Technische Hochschule Ingolstadt, M-APE, Prof. A. Schiendorfer Lukas Lodes M.Sc.

Intro to AI and Neural Networks (Summer 2023)

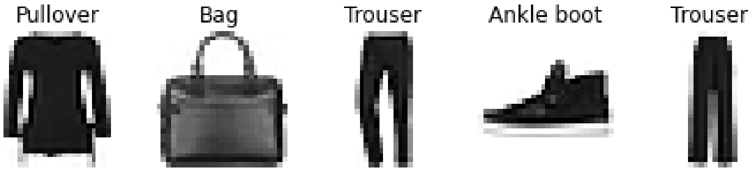
Assignment 07

Exercise 1 *(Iris data set)*

Download the iris data set (`https://archive.ics.uci.edu/ml/datasets/iris` or `https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html`) and learn the classification of the three species using *k-Nearest Neighbor*. For this task, divide the data set properly in training, validation and test set and choose an appropriate loss function. Train models with different values for the hyperparameter k (e.g. $k = 1, ..., 15$) and compare their performance on the validation set. Show the development of the training and validation error for varying k in a *learning curve*. Select the best performing model and compare the validation and test error. What advantages (disadvantages) could have instance-based techniques like k-NN compared to model-based ones? For which conditions are the two techniques particularly suitable?
In the second part of the assignment, compare the instance-based k-NN classifier against the model-based logistic regression model. Which performs better? Additionally, try to decrease the difficulty of the learning problem by applying polynomial transforms to the features. Use can use our provided Jupyter Notebook (`Iris_classification_for_students.ipynb`) as a starting point. If you don't know how to use the functions provided by scikit-learn, take a look at the official documents (`https://scikit-learn.org/stable/modules/classes.html`).

Exercise 2 *(Training a classifier on fashion MNIST)*

Next, we will train our first neural network on the fashion MNIST dataset. This dataset contains 28 × 28-images of items of clothing (and is slightly more challenging than the regular MNIST dataset for handwritten digit recognition):



This exercise will require Keras[1]. Data loading and preparation of train/test/val splits have been prepared in `Fashion_MNIST_base.ipynb`.

- As a baseline, train a random forest classifier on `X_train_vectors` without any restrictions on depth and leave all settings to their default values. Use the following cell to show a random image and your classifier's prediction.
- Using the same dataset, train a logistic regression classifier from sklearn. Which model is better in terms of val. and train accuracy?
- Next, implement a `Sequential` neural network model in Keras. What number of input and output units (neurons) will your model have? Start with two hidden layers, one with 300 units, one with 100. Use `sparse_categorical_crossentropy` as loss function and `adam` as optimizer when compiling the Keras model (`model.compile()`).
- Evaluate your model's summary by calling `model.summary()` and compare this with the block diagram notation from the lecture.
- Follow the notebook and inspect some of the weights (random, as they have not been trained). Do the dimensions make sense?
- Start training the model using `model.fit()` and inspect the loss values. Use the following cell to print the learning curves.

[1] `keras.io` – if you want to run it locally, you need to install TensorFlow `https://www.tensorflow.org/install` or just use Google Colab.