Implementation of a neural network by using data normalisation and pattern reduction in banking dataset

Xiao TIAN

Research School of Computer Science, Australian National University

u6277077@anu.edu.au

Abstract

Some data may be not valid for training back-propagation neural networks, hence affect the implementation of neural networks. In this report, I use PyTorch to build a neural network, and use data normalisation and pattern reduction method to maximise the performance of the network. The task of the network is to predict whether the product will be subscribed based on the clients' personal and financial information. In the process of improvement, I use PyTorch to process the very large number of erratically reliable data, and to produce acceptable level of accuracy and confusion matrix.

The dataset was also used in the article *A data-driven approach to predict the success of bank telemarketing*. The result classification correct rates in this report are higher than the results produced in the article. However, the methods used in the published article are more thoughtful.

Key words: back-propagation, normalised data, pattern reduction, neural network

1 Introduction

Traditional marketing selling strategy focus on sell products to target clients. Marketing selling campaign is one of a business strategy of companies. Before publishing a new product, companies should know who are target customers of this product hence build an effective selling strategy. Directly contact customers by phone is one marketing approach helping companies to focus on possible buyers. However, by use technology, i.e. Neural network, companies access target clients without actual products. We predict whether the client would buy the product by analyse detail information of clients. Rethinking marketing strategy focus on creating product that client want (Rust, Moorman, & Bhalla, 2010).

In this paper I built a neural network to solve a classification problem, that to access whether client will subscribe a term deposit by using back-propagation of error measure, and then use normalise data and reduce pattern to improve the neural network.

1.1 Artificial Neural Networks

Artificial Neural Network is inspired by biological neural network of animal brains (Gerven & Bohte, 2017). An artificial neural network is based on connected units (artificial neurons) collections. Neurons transmit signal between each other by using connection. Back Propagation is a commonly used effective method when training artificial neural networks (Goodfellow, Bengio, & Courville, 2016). In this report, I implement a back-propagation neural network to solve a classification problem.

1.2 Data set

The raw dataset for this report is from the article A data-driven approach to predict the success of bank telemarketing pubulished by Moro, Cortez and Rita in 2014. It has 21 attributes and 41188 instances, which is sufficient enough for the requirement and neural network building. Additionally, the dataset is suitable for training a classification problem-solving neural network. It is related with a marketing campaigns which were proceed by phone inquiring. The marketing campaigns accessed whether the client would subscribe the product or not. The first 20 attributes are age, type of job, marital status, education, default, housing loan, personal loan, contact communication type, last contact month of year, last contact day of the week, last contact duration, number of contacts performed during this campaign and for this client, number of days that passed by after the client was last contacted from a previous campaign, number of contacts performed before this campaign and for this client, outcome of the previous marketing campaign, employment variation rate - quarterly indicator, consumer price index - monthly indicator, consumer confidence index - monthly indicator, euribor 3 month rate - daily indicator and number of employees - quarterly indicator, which were classified as inputs. The last attribute, whether the client subscribed the product, is desired target. The classification goal is to predict if the client will subscribe the product.

1.3 Dataset pre-processing

In this report, I use Python to implement a neural network. Firstly, import all required libraries, here including torch, numpy, pandas, Variables form torch.autograd and activation function. Secondly, read and process data by using PyTorch. Before processing dataset, I delimit it by attributes, i.e. divide dataset into 21 columns. Then, replace input variables to numeric. The banking dataset includes not only rates, balance and other financial figures, but clients' education level, marital status as well. Among 21 attributes there are 11 attributes are not numeric, including job, marital, education, default, housing, loan, contact, month, day_of_week, poutcome, and target "y". In order to implement these data by using neural network, I replaced them to sequential numbers.

Then, convert pandas dataframe to array after collect and convert all non-numeric values to sequential numbers. And then, create x_array and y_array. First, I split arrays into two parts, as the first 20 columns are clients' information while the last one is target. When training a neural network, a huge set of patterns not always be effectiveness (Chauvin, 1990). The risk of overfitting rises as the size of data set increase. In this case, I divided the banking dataset into two parts, which are 80% and 20% of the total dataset, for training and testing respectively. The last step is to create Tensors to hold inputs and outputs and wrap them in Variables since Torch only trains neural network on Variables.

2 Implement a neural network using PyTorch

After pre-process dataset, I define __initi__ function to set up a neural network structure with on hidden layer with 10 neurons and using Sigmoid as activation function. The built input layer includes 20 neurons, representing clients' information, and the output layer includes 2 neurons, representing that if the product would be subscribed or not. Then, define the process of performing forward pass, which is to accept a Variable of input data, x, and return a Variable of output data, y_pred, in the forward function. Use Cross Entropy Loss function to compute the amount by which the prediction deviates from the actual values which will be stored for visualisation in this network. Use Stochastic Gradient Descent (SGD) as an optimiser to train the network, that will hold the current state and will update the parameters based on the computed gradients. The learning rate is set as 0.01. Before performing backward pass, the gradients need to be cleared. And the final step is to call the step function on an optimiser to update its parameters.

2.1 Evaluate the neural network

The accuracy of the target "y" prediction, i.e. if the client will subscribe the product (yes) or not (no) indicates a general validation of the network. The network was initially trained for 1,000 epochs and print out the accuracy every 50 epochs. It showed an accuracy approximately at 93.63%. However, it is insufficient if only accuracy is printed to evaluate the performance of the network. Because, for example, one neuron in target y presents most part (90%), then even if the network predicts every output as the neuron, the accuracy shows 90%. In this case, a confusion matrix is needed to show how well the network performs on different categories, see below:

Epoch [951/1000] Loss: 0.2426 Accuracy: 93.63%
confusion matrix for training:

30850	0
2100	0

Testing Accuracy: 93.63%
confusion matrix for testing:

7979	0
259	0

Figures in the right column are 0, which means the network predicts every client would not subscribe the product. The network did not perform effectively even with results of 93.63% accuracy. The dataset and neural network may need to be further improved.

3 Method

3.1 Data normalisation

Customised neural network structure is crucial for efficient machine learning, and the choice of dataset and dataset pre-processing are vital for a neural network as well. According to Bustos and Gedeon (1995), in some case, the raw data contains a significant number of erratically reliable data may affect the meaningful of prediction. In this part, I will use some method to process erratically reliable figures in banking dataset. After pre-processed as above, the data set is shown as Appendix 1:

No.	attributes	variables
1	age	17, 18, …, 92, 94, 95, 98
2	job	0, 1, …, 11
3	marital	0, 1, 2, 3
4	education	0, 1, …, 7

5	default	0, 1, 2
6	housing	0, 1, 2
7	loan	0, 1, 2
8	contact	0 1
9	month	0, 1, …, 9
10	day_of_week	0, 1, 2, 3, 4
11	duration	0, 1, 2, …, 3785, 4199, 4918
12	campaign	1, 2, …, 42, 43, 56
13	pdays	0, 1, 2, …,25, 26
14	previous	0, 1, …, 7
15	poutcome	0, 1, 2
16	emp.var.rate	-3.4, -3.0, -2.9, -1.8, -1.7, -1.1, -0.2, -0.1, 1.1, 1.4
17	cons.price.idx	92.201, 92.379, …, 94.601, 94.767
18	cons.conf.idx	-50.8, -50.0, -49.5, …, -26.9
19	euribor3m	0.634, 0.635, …, 5.045
20	nr.employed	4963.6, 4991.6, …, 5195.8, 5228.1
21	y	0, 1

The values of attributes, for example "duration" which represents last contact duration, in seconds, are erratically and contain considerable gaps, which may relatively affect the performance of the network, hence should be normalised over the range 0 - 1 for the logistic function. To implement an effective neural network, I processed the values of all the inputs, including age, job, marital, education, default, housing, loan, contact, month, day_of_week, duration, campaign, pdays, previous, poutcome, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr.employed, to be normalised over range 0 – 1. Then, the accuracy and confusion matrix shown as below:

Epoch [951/1000] Loss: 0.2426 Accuracy: 93.63%
confusion matrix for training:

30850	0
2100	0

Testing Accuracy: 93.63%
confusion matrix for testing:

7979	0
259	0

The confusion matrix shows that the network still predicts all the client will not subscribe the product. A thought that some class may have too many instances while others have too few is inspired by Bustos and Gedeon (1995) and Gedeon and Bowden (1992). The banking dataset is unbalanced as 88.7% clients choose not to subscribe the product, i.e. "no" in column y. This may be resulting a significant bias in the network. In this case, I use pattern reduction to improve the performance of the network on the validation dataset.

3.2 Pattern reduction

Among the total 41188 instances, there are only 4640 instances are related to successes while the rest 36548 instances targeted "no". I cut the size of the total non-successes instances, hence only 4640 of them are used in this network, including training dataset and test dataset. Then I got same number of target "yes" and "no".

In order to acquire a clear view of the pattern reduction method benefits, the dataset trained in this part is before normalisation. And the accuracy and confusion matrix shown as below:

Epoch [951/1000] Loss: 0.0529 Accuracy: 97.64%
confusion matrix for training:

3652	60
117	3595

Testing Accuracy: 98.76%
confusion matrix for testing:

905	23
0	928

Finally, the accuracy increases to 97.62%, and confusion matrix shows an acceptable level of correct classification. The technology that reduce pattern considerably improve the performance of the network.

3.3 Reduced size dataset normalisation

To define a maximum performance of the network and the validation of dataset, I normalise the reduce sized dataset. Using the normalised size reduced dataset to train the initial network. Then, the accuracy and confusion matrix shown as below:

Epoch [951/1000] Loss: 0.0489 Accuracy: 97.84%
confusion matrix for training:

3666	36
116	3596

Testing Accuracy: 98.65%
confusion matrix for testing:

903	25
0	928

In this part, the only variable compares to part 3.2 is whether the dataset is normalised or not. The results of accuracy and confusion matrix stay mostly same as when only the pattern reduction method is used. In this case, the technology of data normalisation lacks effectiveness when predict the banking dataset.

4 Results and Discussion

In the final implemented network, there are 20 input units corresponding to the features of the processed clients' personal information data, on hidden layer with 10 neurons, and 2 output units. This network size is suitable when predict the willing of subscribe product. In each run, the network is trained by same number of epochs (1000) and learning rate (0.01). Also, the network is trained by same starting weights to minimise the influence of the unit weights caused by the initial random functionality.

The 20 input units contain many erratically reliable data which are normalised range over 0 – 1. Additionally, to achieve pattern reduction, the original set of 41188 patterns is divided into 2 parts by different results, "yes" and "no". This is because only 11.3% (4640) clients subscribe the product, which make the dataset significantly unbalanced. Then, I randomly choose 4640 patterns, which is the number of patterns in positive part, out of the whole negative part (36543 patterns). Those two parts, contain 9280 patterns in total, are then combined together and become a valid dataset for the neural network.

80% of the valid dataset form a validation training set, and the remaining 20% (1956) form a validation test set which will never be seen by the network during training. The same set of 1956 patterns are used to validate the network trained by both the full size normalised training set and the reduced size normalised training sets. Training and testing results as below:

Epoch [951/1000] Loss: 0.0489 Accuracy: 97.84%
confusion matrix for training:

3666	36
116	3596

Testing Accuracy: 98.65%
confusion matrix for testing:

903	25
0	928

The banking dataset was previously used by Moro, Cortez and Rita in 2014. They used "using the rminer package and R tool and conducted in a Linux server, with an Intel Xeon 5500 2.27 GHz processor" (Moro, Cortez, & Rita, 2014). The results are shown below:

Final set of selected attributes.

Factor	Attributes	Description	AUC
1: Interest rate	nat.avg.rate	National monthly average of deposit interest rate	0.781
	suited.rate	Most suited rate to the client according to bank criteria	
	dif.best.rate.avg	Difference between best rate offered and the national average	
2: Gender	ag.sex	Sex of the agent (male/female) that made (outbound) or answered (inbound) the call	0.793
3: agent experience	ag.generic	If generic agent, i.e. temporary hired, with less experience (yes/no)	0.799
	ag.created	Number of days since the agent was created	
5: Client–bank relationship	cli.house.loan	If the client has a house loan contract (yes/no)	0.805
	cli.affluent	If the client is an affluent client (yes/no)	
	cli.indiv.credit	If the client has an individual credit contract (yes/no)	
	cli.salary.account	If the client has a salary account (yes/no)	
7: Phone call context	call.dir	Call direction (inbound/outbound)	0.809
	call.nr.schedules	Number of previously scheduled calls during the same campaign	
	call.prev.durations	Duration of previously scheduled calls (in *s*)	
8: Date and time	call.month	Month in which the call is made	0.810
9: Bank profiling indicators	cli.sec.group	Security group bank classification	0.927
	cli.agreggate	If the client has aggregated products and services	
	cli.profile	Generic client profile, considering assets and risk	
10: Social and economic indicators	emp.var.rate	Employment variation rate, with a quarterly frequency	0.929
	cons.price.idx	Monthly average consumer price index	
	cons.conf.idx	Monthly average consumer confidence index	
	euribor3m	Daily three month Euribor rate	
	nr.employed	Quarterly average of the total number of employed citizens	

(Moro, Cortez, & Rita, 2014)

In Moro, Cortez and Rita's experiment, the attributes are classified as 8 different factors. Each type of factor is used individually to predict the results. In other words, the possibility of product subscribed by client is affected by different factors, including interest rate, gender, agent experience, client–bank relationship, phone call context, date and time, bank profiling indicators, social and economic indicator. In Moro, Cortez and Rita's article, they considered actual factor in real world. Although those result figures are all lower than mine, those results are more truthfulness. In this case, my results are more general when compare to results in Moro, Cortez and Rita's article.

5 Conclusion and Future Work

In this report, the implemented neural network reads 20 inputs and predicts 2 outputs, contains one hidden layer and perform with 0.01 of learning rate. Four processed datasets were trained by one unchanged network, the initial numeric dataset, the normalised dataset, the pattern reduced dataset and the normalised pattern reduced dataset. Each dataset was separate into training set and testing set by 80% and 20% out of the total set respectively. After normalise dataset, the accuracy and confusion matrix were not improved. While the pattern reduction method has improved the accuracy and confusion matrix in a reasonable level. However, after read A data-driven approach to predict the success of bank telemarketing, I would like to detail analyse and categories data according to actual world for network improvement. Additionally, during my studying of the method to normalised data and the pattern reduction method, other course-mate were studying at shared weights and

6 References

Bustos, R. A., & Gedeon, T. D. (1995). Decrypting Neural Network Data: A Gis Case Study. *Artificial Neural Nets and Genetic Algorithms*, 231-234.

Chauvin, H. (1990). Dynamic Behaviour of Constrained Back-Propagation Networks. *Proc. NIPS-2*, 642-649.

Gedeon, T., & Bowden, T. (1992). Heuristic Pattern Reduction. *International Joint Conference on Neural Networks, 2*, 449-453.

Gerven , M. v., & Bohte, S. (2017). Editorial: Artificial Neural Networks as Models of Neural Information Processing. *Frontiers in Computational Neuroscience*(1662-5188), 114.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems, 62*, 22-31.

Rust, R. T., Moorman, C., & Bhalla, G. (2010). Rethinking Marketing. *Harvard Business Review, 88*(1,2), 94-101.