (get_acc(pred_lr, y_train_MR)))
```

```
The training accuracy is given by: 96.163316
```

6.0.2 Validate Logistic Classifer

[254]:
```
pred_lr = lr.predict(X_val_MR)
print('The validation accuracy is given by: %f' % (get_acc(pred_lr, y_val_MR)))
```

```
The validation accuracy is given by: 95.692308
```

6.0.3 Test Logistic Classifier

[255]:
```
pred_lr = lr.predict(X_test_MR)
print('The testing accuracy is given by: %f' % (get_acc(pred_lr, y_test_MR)))
```

```
The testing accuracy is given by: 95.446154
```