AI - Lab

2. Datasets & Dataloaders

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Prof. Dr. Patrick Baier

SS 24

Datasets

- In the last exercise we used the already existing Dataset `torchvision.datasets.MNIST` and fed it into a data loader:

```
train_loader = torch.utils.data.DataLoader(
  torchvision.datasets.MNIST('/files/', train=True, download=True,
                             transform=torchvision.transforms.Compose([
                               torchvision.transforms.ToTensor(),
                               torchvision.transforms.Normalize(
                                 (0.1307,), (0.3081,))
                             ])),
  batch_size=batch_size_train, shuffle=True)
```

- You can also write your own Dataset classes to load custom data.

Datasets

- A tutorial on this can be found here.

WRITING CUSTOM DATASETS, DATALOADERS AND TRANSFORMS

Author: Sasank Chilamkurthy

A lot of effort in solving any machine learning problem goes into preparing the data. PyTorch provides many tools to make data loading easy and hopefully, to make your code more readable. In this tutorial, we will see how to load and preprocess/augment data from a non trivial dataset.

To run this tutorial, please make sure the following packages are installed:

- `scikit-image`: For image io and transforms
- `pandas`: For easier csv parsing

```
from __future__ import print_function, division
import os
import torch
```

Assignment 2

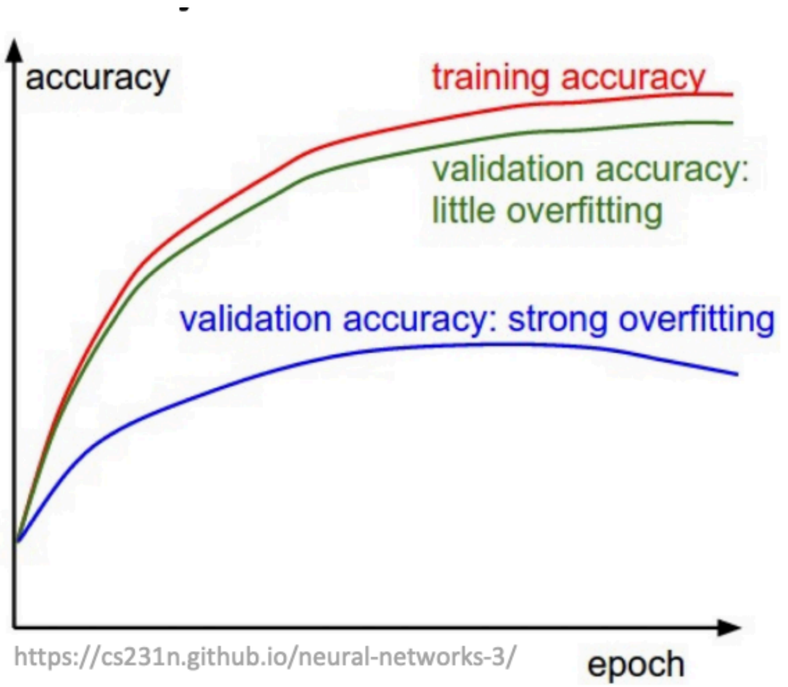
Task:

1. Write a custom dataset class for the titanic data (see the `data` folder on GitHub).
   Use only the features: "Pclass", "Age", "SibSp", "Parch", „Fare“, „Sex“, „Embarked“.
   Preprocess the features accordingly in that class (scaling, one-hot-encoding, etc) and split the data into train and validation data (80% and 20%). The constructor of that class should look like this:
   ```
   titanic_train = TitanicDataSet('titanic.csv', train=True)
   titanic_val = TitanicDataSet('titanic.csv', train=False)
   ```
2. Build a neural network with one hidden layer of size 3 that predicts the survival of the passengers. Use a BCE loss (Hint: you need a sigmoid activation in the output layer).
   Use a data loader to train in batches of size 16 and shuffle the data.
3. Evaluate the performance of the model on the validation data using accuracy as metric.

Assignment 2

Task:

4. Create the following plot that was introduced in the lecture:
5. Increase the complexity of the network by adding more layers and neurons and see if you can overfit on the training data.
6. Try to remove overfitting by introducing a dropout layer.



Submission Guidelines

- Send your solutions as notebook file (.ipynb) to me via e-mail, mentioning the team name and team members in the e-mail.
- Use the following e-mail subject: `[KI-LabSS24] Assignment 2 - <teamname>`
- Deadline is: **04.04.2023 - 11:59 p.m. (23:59)**

Final words

- You can start now working on assignment 2.
- It is up to you and your partners how you organize your time working on this.
- I will be around until the end of the slot at 1 pm for questions.
- Of course, questions can also be asked all time on Mattermost (but do not post any solution there!).

- Any more questions now?
- Happy Coding!