AC50001: Classification and Regression using Keras

TensorFlow is a popular open-source library for developing and evaluating machine learning methods and applications. Keras is a high-level API that makes it easier to build, train and test machine learning models in TensorFlow (see: https://keras.io/about/). In this lab you will be editing and executing Python code, making use of Keras to train deep neural networks to perform classification and regression tasks.

Google Colab notebooks are Jupyter notebooks that are stored on your Google Drive account (you'll need one). Notebooks let you combine code, rich text, plots, image etc. For a quick video intro to Colab see: https://www.youtube.com/watch?v=inN8seMm7UI&ab_channel=TensorFlow. In this lab you will be working in Colab. This means configuration is done for you and you can access GPUs so that training is faster.

The exercises below are from the Jupyter notebooks that accompany Chapter 3 of the book "Deep Learning with Python" by Francois Chollet. They make use of some well-known machine learning datasets. We have updated these notebooks so that they work on Colab with the latest version of Keras.

Setup

You should access the three notebooks at the links below and copy them to your own account.

3.5 https://colab.research.google.com/drive/1LsbqXpzGj8MAA0x6AtVBKk7R1UleueP2?usp=sharing
3.6 https://colab.research.google.com/drive/1p-xEu292xlntUv24YMKHb4hgu5tWfYRx?usp=sharing
3.7 https://colab.research.google.com/drive/102CLIVKmVnNP0gdM9SJ0TUcj6EmMxUFc?usp=sharing

You will then be able to run and edit them using your own profile. You can step through the code, running the code in each cell by hovering over it and pressing the play button, or you can run all the code via the Runtime tab.

Classifying Movie Reviews (Binary Classification)

This notebook trains and tests a neural network to classify movie reviews from IMDb as *positive* or *negative*. The building blocks should be familiar to you, apart from perhaps the following.

- Some pre-processing is used to turn the text reviews into vectors with binary elements. The 0's and 1's in these vectors indicate the absence or presence of particular words.
- The hidden layers use ReLU activation functions. ReLU is short for *Rectified Linear Unit* and is a popular and effective choice of activation function for hidden neurons in deep neural networks.
- The gradient-based optimisation used for learning is RMSprop. This is a refinement of the gradient descent method that automatically adapts the learning rate in a way that works well when training with mini-batches.

Once you have successfully run and understood this Notebook, try the exercises in the *Further experiments* section.

Classifying Newswires (Multi-class Classification)

This notebook is another example of text classification. It trains and tests a neural network to classify short newswires from Reuters into 46 topics. The approach is similar to the previous example except that now we have more than two classes. A softmax activation function is used to produce outputs that could be treated as a class probability distribution. Again, once you have successfully run and understood the Notebook, try the exercises in the *Further experiments* section.

Predicting House Prices (Regression)

In contrast to the previous two notebooks, this one addresses a regression task: predicting the median price of houses in different areas of Boston.

- The input data are varied measures with very different scales. To make learning easier, they are first adjusted so that each input has zero-mean and unit-variance.
- K-fold cross validation is used for evaluation. This splits the data into disjoint training and validation sets in K different ways, and trains K regressors. It does this in such as way that each data point is in exactly one of the K validation sets. This is an effective way to estimate performance when data are limited.

Towards the end of the notebook, the regressor is trained for 500 epochs. This can take quite a long time, so you need not run it – it does not help anyway!

Try setting up and running linear regression. How does performance compare to the neural network? Does the simple linear regression model have any advantages over the neural network (other than being faster to train)?