2019 International Conference on Innovative Computing (ICIC)

# Deep Neural Network a Step by Step Approach to Classify Credit Card Default Customer

1st Waseem Ahmad Chishti
*dept. school of information system*
*university of management and technology*
Lahore, Pakistan
f2017313022@umt.edu.pk

2nd Shahid Mahmood Awan
*dept. school of system and technology*
*university of management and technology*
Lahore, Pakistan
shahid.awan@umt.edu.pk

***Abstract*—This study and research aimed is to classify and predict the credit card default customers payment by means of contemporary approach of artificial neural network (ANN) known as deep neural network. This paper explains the dataset which signifies Taiwan credit card defaults in 2005 and their previous payment histories taken from popular machine learning dataset resource known as UCI. The paper enlightens each and every concept and step require to build, train, validate and test a deep neural network model for classification task that has never been discussed before. Moreover, we tried to elaborate the relevant and important concepts associated with deep neural network model that must be kept in mind during model building. This paper mainly tries to classify the default payment customer with more than 82% accuracy. For this purpose, various deep neural network techniques with different libraries are used to attain maximum accuracy and we have tried to build a best possible model which can be used for future prediction. This study proves deep neural network is the only one that can accurately estimate the real probability of default. So, by using this network model, which is more complex, sophisticated and most widely used than a simple neural network and logistic regression model, the classification simulation shall have a better performance and accuracy**.

***Keywords—Artificial Neural Network, Backpropagation, Classification, Deep Neural Network, Default Customer***

## I. INTRODUCTION

In this study, we used deep neural network model to classify the data taken from UCI machine learning repository named as Taiwan credit card default payment. We tried to cover every step and every concept required to build a comprehensive model. The dataset is related to financial areas where high risk is involved and deep neural networks are used to eliminate these risks. So, we used credit card default payment dataset to nicely predict the correct class of default customer by using step by step approach. The hypothesis for this research is ”How we can achieve high accuracy through classification of default credit card customer using artificial deep neural network?”. The rest of the paper structure is as per the following: Section II shows the writing survey or literature review. Section III explains the different important terms with the help of previous study. Section VI contains methods and techniques for our work of deep neural network. Section V portrays the exploratory setup used to implement the proposed method and its results for classification of default customers. Section VI discusses discussion and Section VII discusses the conclusion and the future scope of the study [23, 24].

## II. LITERATURE REVIEW

[1] This paper explains the classification and neural network very well. The author describes the generalization of the neural network model and explains other classification methods as well but this article contains only theoretical concepts.

[2] This article explains the dataset of default credit card payment with data mining and machine learning tool Rapid Miner.

[3] This article explains the dataset Taiwan credit card default payment with six classification methods and compares each method results with each other. The author concluded that artificial neural network performs well.

[4] This article explains the backpropagation algorithm and artificial neural network. According to this article back propagation is a three tier architecture while the artificial neural network is the best to solve the classification problem.

[5] This article predicts the insurance insolvency with the help of artificial neural network. Moreover, it elaborates the different terms associated with artificial neural network. The author tells about tuning parameters of artificial neural network.

## III. CONCEPTS ASSOCIATED WITH DEEP NEURAL NETWORK

### A. Classification

Classification is a popular method of making decision of human activities [23]. Classification is a technique that is used to group objects in different classes based on their characteristics. It deals with discrete data and predicts discrete or categorical class labels [8]. “A technique where we classify data into a certain number of classes and want to identify a category or class to which new data belongs to or will fall under is known as classification”[16]. Classification model is used for prediction of new class category for unseen or test data. Classification can be in three forms like binary classification, multi class classification, multi label classification. There are three different methods of classification first statistical method include naive Bayes, support vector machine, k-nearest

978-1-7281-4682-9/19/$31.00 ©2019 IEEE

neighbour (non-parametric) second, decision tree (rule-based models), random forest and third one is linking method known as artificial neural network now a days prevalent with the name of deep neural network is the advance form of neural network [4, 22]. It is most widespread classification method that intensifies the model accuracy [16]. Among all classification techniques deep neural network is the most popular in the fields of finance and health. Previous research also shows deep neural network performs better than any other classification method [4, 23].

### *B. Artificial Neural Network*

Artificial neural networks are erected or modelled by human brain and made up of large number of artificial neurons called processing elements [5, 25]. But in artificial neural networks, neurons compare to human's brain neurons have less roles and connections. Also, they are less in numbers than human brain neurons. However, working of both networks is the same. Like in artificial neural network the first step is to receive inputs from other neurons or perceptrons which are connected to the other end of network. After that an activation or transfer function is applied to these received inputs which leads to the activation level of the neurons which is actually the output value of the neuron (basically a neuron performs mapping from input to output) [26]. There are different activation functions that can be applied to create non linearity in neural network like step function, sigmoid function, linear function. In multilayer back propagation network sigmoid function is used and its activation value is between -1 and 1. Mathematically sigmoid function can be defined as: $y = 1/1+e^{-x}$ [25, 6]. Sigmoid function after differentiation can be written as: $dy/dx = y\ (1-y)$. There are various properties associated with sigmoid function like:

- The sigmoid function can be used to approximate a step function.
- It is monotonic.
- It is the solution of first order non-linear differential equation.

Sigmoid function is also known as logistic function and it looks like S-shaped [12]. Artificial neural networks are data preparing self-adaptive frameworks whose structure and functionality is spurred by the cognitive processes [23].

### *C. Deep Neural Network*

The network which consists of more than three layers of neurons i.e. input layer, hidden layers also called dense layers and output layer is known as deep neural network. Fundamentally, deep neural network is based on the idea of successive or stack of layers of representation, these layers define the depth of the model that how many layers contribute to build the model of the data [6]. DNNs are kind of feed forward network or feed forward multilayer perceptron (MLPs) where data flows from input layer (first stage) to output layer (final stage) [19]. Moreover, it is a multistage information purification process, where information goes through successive filters and comes out with gradually clean or some kind of magical solution of many tasks. To get the better solutions of complex problems neural networks have to come up with complex models. By complex model usually it is considered the additional number of hidden layers or more neurons in hidden layers between input and output layers so it

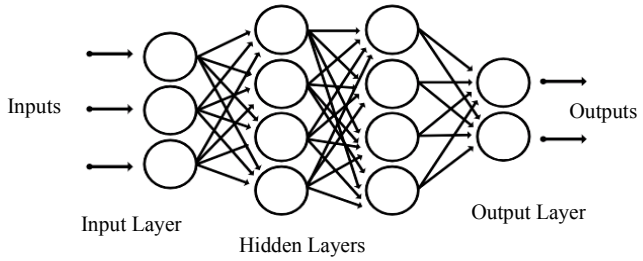


Fig. 1 Deep neural network graphical model [28]

can learn representations of data with multiple levels of abstraction that is why it is called deep neural network [20]. Although it will increase the computation power but will appear with an improved solution where the number of features in thousands or in hundred thousand [6, 5, 26, 27, 28].

### *D. Backpropagation*

Backpropagation was first appeared in 1970 but it is considered the workhorse of learning in neural network in today's environment" An algorithm of artificial neural network for supervised learning tasks using gradient descent for training multi-layer perceptron" [6]. A central Algorithm in deep neural network to develop a meaningful relationship among input and output through learning process. Back propagation is broadly used in multilayer perceptron for the adjustment of weights because in MLP neurons have weights associated with inputs, these weights are assigned by random values. So, when error is made a large number of weights need to be adjusted with training data examples. Below is a model of multilayer perceptron which consists of three layers first layer is input layer, second layer is hidden layer (a model may have more than one hidden layer) and last is output layer. [6, 28].

*1) Importance of backpropagation:* An algorithm that is used to find minimum value of error function by adjusting weights assigned randomly using gradient descent or delta rule. The aim behind backpropagation is to optimize and adjust the weights so that neural network can learn how to map input to output correctly. Three steps involved in backpropagation.

a) *Forward propagation*
*b) Backward propagation*
*c) By putting all values and calculating new updating weight values*

Forward Propagation or Forward Pass: In the forward propagation pass we apply input vector to the nodes of the

network. A set of output is produced during this process so, all weights (synaptic) of the network are fixed [7].

Back Propagation or Backward Pass: First of all, error is calculated (error in this case is of two types function signal and error signal) and propagated through the network after that synaptic weights are adjusted by following error correction rule.

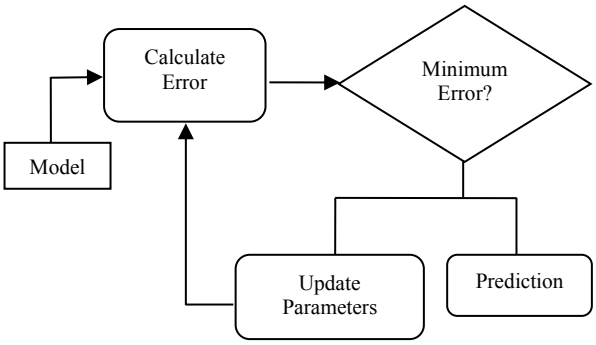


Fig. 2. Working of backpropagation

Function signal is an input signal and it propagates forward through the network whereas, error signal propagates backward through the network [25]. The first step in backpropagation is to train a model by adjusting or updating the weights of the model after adjustment of input's weight which is an Iterative procedure next step is to find the minimum error. The model is considered best where the error is minimum otherwise the procedure of updating parameters will continue till the minimum error. In short, the goal of backpropagation is to reach at global minima that is why its value should be realized [13].

## IV. METHODOLOGY

The dataset we used for our research is named as Taiwan credit card defaults in 2005 valuable dataset especially for banks. The dataset consists of 30,000 instances (observations) with 24 variables one is dependent and 23 are independent variables.

The dependent variable named as default payment next month (Yes = 1, No = 0). A deep neural network model is used to classify the data and all work is done using open source language and libraries such as Python, Keras, Tensorflow, data analysis and visualisation libraries matplot, seaborn, pandas, pandas profiling and numpy. All work will be available on our github repository https://github.com/waseemchishti. Detail of the dataset is given in the table 1.

## V. DEEP NEURAL NETWORK A STEP BY STEP APPROACH

To make the neural network ready for training, we need to choose four things.

- Input Data: The input data and their corresponding targets.
- Layers: Input layers, hidden or dense layers, output layers.
- Loss Function: A function that highlights and makes the network mature to measure its performance and how it can transform itself in the exact location. It is used to define the feedback signal used in the training procedure.

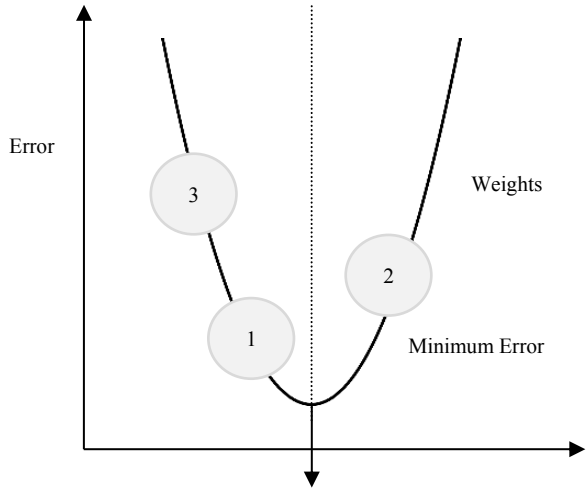


Fig. 3 Backpropagation Error Model

- Optimizer: An optimizer parameter determines how the network would be updated based on loss function used in the network [6].

### A. Model Specification

The model specification has the following steps:

*1) No of inputs:* How many inputs you want to give to your model or how many inputs that make sense for model training it could be selected after checking relationship. In our neural network model, we use total 23 input parameters.

*2) Model Type:* First one is Sequential (Each layer is connected with every other layer in the network directly) and second is Functional APIs (One input-multiple outputs, multiple inputs-multiple outputs). We used Sequential model as we have single type of input and single output in case of more than one input or output use keras Functional APIs).

*3) Dense Layer:* In dense layer each node in current layer is in contact with next layer's all nodes and is called densely connected or fully connected network [6].

### B. Implementation of Deep Neural Network

The Implementation of deep neural network comprises of several steps. We tried to illuminate each and every step required to implement deep neural network in a simple and comprehensive manner. Besides, these steps are not specific for one problem but general where every next step followed by previous. So, all these steps make a valuable contribution in order to build a best deep neural network model.

*1) Importing or reading data*: Importing or reading the dataset from working directory to whom you want to apply

classifier is the first step. Many methods to import or read the dataset pandas is an important library available in python that provides flexibility and lot of functionality to deal with data by using pandas "DataFrame" method.

*2) Type Checking:* The second step is to check the data type of all variables referred as features. If the data type differs then it needs to be converted. In our case all variables are of integer type and also dataset is in categorical variable we don't need to convert into categories like in Gender (male=1, female=2) in other case we can use "Label Encoder" library to do this job.

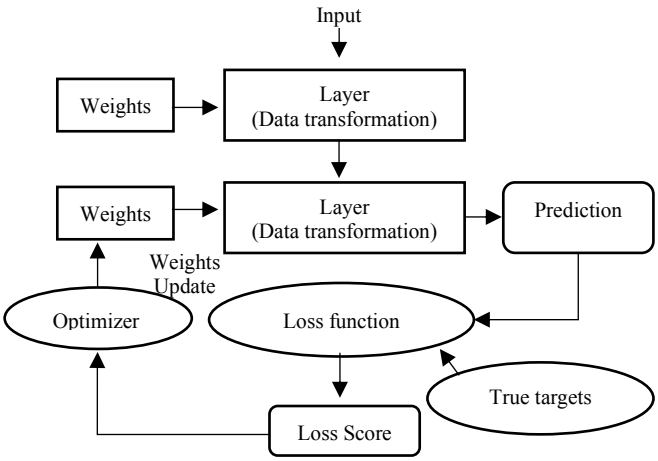


Fig. 4. Relationship between the network layers, loss function and optimizer

*3) Relationship Checking:* The third step is to check the relationship of variables by using correlation command. We select all variables such as independent variables and one dependent variable named as 'default payment next month'. All selected variables are shown through correlation matrix some variables are not so convinced in case of correlation and have weak correlation but still have importance in our dataset and in model training. "Matplot and seaborne" are python's libraries used to display correlation graph.

*4) Dummy Variable:* In the dataset the variable MARRIAGE and EDUCATION are converted into dummy variables before dummy variables there were 23 independent variables now extended up-to 28 variables such as 3 for (MARRIAGE) and 5 for (EDUCATION). We used pandas get dummies method to create dummy variables.

*5) Splitting the dataset:* Splitting dataset means dividing the dataset into training and testing ratio. We split the dataset for training and testing purpose into 75-25 ratio 75 for training and validation (the ratio between training and validation is 55-20) and 25 percent dataset used for testing the model. ScikitLearn python library is used for splitting data into training and testing purpose using "train_test_split" function.

*6) Transforming and Scaling Data:* Transforming and scaling of data is also necessary before going to implement model. It can be done on training data by using "Standard Scaler" method available in ScikitLearn. Actually, we have to normalize our data so we used "StandardScaler" method, it transforms values into similar range variable mean is 0 and variable variance is 1. We used identical fitted method to transform test data.

TABLE I. Detail of dependent variables

| Sr No | Variable | Detail |
|---|---|---|
| 1 | LIMIT BAL | Amount of the given credit (NT dollar), it includes both the individual consumer credit and his/her family (supplementary) credit |
| 2 | SEX | Gender (1 = male; 2 = female) |
| 3 | EDUCATION | Education (1 = graduate school; 2 = university; 3 = high school; 4 = others) |
| 4 | MARRIAGE | Marital status (1 =married; 2 = single; 3 = others) |
| 5 | AGE | Age (year) |
| 6 | PAY 0 | Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, 8= payment delay for eight months, 9=payment delay for nine months and above) |
| 7 | PAY 2 | Repayment status in August, 2005 (scale same as above in PAY 0) |
| 8 | PAY 3 | Repayment status in July, 2005 (scale same as above in PAY 0) |
| 9 | PAY 4 | Repayment status in June, 2005 (scale same as above in PAY 0) |
| 10 | PAY 5 | Repayment status in May, 2005 (scale same as above in PAY 0) |
| 11 | PAY 6 | Repayment status in April, 2005 (scale same as above in PAY 0) |
| 12 | BILL AMT1 | Amount of bill statement in September, 2005 (NT dollar) |
| 13 | BILL AMT2 | Amount of bill statement in August, 2005 (NT dollar) |
| 14 | BILL AMT3 | Amount of bill statement in July, 2005 (NT dollar) |
| 15 | BILL AMT4 | Amount of bill statement in June, 2005 (NT dollar) |
| 16 | BILL AMT5 | Amount of bill statement in May, 2005 (NT dollar) |
| 17 | BILL AMT6 | Amount of bill statement in April, 2005 (NT dollar) |
| 18 | PAY AMT1 | Amount of previous payment in September, 2005 (NT dollar) |
| 19 | PAY_AMT2 | Amount of previous payment in August, 2005 (NT dollar) |
| 20 | PAY_AMT3 | Amount of previous payment in July, 2005 (NT dollar) |
| 21 | PAY_AMT4 | Amount of previous payment in June, 2005 (NT dollar) |
| 22 | PAY_AMT5 | Amount of previous payment in May, 2005 (NT dollar) |
| 23 | PAY_AMT6 | Amount of previous payment in April, 2005 (NT dollar) |

*7) Initializing Neural Network:* This step involves initializing the neural network or importing the required libraries. This involves two important things first is "Sequential" module which is used to initialize neural network, second "Dense" module use to add hidden layers. Both Sequential and Dense modules can be imported from keras library.

*8) Model Initializing:* We gave the model name as 'Classifier' which is just a defining variable you can use according to your choice this initialization is made by using keras "Sequential" module.

*9) Adding Dense Layers:* Adding dense layers is also an important point as we know in deep neural network there should be more than two hidden layers but it is not mandatory that the more layers mean more accuracy some time we got accuracy with one or two hidden layers but if we have large set of observation and more parameters then more hidden layers should be added. In our first model we added three hidden layers with 28 nodes each by using "Dense" function available in keras. In our model the first parameter 'output_dim' is the number of nodes you want to add to this layer the second parameter is 'input_dim' used in the first layer to know how many input variables are, the third parameter is the 'kernel-_initializer' initialization of weights. In Neural Network we have to initialize the weights randomly by using uniform function that is the 'kernel_initializer' is equal to uniform. The second layer does't receive an input shape or 'input_dim' argument instead, it automatically inferred its input shape as being the output shape of the layer that came before. In our model, the total number of input variables are 28 including dummy variables but actual variables were 23. The fourth parameter is 'activation' a very significant concept in deep neural network. Neuron applies activation function in intermediate layers like biological neuron to activate neuron as the value close to 1 the chances of activation of neuron increases. The choice of activation function is also a little tricky. Of course, there are some solid reasons on the basis of these reasons activation function is selected for model. In our deep neural network model, we used rectifier (relu) function in hidden layers it maps input x to max (0,x) negative to 0 output and positive to same whereas in output layer we used sigmoid function due to binary classification problem it maps input between 0 and 1 value [6,5,26].

*10) Compilation step:* There are some parameters that govern the whole network of the model. The first argument is 'optimizer' (In step 8 we just initialized weights now we are applying some sort of algorithm which will optimize weights that will lead to making our neural network more powerful. This algorithm is known as gradient descent, several variants of gradient descent such as batch gradient descent, mini batch and stochastic gradient descent [30]. In our model we used (SGD) algorithm it is called stochastic because it picks one random unit at a time. SGD algorithm has several types of which we used 'adam'. Second argument is 'loss' (loss function), because the dependent variable 'default payment next month' is binary categorical variable so, we used logarithmic loss function named as 'binary_crossentropy' while in case of more than 2 categories of dependent variable then 'categorical crossentropy' will be used. Third argument is 'accuracy' we used metrics as accuracy to enhance the performance of our neural network [6, 7, 5, 26].

*11) Model Training:* For training the model we used fit method 'fit' method contains several parameters each of which has importance and cannot be excluded. The first parameter is training data given as 'X_train' second corresponding training target examples 'y_train'. Third parameter is 'batch_size' which is used to update the weights after certain observations we can use by observing the total input examples and can update it, depends on model data examples. Fourth parameter is epochs which is the number to how many times you want to compile the model sometime more epochs leads to over fitting the model but there is no specific rule to select epochs as well as batch size. Choosing the value of batch size and epochs is trial and error. We fit the model by using different epochs and batch. We used in first model 120, 30, in second model 40, 120, in third model 30, 250 and in fourth model 80,40 respectively. Remember that fitting the model not limited to these four models we built almost 30 models with different batch size and epochs because the purpose was to create the overfitting in the model so that different models are fitted to depicts and highlight the over fitting concept in deep neural network. The model below is fitted with 30 batch size and 250 epochs. Out of 22500 training examples we used 20 percent for validation in the diagram named as test upper right corner of the diagram. Model architecture and results with training and test data. Model architecture can be shown using 'summary' function available in keras 'Sequential' module. The detail of the model already described in step 10 above.

Training Data Confusion Matrix:

| | | Actual Values | |
|---|---|---|---|
| | | 1 | 0 |
| Predicted Values | 1 | TP [[16715 | 776] FP |
| | 0 | FN [2802 | 2207]] TN |

Accuracy Score: 0.8409777778

Classification Report:

TABLE II. TESTING DATA REPORT

| | **precision** | **recall** | **f1-score** | **support** |
|---|---|---|---|---|
| 0 | 0.86 | 0.96 | 0.90 | 17491 |
| 1 | 0.74 | 0.44 | 0.55 | 5009 |

Testing Data Confusion Matrix:

[[5453 420]
[1087 540]]

Accuracy Score: 0.7990666667

Classification Report:

TABLE III. TESTING DATA REPORT

| | **precision** | **recall** | **f1-score** | **support** |
|---|---|---|---|---|
| 0 | 0.83 | 0.93 | 0.88 | 5873 |
| 1 | 0.56 | 0.33 | 0.42 | 1627 |

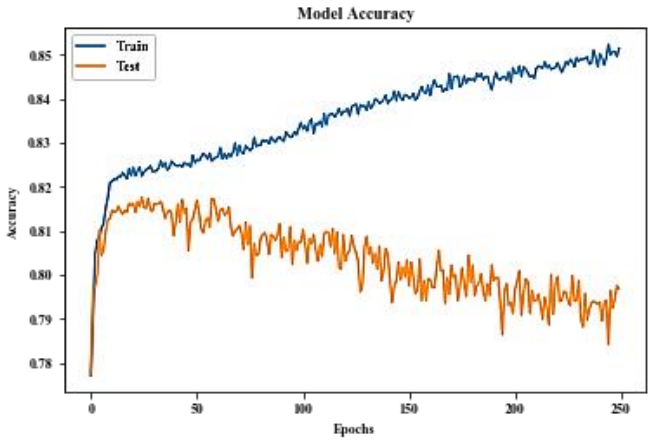


Fig. 5. Model accuracy graph with overfitting

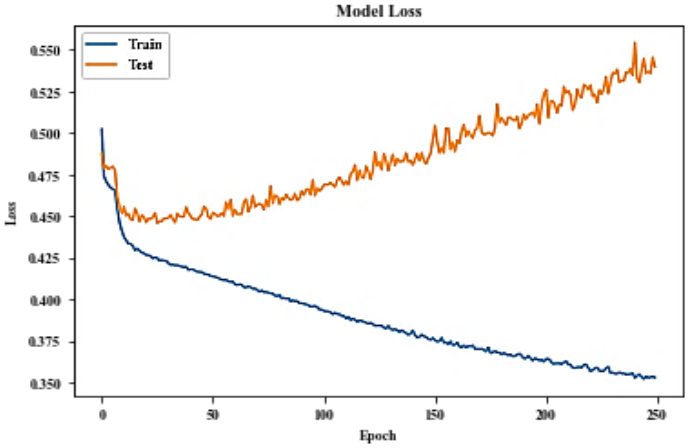


Fig 6. Model loss graph with overfitting

*1) Overfitting the Model:* Overfitting is the major issue, which occurs in almost every machine learning problem and leads to fall the generalization ability of the model [29]. How to deal with overfitting defines master of machine learning. We can understand the model's overfitting in two diagrams where the model accuracy is decreasing in first diagram and loss is decreasing in the second diagram on validation data. First diagram and loss is decreasing in the second diagram on validation data.Two concepts occur in over fitting optimization and generalization. If the model performs good at training data, it is called optimization and if it also performs well at testing data then it is known as generalization [23]. If a model's performance degrades on testing data and memorize the dataset then it is called overfitting and we can say that the model is overfit, it must be overcome by using different techniques or adjusting hyper-parameters [6, 5, 29]. The model overall accuracy score is 84.14 calculated by using 'evaluate method' with training examples 'X_train' and training target 'y_train'. Confusion matrix shows training accuracy is 84.09 while on testing data it gives accuracy up-to 0.7990 but it is not impressive. There is huge difference between training and validation accuracy and loss, this is the example of overfitting.

*2) Fighting against Overfitting:* To overcome overfitting many methods and mechanisms are used depends on which neural network you are performing, Sequential Neural Network, Convolution Neural Network, Recurrent Neural Network etc, We used three methods one by one and pair of both batch normalization and dropout to overcome the overfitting but the dropout method performed better than others. It is most popular used technique of regularization to fight against overfitting. Actually, there are many ways to overcome the overfitting dependings upon the model architecture and nature of model. One simple way is to decrease the network size by changing the epochs and batch size but it cannot work if you have large dataset. In that case, we used different methods and ways but here putting only one which is more valuable and efficient way to fight against overfitting. We added dropout layer after first , second and third hidden layer with0.5 dropout fraction rate also, we checked the rate from 0.2 to 0.5 and found that 0.5 rate is good for our model now the results are much different than the previous results and quite satisfactory. The graphical details of the results are shown in diagram 7 and 8 [6, 7, 5, 29].

*3) Prediction:* The prediction is done by using threshold value. We used 0.5 above this or equal to this value will be converted to 1 and below this value will be converted to 0. This prediction result is the probability of default customer payment next month converted to binary numbers 0 and 1. This is also done in step 11 above. Model architecture and results with training and test data is given below:

Architecture of the deep neural network model

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense_1 (Dense) | (None, 28) | 812 |
| dropout_1 (Dropout) | (None, 28) | 0 |
| dense_2 (Dense) | (None, 28) | 812 |
| dropout_2 (Dropout) | (None, 28) | 0 |
| dense_3 (Dense) | (None, 28) | 812 |
| dropout_3 (Dropout) | (None, 28) | 0 |
| dense_5 (Dense) | (None, 1) | 29 |

Total params: 2,465
Trainable params: 2,465
Non-trainable params: 0

Both training and testing data confusion matrix results are given below.

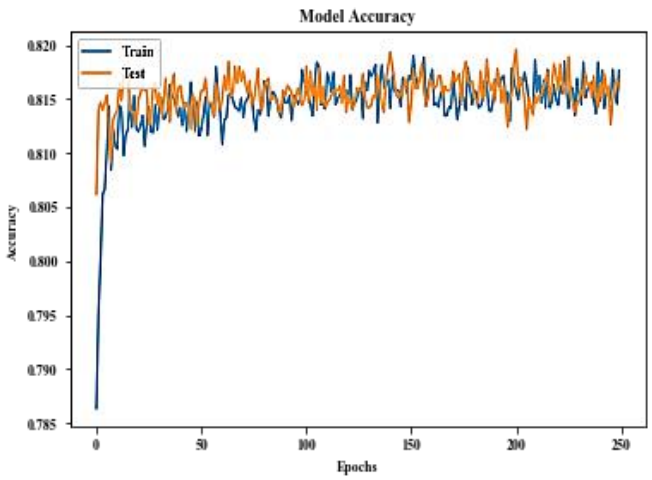


Fig.7. Model accuracy graph without overfitting

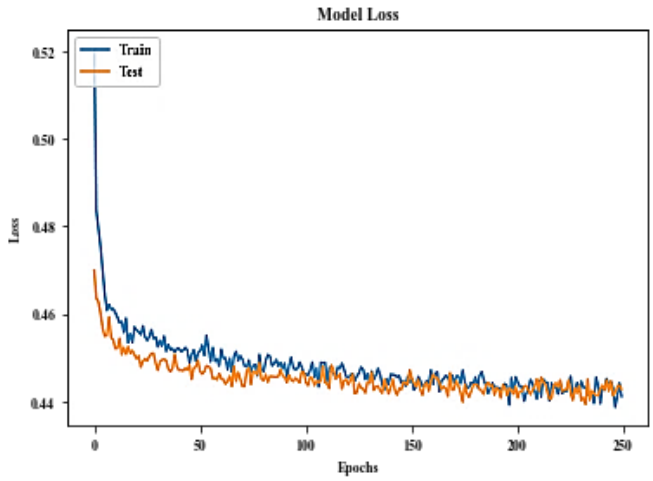


Fig. 8. Model loss graph without overfitting

Training Data Confusion Matrix:

[[16796 695]
[3270 1739]]

Accuracy Score: 0.8237777777777778

Classification Report:

TABLE IV. TRAINING DATA REPORT

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.84 | 0.96 | 0.89 | 17491 |
| 1 | 0.71 | 0.35 | 0.47 | 5009 |

*4) Model Visualisation:* Model visualization is also an important step especially when you want to keep update with the model's performance and working of the model behind the scene. Tensorflow provides a very useful library and platform for visualisation of model's training named as tensorboard. To view the working of the model, updation of loss and accuracy, there are few steps provided by tensorflow that can be used with python.

a) *Define directory of dataset:* By using python, working directory can be defined as 'mkdir (working directory name)'
b) *Callback function:* Callback function has certain parameters. The choice of using these parameters depend on which type of model we are going to use or we have used.
c) *Model Visualization:* To View the entire working of the model. We need a command 'tensorboard - - logdir (working directory name)'.

*5) Model Evolution:* In the last step performance of the model is evaluated. We checked the accuracy of the model by building confusion matrix [9].

Testing Data Confusion Matrix:

[[5620 253]
[1110 517]]

Accuracy Score: 0.8182666666666667

Classification Report:

TABLE V. TRAINING DATA REPORT

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.84 | 0.96 | 0.89 | 5873 |
| 1 | 0.67 | 0.32 | 0.43 | 1627 |

Four concepts are involved in confusion matrix and two types' actual and predicted results are obtained such as True Positive, True Negatives, False Positives, and False Negatives. TP (True Positive) those tuples/records/rows that are correctly mapped by the machine/classifier.TN (True Negatives) those tuples/records/rows that are correctly mapped by the machine/ classifier and these were actually negatives and classifier did the same job. FP (False Positives) refer to positives tuples that were incorrectly mapped by the classifier as positives but these were actually negatives. FN (False Negatives) are those tuples that are incorrectly mapped by the classifier as negatives but actually these were positives [8]. Two classification reports are shown above of the models.

## VI. RESULTS

Training data:
Accuracy: (TP + TN) / (P+N) (16796 + 1739) / 22500 =0.8238 nearly 82.5%. Specificity: True Negative recognition rate TN/N 1739/2434 = 0.7145 nearly 72%. Sensitivity: True Positive recognition rate TP/ (TP+FN) 16796/17491 = 0.9602 nearly 84%. Recall: fullness – What percentage that the classifier labeled as positive of positive tuples? TP / (TP + FN) 16796 / (16796+695) = 0.9602 above 96%. Precision: correctness – what percentage of tuples are actually positive that the classifier labeled as positive. TP / (TP + FP) 16796 / (16796 + 3270) =

0.8370 nearly 84%. Error Rate: (FP + FN) /All or 1- accuracy 0.1762. F-Score: The harmonic mean of precision and recall and it can be calculated as F=2* precision *recall /precision + recall F=2*0.8372*0.9602/0.8372+0.9602 =0.8944 [7, 8].

Testing Data:
Accuracy: (TP + TN)/All (5620 + 517) / 7500 =0.8182 nearly 82%. Specificity: True Negative recognition rate TN/N 517/770 = 0.6714. Sensitivity: True Positive recognition rate TP/ (TP+FN) 5620/5873 = 0.9569 nearly 84%. Recall: fullness – What percentage that the classifier labeled as positive of positive tuples? TP / (TP + FN) 5620 / (5620+253) =0.9569 about 96%. Precision: correctness – what percentage of tuples are actually positive that the classifier labeled as positive TP / TP + FP 5620 / (5620 + 1110) = 0.8350 nearly 84%. Error Rate: (FP + FN) / All or 1- accuracy 0.1817. F-Score: The harmonic ean of precision and recall and it can be calculated as F=2* precision * recall / precision + recall F=2*0.8350*0.9569/0.8350+0.9569=0.8918 [7, 8].

## VII. DISCUSSION

In this work we explained deep neural network model and Taiwan credit card default customer's dataset. Previous work does not explain deep neural network model and the overfitting issue during training the model in this way. We tried to explain the different regularization methods which can be used to generalize a better and reliable model [29]. Moreover, a complete step by step approach is implemented. We proved that by using deep neural network, we can classify default credit card customers and get the accuracy around 82-83.This study prevails by following the procedure, we can make a comprehensive decision making model of deep learning.

## CONCLUSIONS

The deep neural network is very popular now a days. It is being implementing in different fields for prediction and classification. In this paper we described deep neural network (DNN) a step by step approach to classify customers default payment in a complete and accurate manner. We achieved better results than previous research. We mentioned all steps that are required to build best deep neural network model with the help of Taiwan credit card default customer's dataset and proved our hypothesis very true.

## REFERENCES


[1] Guoqiang Peter Zhang. "Neural networks for classification: a survey", ieee transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 30, no. 4, November 2000.
[2] W.S.Kanmani, B.Jayapradha, "Prediction of default customer in banking sector using artificial neural network", IJRITCC, vol. 5, pp. 293 – 296, July 2017.
[3] Cheng Yeh, Che-hui Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients", ACM Expert Systems with Applications, vol. 36, pp. 2473–2480, March 2009.
[4] Xiao-feng gu1, lin liu1, jian-ping li1, yuan-yuan huang1, jie bin1. "Data classification based on artificial neural networks", 2008, IEEE, p. 223-226.
[5] Ben Coppin, Artificial intelligence illuminated, ISBN 0-7637-3230-3.
[6] Franc¸ois chollet, Deep Learning with Python, ISBN 9781617294433.
[7] Daniel Jurafsky, James H. Martin, Speech and Language Processing, 2017.
[8] Jiawei Han, Micheline Kamber, Jian Pei, Data mining concepts and techniques, Elsevier, ISBN 978 0-12-381479-1.
[9] Pushkar Mandot. (2017) pushkarmandot homepage on MEDIUM [Online]. Available: https://medium.com/@pushkarmandot/build-your firstdeep-learning-neural-network-model-using-keras-inpythona90b586411 6d.
[10] (2002) The IBM cognitiveclass.ai website. [Online]. Available: https://cognitiveclass.ai/courses/data-analysispython/
[11] John McGonagle, George Shaikouski, Christopher Williams, and two others contributed. (2018).
Backpropagation on Brillient.org. [Online]. Available: https://brilliant.org/wiki/backpropagation/.
[12] Saurabh. (2018) Backpropagation on edureka.co. [Online]. Available: https://www.edureka.co/blog/backpropagation/.
[13] Matt Mazur. (2017) A Step by Step Backpropagation on mattmazur.com. [Online]. Available: https://mattmazur.com/2015/03/17/a-step-by-stepbackpropagation- example/.
[14] Luca Basanisi. (2018) Credit Card Default: a very pedagogical notebook on Kaggle.com. [Online]. Available: https://www.kaggle.com/lucabasa/credit-card-default-avery- pedagogical-notebook/notebook/.
[15] Drig˘a, I. and Dura, C, "The financial sector and the role of banks in economic development in 6th International Multidisciplinary Symposium" Universitaria SIMPRO, pp. 1-6. 2014.
[16] Rohit Garg. (2018) 7 types of classification algorithm on analyticsindiamag.com [Online]. Available: https://analyticsindiamag.com/7-typesclassification- algorithms/.
[17] Abhishek Srivastava, "Product and service innovation in Retail banking industry", thesis, Jagran Institute of Management (Jim), India, 2008.
[18] (2015) The Stack Exchange website. [Online]. Available: https://stats.stackexchange.com.
[19] (2018) The wikipedia website. [Online]. Available: https://en.wikipedia.org.
[20] Afaan Bilal. (2018) Artificial Neural Networks and Deep Learning on https://becominghuman.ai/ [Online]. Available: https://becominghuman.ai/artificialneural-networks-and-deep-learning-a3c9136f2137.
[21] Jason Brownlee j. (2016) how to implement the backpropagation algorithm from scratch in python [Online]. Available: https://machinelearningmastery.com/implementbackpropagation-algorithm - scratch-python/.
[22] Stephan Dreiseitl, Lucila Ohno-Machado. "Logistic
regression and artificial neural network classification models: a methodology review", Journal of Biomedical Informatics, 35 (2002) 352–359, February 2003.
[24] George E. Dahl, Tara N. Sainathy, Geoffrey E. Hinton. "Improving deep neural networks for lvcsr using rectified linear units and dropout", IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.
[25] A. Ibiwoye, O. O. E. Ajibola, A. B. Sogunro. "Artificial neural network model for predicting insurance insolvency", University of Lagos, Lagos, Nigeria 2012.
[26] Stephen Moore, Deep Learning for Computer Vision, ISBN 978-1-78829-562-8.
[27] Yann LeCun, Yoshua Bengio, Geoffrey Hinton "Deep learning", doi: 10.1038 / nature14539.
[28] What is deep learning? 3 things you need to know on https://www.mathworks.com/ [Online]. Available: https://www.mathworks.com/discovery/deep learning.html.
[29] Adam P. Piotrowski, Jarosław J, Napiorkowski. "A Comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling", Journal of Hydrology 476 (2013) 97–111, October 2012.
[30] Sebastian Ruder. "An overview of gradient descent optimization algorithms", June 2017. [19] (2018) The wikipedia website. [Online]. Available: https://en.wikipedia.org.