Deep Learning KU (DAT.C302UF), WS24
Assignment 2
Training Simple Neural Networks

Thomas Wedenig
thomas.wedenig@tugraz.at

Teaching Assistant:	Hade Mohamed
Points to achieve:	25 pts
Deadline:	04.12.2024 23:59
Hand-in procedure:	You can work in groups of **at most two people**. **Exactly one** team member uploads two files to TeachCenter: **The report (.pdf)** and **the Python code (.py)**. The first page of the report must be the **cover letter**. Do not upload a folder. Do not zip the files.
Plagiarism:	If detected, 0 points for all parties involved. If this happens twice, we will grade the group with "Ungültig aufgrund von Täuschung"

We will use PyTorch to train a neural network for a regression task on the California Housing Dataset. Our goal is to predict the values of the houses based on various predictive variables. We recommend that you set up a conda environment (as detailed in the `P2_Intro_to_PyTorch.ipynb` Jupyter Notebook) and install the dependencies using `pip install -r requirements.txt`. Your submitted code must not use any external libraries that are not included in this file.-> he should be able to run code, don't use any other modules, than listed

Use the following code to fetch the dataset and split it into a training and test dataset:

```
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
X, y = fetch_california_housing(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=302)
```

There are 15,480 training and 5,160 test examples with 8 predictive input features and 1 target variable. Input features consist of the spatial locations of the districts that data is collected from (latitude, longitude), demographic information in the districts (income, population, house occupancy), and general information regarding the houses (number of rooms, number of bedrooms, age of the house). Since these statistics are measured per district, the features correspond to averages or medians across "block groups" (a geographical unit with a population of 600 to 3,000 people).

The input features (in `X_train` and `X_test`) are provided in the following order:

- `MedInc` : median income in block group - columns of the data set
- `HouseAge`: median age of a house within a block
- `AveRooms`: average number of rooms per household
- `AveBedrms`: average number of bedrooms per household
- `Population`: block group population
- `AveOccup`: average number of household members
- `Latitude`: a measure of how far north a house is
- `Longitude`: a measure of how far west a house is

The target variable is the *median house value* for California districts, expressed in hundreds of thousands of dollars ($\$100,000$).

For all the tasks below, create the appropriate code and discuss your experimentation and reasoning process, findings and choices in your report. All results should be shown in the PDF report.**The submitted code should be able to reproduce your results**. Therefore, make sure you set the random seed appropriately to ensure deterministic behaviour of your experiments.

-> for reproducability
-> need to get same results as you show in report

Task details:

a) (3 pts) : Get familiar with the dataset. Again split `X_train` into a *proper training set* and a *validation* set. We will use this validation set during the model selection process. We will only use the test set to evaluate the final model (i.e., *after* model selection). Investigate the feature distributions and include all plots in your report. Normalize the data if it is necessary. When you choose to normalize features, make sure you do not compute statistics using the test set.

add plots of the feature distributions
-> tell him if you decide to normalize features
-> but only based on training sets

b) (7 pts) : Design your feed-forward neural network architecture for the regression task. Explain your choices for the output layer and the loss function that you will use. Minimize the training loss by using mini-batches of suitable size. Try at least 5 different architectures with varying numbers of hidden units and hidden layers. Compare these choices and report training and validation set loss in a table.

-> basically manual tuning of hyperparameters
ech row one architecture, trianing lloss, test loss

c) (4 pts) : Investigate and compare different optimizers such as stochastic gradient descent (SGD), momentum SGD, and ADAM. Accordingly, try a number of learning rates and also try out adapting the learning rate during training by scheduling. Provide a table where training and validation losses of various optimization hyper-parameters are compared.

also try scheduling
table with hyperparemeters, optimiezers, schedulers, training and evaluation
if you normalize features, also use normlization on test set(?)

d) (3 pts) : Clearly summarize your final model once your architecture choices are fixed. Provide a plot where the evolution of the training and validation loss during training are shown throughout iterations. Perform a final training with this model on the whole training set (i.e., combine the proper training set and the validation set). Report and comment on the final test loss. Provide a scatter plot in which you compare model predictions (x-axis) with their ground truth values (y-axis) on the test set.

-> do parameters search only for the best(or the best two)
number of epochs is also a hyperparameter
-> try out different numbers and discuss this
numbere or iterations/epochs on x-axis, training/validation loss on y-axis
think about, discuss, if you should normalize all features

e) (3 pts) : Denote the training dataset with $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, and $y_i \in \mathbb{R}$. Let $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}$ denote the neural network. Show that finding the minimizer for the `MSELoss` is equivalent to finding the *maximum likelihood estimate*, i.e.,

- now use entierety of the data, also including validation set, because wee don't need thaat aanymoree. still not including test set

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (f_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)^2 = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(y_1, \ldots, y_n \,|\, \mathbf{x}_1, \ldots, \mathbf{x}_n)$$

- under somee assumptions(i.i.d.)

2D scatter plot -> perfect model looks like identity (line with slope 1, offset 0

with the following assumptions: All $y_i \sim p^*(y_i \,|\, \mathbf{x}_i)$ are independently sampled from their corresponding conditional distributions and $p_{\boldsymbol{\theta}}(y_i \,|\, \mathbf{x}_i) = \mathcal{N}(y_i; f_{\boldsymbol{\theta}}(\mathbf{x}_i), \sigma^2)$ is Gaussian with mean $f_{\boldsymbol{\theta}}(\mathbf{x}_i)$ and fixed variance σ^2 (independent of $\mathbf{x}_i$). Clearly show all steps of your derivation in your report.

TIP: - independent assumption needed to rewrite into product and samplewiseee likelihood, then apply log

f) (5 pts) : Now assume that we want to use a similar architecture for the binary classification problem of determining if the median house value is below or over $\$200,000$. Explain which parts of the architecture and the training pipeline you would need to change, and which test set evaluation metrics should be investigated in this case. Explain the reasons for these differences.

Implement these changes to the architecture and the training pipeline you had in Task (d), and train a single model for this binary classification task using the whole training set. Note that you will also need to redefine your target variables by simply executing the following lines:

```
y_train[y_train < 2], y_test[y_test < 2] = 0, 0
y_train[y_train >= 2], y_test[y_test >= 2] = 1, 1
```

Evaluate your model on the test set, report and comment on its performance.--> tell him which metric you use
t-> only use the final model/no model selection anymore

-> trained again on the whole data, including valication se