DATA.ML.100 Introduction to Pattern Recognition and Machine Learning
TAU Computing Sciences
Exercise - Week 5: *Neural network learning (CIFAR-10)*

Be prepared for the exercise sessions (watch the demo lecture). You may ask TAs to help if you cannot make your program to work, but don't expect them to show you how to start from the scratch.

1. **CIFAR-10 – get to know** (no points)

   CIFAR-10 is a benchmark popular datasets used in machine learning literature for bechmarking different methods. For brief introduction check https://www.cs.toronto.edu/~kriz/cifar.html.

   Download the dataset to your local computer (perhaps your prefer */worktemp* instead of your home directory) from here (163MB):

   - https://tuni-my.sharepoint.com/:u:/r/personal/joni_kamarainen_tuni_fi/Documents/cifar-10-python.tar.gz?csf=1&web=1&e=4wmDah

   Extract the files and then edit configuration paths in the provided *cifar10_illustrate.py*. Run the code:

   ```
   (intro_prml) kamarain@Joni-Precision-5520:~/exercises$ python cifar10_illustrate.py
   ```

   The code should randomly plot images from the dataset. Note that the code loads only one of the training batches that each contains 10,000 training samples. Test data is also provided separately (10,000 samples, 1,000 per each class).

2. **CIFAR-10 – Neural Networks** (60 points)

   Data: We will use the CIFAR-10 dataset.

   Study the neural network examples in the neural networks Jupyter notebook.

   Note that with neural networks we prefer to use so called "one-hot encoding". That means that the network does not output a single value between 0-9 but there is a separate output for each class.

   For network training, the target value is a vector of 10 elements. For that purpose, you need to convert the class ids to one-hot vectors:

   $$0 \rightarrow (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$
   $$1 \rightarrow (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$
   $$2 \rightarrow (0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$
   $$\ldots$$

   Write Python code that 1) makes a full connected neural network that take a 32×32 Cifar-10 image as a 3,072 dimensional input. The network produces 10 outputs each representing one Cifar-10 class. The hidden layer for a simple MLP can, for example, contain five neurons only, in the beginning.

   Then, 2) code trains the network with Cifar-10 trainining data. Set a suitable learning rate and number of epochs. Plot the training loss curve after training to confirm that the network learns.

   Play with the parameters, and after you find good parameters, 3) test the model with Cifar-10 test samples.

   Finally, print classification accuracy for the training data and test data.

   During the lectures *Keras* (includes *TensorFlow*) was used to train neural networks, but also PyTorch can be used.

   In Keras a layer of 5 full-connected neurons can be added as

   ```
   model.add(Dense(5, input_dim=3072, activation='sigmoid'))
   ```

   For the last layer, you need to always add a full-connected (dense) layer of 10 sigmoid units.

   Return the following items:

   - Python code (single file): `<surname>_cifar10_mlp.py`
   - A full desktop screenshot that includes a terminal window executing your code and printing the results:
     `<surname>_cifar10_mlp_desktop.png`