DATA.ML.100 Introduction to Pattern Recognition and Machine Learning
TAU Computing Sciences
Exercise 4 *Visual classification with NNets (CIFAR-10 dataset)*

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.

   A good introduction to this exercise is the Jupyter notebook example from the lectures. In neural networks we define one output per class. You need to convert the class labels to "one-hot" 10-vectors:

$$
\begin{aligned}
0 &\rightarrow (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)\\
1 &\rightarrow (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)\\
2 &\rightarrow (0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0)\\
&\qquad\qquad\qquad\qquad \cdots
\end{aligned}
$$

   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.

   Note that the tools for neural networks learning in Python are TensorFlow and Keras. You may images as one long vector:

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

   or images:

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

   With the image input you may use the convolutional layer type which is at the core of all deep neural network architectures:

```
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.