

Introduction to Data Science and AI, LP1 2021,

Assignment 7 Neural Networks with Keras and TensorFlow:

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

This assignment will work with the MNIST data set. The MNIST dataset is a standard benchmark where small 28x28 pixel grayscale images of handwritten digits. Each image was manually assigned to a class label, an integer from 0 to 9, by the US Census Bureau. The task associated with the dataset is building **a model that takes a new image (of the same size) and returns a class label – that is, an integer from 0 to 9.**

For this assignment, we will 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. In this assignment, you can use free GPU leases on Google Colab or deepnote to speed up training.

If there are any questions regarding the assignment or the Keras/TensorFlow framework, please contact the TAs. Use the lab-sessions to ask questions!

The grade for this assignment is pass/fail.

1) *Preprocessing.*

In the notebook, the data is downloaded from an external server imported into the notebook environment using the `mnist.load_data()` function call.

Explain the data pre-processing high-lighted in the notebook.

2) *Network model, training, and changing hyper-parameters.*

A) How many layers does the network in the notebook have? How many neurons does each layer have? What activation functions and why are these appropriate for this application? What is the total number of parameters for the network? Why does the input and output layers have the dimensions they have?

B) What loss-function is used to train the network? What is the functional form (mathematical expression) of the loss function? and how should we interpret it? Why is it appropriate for the problem at hand?

C) Train the network for 10 epochs and plot the training and validation accuracy for each epoch.

D) Update model to implement a three-layer neural network where the hidden-layers has 500 and 300 hidden units respectively. Train for 40 epochs. What is the best validation accuracy you can achieve? – Geoff Hinton (a co-pioneer of Deep learning) claimed this network could reach a validation accuracy of 0.9847 (<http://yann.lecun.com/exdb/mnist/>) using **weight decay** (L2 regularization of weights (kernels): <https://keras.io/api/layers/regularizers/>). Implement weight decay on hidden units and train and select 5 regularization factors from 0.000001 to 0.001. Train 3 replicate networks for each regularization factor. Plot the final validation accuracy with standard deviation (computed from the replicates) as a function of the regularization factor. How close do you get to Hinton's result? – If you do not get the same results, what factors may

influence this? (hint: What information is not given by Hinton on the MNIST database that may influence Model training)

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).

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.