Machine Learning (WS 24 / 25) – Project 2

HfT Stuttgart, Prof. Dr. Laura von Rueden

Deliverable:

- Python notebook (.ipynb) with code and text cells where you describe both your analysis steps and your results interpretation. An extra report is not needed when you describe everything in the notebook.
    - As last time: Structure your notebook along the tasks. Use proper commenting. Describe your observations in text cells after the code cells.
- Hand-in via Moodle until **Friday, 17.01.25., 6 pm**

Goal:

- Train neural networks for image classification of handwritten digits and investigate effects of the architecture settings. Have fun ☺ !

Datasets:

- MNIST. Download with

```
from sklearn.datasets import fetch_openml
mnist = fetch_openml('mnist_784', as_frame=False)
```

Hints:

- You can expect accuracies of more than 90%.
- Each training of a neural network can take a few minutes (depending on the used hardware). It's good to start early to have enough time.
- These python notebooks contain useful code snippets:
    - https://github.com/ageron/handson-ml3/blob/main/03_classification.ipynb (Lines 5-16)
    - From the lecture: ML_P07_neural_networks.ipynb
    - https://github.com/ageron/handson-ml3/blob/main/10_neural_nets_with_keras.ipynb (Lines 23-48)

Tasks:

1. Data collection, exploration, and preparation
    a) Get the data. How many data instance does the data set have? How many features does and how many target variables are there?
    b) Describe what the features and the target variables are. How are the images stored?
    c) For exploration: Plot a few images.
    d) Split the data into data subsets.
    e) Is feature scaling needed?

2. Feed-forward neural network
    a) Train an MLPClassifier using scikit-learn. Set verbose=True. Set the number of epochs to 10. Otherwise, use the default settings.
        i. What is the architecture? How many model parameters does the network have?
        ii. How is the model accuracy on the validation set?
    b) Train a new network with 100 epochs. What happens?
    c) Implement partial fitting, measure the error during training and plot the learning curves on train and validation set. After how many epochs is the best model achieved? What is its validation accuracy?
    d) Investigate the effect of the activation function of the hidden layer. Try two other activation functions. What do you observe in the learning curves? How does the validation accuracy change?
    e) Save the best model of a-c). This is now your model 1.

3. Effect of architecture and training size
    a) Use neural networks with the same settings as model 1, but now modify the size of the hidden layer. Train two models with smaller, and two with larger size. Measure the validation performance and plot them against the hidden layer size. What do you observe?
    b) Select one architecture of 3a). Let's say it has X neurons on the hidden layer. Now select two new architectures that also have a total number of X neurons but have more hidden layers.

How many parameters does these architectures have? Train the model. Compare the validation performance and the learning curves. What do you observe?

c) Take again your model 1. Now investigate the effect the data size. Train with only 10% and 1% of the training data. Again: what do you observe?

d) Save the best model of a-c). This is now your model 2.

4. Python Keras
    a) Make sure you have Tensorflow and Keras installed.
    b) Now train a neural network with the same settings as model 2, but now in Keras. Do you achieve exactly the same validation performance? Save the Keras model. This is now your model 3.
    c) Research and describe: When is Keras useful, especially compared to Scikit-Learn?

5. Convolutional neural network[1]
    a) Implement the LeNet5 architecture in Keras and train the model. How is the validation accuracy? How is it compared to model 2?
    b) Is there a performance improvement for specific classes? Explain your observation.
    c) How many parameters does the architecture have?
    d) Optional: If you have access to a GPU: Train the model on it. How does the training time change?
    e) Last but not least, train one more neural network: Train LeNet5 with Dropout regularization. If you already had it turned on before, turn it off now. What do you observe?
    f) Save the best model of a-e). This is now your model 4.

6. Final comparison
    a) Evaluate the test performance of models 1-4. Which is the best? Explain why.
    b) Demonstrate how it can be used for image classification.

[1] Will be discussed in lecture 9.