DATA.ML.100 Introduction to Pattern Recognition and Machine Learning
TAU Computing Sciences
Exercise 4 *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 – Neural Networks** (60 points)

   Data: We will use the CIFAR-10 dataset.

   Study the neural network examples in the course Jupyter notebook. Note that with neural networks we prefer to use "one-hot" output vectors instead of a single value that defines the class. Therefore, you need to convert class numbers 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 inputs a Cifar-10 image (or 3,072 dimensional vector) and outputs 10 outputs, one for each class (you may start with one layer of 5 neuros). 2) train your neural network with Cifar-10 train data (select a suitable learning rate and number of epochs) and 3) test with Cifar-10 test data. Again report the performance and compare it to 1-NN and Bayes classifier.

   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'))
   ```

   or Keras also allows to define inputs as images

   ```
   model.add(Dense(5, input_shape(32,32,3), activation='sigmoid'))
   ```

   If you want to play with convolutional layers you may add them as

   ```
   model.add(keras.Input(shape=(32, 32, 3)))
   model.add(layers.Conv2D(32, 5, strides=2, activation="relu"))
   ...
   ```

   The last layer is always 10 output sigmoid.