Programming with neural networks: Exercise sheet 5 SS 2020

University of Würzburg - Chair for Computer Science VI Prof. Dr. F. Doll, A. Hekalo, M. Krug

Exercise sheet: 5

Processing on June 4th

Task 1: Setting up Tensorflow and more

Install Tensorflow, Keras, Matplotlib and scikit-learn on your computer.
The operating system does not matter, older computers may need an older
sorflow version, e.g. B. 1.7 or 1.3, installed (pip install tensorflow == 1.3)
Optional: Install all the libraries you need to run Tensorflor on your GPU
to run.

- https://www.tensorflow.org/install
- https://www.tensorflow.org/install/gpu

Keras is automatically installed when the Tensorflow package is installed.
Install matplotlib and scikit-learn analogously to Tensorflow using
pip install matplotlib or pip install scikit –learn.

Task 2: MNIST training

(a) Run the MNIST tutorial for Keras locally on your computer (https:
//www.tensorflow.org/tutorials) .

(b) At the end, the test data is evaluated in the Turoial. Evaluate additionally
on the training data and compare the two results. What falls
To you and how can you explain that?

(c) Add another layer with 512 nodes and compare the
results with the default settings.

(d) Convert the MLP into a simple perceptron, i.e. construct that
Network in such a way that it only consists of an input and an output layer, and
compare the results with the default settings.

(e) Use a tanh activation function instead of the ReLU and compare
The results

(f) For example, save your best result (model.save model)

Task 3: MNIST evaluation

(a) Create another script in which you use the model that has already been trained

Page 2

load (model.load model) and to check the test data again eva-
calculate (it must give the same result).

(b) Take a look at the documentation of the Keras model (tf .keras.model on https://tensorflow.org). You are looking for a role around the class predict each test sample.

(c) Now calculate manually (without using the model.evaluate function) the Accuracy and compare the value with the one calculated from Keras.

(d) Calculate the individual Precision, Recall and F_1 scores for each class k . Count TP, FP and FN as discussed in the lecture.

(e) Calculate the confusion matrix M_{ij} , which counts the number of times that numbers come together were confused. I. E. M_{00} counts all correct zeros, M_{01} counts how often one 1 was predicted instead of a zero.

(f) Plot the matrix as an image using Matplotlib. Where do the most common Confusion on? (You may have to set the diagonal to 0 so that you can see more.)

(g) Use the in scikit-learn.metrics (import as import sklearn.metrics) available functions to compute the confusion matrix and metrics and compare with your results.