VOL 1 (2017) NO 3

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Bike Sharing Prediction using Deep Neural Networks

Chandrasegar Thirumalai[#], Ravisankar Koppuravuri[#]

[#] *School of Information Technology and Engineering, VIT University, India*
E-mail: chandru01@gmail.com, koppuravuri.ravisankar2013@vit.ac.in

Abstract— In this paper, we will use deep neural networks for predicting the bike sharing usage based on previous years usage data. We will use because deep neural nets for getting higher accuracy. Deep neural nets are quite different from other machine learning techniques; here we can add many numbers of hidden layers to improve the accuracy of our prediction and the model can be trained in the way we want such that we can achieve the results we want. Nowadays many AI experts will say that deep learning is the best AI technique available now and we can achieve some unbelievable results using this technique. Now we will use that technique to predict bike sharing usage of a rental company to make sure they can take good business decisions based on previous years data.

Keywords: Accuracy, Artificial Intelligence, Neural network, Deep learning, Hidden layer, Sharing

I. INTRODUCTION

Artificial Intelligence has been everyone's dream from decades, but we are not able to reach that stage due to lack of computation power and data, but today in the present world environment where the internet is playing a key role we are generating a lot of data and processors which can do complex tasks has been developed and in future we can see quantum computers too, so with these emerging trends and technologies AI has come into limelight and it has been used (or) implemented in various domains and the best present AI available is MachineLearning.It is quite the reverse way of traditional programming practice, normally we will try to tell the program how it needs to implement calculations and functions to get the required outcome, but here we will develop a model to which we will give original data as input and we will allow it to take decisions. Here we can tell the machine what we want to do with the data (or) we will allow the machine to take decisions itself. And there are 3 types of learning methods

1) Supervised Learning: Here model will do what we want it to do. We will clearly specify the input variables, data types and what type of operations to be done. We will specify the model, what we are expecting as output from the system. This is a sort of normal way of programming style.

Eg: Linear regression, Logistic regression etc.

2) Unsupervised Learning: This is the most useful and unorthodox way of machine Learning, here we won't specify the model what to be done clearly (or) what we are expecting as output, here pure raw data will be given as input and model will decide itself of what to be done with data, this can be possible only with intelligent way of programming since here some decisions has to be solely taken by model itself. This is the most required one too because the data generated these days is mostly unstructured and its almost impossible for humans to label these huge datasets.

Eg: clustering etc.

3) Reinforcement Learning: this is one of the places huge amount of development has been going on, mainly in automated chat bots and game bots, where a move will be taken by the model and depending upon the result of that move next step will be taken and the previous move will be learned by the model and it will use that knowledge in next attempts.

Eg: Google Alpha go, IBM Watson.

Neural networks are built using smaller units called neurons (or) perceptrons which will work like neurons in our brain and thus named neural nets and just they will do some calculation and data they got and will pass the output of that operation to other. Deep Learning is a type of Deep Neural Networks.

This enlightening dataset [10] is used as the commitment to figure the direct relapse [2], [31], [32] and Pearson [11], [15], [19], [21], [27], [29]. In the present days, there are colossal measures of data recorded by the banks and examining them requires complex estimations. A few machine learning calculations [1], [3], [4], [5], [6], [7], [8], [9], [45], assumes an imperative part in settling on an astute choice to anticipate the bicycle sharing data. We played out the item metric examination on the given instructive gathering. From the data examination [13], [14], [17], [22], [23], [25] we can pick which property can be viewed as and which quality can be disregarded. For instance, in the Pearson procedure if the

estimation of r is more than 0.5 then the attributes are thought to be immovably related and if it is underneath 0.3 the characteristics are insufficiently related. A part of the past procedures to appraise the decisions in perspective of their relationship of value are Spearman [6], Analytical Hierarchical Process (AHP) [12], [18], [20] and Traveling Salesman Problem (TSP) [43]. The sensitive information's among various substances [24], [26], [33], [35], [37], [29], [31] among the bank stock model are dealt with by late secured strategies [28], [30], [34], [36], [38], [40], [42], [44]..

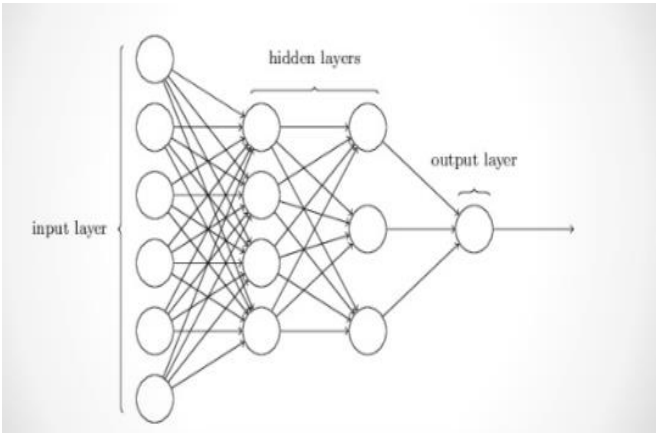

Fig. 1. Fig. 1. Deep Learning Architecture.

Perceptron: Data is fed into a network of interconnected nodes. These individual nodes are called perceptron's or neurons, and they are the basic unit of a neural network. Each one looks at input data and decides how to categorize that data. In the example above, the input either passes a threshold for grades and test scores or doesn't and so the two categories are: yes (passed the threshold) and no (didn't pass the threshold). These categories then combine to form a decision.

We will multiply the data coming into perceptrons with some value called weights. When input data comes into a perceptron, it gets multiplied by a weight value that is assigned to this particular input. For example, the perceptron having two inputs will have two individual weights for each input, so it has two associated weights that can be adjusted individually. These weights start out as random values, and as the neural network learns more about what kind of input data leads to a student being accepted into a university, the network adjusts the weights based on any errors in categorization that the previous weights resulted in. This is called training the neural network.

A higher weight means the neural network considers that input more important than other inputs, and lower weight means that the data is considered less important. Now a bike sharing company's previous data will be taken to find out what business decisions needed to be taken in future by applying Deep neural nets

TABLE I. DATASET PART-1

instant	*day*	*season*	*Year*	*Month*	*Hour*
1	1/1/2011	1	0	1	0
2	1/1/2011	1	0	1	1
3	1/1/2011	1	0	1	2
4	1/1/2011	1	0	1	3
5	1/1/2011	1	0	1	4
6	1/1/2011	1	0	1	5
7	1/1/2011	1	0	1	6
8	1/1/2011	1	0	1	7
9	1/1/2011	1	0	1	8

TABLE II: DATASET PART-2

holida	*weekda*	*working*	*weather*	*tem*	*temp*
0	6	0	1	0.24	0.287
0	6	0	1	0.22	0.272
0	6	0	1	0.22	0.272
0	6	0	1	0.24	0.287
0	6	0	1	0.24	0.287
0	6	0	2	0.24	0.257
0	6	0	1	0.22	0.272

The process of building neural network can be called as model building also. We will build our model in small steps to make sure everything is going well. Technically two important processes are needed to be implemented.

1) Forward pass: multiplying weights with input data and adding those products together with bias and passing that output to next neuron.

2) Backward pass: in this phase, we will calculate the error and then we will backpropagate the error to update the weights.

1) Load and prepare data

This dataset has the number of riders for each hour of each day from January 1, 2011, to December 31, 2012. The number of riders is split between casual and registered, summed up in the cnt column. You can see the first few rows of the data above. Below is a plot showing the number of bike riders over the first 10 days in the data set. You can see the hourly rentals here. This data is pretty complicated! The weekends have lower overall ridership and there are spikes when people are biking to and from work during the week. Looking at the data above, we also have information about temperature, humidity, and wind speed, all of this likely affecting the number of riders.

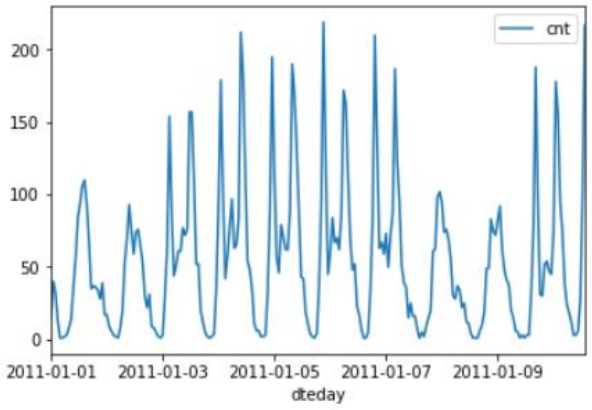

Fig. 2. Chart showing number of riders over first 10 days

2) Dummy variables

Here we have some categorical variables like season, weather, and month. To include these in our model, we'll need to make binary dummy variables. This is simple to do with Pandas using a function called get_dummies().

TABLE III. DIFFERENT WEEKS OF BIKE SHARING.

Wday	*Wday*	*Wday*	*Wday*	*Wday*	*Wday*	*Wday*
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	1
0	0	0	0	0	0	1

3) Scaling target variables

To make training the network easier, we'll standardize each of the continuous variables. That is, we'll shift and scale the variables such that they have zero mean and a standard deviation of 1. The scaling factors are saved so we can go backward when we use the network for predictions.

4) Splitting the data into training, testing, and validation sets

We'll save the last 21 days of the data to use as a test set after we've trained the network. We'll use this set to make predictions and compare them with the actual number of riders. We'll split the data into two sets, one for training and one for validating as the network is being trained. Since this is time series data, we'll train on historical data, then try to predict on future data (the validation set).

5) Building Neural net

The network has two layers, a hidden layer, and an output layer. The hidden layer will use the sigmoid function for activations. The output layer has only one node and is used for the regression, the output of the node is the same as the input of the node. That is, the activation function is f(x)=x. A function that takes the input signal and generates an output signal, but takes into account the threshold, is called an activation function. We work through each layer of our network calculating the outputs for each neuron. All of the outputs from one layer become inputs to the neurons on the next layer. This process is called forward propagation. We use the weights to propagate signals forward from the input to the output layers in a neural network. We use the weights to also propagate error backward from the output back into the network to update our weights. This is called back propagation.

II. SAMPLE CODE

inputs = np.array(inputs_list, ndmin=2).T

targets = np.array(targets_list, ndmin=2).T

 # print('inputs at start of train', inputs.shape)

 # print('targets at start of train ', targets.shape)

Implement the forward pass here

Forward pass

 # Hidden layer

 # signals into hidden layer

hidden_inputs = np.dot(self.weights_input_to_hidden, inputs)

 # signals from hidden layer

hidden_outputs = self.activation_function(hidden_inputs)

 # TODO: Output layer

 # signals into final output layer

final_inputs = np.dot(self.weights_hidden_to_output, hidden_outputs)

 # signals from final output layer

final_outputs = final_inputs

Implement the backward pass here

Backward pass

 # TODO: Output error

output_errors = targets - final_outputs

 # Output layer error is the difference between desired target and actual output.

 # TODO: Backpropagated error

hidden_errors = np.dot(self.weights_hidden_to_output.T, output_errors)

 # errors propagated to the hidden layer

hidden_grad = hidden_outputs * (1 - hidden_outputs)

 # hidden layer gradients

 # TODO: Update the weights

output_grad =1

self.weights_hidden_to_output += self.lr * np.dot(output_errors, hidden_outputs.T)

 # update hidden-to-output weights with gradient descent

self.weights_input_to_hidden += self.lr * np.dot(hidden_grad * hidden_errors , inputs.T).

III. METRICS

$$\text{sigmoid}(x) = 1/(1+e^{-x})$$

$$y = f(h) = \text{sigmoid}(\textstyle\sum_i w_i x_i + b)$$

Gradient Descent formula:

Error calculation for finding out the difference between predicted and original output. A common metric is the sum of the squared errors (SSE).

$$E = \tfrac{1}{2} \textstyle\sum_\mu \sum_j \left[y_j^\mu - \hat{y}_j^\mu \right]^2$$

$$E = \frac{1}{2} \sum_{\mu} \sum_{j} \left[y_j^{\mu} - f\left(\sum_i w_{ij} x_i^{\mu} \right) \right]^2$$

$$\hat{y}_j^{\mu} = f\left(\sum_i w_{ij} x_i^{\mu} \right)$$

Now we will train the network using training data and we will test using testing data.

After training:

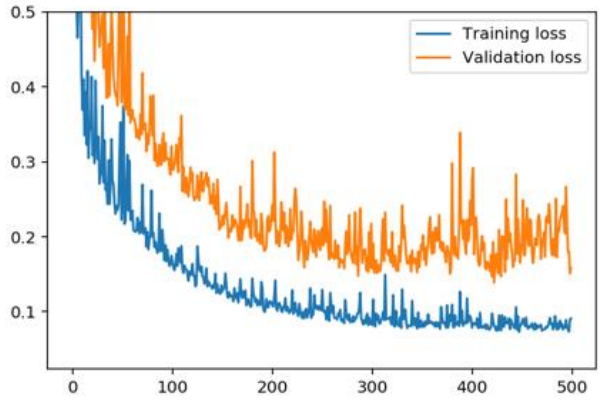

Fig. 3. Comparison between original and predicted data

Now we will compare our result with original data to check how far we predicted correctly.

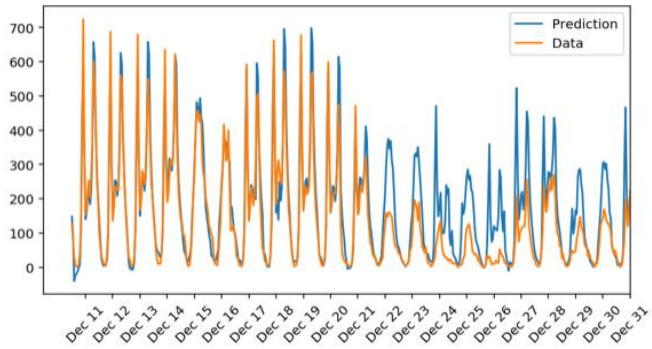

Fig. 4. Validation accuracy on testing data.

IV. Conclusion

In this solution, we have used deep neural nets instead of general neural nets (or) single layer algorithms such that hidden layers can be added in between input and output layers so that the network can identify relations between the data and can properly update the weights of respective features. Although sometimes the improvement in accuracy cannot vary much from general techniques, but when model is properly trained with proper training data and best number of hidden layers then we can achieve much better results, in this business problem with only one hidden layer we clearly achieved an accuracy of approx 70%, which is very good considering the type of business problem.ence we can see that by using deep neural nets we almost got 80% accuracy, which is very good for this sort of business problem.

References

[1] Hinton, G. E., Osindero, S., & Teh, Y.¬W. (2006). A fast learning algorithm for deep belief nets.

[2] Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines

[3] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors.

[4] Bengio, Y. and LeCun, Y. Scaling learning algorithms to-wards AI. 2007.

[5] Freund, Y. and Haussler, D. Unsupervised learning of distributions on binary vectors using two layer networks.

[6] Technical report, Santa Cruz, CA, USA, 1994.

[7] Salakhutdinov, R. and Hinton, G. E. Replicated softmax: an undirected topic model. In Advances in Neural In-formation Processing Systems 22, 2009.

[8] Hahnloser, Richard H. R., Seung, H. Sebastian, and Slotine, Jean-Jacques. Permitted and forbidden sets in symmetric threshold-linear networks. Neural Computation,15(3):621–638, 2003. ISSN 0899-7667.

[9] Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio., Y. An empirical evaluation of deep architectures on problems with many factors of variation. In ICML, pp. 473–480, 2007.

[10] https://archive.ics.uci.edu/ml/datasets/Bike Sharing Dataset

[11] Hauke J., Kossowski T., Comparison of values of Pearson's and Spearman's correlation coefficient on the same sets of data. Quaestiones Geographicae 30(2), Bogucki Wydawnictwo Naukowe, Poznań 2011, pp. 87–93, 3 figs, 1 table. DOI 10.2478/v10117-011-0021-1, ISBN 978-83-62662-62-3, ISSN 0137-477X.

[12] Piovani J.I., 2008. The historical construction of correlation as a conceptual and operative instrument for empirical research. Quality & Quantity 42: 757–777.

[13] P. Dhavachelvan, Chandra Segar T, K. Satheskumar, "Evaluation of SOA Complexity Metrics Using Weyuker's Axioms," IEEE International Advance Computing (IACC), India, pp. 2325 – 2329, March 2009

[14] Halstead Metric for Intelligence, Effort, Time predictions, DOI:10.13140/RG.2.2.17988.42881

[15] Fisher R.A., 1921. On the "probable error" of a coefficient of correlation deduced from a small sample. Metron 1: 3–32.

[16] Spearman C.E, 1904b. General intelligence, objectively determined and measured. American Journal of Psychology 15: 201–293.

[17] Software metric Numerical Data analysis using Box plot and control chart methods, VIT University, DOI:10.13140/RG.2.2.27422.95041

[18] Vaishnavi B, Karthikeyan J, Kiran Yarrakula, Chandrasegar Thirumalai, "An Assessment Framework for Precipitation Decision Making Using AHP", International Conference on Electronics and Communication Systems (ICECS), IEEE & 978-1-4673-7832-1, Feb. 2016

[19] Griffith D.A., 2003. Spatial autocorrelation and spatial filtering. Springer, Berlin.

[20] Chandrasegar Thirumalai, Senthilkumar M, "An Assessment Framework of Intuitionistic Fuzzy Network for C2B Decision Making", International Conference on Electronics and Communication Systems (ICECS), IEEE & 978-1-4673-7832-1, Feb. 2016

[21] Rodgers J.L. & Nicewander W.A., 1988. Thirteen ways to look at the correlation coefficient. The American Statistician 42 (1): 59–66.

[22] F. Fioravanti, P. Nesi, "A method and tool for assessing object-oriented projects and metrics management," Journal of Systems and Software, Volume 53, Issue 2, 31 August 2000, Pages 111-136

[23] Galton F., 1875. Statistics by intercomparison. Philosophical Magazine 49: 33–46

[24] Chandrasegar Thirumalai, Viswanathan P, "Diophantine based Asymmetric Cryptomata for Cloud Confidentiality and Blind Signature applications," JISA, Elsevier, 2017.

[25] Galton F., 1877. Typical laws of heredity. Proceedings of the Royal Institution 8: 282–301.

[26] Chandrasegar Thirumalai, Sathish Shanmugam, "Multi-key distribution scheme using Diophantine form for secure IoT communications," IEEE IPACT 2017.

[27] Galton F., 1888. Co-relations and their measurement, chiefly from anthropometric data. Proceedings of the Royal Society of London 45: 135–145.

[28] Chandrasegar Thirumalai, Senthilkumar M, “Spanning Tree approach for Error Detection and Correction,” IJPT, Vol. 8, Issue No. 4, Dec-2016, pp. 5009-5020.
[29] Galton F., 1890. Kinship and correlation. North American Review 150: 419–431.
[30] Chandrasegar Thirumalai, Senthilkumar M, “Secured E-Mail System using Base 128 Encoding Scheme,” International journal of pharmacy and technology, Vol. 8 Issue 4, Dec. 2016, pp. 21797-21806.
[31] Yule G.U., 1897a. On the significance of Bravais’ formulae for regression, in the case of skew correlation. Proceedings of the Royal Society of London Ser. A 60: 477–489
[32] Chandramowliswaran N, Srinivasan.S and Chandra Segar.T, “A Note on Linear based Set Associative Cache address System” International J. on Computer Science and Engg. (IJCSE) & India, Engineering Journals & 0975-3397, Vol. 4 No. 08 / pp. 1383-1386 / Aug. 2012.
[33] T Chandra Segar, R Vijayaragavan, “Pell's RSA key generation and its security analysis,” in Computing, Communications and Networking Technologies (ICCCNT) 2013, pp. 1-5
[34] Chandrasegar Thirumalai, Senthilkumar M, Vaishnavi B, “Physicians Medicament using Linear Public Key Crypto System,” in International conference on Electrical, Electronics, and Optimization Techniques, ICEEOT, IEEE & 978-1-4673-9939-5, March 2016.
[35] Chandrasegar Thirumalai, “Physicians Drug encoding system using an Efficient and Secured Linear Public Key Cryptosystem (ESLPKC),” International journal of pharmacy and technology, Vol. 8 Issue 3, Sep. 2016, pp. 16296-16303
[36] Malathy, Chandra Segar Thirumalai, "Review on non-linear set associative cache design," IJPT, Dec-2016, Vol. 8, Issue No.4, pp. 5320-5330
[37] “DDoS: Survey Of Traceback Methods”, International Joint Journal Conference in Engineering 2009, ISSN 1797-9617.
[38] Chandrasegar Thirumalai, Senthilkumar M, Silambarasan R, Carlos Becker Westphall, “Analyzing the strength of Pell’s RSA,” IJPT, Vol. 8 Issue 4, Dec. 2016, pp. 21869-21874.
[39] Chandramowliswaran N, Srinivasan.S and Chandra Segar T, “A Novel scheme for Secured Associative Mapping” The International J. of Computer Science and Applications (TIJCSA) & India, TIJCSA Publishers & 2278-1080, Vol. 1, No 5 / pp. 1-7 / July 2012
[40] Chandrasegar Thirumalai, “Review on the memory efficient RSA variants,” International Journal of Pharmacy and Technology, Vol. 8 Issue 4, Dec. 2016, pp. 4907-4916.
[41] Vinothini S, Chandra Segar Thirumalai, Vijayaragavan R, Senthil Kumar M, “A Cubic based Set Associative Cache encoded mapping,” International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 02 May -2015
[42] Chandrasegar Thirumalai, Himanshu Kar, “Memory Efficient Multi Key (MEMK) generation scheme for secure transportation of sensitive data over Cloud and IoT devices,” IEEE IPACT 2017.
[43] M.Senthilkumar, T.Chandrasegar, M.K. Nallakaruppan, S.Prasanna, “A Modified and Efficient Genetic Algorithm to Address a Travelling Salesman Problem,” in International Journal of Applied Engineering Research, Vol. 9 No. 10, 2014, pp. 1279-1288
[44] Nallakaruppan, M.K., Senthil Kumar, M., Chandrasegar, T., Suraj, K.A., Magesh, G., “Accident avoidance in railway tracks using Adhoc wireless networks,” 2014, IJAER, 9 (21), pp. 9551-9556.
[45] Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1), 221–231.