Homework 05

IANNwTF

This week's deadline is *December 10th, 11.59pm.*

Submit your homework via https://forms.gle/HM7L1P2Xc3Y7LhB67

Remember that you now have to review the homework submission of two other groups.

Contents

1 Assignment: CIFAR-10 Classification

During the last two weeks you learned how to make use of TensorFlow functionalities to build neural networks. This week you will do the same implementing a *Convolutional Neural Network*. So far, the instructions in the previous homework assignments have been rather specific about how things should be done. This week's however is going to be a little bit more free, which should give you the opportunity to play around with your model setup and hyperparameters. In the end you can share your best results or interesting results for comparison with your fellow students!

1.1 Prepare the Dataset

First familiarise yourself with the dataset. It is included as a TensorFlow Dataset but can also be obtained from other sources. Try to understand how it works (information in the data, dataset size, image size, are there color channels, type the data is stored as, etc.). Understanding the data you are working with and applying respective preprocessing steps is a major part of the process for real life applications.

Now that you familiarised yourself with it, load the data and apply respective preprocessing steps using a data pipeline (In case of problems refer to your old homework tasks or the Courseware). [1] Visualize a sample of the dataset.

1.2 The CNN Model

Build a *Convolutional Neural Network* inspired by the information gained in this weeks Courseware. You are going to need more than the dense-layers used so far. Start with a basic structure and don't necessarily think about optimizing the architecture too much, as this will be part of the last step. [2]

1.3 Training the network

Define a training loop function. Define your hyperparameters, think of an appropriate loss function and optimizer and initialize everything. Store loss and accuracy for training and test data. Training your network for a maximum of 15 epochs should be enough.

[1] You should be able to mostly recycle one of the pipelines from old homeworks.

[2] Try to think of an appropriate structure of your own or get inspiration from lecture or guides on the internet. Only if you are totally stuck you can scroll down to subsection 3.1, an example of what your network structure and hyperparameters could look like can be found in the google doc linked there.

1.4 Visualization

After training visualize the performance of your model and the values that you collected during training and testing. [3]

2 Adjust hyperparameters and architecture

Now it is time to learn about the influence hyperparameter and training choices can make. Try out:

- Two different Learning Rates
- Two different Architectures (e.g. number and size of layers)
- Two different optimizers (Including Adam and one other)

In total this should create 8 respective settings, for each of which we ask you to plot:

- Training Accuracy
- Training Loss
- Test Accuracy
- Test Loss

To be explicit: This should result in a total of 8 times four plots!

2.1 Sharing your results

A strength of the AI-community is the open exchange of knowledge. This open-source spirit is important. We have prepared a google form in which you we encourage you to post your model-architecture as well as a list of your hyper-parameters and a plot of your results (loss & accuracy). This way you can see what your peers did and how different implementations perform.

Link to: IANNwTF homework 05 CNNS - Collaboratory results.

Each group should try to at least post one architecture of their best results but feel free to post more. Coding is all about the community.

And most importantly: **Enjoy the process!**

[3] You can use matplotlib or seaborn here, or try to go the extra mile and implement a TensorBoard to track your models performance.