	PUSAT PENGAJIAN SAINS KOMPUTER ***(School of Computer Sciences)*** **SEMESTER II 2018/19** HELAIAN KULIT TUGASAN *(ASSIGNMENT COVER SHEET)*

Nama Isaac Teoh Yi Zhe Selvaraj kanimozhi *(Name)*		Nombor Matrik P-COM0001/19 *(Matric Number) P-COM0155/19*
Masa/Kumpulan Tutorial (Jika Berkenaan) *(Tutorial Time/Group (If Applicable))* -	Tutor (Jika Berkenaan) *(Tutor (If Applicable))* -	Tarikh Akhir Penyerahan Tugasan *(Date Assignment is Due)* 21 Dec 2019

Kod Kursus *(Course Code)* CCS516	Tajuk Kursus *(Course Title)* *Computational Intelligence*	No./Tajuk Tugasan *(Assignment No./Title)* Prediction
Diterima Oleh: Tandatangan Staf *(Received By: Staff Signature)*		Tarikh/Masa Diterima *(Date/Time Lodged)*

Pengisytiharan

Saya/~~Kami~~* mengesahkan bahawa tugasan ini adalah kerja saya/~~kami~~* seluruhnya kecuali apabila saya/kami* memberikan rujukan lengkap kepada kerja yang dilakukan oleh orang lain, dan bahan di dalam tugasan ini tidak pernah diserahkan untuk penilaian dalam mana-mana kursus pengajian formal yang lain.

Declaration

I/~~We~~ certify that this assignment is entirely my/~~our~~* own work, except where I/we* have given fully documented references to the work of others, and that the material contained in this assignment has not previously been submitted for assessment in any formal course of study.*

*Potong mana yang tidak berkenaan (bergantung pada sama ada tugasan individu atau kumpulan).
Delete where applicable (depends on whether individual or group assignment).

Tandatangan Pelajar :
(Signature of Student)

Gred *(Grade)*	Pemeriksa *(Marker) Dr. Mohd Nadhir Ab Wahab*
Ulasan Penanda *(Marker's Comments)*	

✄---

Resit Pelajar
(Student Receipt)

Untuk diisi oleh pelajar
(To be completed by the student)

Nama Pelajar *(Student Name)*		Nama Pensyarah *(Name of Lecturer) Dr. Mohd Nadhir Ab Wahab*
Kod Kursus *(Course Code)* CCS516	Tajuk Kursus *(Course Title)* *Computational Intelligence*	No./Tajuk Tugasan *(Assignment No/Title)* Prediction
Diterima Oleh: Tandatangan Staf *(Received By: Staff Signature)*		Tarikh/Masa Diterima *(Date/Time Lodged)*

Neural Network

Neural network (NN) is used for prediction of the classification of data. The NN emulates the brain structure. NN is created with linkage of nodes using weight value just as the brain are comprised with many linkages of neurons the especially the neurons association using weight value (Kim, 2017).

The figure 1 shows single perceptron process:

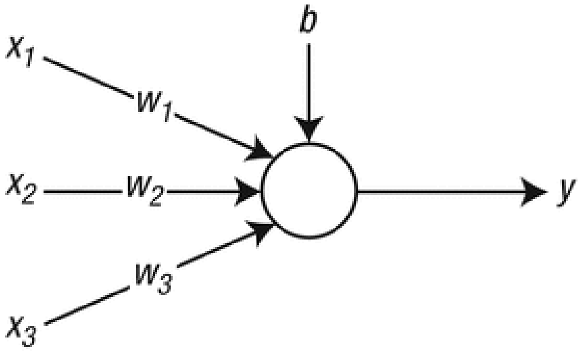

Figure 1 Single perceptron process (Kim, 2017)

The arrow and circle represent the node and single flow while the input vector x=(x_1, x_2, x_3) and weight w= (w_1, w_2, w_3) and lastly b is bias, which involve with storage of data. The weighted sum is provided as v=w_1.x_1+w_2.x_2+w_3.x_3+b.

“One of the most commonly” presented model in NN is as in Figure 2:

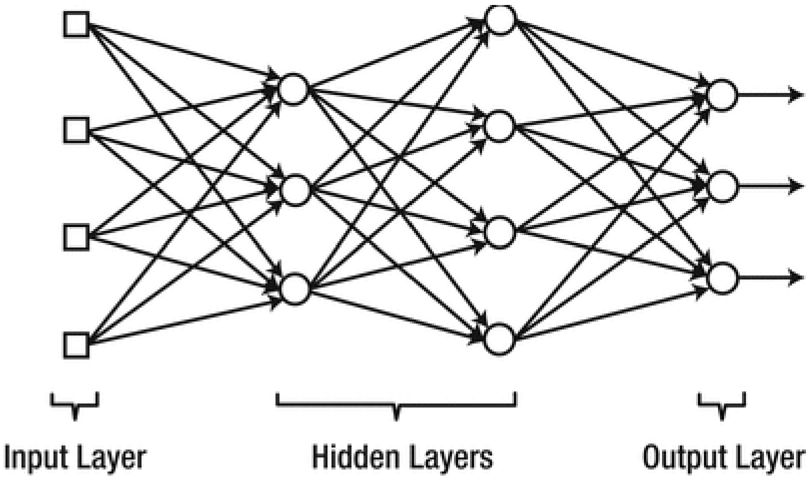

Figure 2 Neural Network model (Kim, 2017)

The input layer do not compute the “weighted sum and activation function “as it merely provides passage to “transfer input signals to the next nodes”. The middle layer is known as the hidden layers as it is not reachable from NN outer part. The output layer produce the concluding result of the system. Neural networks with “very simple architecture” that has only “input and output” are known as “single layer neural networks”. “Single layer neural networks” with added “hidden layers are called multi-layer neural network”. “Multi-layer neural network” that has only one hidden layer is called shallow/ vanilla neural network and when it has more than one hidden networks it is “called deep neural” networks. Most of the existing neural network applications are using deep neural networks. (Kim, 2017).

Neural network could be trained to learn the mapping that represents the required relation, by fine-tuning the weight values between the elements and consequently the internal structure is modified. One of Neural Networks benefit is to generalise the trained knowledge and offer accurate responses to new data, making it a more flexible and convenient tool of data modelling.

The common applications in neural networks are classification, clustering, data compression, feature extraction, function approximation prediction, nonlinear system modelling and control and etc (Liu et al., 2017).

Neural Networks have shown to outclassed conservative practices in solving complex problems. This is because of NN ability to extract the required information from the data when train especially when the correlation between the inputs and outputs are highly non-linear. The goal of neural networks is to bring the conventional computers to imitate the human brain mechanism.

For the prediction of data, data cleaning and processing needs to be made to ensure the data quality and it improves the productivity.

The following column are removed: '%Skor ','Input B','Jenis Pemeriksaan','Nama Pemeriksaan’ as it is redundant data which have no value or only have 1 value.

In data cleaning all the blank and null data are removed. In data processing the data are converted to numerical for prediction of data.

Classification- Python and Anaconda library Keras

Cross validation: Percentage split-67% training set/33% test set. This is important to assess the predictive performance of models to judge how the new sample perform outside the sample to a new data.

The following are the configuration of neural network in keras:

```
model = Sequential()
init = TruncatedNormal(stddev=0.01, seed=10)
model.add(Dense(units=50, input_dim=27, activation='relu',init =
TruncatedNormal(stddev=0.01, seed=10)))
model.add(Dense(units=4, activation='softmax', kernel_initializer=init))
adam = Adam(lr=0.007)
model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=['accuracy'])
estimator = KerasClassifier(build_fn=baseline_model, nb_epoch=200, batch_size=5,
verbose=0)
estimator.fit(X_train, Y_train, epochs=800, validation_data=(X_test, Y_test), shuffle=False,
verbose=0)
```

The Keras library provides wrapper classes provide the usage of neural network models developed with Keras in scikit-learn.

There is a KerasClassifier class in Keras that can be used as an Estimator in scikit-learn, the base type of model in the library. The KerasClassifier takes the name of a function as an argument. This function must return the constructed neural network model, ready for training.

The network topology of 1 layer neural network can be summarized as:

27 inputs -> [50 hidden nodes] -> 4 outputs

"softmax" activation function are used in the output layer. This is to ensure the output values are in the range of 0 and 1 and may be used as predicted probabilities.

Finally, the network uses the efficient Adam gradient descent optimization algorithm with a logarithmic loss function, which is called "categorical_crossentropy" in Keras.

The benefits of using Adam on non-convex optimization problems are as follows:

- Straightforward to implement.
- Computationally efficient.
- Little memory requirements.
- Invariant to diagonal rescale of the gradients.
- Well suited for problems that are large in terms of data and/or parameters.
- Appropriate for non-stationary objectives.
- Appropriate for problems with very noisy/or sparse gradients.
- Hyper-parameters have intuitive interpretation and typically require little tuning.

  (Brownlee, n.d.-a)

The arguments in the construction of the KerasClassifier class will be passed on to the fit() function internally used to train the neural network. The number passed are of epochs as 200 and batch size as 5 to use when training the model. Debugging is also turned off when training by setting verbose to 0.(Brownlee, n.d.-b) Then the model are fitted for 800 training epochs with the default batch size of 32 samples and performance of the model are evaluated at the end of each training epoch on the test dataset.

Results

Overall Accuracy : 91.36163982430455

confusion_matrix :

```
 [[267   7   0   3]
 [  7  31   0   0]
 [  0   0   8   0]
 [ 41   0   1 318]]
```

The following figure 3 and 4 are the model accuracy and model loss:

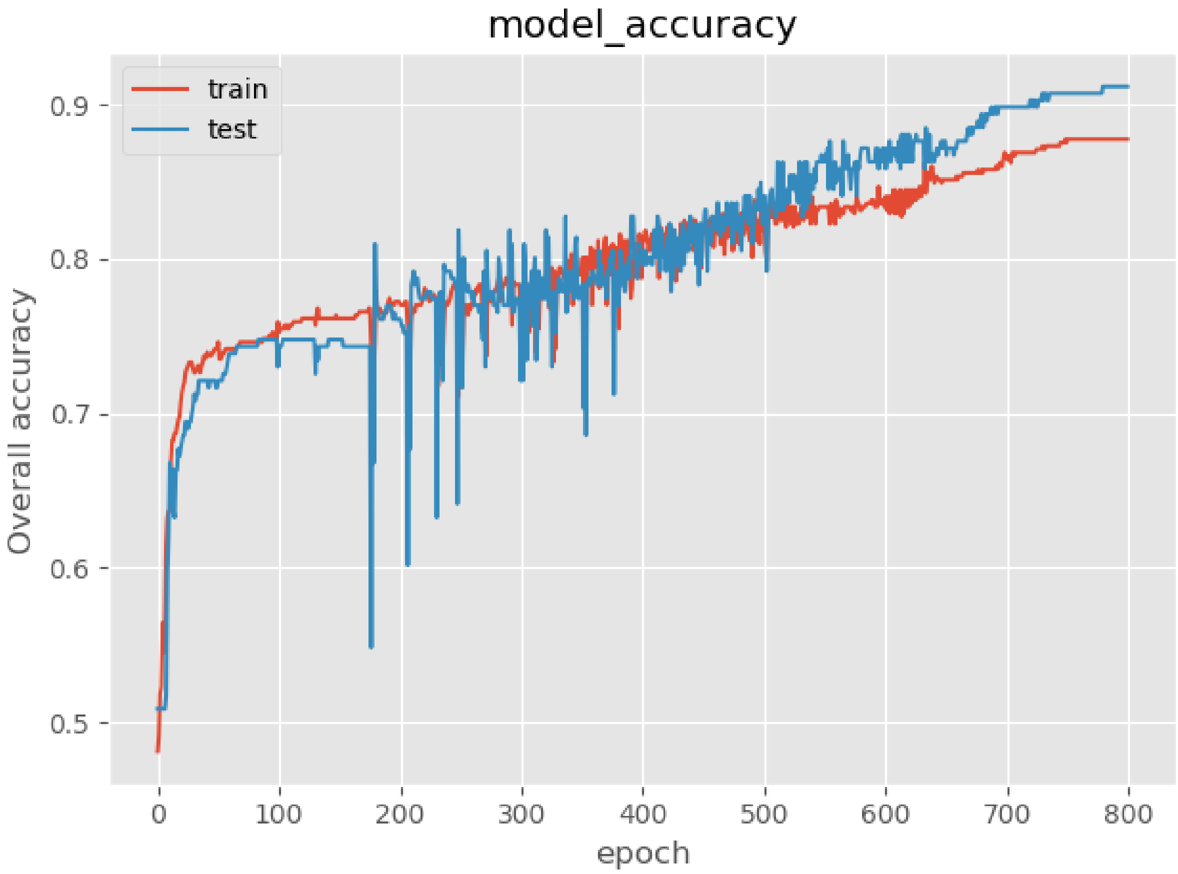

Figure 3: Model accuracy

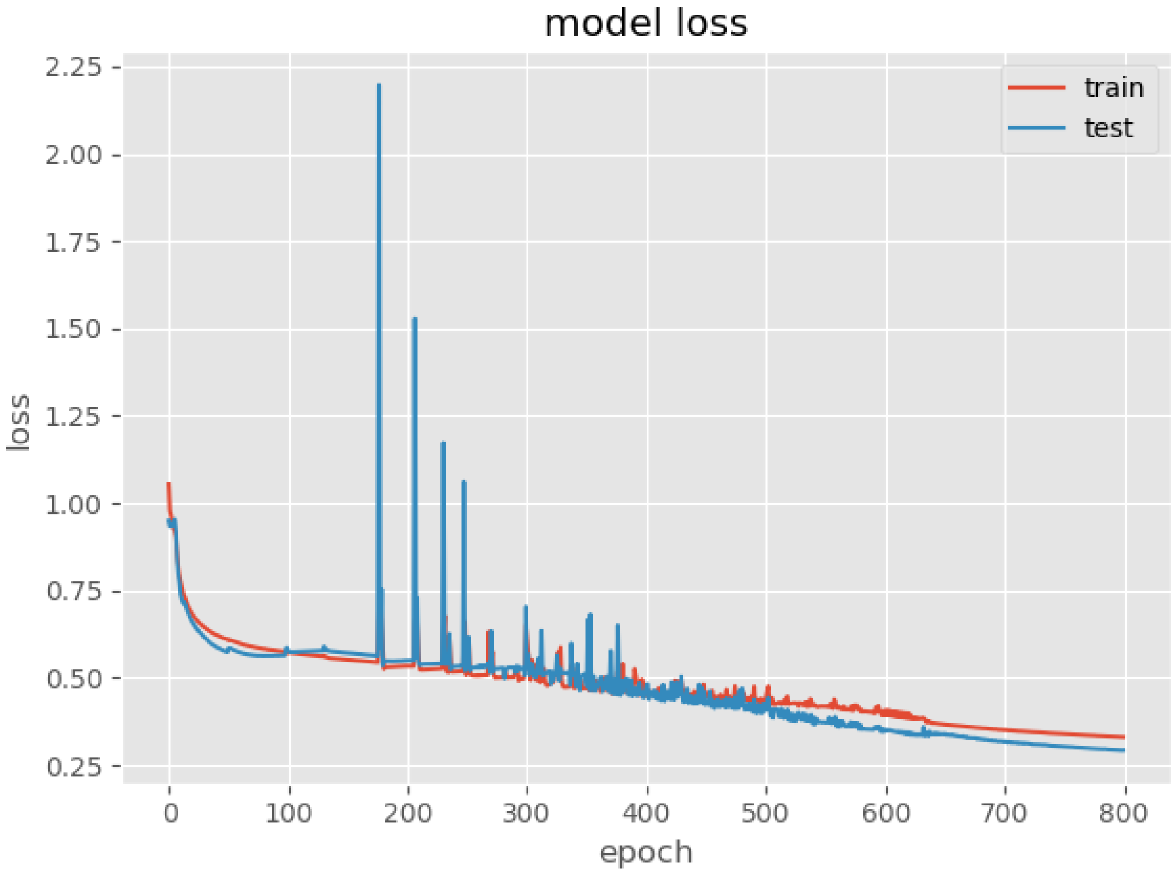

Figure 4: Model Loss

Prediction of data classification can be found from the Export folder of the source code.

Reference

Brownlee, J. (n.d.-a). Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. Retrieved December 20, 2019, from https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Brownlee, J. (n.d.-b). Multi-Class Classification Tutorial with the Keras Deep Learning Library. Retrieved December 20, 2019, from https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/

Kim, P. (2017). *MATLAB Deep Learning_ With Machine Learning, Neural Networks and Artificial Intelligence*. Apress.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*(October 2016), 11–26. https://doi.org/10.1016/j.neucom.2016.12.038