Scientific Research Publishing

Journal of Computer and Communications, 2019, 7, 44-54
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Deep Learning Approach for Customer Records Prediction for Future Products Issues

Sriramakrishnan Chandrasekaran

KTech, Regional Delivery Center, KPMG LLP, Montvale, NJ, USA
Email: sriramakrishnanchand@kpmg.com

How to cite this paper: Chandrasekaran, S. (2019) Deep Learning Approach for Customer Records Prediction for Future Products Issues. *Journal of Computer and Communications*, **7**, 44-54.
https://doi.org/10.4236/jcc.2019.73005

Received: January 9, 2019
Accepted: March 25, 2019
Published: March 28, 2019

Abstract

Artificial neural networks consist of the mechanisms of identifying the solution for the problem with the layers design and implementation, and the hidden layers will have rules to be done by the model. The model consists of different approaches, and based on the priority and the requirement, we need to create a model, and this article is dealing with the prediction related to bank customers and their loan processing, etc. In this article, we have a global dataset collected related to 10000 customers of a single bank and their account and other details related to the customers of the bank. Here, we are implementing using tensor flow and Keras libraries to create an artificial neural network which will work on the model and hidden layers. The hidden layers are the most import part of the presentation, and the virtual environment in the field can be helpful for the better prediction of the related things. Machine learning implementations with the combination of deep learning artificial neural networks and also with tensor flow and Keras will be the most exciting and attractive portion of the research work in any field of science and technology. This architecture and application will help to predict future bank applications, and this can be helpful for the customers to understand their level of applications and products usage based on their account weight.

Keywords

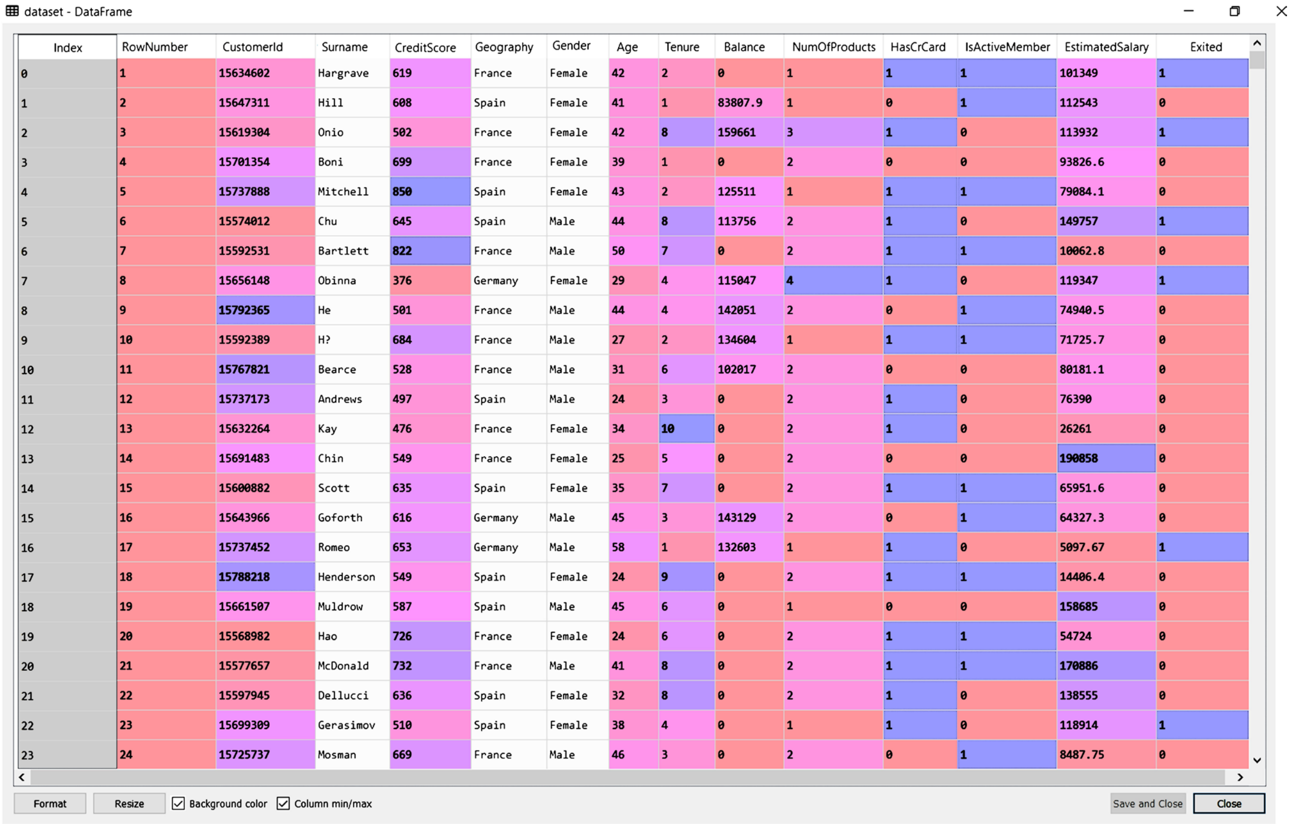
Machine Learning, Neural Networks, Tensor Flow, Keras, Virtual Environment, Hidden Layers

1. Introduction

Machine learning poses a unique role in science and technology and remains us that we have to achieve lot more in the technology development which makes the life simpler day to day. In this article, we are concentrating on the data set of a bank customer which is dealing with the number of products the customer handles from that bank and whether he is capable of being given the future products or services based on the necessary information we store related to the customers. That information is related to the age, gender and salary [1]. There are a number of independent variables which focus on the implementation of the customer related things in machine learning which will help them to predict something related to the customer information. Even this kind of machine learning and deep learning applications must be implemented in real life applications to make simpler life of the human [2].

In this application, we are considering churn analysis of the bank users details, and we create a few interesting things in this scenario. Here, we need to encode the categorical variables. There are few numbers of categorical variables in this dataset. As mentioned in Figure 1, this represents the structure of the dataset. In this, we have to implement something new life categorizing the gender, country, etc. Which are related to the string data types? The string types are considered as objects in the basic installation or importing the dataset, and these have to convert from unsupported format to supported format. Next, we need to build a network based on the data we have. In further sections, we will let you understand the complete structure of the project [3] [4] [5].

Deep learning mechanism in different streams performed different operations and we also tried to implement deep learning mechanism CNN, and we



Index	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101349	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.9	1	0	1	112543	0
2	3	15619304	Onio	502	France	Female	42	8	159661	3	1	0	113932	1
3	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.6	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125511	1	1	1	79084.1	0
5	6	15574012	Chu	645	Spain	Male	44	8	113756	2	1	0	149757	1
6	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
7	8	15656148	Obinna	376	Germany	Female	29	4	115047	4	1	0	119347	1
8	9	15792365	He	501	France	Male	44	4	142051	2	0	1	74940.5	0
9	10	15592389	H?	684	France	Male	27	2	134604	1	1	1	71725.7	0
10	11	15767821	Bearce	528	France	Male	31	6	102017	2	0	0	80181.1	0
11	12	15737173	Andrews	497	Spain	Male	24	3	0	2	1	0	76390	0
12	13	15632264	Kay	476	France	Female	34	10	0	2	1	0	26261	0
13	14	15691483	Chin	549	France	Female	25	5	0	2	0	0	190858	0
14	15	15600882	Scott	635	Spain	Female	35	7	0	2	1	1	65951.6	0
15	16	15643966	Goforth	616	Germany	Male	45	3	143129	2	0	1	64327.3	0
16	17	15737452	Romeo	653	Germany	Male	58	1	132603	1	1	0	5097.67	1
17	18	15788218	Henderson	549	Spain	Female	24	9	0	2	1	1	14406.4	0
18	19	15661507	Muldrow	587	Spain	Male	45	6	0	1	0	0	158685	0
19	20	15568982	Hao	726	France	Female	24	6	0	2	1	1	54724	0
20	21	15577657	McDonald	732	France	Male	41	8	0	2	1	1	170886	0
21	22	15597945	Dellucci	636	Spain	Female	32	8	0	2	1	0	138555	0
22	23	15699309	Gerasimov	510	Spain	Female	38	4	0	1	1	0	118914	1
23	24	15725737	Mosman	669	France	Male	46	3	0	2	0	1	8487.75	0

Figure 1. Dataset of the churn analysis.

tried to compare both ANN and CNN mechanisms. There are many other cases which are available to achieve CNN. The feature extraction mechanism will identify the strong features for the model and as mentioned above, the difference between the Y and Y^ has to be minimum to implement the CNN mechanism. There must be a valid hypothesis for identifying whether the model which we are writing is showing the factual accuracy or not. Accuracy between ANN and CNN will be compared in later articles, and we identified similar accuracy rate among different algorithms [6] [7]. Till now the existing architectures on the backing system didn't focused on churn analysis as it is in use of discussions. There is a possibility of implementing hybrid analysis for the banking system, and now in this approaches we are implementing using deep learning and implement some proper and optimal rules on the model and assign some of the proper inputs from the user and based on that we can predict the specific person can get the future products or implementations of bank based on his or her performance through the stages of transactions and operations made in the back.

Next section will explain the libraries and architecture of libraries in creating neural networks, later the proposed flow of the structure, next with sample output and conclude with the last section with a future scope as also.

The existing mechanism considers the classification and regression mechanisms to implement the feature extraction and implementations. In this feature extraction, researchers will achieve forward elimination as the underlying methodology to identify the essential features on which we need to implement the model. The feature extraction will estimate the P and SL value on which we need to determine the low P-value features to perform the best model with the compelling features [8] [9].

2. Working Procedure

The effective procedure consists of the explanation of the architecture with libraries included in this implementation. The libraries involved in these are as follows:

1) Pandas

2) Keras

3) Matplotlib

4) Numpy

5) Tensorflow

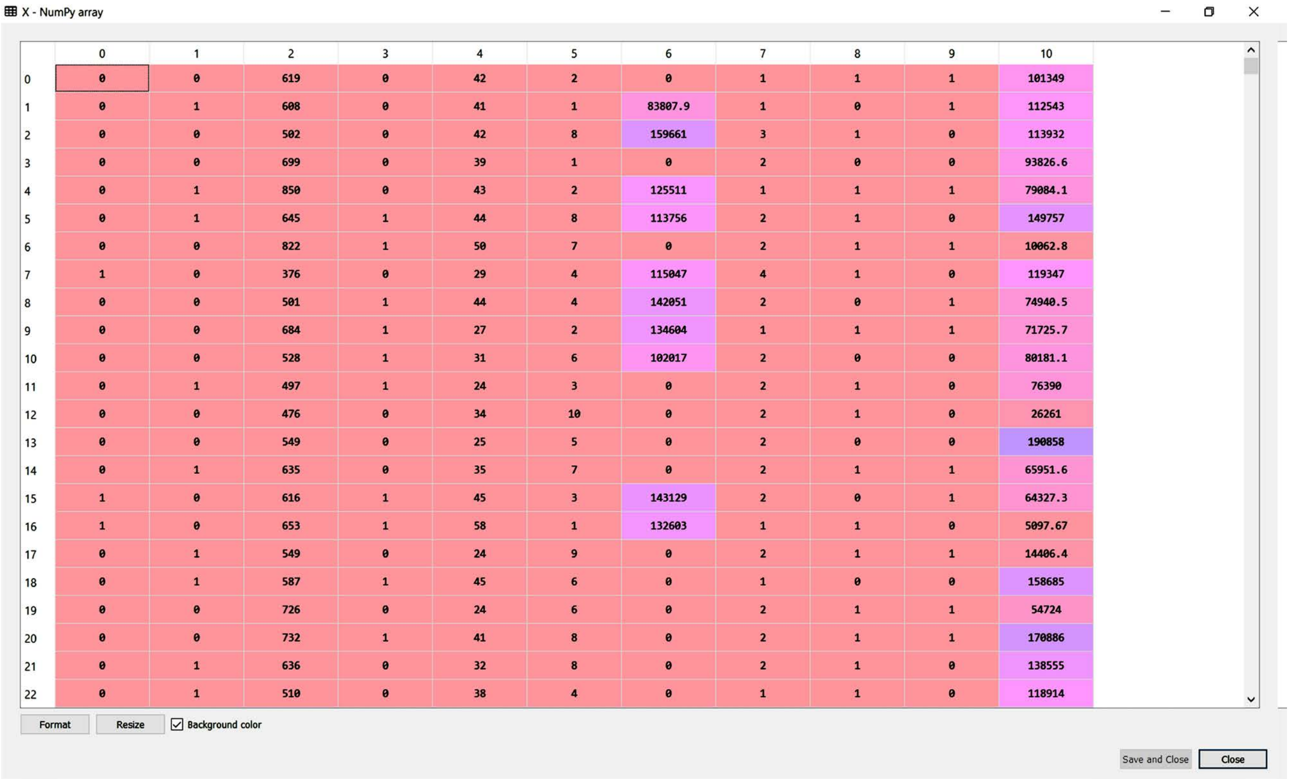
Based on the categorical variables identified we converted the strings to the basic categories and they are mentioned in **Figure 2** below. The categorical variables are then converted from string s to the numerical. If there are two categorical variables, we need to include only one for the code to avoid the definite loop [10] [11] [12].

Image 4 will explain the architecture of Keras considered for this application development. In this scenario for creating the hidden layers, we are implementing this current project. This article will discuss the different libraries like tensorflow, numpy, and pandas. Matplot lib is used for plotting. **Figure 4** will show the combination of tensorflow and Keras [13] [14] [15] [16].

The below image will explain the architecture of Keras library. This contains basic things related to Keras featuring input shape; hidden layers shape with LSTM, output shape. The difference between the expected output value and the actual obtained value will state that the model is accurate or not.

Figure 3 explains the conceptual architecture of this article with the necessary additions of the other libraries. NLTK a natural language toolkit is the additional library which can be included for converting some of the native languages to digital texts. Theano is also considered, and this can be covered with Kears API. Keras API will work on the top of the layer next layer consists of TensorFlow, Theano and CNTK, the final Layer consists of TensorFlow Workflow additional libraries. Figure 3 will give you a clear idea [17] [18] [19].

Consider the Hidden layers count of 500. This much layer is to apply different rules to different applicational data. For suppose, we can identify the final exists variable based on salary and age as one combination, have a credit card or not as other scenario and remaining as single scenarios, etc. For every rule and the



	0	1	2	3	4	5	6	7	8	9	10
0	0	0	619	0	42	2	0	1	1	1	101349
1	0	1	608	0	41	1	83807.9	1	0	1	112543
2	0	0	502	0	42	8	159661	3	1	0	113932
3	0	0	699	0	39	1	0	2	0	0	93826.6
4	0	1	850	0	43	2	125511	1	1	1	79084.1
5	0	1	645	1	44	8	113756	2	1	0	149757
6	0	0	822	1	50	7	0	2	1	1	10062.8
7	1	0	376	0	29	4	115047	4	1	0	119347
8	0	0	501	1	44	4	142051	2	0	1	74940.5
9	0	0	684	1	27	2	134604	1	1	1	71725.7
10	0	0	528	1	31	6	102017	2	0	0	80181.1
11	0	1	497	1	24	3	0	2	1	0	76390
12	0	0	476	0	34	10	0	2	1	0	26261
13	0	0	549	0	25	5	0	2	0	0	190858
14	0	1	635	0	35	7	0	2	1	1	65951.6
15	1	0	616	1	45	3	143129	2	0	1	64327.3
16	1	0	653	1	58	1	132603	1	1	0	5097.67
17	0	1	549	0	24	9	0	2	1	1	14406.4
18	0	1	587	1	45	6	0	1	0	0	158685
19	0	0	726	0	24	6	0	2	1	1	54724
20	0	0	732	1	41	8	0	2	1	1	170886
21	0	1	636	0	32	8	0	2	1	0	138555
22	0	1	510	0	38	4	0	1	1	0	118914

Figure 2. Categorical variables.

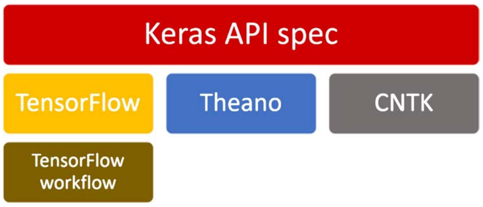


Figure 3. Architecture of the project.

combination, we impose on the model we need to maintain a hidden layer to map the related things in the model.

X represents input layers, Y Represents output and Y^ represents obtained output. The difference between the expected output and the obtained output defines the accuracy of the algorithm [20] [21]. Here **Figure 4** explains Keras library architecture and **Figure 5** explains TensorFlow architecture.

$$\text{The Accuracy of algorithm} = \text{Differece}\left(Y, Y\,^\wedge\right)$$

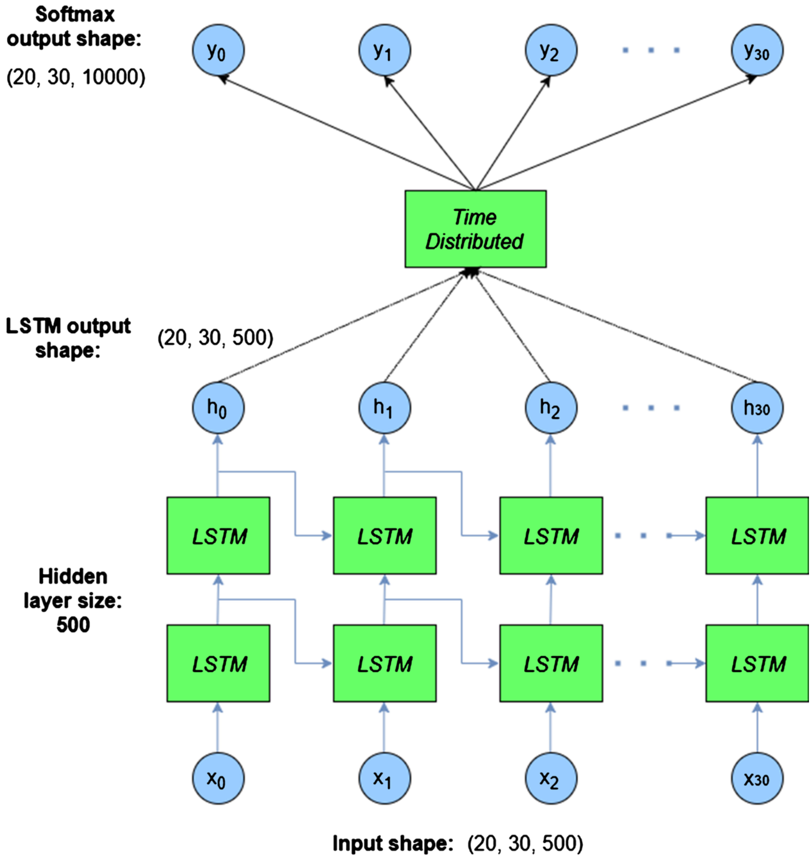


Figure 4. Keras architecture.

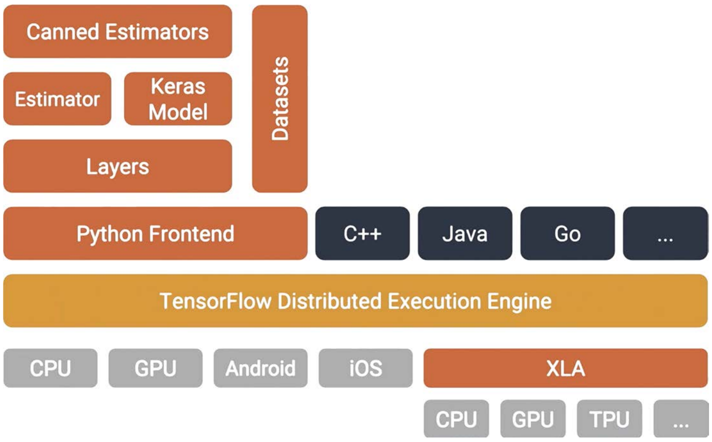


Figure 5. TensorFlow architecture.

Tensorflow architecture consists of the estimators which help for hidden layers identification and below that we need to map the Keras model. Then the model will form. Then Frontend model is connected to python. We can also implement the same using C++ or GO also [22].

3. Architectural Implementation

The architecture consists of pre-processing as the initial stage and then we need to convert the objects into categorical variables and have to assign a boolean value to those variables. Then we need to identify the test and train data from the given dataset. Later we need to connect the data and import the keras and tensorflow libraries. Then make a neural network with 500 hidden layers, add input layer, output layer next and implement ANN with all data. **Figure 6** is representing the architectural overview [23].

In this architecture, we need to identify what is the best layer and the best pair of rules we need to implement on the hidden layer. In those rules, we determined that triplets age, salary and tenure are the most useful rule combinations. The output graphs will represent the states. The following will be considered architecture.

The first step is to identify the dataset, and the implementations on the data can be considered as the pre-processing mechanisms, and the pre-processing mechanism can be used to implement elimination of the noisy data and the missing values in the dataset and there may be implementation of identification of the unwanted symbols in the dataset and the symbols can be eliminated using pre-processing mechanisms. Next, we need to split the dataset into related parts. As we follow the professional implementation, we have to break the dataset into 20/80 formats, that is 20% of the test data and the remaining 80% to train the model as the training dataset. The machine will learn from the knowledge we include in the rules, and the test set will help to identify the accuracy of the model.

Later we need to encode the categorical variables. The categorical variables places a crucial role in the model design as some of the features will have the string values which are needed to be converted to the accepted format by the modal. By default, all the string variables are considered as the object category, and after turning them to the boolean variables, we need to utilize them for the feature extraction [24] [25] [26].

In the feature scaling, we are implementing the concept of identifying the features which are mostly used for the mode to bring the highest accuracy for the model. In this scenario, we are implementing a feature scaling mechanism to identify the features. Later we need to import the necessary libraries which are mostly used for the model's elevation. Libraries which we are using are mentioned in the earlier phases of this article. Pandas are used for dataset importing, matplotlib is for plotting the graphs from the output, numpy is for mathematical operations from the model, Keras is to implement text classification and the implement

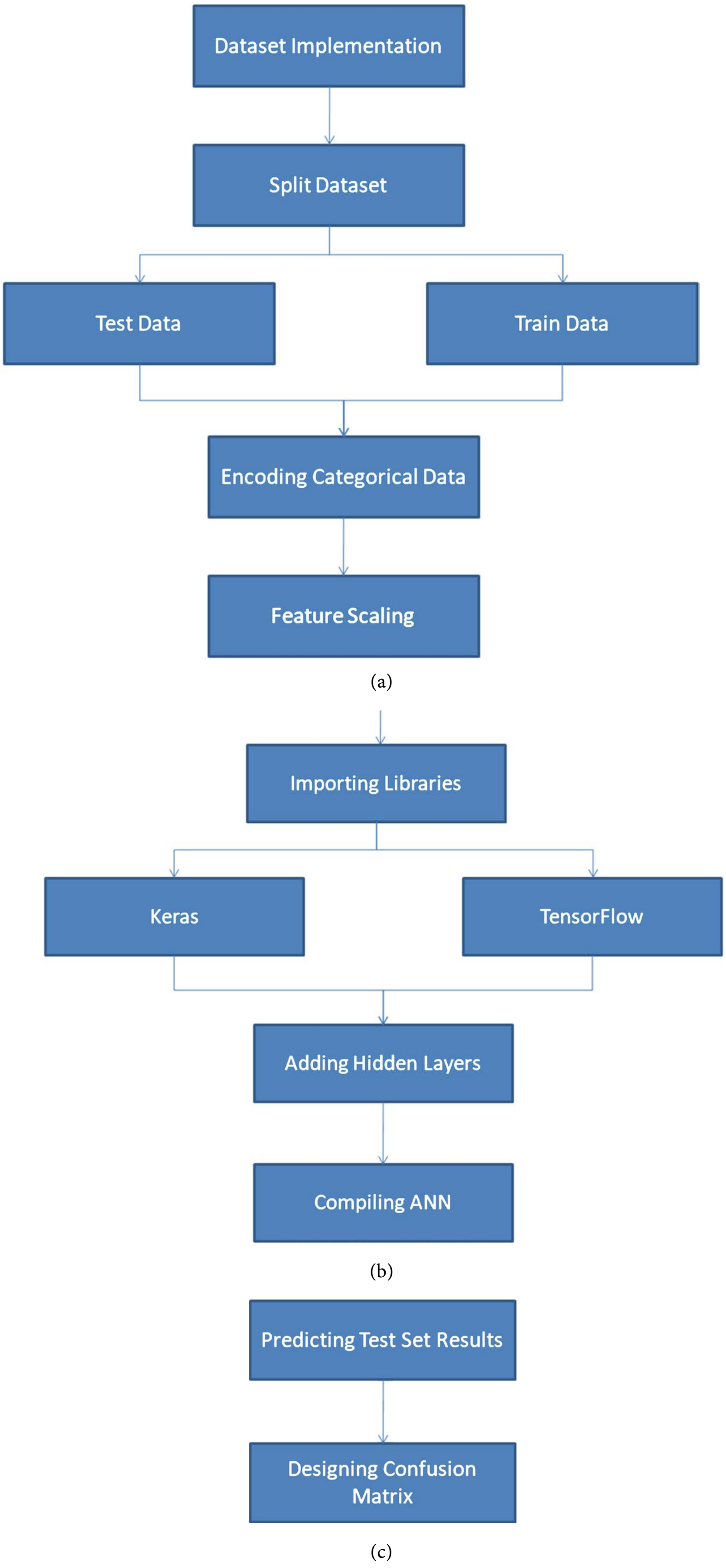


Figure 6. (a) Architecture; (b) Architecture Part 2; (c) Architecture Part 3.

tensorflow mechanism in the model. Using tensorflow and keras, we will make the neural network efficiently [10] [11] [12] [13] [14].

Now add the hidden layers using keras and tensorflow and make them take the rules and compile the rules. The laws we impose on the model must be capable of receiving a large amount of variable and the implement the features and build the model. The model consists of one input layer, number of hidden layers and only one output later. The difference between the obtained output value to the actual output value will differ in the accuracy of the mode. If the accuracy is not much acceptable, then we have to build the entire model again starting from scratch [1] [2].

After that process, we need to compile the entire neural network model to understand the working nature of it. The implementation of the neural network is easy with keras and tensor low. There are other implementable libraries for neural networks. But these two are most acceptable in reality.

In this last but one step we need to predict the test set results and identify the required result from the test set predictor. So when the same kind of data is obtained by the model in the future, then the same mechanism will be used by the developers. Finally, we need to identify the performance of the model and the classifier. For that, we need to build the confusion matrix which will determine the real positive, false positive, true negative and false negative values of the classifier [1] [2].

4. Results

The above architecture is being used in this article, and the implementation achieved an accuracy of algorithm up to 96%. The graphs presenting below will be the final outputs related to hidden layer creation with input and output layers. The number of hidden layers can be increased and all the hidden layers we utilized cannot be shown in the output for space constrain. The main purpose of this is to identify whether the customer in the current bank can exist with the same track record furthermore or not and also to identify whether there is an alternative for the methodology of identification of the customer related issues with the neural network. We also plan to execute the same thing with CNN. CNN is also an efficient manner of identifying the customer related issues and the main understanding between ANN and CNN is an implementation of hidden layers in an efficient way. **Figure 7** explains the sample of output from ANN for the prediction model with number of hidden layers with knowledge representation.

Here the inputs for the hidden layer are the rules which we impose and the features which we extract while doing any operations. If we consider this research work results, we can take some of the optimal rules but to form the optimal solutions we need to prove the logic of the concept. There comes predicate logic into the picture, which is domain dependent. In this result, we have sample inputs of the features and the predicate logic will identify the correctness of the rules and form the result.

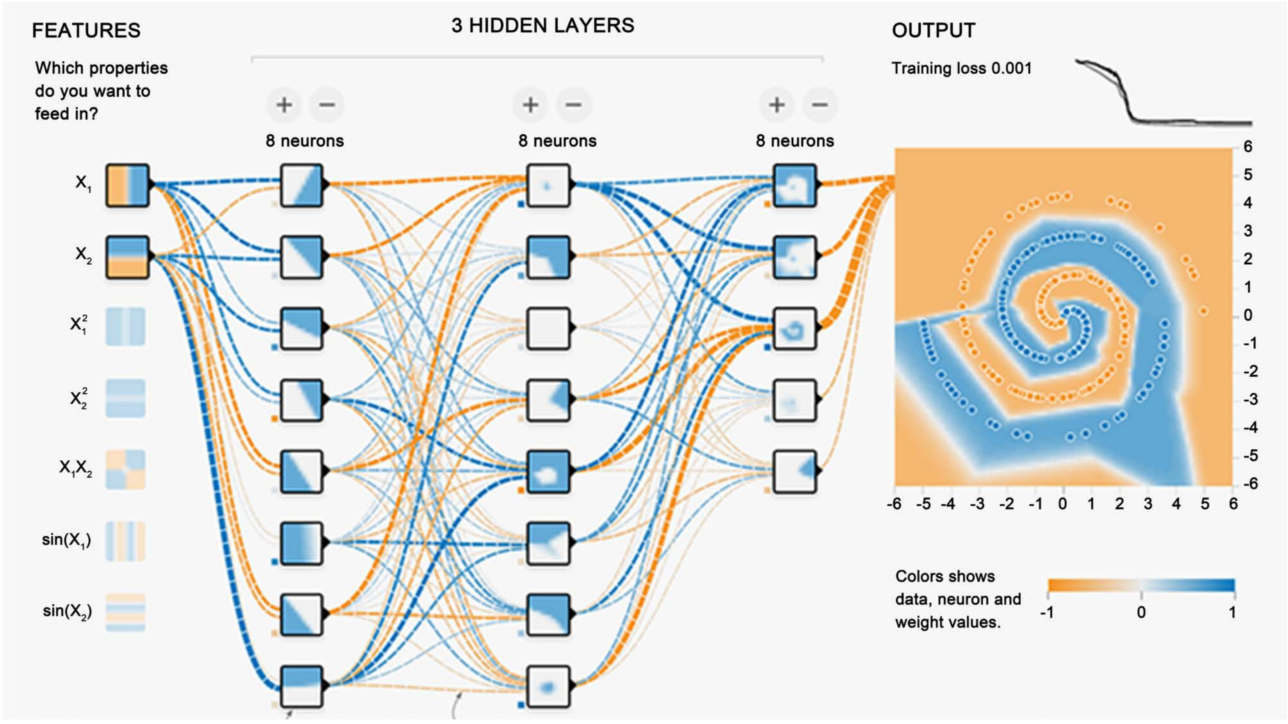


Figure 7. Sample ANN.

5. Conclusion

In this architecture, we are predicting the customer future product utilization and whether the customer can refund the loan amount to the bank and some other bank related things with regards to the customer transactions and the salary criteria. Here, one of the main content is using the feature of whether the customer has the credit card or not. If the person has a credit card, then whether he can pay or not is another constraint. Then we can identify whether the model we are designing will be used for further protocols or not. Machine learning implementations are being used with deep learning applications. Here, we used Keras and tensor flow mechanisms to predict the factors. And we acquired 96% accuracy in the triplet rule scenario mentioned in the previous sections. The accuracy of the model differs when there is more number of inputs appearing and a new logic combination has appeared. Irrespective of predicate logic, we are assigning the concept deep learning mechanism which will consider the formation of rules and models which will either form new hidden layers based on requirement of number of rules and the optimality of those paths.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Rawat, K., Kumar, A. and Gautam, A.K. (2014) Lower Bound on Naïve Bayes Clas-

sifier Accuracy in Case of Noisy Data. In: Babu, B., *et al.* Eds., *Proceedings of the Second International Conference on Soft Computing for Problem Solving* (*SocProS* 2012), **236**, 639-644. https://doi.org/10.1007/978-81-322-1602-5_68

[2] Kumar, P., Sasubilli, S. and Khurshudyan, A.Z. (2018) Approximate Analytical Solution to Non-Linear Young-Laplace Equation with an Infinite Boundary Condition. *International Conference on Computing, Mathematics and Engineering Technologies* (*iCoMET*), Sukkur, 3-4 March 2018, 1-5. https://doi.org/10.1109/ICOMET.2018.8346349

[3] To, K.B. and Napolitano, L.M. (2012) Common Complications in the Critically Ill Patient. *Surgical Clinics of North America*, **92**, 1519-1557.

[4] Wollschlager, C.M. and Conrad, A.R. (1988) Common Complications in Critically Ill Patients. *Disease-a-Month*, **34**, 225-293.

[5] Desai, S.V., Law, T.J. and Needham, D.M. (2011) Long-Term Complications of Critical Care. *Critical Care Medicine*, **39**, 371-379.

[6] Halpern, N.A., Pastores, S.M., Oropello, J.M. and Kvetan, V. (2013) Critical Care Medicine in the United States: Addressing the Intensivist Shortage and Image of the Specialty. *Critical Care Medicine*, **41**, 2754-2761.

[7] Johnson, A.E.W., Ghassemi, M.M., Nemati, S., Niehaus, K.E., Clifton, D.A. and Clifford, G.D. (2016) Machine Learning and Decision Support in Critical Care. *Proceedings of the IEEE*, **104**, 444-466. https://doi.org/10.1109/JPROC.2015.2501978

[8] Badawi, O., *et al.* (2014) Making Big Data Useful for Health Care: A Summary of the Inaugural MIT Critical Data Conference. *JMIR Medical Informatics*, **2**, e22. https://doi.org/10.2196/medinform.3447

[9] Reddy, C.K. and Aggarwal, C.C. (2015) Healthcare Data Analytics. Vol. 36, CRC Press, Boca Raton, FL. https://doi.org/10.1201/b18588

[10] Gotz, D., Stavropoulos, H., Sun, J. and Wang, F. (2012) ICDA: A Platform for Intelligent Care Delivery Analytics. *AMIA Annual Symposium Proceedings*, **2012**, 264-273.

[11] Perer, A. and Sun, J. (2012) MatrixFlow: Temporal Network Visual Analytics to Track Symptom Evolution during Disease Progression. *AMIA Annual Symposium Proceedings*, **2012**, 716-725.

[12] Mao, Y., Chen, W., Chen, Y., Lu, C., Kollef, M. and Bailey, T. (2012) An Integrated Data Mining Approach to Real-Time Clinical Monitoring and Deterioration Warning. *Proceedings of the* 18*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, 12-16 August 2012, 1140-1148. https://doi.org/10.1145/2339530.2339709

[13] Wiens, J., Horvitz, E. and Guttag, J.V. (2012) Patient Risk Stratification for Hospital-Associated C. Diff as a Time-Series Classification Task. *Advances in Neural Information Processing Systems*, 467-475.

[14] Saria, S., Koller, D. and Penn, A. (2010) Learning Individual and Population Level Traits from Clinical Temporal Data. *Neural Information Processing Systems* (*NIPS*), *Predictive Models Personalized Med. Workshop*.

[15] Dürichen, R., Pimentel, M.A.F., Clifton, L., Schweikard, A. and Clifton, D.A. (2015) Multitask Gaussian Processes for Multivariate Physiological Time-Series Analysis. *IEEE Transactions on Biomedical Engineering*, **62**, 314-322.

[16] Ghassemi, M., *et al.* (2015) Amultivariate Timeseries Modeling Approach to Severity of Illness Assessment and Forecasting in ICU with Sparse, Heterogeneous Clini-

cal Data. *AAAI Conference on Artificial Intelligence*, **2015**, 446-453.

[17] Batal, I., Valizadegan, H., Cooper, G.F. and Hauskrecht, M. (2011) A Pattern Mining Approach for Classifying Multivariate Temporal Data. *IEEE International Conference on Bioinformatics and Biomedicine*, **2011**, 358-365.

[18] Lasko, T.A. (2014) Efficient Inference of Gaussian-Process-Modulated Renewal Processes with Application to Medical Event Data. *Uncertainty in Artificial Intelligence*, **2014**, 469-476.

[19] Barajas, K.L.C. and Akella, R. (2015) Dynamically Modeling Patient's Health State from Electronic Medical Records: A Time Series Approach. 21*st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, 10-13 August 2015, 69-78.

[20] Wang, X., Sontag, D. and Wang, F. (2014) Unsupervised Learning of Disease Progression Models. Proc. 20*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 24-27 August 2014, 85-94. https://doi.org/10.1145/2623330.2623754

[21] Cohen, M.J., Grossman, A.D., Morabito, D., Knudson, M.M., Butte, A.J. and Manley, G.T. (2010) Identification of Complex Metabolic States in Critically Injured Patients Using Bioinformatic Cluster Analysis. *Critical Care*, **14**, 1. https://doi.org/10.1186/cc8864

[22] Mylavarapu, B.K. (2018) Implementing Machine Learning in Intensive Care Units: For Avoiding Over Medication. *International Journal of Pure and Applied Mathematics*, **118**, 4799-4811. https://acadpubl.eu/hub/2018-118-21/articles/21f/33.pdf

[23] Mylavarapu, B.K. and Mylavarapu, R.T. (2018) A Framework for Hierarchical Big Image Data. 2018 *Second International Conference on Inventive Communication and Computational Technologies* (*ICICCT*), Coimbatore, 20-21 April 2018, 169-173. https://doi.org/10.1109/ICICCT.2018.8473339

[24] Zhou, J., Liu, J., Narayan, V.A. and Ye, J. (2012) Modeling Disease Progression via Fused Sparse Group Lasso. 18*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, 12-16 August 2012, 1095-1103. https://doi.org/10.1145/2339530.2339702

[25] Choi, E., Du, N., Chen, R., Song, L. and Sun, J. (2015) Constructing Disease Network and Temporal Progression Model via Context-Sensitive Hawkes Process. *Proceedings of the IEEE International Conference on Data Mining*, Atlantic City, 14-17 November 2015, 721-726. https://doi.org/10.1109/ICDM.2015.144

[26] Pivovarov, R., Perotte, A.J., Grave, E., Angiolillo, J., Wiggins, C.H. and Elhadad, N. (2015) Learning Probabilistic Phenotypes from Heterogeneous HER Data. *Journal of Biomedical Informatics*, **58**, 156-165.