DAT405 Introduction to Data Science and AI, 2020,

Assignment 7:

Download the notebook for this assignment. The notebook will provide a basis that you can use to solve the exercises.

In this assignment we will work with the MNIST data set and use the Keras framework to construct our neural networks. You can read more about the Keras framework here https://keras.io/ . You can find information regarding the different layers and regularizers there. If it takes too long time to train a neural network on your computer we recommend that you visit Google Colab where you can use a notebook with a GPU for free, a short guide to using Google Colab is given at the end of this assignment.

If there are any questions regarding the assignment or the Keras framework, contact the Tas.

There will be 10 points for this assignment.

1)

- A) How many layers does the network in the notebook have? How many neurons does each layer have? What are the total number of parameters for the network?
- B) Train the network for 10 epochs and plot the training and validation accuracy for each epoch.

2)

- A) On the same data as before, plot the training and validation accuracy for each epoch for a neural network with one hidden layer with 100 neurons. Train the network for 30 epochs with a learning rate of 0.1.
- B) Change the number of epochs to 10. Try different learning rates, ranging from 0.001 to 1. Repeat this 3 times for each learning rate. From your results, which learning rate is optimal?
- C) Fix the number of epochs to 10. Create a chart over the performance of your network when varying the number of neurons (between 10-1000) in your single hidden layer and the learning rate (between 0.001 to 0.1). For the optimal model, plot the validation and training accuracy for each epoch when the network is trained for 30 epochs.

3)

Use a neural network with a single hidden layer with 100 neurons. Train for 30 epochs with a learning rate of 0.1.

- A) Add Gaussian noise to each batch during training. Try the standard deviations 0.1, 1 and 10 for the noise layer. Report the final prediction scores. Can you come up with an argument for why adding noise like this could be a good idea in certain situations?

  B) Implement l_2 norm regularization: We want to change the risk minimization framework to a regularized one:

$$\min \frac{1}{2N} \sum_{i=1}^{N} \left(y_i - f_{W,b}(x_i) \right)^2 + \alpha \|W\|_2^2$$

Where N=50000 is the number of training data points, W, b are respectively the vectors of all weights and biases in the network, $\|W\|_2^2$ is the squared l_2 norm of the weights (sum of squares of weights) and α is a regularization parameter.

Does adding l_2 regularization to your layer improve the final prediction score? Try regularization parameters between 0.001-0.1.

4)

A) Use at least one convolutional layer and try and create a network that can reach 99% accuracy on the validation data. If you choose to use any layers except convolutional layers and layers that you used in previous exercises, you must describe what they do. If you do not reach 99% accuracy, report your best performance and explain your attempts and thought process.

B) What is a benefit of using convolutional layers over fully connected ones?

What to submit

- All Python code written.
- A report that includes the figures produced and the descriptions/discussions that are requested in the questions.

If you upload a zip file, please also upload any PDF files separately (so that they can be viewed more conveniently in Canvas).

Deadline: Monday, October 19 at 23:59

Self-check

Is all the required information on the front page? Have you answered all questions to the best of your ability? Anything else you can easily check? (details, terminology, arguments, clearly stated answers etc.?)

Grading will be based on a qualitative assessment of each assignment. It is important to:

- Present clear arguments
- Present the results in a pedagogical way
    - Should it be table/plot? What kind of plot? Is everything clear and easy to understand?
- Show understanding of the topics
- Give correct solutions.
- Make sure that the code is well commented.
    - Important parts of the code should be included in the running text and the full code uploaded to Canvas.

Google Colab Instructions: (Requires signing into google)

- Visit https://colab.research.google.com/notebooks/intro.ipynb
- Go to “File” and either upload the notebook or create a new one.
- After the notebook has been created, go to “Edit”, then press “Notebook Settings”.
- Under “Hardware accelerator”, choose “GPU”.
- You can now run your notebook with access to a GPU.