8. Fully-Connected Neural Networks

Project due Apr 5, 2023 08:59 -03 Completed

First, we will employ the most basic form of a deep neural network, in which the neuron fully connected to one another.

You will be working in the files`part2-mnist/nnet_fc.py`**in this problem**

Training and Testing Accuracy Over Time

1.0/1.0 point (graded)

We have provided a toy example **nnet_fc.py** in which we have implemented for you a si This network has one hidden layer of 10 neurons with a rectified linear unit (ReLU) nonl output layer of 10 neurons (one for each digit class). Finally, a softmax function normali the output neurons so that they specify a probability distribution. Reference the PyTorc read through it in order to gain a better understanding of the code. Then, try running th computer with the command `python3 nnet_fc.py`. This will train the network with 10 epoch is a complete pass through the training dataset. Total training time of your netwo than a couple of minutes. At the end of training, your model should have an accuracy o data.

Note: We are not using a softmax layer because it is already present in the loss: PyTorc `nn.CrossEntropyLoss` combines `nn.LogSoftMax` with `nn.NLLLoss`.

Report the test accuracy below.

Test Accuracy = 0.9204727564102564 ✔

Submit You have used 2 of 3 attempts

Improving Accuracy

5.0/5.0 points (graded)

We would like to try to improve the performance of the model by performing a mini grid parameters (note that a full grid search should include more values and combinations). our **baseline model (batch size 32, hidden size 10, learning rate 0.1, momentum 0 and function)** and modify one parameter each time while keeping all others to the baseline validation accuracy of the model after training for 10 epochs. For the LeakyReLU activa default parameters from pyTorch (negative_slope=0.01).

Note: If you run the model multiple times from the same script, make sure to initialize th **random seeds to 12321 before each run**.

Validation Accuracy = 0.939852 ✔

Does the model variation that achieved the highest validation accuracy achieved also t
accuracy?

- (●) Yes
- ( ) No

✔

Submit You have used 1 of 3 attempts

Improving Accuracy - Hidden 128

3.0/3.0 points (graded)

Modifying the model's architecture is also worth considering. Increase the hidden repre
to 128 and repeat the grid search over the hyper parameters. This time, what modificat
validation accuracy?

- ( ) baseline (no modifications)
- ( ) batch size 64
- ( ) learning rate 0.01
- ( ) momentum 0.9
- (●) LeakyReLU activation

✔

Submit You have used 1 of 3 attempts

‹ Previous Next ›

- where in the code is the heavy lifting?
  It took me a while to sort this out, and I may be wrong. Maybe my thoughts will help others figure this out, ar
- Validation accuracy - is it the maximal or the last value?
  Improving accuracy - there are 10 epochs in each run. For some parameters the accuracy changes for better
- where is toy example nnet_fc.py
  I am not getting where is this toy example nnet_fc.py. means they have written that we have provided you th
- test accuracy not accepted?..
  How come my test accuracy of nnet_fc.py is not accepted?.. Anyone has the same problem?
- don't lose those random seeds!
  To better understand the code in , I was running through the function a few lines at a time i
- how to fix this problem?
  ModuleNotFoundError: No module named 'tqdm'
- Hidden Representation Size
  Where is the hidden representation size defined?
- why does batchsize have an effect if no. epochs is the same?