University of Southern Denmark
IMADA

DM566: Data Mining and Machine Learning
Spring term 2022

Exercise 13

Exercise 13-1 Handwritten digit recognition in PyTorch with neural nets
In this exercise session we will be discussing neural networks and develop a handwritten digit classifier from scratch. For this, we will be using PyTorch with nn.Linear, nn.ReLU, (optional) nn.Sequential.

We will be using the popular MNIST database. It is a collection of 70000 handwritten digits split into training and test set of 60000 and 10000 images respectively.

You will need to build a neural net (multilayer perceptron) with

- 0 hidden layers (linear regression)
- 1 hidden layer of for example 100 neurons
- 2 hidden layers of 100 hidden neurons

Get it to run on the MNIST data set. Tutorials and other helpful resources can be found on the PyTorch website.

Observe how batch size and learning rate should be changed together, and how adjusting the parameters affects the accuracy. Observe how training set size affects accuracy. Observe how different levels of depth affect accuracy (0-1-2- hidden layers).

Suggested solution:
Take a look at the three files linearRegression.py, oneHiddenLayer.py and twoHiddenLayers.py to see the three neural networks.

For the linear regression you should aim for an accuracy of about 91%.
For the deeper neural nets you should achieve circa 95% accuracy.

We set the batch size to 100 and the learning rate to 0.01.

For the linear regression we get the following output:

```
Epoch 0 - Training loss: 1.1461583495140075
Epoch 1 - Training loss: 0.6435175343851248
Epoch 2 - Training loss: 0.5372771553198497
Epoch 3 - Training loss: 0.4859272279838721
Epoch 4 - Training loss: 0.4543911881248156
Epoch 5 - Training loss: 0.4324587963024775
Epoch 6 - Training loss: 0.4162375754366318
Epoch 7 - Training loss: 0.40350041173398493
Epoch 8 - Training loss: 0.393119697868824
Epoch 9 - Training loss: 0.38453388152023155
Epoch 10 - Training loss: 0.377195683841904
Epoch 11 - Training loss: 0.3708817851791779
Epoch 12 - Training loss: 0.3653448565552632
Epoch 13 - Training loss: 0.36045041332642236
Epoch 14 - Training loss: 0.3560842328270276
Number Of Images Tested = 10000

Model Accuracy = 0.909
```

For the one hidden layer we get the following output:

```
Epoch 0 - Training loss: 1.531797177096208
Epoch 1 - Training loss: 0.6390927668909232
Epoch 2 - Training loss: 0.46941557213664054
Epoch 3 - Training loss: 0.40717643169065315
Epoch 4 - Training loss: 0.37363902737696963
Epoch 5 - Training loss: 0.3517376111944516
Epoch 6 - Training loss: 0.33549681268632414
Epoch 7 - Training loss: 0.32254696791370707
Epoch 8 - Training loss: 0.311447286978364
Epoch 9 - Training loss: 0.30184907679756484
Epoch 10 - Training loss: 0.2931606252739827
Epoch 11 - Training loss: 0.285176442625622
Epoch 12 - Training loss: 0.2776243710021178
Epoch 13 - Training loss: 0.2706863403817018
Epoch 14 - Training loss: 0.26408729936927555
Number Of Images Tested = 10000

Model Accuracy = 0.9279
```

For the two hidden layers we get the following output:

```
Epoch 0 - Training loss: 2.0492301162083946
Epoch 1 - Training loss: 0.8995324766635895
Epoch 2 - Training loss: 0.5114810182154179
Epoch 3 - Training loss: 0.4113603740930557
Epoch 4 - Training loss: 0.3663455333809058
Epoch 5 - Training loss: 0.33947630842526755
Epoch 6 - Training loss: 0.3198274608204762
Epoch 7 - Training loss: 0.30403517658511797
Epoch 8 - Training loss: 0.2903701156750321
Epoch 9 - Training loss: 0.2783450056364139
Epoch 10 - Training loss: 0.26792085708429414
Epoch 11 - Training loss: 0.25805873356759546
Epoch 12 - Training loss: 0.24862257429709037
Epoch 13 - Training loss: 0.2401365895072619
Epoch 14 - Training loss: 0.23187484511484702
Number Of Images Tested = 10000

Model Accuracy = 0.9367
```