Deep Learning first assignment

Michel Valentini
Elma Nevala
Dovile Umbrasaite

Group 27

April 2023

1 Description Practice

1.1 MNIST

Take the Fashion MNIST data set and a suitable model architecture. Display the effect of the choice of the activation function, i.e., investigate different learning and generalization behavior if sgd or a modern activation function (e.g. relu, selu, ...) is used. Compare the behavior of at least three different acivation functions.

First we import the needed Pytorch and Torchvision libraries. Then we define the initialization and forward part of the class FashionMNISTNet.

Then we choose our activation function. For this practise we use 3 different activation functions:

- Relu : ReLu is a piecewise linear function that given the input value, if its positive and zero if its negaive.
- Selu : SeLu is a scaled exponential linear unit activation function. When x is smaller than zero, we take the exponential of the x-value minus 1, then we multiply it with alpha and lambda.
- Sigmoid : This function takes any real value as input and outputs values in the range of 0 to 1 The larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to 0.0.

After choosing the activation function we can proceed with loading the dataset. First we create a variable "transform" that is composed by toTensor and Normalize functions. We use the function ToTensor to transform an image in a torch.FloatTensor, and scale the value. Normalize function is used to Normalize a tensor image given mean and standard deviation.
Then we load the trainSet and TestSet from the dataset. Here the important attribute to give are, the Boolean "train" to have the network balance with the weights, and the number of "batch_size" that we'll use in the next part.

Other two things to define before starting the training are the Loss Function and Optimizer. For the loss function the standard "CrossEntropyLoss" is used but for the Optimizer we use SGD (stochastic gradient descent) like in the tutorial example.

Finally is possible to start with the training and testing part.

- Training : is use to train our network given the inputs data and labels and obtain an result that given at the loss function we can reduce the distance between our results to ground truth. After that, using the loss.backward is possible to adjust the weight for each node of out network in such a way to reduce the errors during the test part.

- Testing : here we can valuate our network simply given the test data in input and waiting the prediction in output. Also we sum all the labels and the right prediction.

Finally, we divide the correct labels by the total labels, multiply for 100 and obtain the precision of our model.

1.2 Results

Here we show some results of our model, divide by different Activation Functions and all with SGD optimizer and batch size of 64:

- RELU : Epoch [1/10], Loss: 0.6396
  Epoch [2/10], Loss: 0.3990
  Epoch [3/10], Loss: 0.3565
  Epoch [4/10], Loss: 0.3295
  Epoch [5/10], Loss: 0.3106
  Epoch [6/10], Loss: 0.2943
  Epoch [7/10], Loss: 0.2802
  Epoch [8/10], Loss: 0.2653
  Epoch [9/10], Loss: 0.2565
  Epoch [10/10], Loss: 0.2448
  Accuracy of the network test images: 87.54 %
- SELU : Epoch [1/10], Loss: 0.5469
  Epoch [2/10], Loss: 0.4080
  Epoch [3/10], Loss: 0.3753
  Epoch [4/10], Loss: 0.3503
  Epoch [5/10], Loss: 0.3337
  Epoch [6/10], Loss: 0.3154
  Epoch [7/10], Loss: 0.3016
  Epoch [8/10], Loss: 0.2930
  Epoch [9/10], Loss: 0.2792
  Epoch [10/10], Loss: 0.2740
  Accuracy of the network test images: 87.81 %
- SIGMOID : Epoch [1/10], Loss: 2.2412
  Epoch [2/10], Loss: 1.1376
  Epoch [3/10], Loss: 0.8087
  Epoch [4/10], Loss: 0.6981
  Epoch [5/10], Loss: 0.6191
  Epoch [6/10], Loss: 0.5622
  Epoch [7/10], Loss: 0.5170
  Epoch [8/10], Loss: 0.4824
  Epoch [9/10], Loss: 0.4543
  Epoch [10/10], Loss: 0.4294
  Accuracy of the network test images: 83.91 %

How we can see, the Selu and Relu results are close each other.

1.3 ADAM

Take the same setup and the ADAM optimizer, display the result of the choice of different batch sizes for mini-batch training. Take at least three ch
ces and shortly discuss the differences.

How describe we change the optimizer function from SGD to ADAM with a learning rate equal at 0.01. Then we modify the batch size from 64 to 128 to 256 obtain these results:

1.4 Activation Function Relu

- Optimizer ADAM batch size 64 : Epoch [1/10], Loss: 0.5731
  Epoch [2/10], Loss: 0.4622
  Epoch [3/10], Loss: 0.4346
  Epoch [4/10], Loss: 0.4102
  Epoch [5/10], Loss: 0.4062
  Epoch [6/10], Loss: 0.3974
  Epoch [7/10], Loss: 0.4039
  Epoch [8/10], Loss: 0.3859
  Epoch [9/10], Loss: 0.3818
  Epoch [10/10], Loss: 0.3692
  Accuracy of the network test images: 85.32 %

- Optimizer ADAM batch size 128 : Epoch [1/10], Loss: 0.5525
  Epoch [2/10], Loss: 0.4242
  Epoch [3/10], Loss: 0.3945
  Epoch [4/10], Loss: 0.3818
  Epoch [5/10], Loss: 0.3679
  Epoch [6/10], Loss: 0.3643
  Epoch [7/10], Loss: 0.3516
  Epoch [8/10], Loss: 0.3394
  Epoch [9/10], Loss: 0.3349
  Epoch [10/10], Loss: 0.3311
  Accuracy of the network test images: 86.21 %

- Optimizer ADAM batch size 256 : Epoch [1/10], Loss: 0.5786
  Epoch [2/10], Loss: 0.4076
  Epoch [3/10], Loss: 0.3718
  Epoch [4/10], Loss: 0.3463
  Epoch [5/10], Loss: 0.3373
  Epoch [6/10], Loss: 0.3178
  Epoch [7/10], Loss: 0.3163
  Epoch [8/10], Loss: 0.3059
  Epoch [9/10], Loss: 0.3016

Epoch [10/10], Loss: 0.2883
Accuracy of the network test images: 86.71 %

1.5 Activation Function Selu

- Optimizer ADAM batch size 64 : Epoch [1/10], Loss: 0.6348
  Epoch [2/10], Loss: 0.5304
  Epoch [3/10], Loss: 0.5354
  Epoch [4/10], Loss: 0.5789
  Epoch [5/10], Loss: 0.4528
  Epoch [6/10], Loss: 0.4482
  Epoch [7/10], Loss: 0.4328
  Epoch [8/10], Loss: 0.4313
  Epoch [9/10], Loss: 0.4231
  Epoch [10/10], Loss: 0.4154
  Accuracy of the network test images: 84.11 %

- Optimizer ADAM batch size 128 : Epoch [1/10], Loss: 0.5965
  Epoch [2/10], Loss: 0.4513
  Epoch [3/10], Loss: 0.4273
  Epoch [4/10], Loss: 0.4297
  Epoch [5/10], Loss: 0.3914
  Epoch [6/10], Loss: 0.3811
  Epoch [7/10], Loss: 0.3702
  Epoch [8/10], Loss: 0.3664
  Epoch [9/10], Loss: 0.4904
  Epoch [10/10], Loss: 1.1954
  Accuracy of the network test images: 74.78 %

- Optimizer ADAM batch size 256 : Epoch [1/10], Loss: 0.6074
  Epoch [2/10], Loss: 0.4192
  Epoch [3/10], Loss: 0.3738
  Epoch [4/10], Loss: 0.3638
  Epoch [5/10], Loss: 0.3525
  Epoch [6/10], Loss: 0.3388
  Epoch [7/10], Loss: 0.3227
  Epoch [8/10], Loss: 0.3214
  Epoch [9/10], Loss: 0.3180
  Epoch [10/10], Loss: 0.3023
  Accuracy of the network test images: 87.09 %

1.6 Activation Function SIGMOID

- Optimizer ADAM batch size 64 : Epoch [1/10], Loss: 0.7169
  Epoch [2/10], Loss: 0.5585
  Epoch [3/10], Loss: 0.5319
  Epoch [4/10], Loss: 0.5026

Epoch [5/10], Loss: 0.5252
Epoch [6/10], Loss: 0.5084
Epoch [7/10], Loss: 0.5064
Epoch [8/10], Loss: 0.4941
Epoch [9/10], Loss: 0.4997
Epoch [10/10], Loss: 0.4911
Accuracy of the network test images: 80.64 %

- Optimizer ADAM batch size 128 : Epoch [1/10], Loss: 0.7339
  Epoch [2/10], Loss: 0.5239
  Epoch [3/10], Loss: 0.4832
  Epoch [4/10], Loss: 0.4707
  Epoch [5/10], Loss: 0.4500
  Epoch [6/10], Loss: 0.4389
  Epoch [7/10], Loss: 0.4276
  Epoch [8/10], Loss: 0.4249
  Epoch [9/10], Loss: 0.4163
  Epoch [10/10], Loss: 0.4173
  Accuracy of the network test images: 83.80 %

- Optimizer ADAM batch size 256 : Epoch [1/10], Loss: 0.7785
  Epoch [2/10], Loss: 0.5014
  Epoch [3/10], Loss: 0.4509
  Epoch [4/10], Loss: 0.4334
  Epoch [5/10], Loss: 0.4122
  Epoch [6/10], Loss: 0.3987
  Epoch [7/10], Loss: 0.3920
  Epoch [8/10], Loss: 0.3910
  Epoch [9/10], Loss: 0.3763
  Epoch [10/10], Loss: 0.3676
  Accuracy of the network test images: 85.06 %