**ADTA 5550 - Assignment 3**

**Sri Charan (11661666)**

**Part 1: Answers**

**Question1.1**:

Yes, it is necessary to perform encoding. Since few algorithms in machine learning cannot handle string type categorical values. Algorithms like Random Forest can support categorical values as is but for deep learning project string values cannot be parsed hence it is important to perform encoding.

**Question 1.2**:

As the class labels in the target field have 10 values, it is best to perform one hot encoding on the data before using for deep learning project. We can also perform Label Binarizer (from Scikit-Learn library) but it is similar to the one hot encoding.

**Question 1.3**:

As I considered One hot encoding, the steps performed on the target variable is as below. Each class label type is considered as different, and their respective columns will be created. These new columns can hold binary values. For any data row, only the respective label’s column is one and others will be zero. We can create one less than number of columns new columns in one hot encoding.

**Question 1.4**:

We can see below example of one hot encoding. There are three categories in the data. we can see 3 different columns created which contains binary values. Also, we can see only one 1 column have 1 as value at any given row.

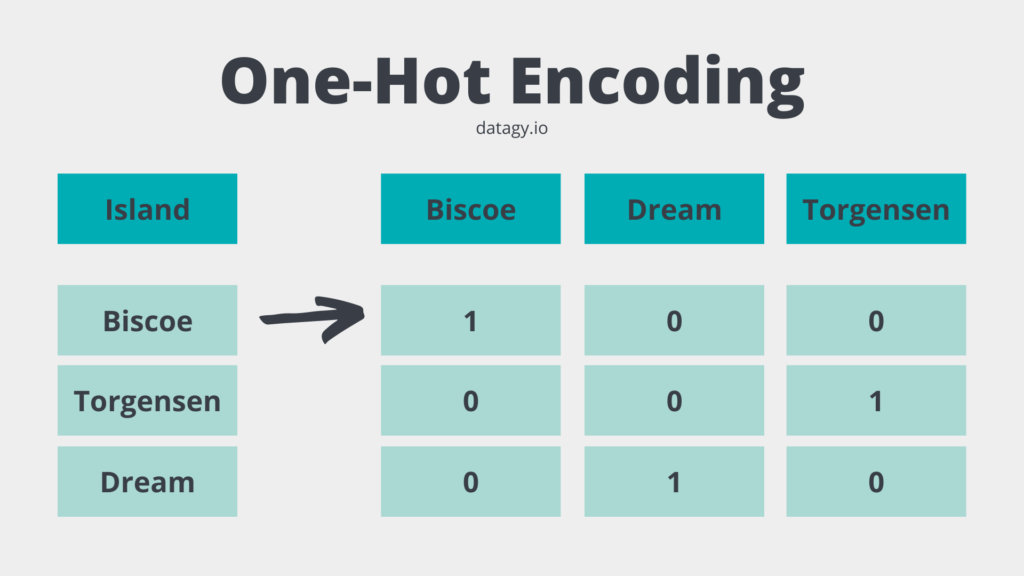


Image taken from datagy.io.

**Part II: MLPs with Keras**

Below code shows about model with 1 hidden layer.

A screenshot of a computer program

Description automatically generated

In the below screenshot we can see the results generated from model with 1 hidden layer. We can see that highest valuation accuracy found is 93% with 150 epochs. We can also see loss value reduces and the final loss value is at 0.34.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

We can see charts plotted from the results of model 2. In the beginning loss has higher value but it falls off abruptly. Later, it gradually reduces until 150 epochs. While the accuracy is highly variant up to 20 and increases suddenly until 100 epochs and the accuracy shoots to 90%. Thereafter, the accuracy does not show major differences after 80 epochs.

Training accuracy is higher than valuation accuracy in the chart. Which means the model is overfitting. The best epoch we found is 80.

**A diagram of a network

Description automatically generated**

(Created in google slides)

**Part III**

Now, the model has 2 hidden layers and 1 input while the other is output layer.

A screenshot of a computer code

Description automatically generated

In the below screenshot we can see the results generated from model with 1 hidden layer. We can see that highest valuation accuracy found is 96.67% with 150 epochs. We can also see loss value reduces and the final loss value is at 0.157.

**A screenshot of a computer program

Description automatically generated**

We can see charts plotted from the results of model. In the beginning loss has higher value but it falls off abruptly. Later, it gradually reduces until 150 epochs. While the accuracy is rises up at low epochs and stays steadily at 68% accuracy until 35 epochs and increases until 100 epochs and the accuracy shoots to 90%. Thereafter, the accuracy does not show major differences after 80 epochs.

we can see that model is having validation accuracy higher than the training accuracy. Which means model is working fine until 150 epochs. However the accuracies do not change much.

**A screenshot of a graph

Description automatically generated**

**A diagram of a network

Description automatically generated**

(Created in google slides)