EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATH. - NATURWISS. FAKULTÄT
FACHBEREICH INFORMATIK
KOGNITIVE SYSTEME · PROF. A. ZELL

Introduction to Neural Networks

Assignment 5

Assignment due by Friday: 17th June 2022, Discussions on: 23rd June 2022

Question 1 Boat-MNIST: Compete on an image classification task on SeaDronesSee

We created a machine learning challenge on our SeaDronesSee benchmark, in which you have to classify images with a neural network. For this challenge you should work in your usual group. If you do not have sufficient hardware to train your neural networks, you can have one TCML cluster account per group to train your networks. For that, please ask in the moodle-forum. For more information on the TCML-Cluster please see the documentation of the TCML-Cluster, including a tutorial of how to train a network here: `https://tinyurl.com/yyje2e37`.

The SeaDronesSee benchmark hosting a leaderboard and the competition data is found on

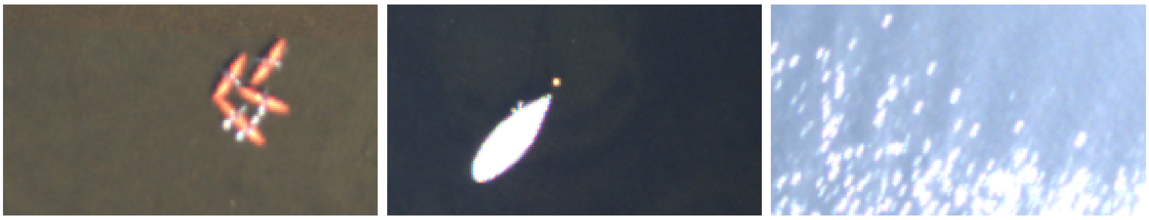
`https://seadronessee.cs.uni-tuebingen.de/`.

You can find a starter code on git:

`https://tinyurl.com/2p8s64t3`.

Boat-MNIST is a toy data set for the task of binary image classification, i.e. given an image, your task is to classify it into one of two classes. It aims at providing a simple hands-on benchmark to test small neural networks. There are the following two classes:

- 1 - if the image contains any watercraft, including boats, ships or surfboards ON the water
- 0 - all the rest, i.e. just water or anything on the land (could also be boats).



As metrics, we employ the prediction accuracy (number of correctly predicted images divided by number of all images) and the number of parameters of the model. For this benchmark, you can upload your trained ONNX model to be ranked on the leaderboard. For that, please refer to the sample script in git. It trains a simple single-layer perceptron architecture on this data set upon saving and exporting the Pytorch model as an ONNX file. Make sure the exported model uses the transformation provided in this code, as this is the transformation used for the webserver evaluation.

In the following, you find a sequence of tasks that you should do in order to participate in the challenge. For this assignment sheet you are supposed to upload your solution to the git repository listed above. For that, you should clone the repository to your local computer and once you completed the tasks below, you should upload your code and your documentation by creating a merge request.

(a) Go to the git repository and download the sample script Boat_MNIST/challenge_nn.py. Make sure you install all the packages needed to run this script. You can create a conda or virtualenv (as described in the tutorial) environment for that.

(b) Roughly familiarize yourself with the script and its workings. Make sure you understand that the script writes two model files to the script directory: a Pytorch model and an ONNX model. The ONNX file is later needed to upload on the webserver to evaluate your model. Download the data set at https://seadronessee.cs.uni-tuebingen.de/dataset and adjust the paths in the script. Run the script and note that you should obtain a validation accuracy of roughly 66%. You can already use the saved ONNX file for part (d).

(c) The neural network architecture in the script is a very basic single-layer neural network with sigmoid activation function. The sigmoid activation function at the end is needed to limit the output between 0 and 1. At evaluation, the output is rounded which yields the predicted class. Now, try to adapt the script by modifying the neural network architecture and the training configurations including the hyperparameters. For example, add a hidden layer in your neural network followed by a ReLU activation function.

(d) Take part in the SeaDronesSee Boat-MNIST challenge. For that, go to the SeaDronesSee webpage and register with your student e-mail. Note that this is a separate registration to your usual university account. Once registered and logged in, you can upload your ONNX file on https://seadronessee.cs.uni-tuebingen.de/upload. Choose an appropriate project name, optionally add a link to a resource that you referred to and provide your group number from moodle. Once you click upload, it will run the script on the webserver. Note that the evaluation can take up to two minutes. If your model is too big and the evaluation takes longer, it will cancel the evaluation. If the webserver does not throw an error within roughly the first 10 seconds, you can leave the page. It will add the validation accuracy and number of parameters of your model to the leaderboard right away. It will also compute the test set accuracy but that will only be revealed at the end of the competition. Note that there is an upload limit of three uploads per day each not bigger than 200 MB. If you encounter problems when uploading your file, contact us via moodle, git or mail.

(e) You can upload three times a day until the end of the challenge. Then, the winner is the group with the highest test set accuracy. In case of a draw, the smaller model is considered better. The top 5 teams will get (5/4/3/2/1) bonus exam points. They are expected to give a short presentation (≈5 mins) on their process and methods at the date of discussion.

(f) Document your results. Add a short explanation of your chosen architecture and a documentation of your training procedure. This should at least include a description of the architecture, the used hyperparameters, the used optimizer, any data augmentation and normalization techniques, and the final validation loss. Make a merge request by creating a folder with your group number in the git folder Boat_MNIST and include your code and a PDF-documentation. Also upload you files to moodle.

Hints:

- You are allowed to change all parts of the code if it provides you some benefits.
- Some adaptations for potentially better performance are: learning rate (scheduler), learning rate decay, number of layers, batch size, loss function, activation function.