COMP2002 - Artificial Intelligence

Week 3 - Neural Network Exercises

Introduction

The aim of this sheet of exercises is to get you started with implementing machine learning software. You should complete the exercises ahead of the week 4 seminar session. The model answers will be published shortly after, giving you enough time to re-attempt the exercises after the demonstration in the seminar. You should complete the exercises in a Jupyter Notebook. You can either install Jupyter on your own device or use the version available in the labs.

Activities

Your task is to go through the following tasks. Please note, you are expected to complete some work on this outside of the timetabled sessions.

Exercise 1 - Classify the Iris Data

The first task this week is to use a neural network to classify the iris data. Begin by loading the data and then use the MLPClassifier class to train a classifier for the data.
Once you have trained the neural network, compare the results with the *confusion_matrix* function and plot the result with the Matplotlib *imshow* function.

Exercise 2 - Regression with Neural Networks

Having tried a classification problem, this exercise requires you to train a neural network to solve a regression problem – the California house price data we discussed in the lecture. Begin by loading the data as follows:

```
from sklearn.datasets import fetch_california_housing
# Load the data.
data = fetch california housing()
inputs = data["data"]
targets = data["target"]
```

Once you've loaded the data, plot any two of the columns against each other as follows:

```
import matplotlib.pyplot as plt

# Plot columns 0 and 1 (can replace with any pair of inputs).
plt.scatter(inputs[:,0], inputs[:,1], c=targets, cmap="viridis")
plt.savefig("california_data.pdf", bbox_inches="tight")
plt.show()
```

Exercise 3 - Normalise the Data

You've loaded the data, it's now time to pre-process the data by normalising. Use the *MinMaxScaler* to normalise the input values lie between 0 and 1.
Once you've normalised the values you can use the *ptp* method to print the difference between the maximum and minimum values for each input as follows:

```
# Print the range of the variables to show the normalisation effect.
print(inputs.ptp(axis=0))
print(scaled.ptp(axis=0))
```

Exercise 4 - Train the Model

The data is now pre-processed and ready for modelling. Use the *MLPRegressor* as shown in the lecture to train a neural network that predicts house prices.