Homework 06

IANNwTF

November 25, 2022

This week's deadline is *11.12., 23:59*.

Submit your homework via https://forms.gle/ApAZ5ubY8ewgNmJA9

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

Contents

1 Recap on Reviews

Welcome back to the sixth homework for IANNwTF. Here's just a short refresher on how to do reviews.

In addition to handing in your homework, you will have to review last week's homework of two groups and note down which group you reviewed on the homework submission form. This requires you to find two other groups and have a short (10 min) meeting where you go over each others homework submission, discuss it and then write a short summary of what you discussed on each others forum pages. We recommend using the Q&A timeslots for this purpose, but you can meet however and whenever you like. The main review part of this should be the discussion you have together. The written review in the forum should merely be a recap of the main points you discussed so that we can see that you did something and the reviewed group has access to a reminder of the feedback they received. **If there are any open questions regarding your or any other groups code afterwards, please feel invited to discuss this with us in the QnA sessions, so we can help you sort out how a perfect solution would have looked like!** Just to make this obvious: We will not penalize any submission (by you or any of the groups you reviewed) based on such questions. The purpose is helping you understand and code better only.

As to how a discussion could look like, you could for example have the group being reviewed walking the other two groups through their code. The other two groups should try to give feedback, e.g. "I like how you designed your data pipeline, looks very efficient.", "For this function you implemented there actually exists a method in e.g. NumPy or TensorFlow you could've used instead." or "Here you could've used a slightly different network architecture and higher learning rate to probably achieve better results.", and note down their main points in the forum.

Important! Not every member of every group has to be present for this. You could for example implement a rotation where every group member of yours only has to participate in a review every third week.

2 Assignment: CIFAR-10 Classification

Last week you already touched on this dataset and we asked you to build a CNN in order to classify the different pictures. This task will be about the same dataset. As you might have realised, your model possibly did not perform too well (at least on the test data) using the free structure you could choose. The reason for that is probably that your network *overfitted* on the test data. This week we'll tackle this problem and apply further optimisation steps.

2.1 Prepare the Dataset

First, as always, familiarise yourself with the dataset. Even if you still remember it from last week, take a close look again. **Always know the data your**

working with by heart! Honestly it just makes your life as data scientist easier. Further, this week you might want to apply some data augmentation.

If you haven't done the homework last week, remember Cifar10 is included in TensorFlow Dataset. Be aware of the relevant information (dataset size, image size, are there color channels, type the data is stored as, etc.).

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

2.2 Optimise your model

You probably still have your model architecture from last week. We would advise you to keep it the same as last week in order to see the effect of the changes you are about to make. After applying the respective steps you can of course feel free to toy around with different concepts like *ResNet* or *DenseNet*. Still, please document the effects of your changes with the old network architecture in a separate document.

So what can you do?

1. Use the respective steps from the Courseware to find out whether your model overfits the test data

2. Apply at least **five** different optimisation techniques (You might have already done that in the first part)

For each technique you choose write down how it works and why you are specifically using it (And perhaps why you're using it over another technique). Also add this to the document mentioned in the prior paragraph. The purpose is to actually think about what techniques to apply and the effect they have. In this toy problem you might be able to just blindly try different concepts (Even though this probably takes more time). In bigger projects, with more training time, you will not be able to this and need to actually understand what is happening. Optimization techniques also can worsen the performance of your network if applied incorrectly.

2.3 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.

2.4 Visualisation

After training visualize the performance of your model and the values that you collected during training and testing. You are not allowed to use the exact same visualisation as last week. [1]

As always, keep the most important part in mind: **Enjoy the process!**

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