Module Code: CS3AI18

Assignment report Title:

Coursework Date (when the work completed): Actual hrs spent for the assignment:

Contents

[Machine Learning 4](#_Toc150887725)

[SEQUENTIAL MODEL USING TENSORFLOW KERAS LIBRARY: 4](#_Toc150887726)

[**Model Training** 4](#_Toc150887727)

**Introduction**

In terms of data visualisation and analysis, the ability to extract specific parts of data and analyse them individually. This coursework would display a CSV of my choice and the methods that I have used to analyse and produce different data which serve as the foundation for training a machine learning algorithm. The ultimate goal is to train the algorithm to make accurate predictions based on the insights derived from the CSV provide

For data extraction I will be using different python scripts, that depict and display diagrams relating to the data e.g. bar charts/box plots.

**Visualisation of the datasets**

I have used a variety of different figures to represent the data presented by the Pokemon.CSV. This involved the creation of 5 different visualisations that would eventually be used to train the machine learning model. In order to create these visualisations I incorporated the imports of Seaborn and pandas, these applications would allow for the data to be manipulated and visualised correctly as seen in figure 1 below:

### Machine Learning

### SEQUENTIAL MODEL USING TENSORFLOW KERAS LIBRARY:

In order to create this machine learning technique it involved the use of the keras library. This library would be the store that provides the interface for Python neural networks. This required its own specific import and download. This is then set to a sequential model using **model = Sequential(),** a linear stack of layers. Which is then compiled, using the operator ‘adam’ and the loss function Mean Square Error. Aswell as this, after creating the model, data will need to be loaded and it is required to be trained.

**ReLU**

ReLU (Rectified Linear Unit) is an essential activation function in deep learning. Its primary role is to introduce non-linearity, crucial for enabling neural networks to learn intricate patterns within data. Without non-linear activation functions like ReLU, a neural network would essentially perform similar to a linear regression model. The function would be used to replace negative input values with zeros, leaving positive values unchanged. In doing so, this allows for the model to process and interpret patterns/representations within the specified data. In the context of 'Pokemon.CSV', this activation function contributes to recognising trends, such as how attack and defence values correlate with the total stats.  
I utilised the ReLU activation function while adding an additional densely connected layer. This layer comprises 32 units, each employing ReLU activation.

**Model Training**

***Model Fit, X\_Train, Y\_Train***

The **model.fit()** function serves as a pivotal step in training a neural network model. It enhances the learning process by iterating through the provided training dataset and refining the model's parameters based on the observed patterns. Aswell as this **X\_train** and **Y\_train** would be used to represent the training features, aswell as the corresponding target labels, which are then used to teach the model allowing for it to understand patterns between the input and output features.

**Epoch**

Epoch would be the amount of iterations the entire dataset is presented to the model for learning, the higher the Epochs, the higher number of iterations that the dataset has been passed through the model, therefore allowing for the model to refine its internal parameters respectively. In my case, I presented the model with 4 iterations via the code line ‘Epoch = 4’, this would show an indication of 4 repeats of the dataset.

***Batch\_size = 32***

This would be the parameter used to partition the dataset into smaller subsets i.e. batches, each containing 32 samples. The model would then process these batches sequentially, with the parameters updating after each of the batches have been completed. This approach assists in optimising the model's learning process.

***Summary***

Overall, [model.fit](http://model.fit)() illustrates the training process, allowing for the model to learn from the training data across multiple epochs and batches whilst also providing insights on the validation dataset, and how it has improve and make more accurate predictions methods overtime.

A graph of blue and orange dots

Description automatically generated

Figure Epoch 1, First iteration of model

A graph showing a number of blue and orange dots

Description automatically generated

Figure Epoch 3, 3rd iteration of model