Journal of Financial Risk Management, 2017, 6, 364-374
http://www.scirp.org/journal/jfrm
ISSN Online: 2167-9541
ISSN Print: 2167-9533

Machine Learning Approaches to Predicting Company Bankruptcy

Wenhao Zhang

Catholic Memorial School, West Roxbury, USA
Email: wenhaozhang@catholicmemorial.org

How to cite this paper: Zhang, W. H. (2017). Machine Learning Approaches to Predicting Company Bankruptcy. *Journal of Financial Risk Management, 6*, 364-374.
https://doi.org/10.4236/jfrm.2017.64026

Received: September 11, 2017
Accepted: December 10, 2017
Published: December 13, 2017

Abstract

Machine Learning has undergone a tremendous progress, which is evolutionary over the last decade. It is widely used to make predictions that lead to the most valuable decisions. Many experts in economics use models derived from Machine Learning as important assistance, and many companies would use Neural Network, a model in bankruptcy prediction, as their guide to prevent potential failure. However, although Neural Networks can process a tremendous amount of attribute factors, it results in overfitting frequently when more statistics is taken in. By using K-Nearest Neighbor and Random Forest, we can obtain better results from different perspectives. This paper testifies the optimal algorithm for bankruptcy calculation by comparing the results of the two methods.

Keywords

Neural Networks, Random Forest, KNN, Bankruptcy Prediction

1. Introduction

Since the discovery of the very first machine learning algorithms, machine learning has unprecedentedly developed. New models emerge out of obscurity. Business companies nowadays are struggling to determine models for a better prediction. As a consequence, there has been a growing interest in data mining models, which are capable of predicting results from a huge amount of data.

K Nearest Neighbor (KNN) is an algorithm that classifies all available statistics based on a similarity measurement (Saravanan, 2010). First introduced in the 1970s, KNN is commonly used in statistic estimation and pattern recognition fields (Jóźwik, 1983). The NN rule is first called "minimum distance classifier" or "proximity algorithm" by Sebestyen, an expert in neurocomputing field. Essentially, it is based on a fundamental principle called Ockham's razor: a heuristic guide for scientist to build theoretical models. Conceptually understandable, KNN is able to solve complex tasks with relatively small amount of statistics.

Random Forest (RF) is another classification method based on decision tree. The first related algorithm is called "random subspace method". Created by Tin Kam Ho (Ho, 1995), it is able to reduce the correlation of the estimators by training the model with random samples. An extension of this algorithm is created by Leo Breiman who was a statistics professor at University of California Berkeley. He combined several methods and constructed a collective decision tree (a tree-like model that graphs possible consequences) (Beriman, 2001). Instead of growing one decision tree, random forest generates ensemble of unique trees and rolls out the most popular class by voting. This makes random forest one of the most accurate algorithms. However, this accuracy has the problem of overfitting which occurs when the model predicts well using the training dataset but performs unsatisfying with some noisy classification tasks.

Neural Networks (NN) is a computing system inspired by animal brains that can improve the model performance progressively as the model is trained. It is mainly used in such fields as image recognition and social networks (Nielsen, 2015). Neural Networks consist of a collection of artificial neurons and connections are made to transmit signals among them. A typical model has several layers which perform different transitions, where the signal would travel from the input layer to the output layer and bypassing many hidden layers. Neural Networks can fit into any functions regardless of its linearity, but, as this paper will discuss, it is often used in cases where easier solutions would perform better.

2. Motivation

Economy is a major part of our society, and we rely heavily on businesses that form our economy. However, major problems such as over-speculation, the belief that the value of one company will always get better, harm the entire system. Thus, a way of modeling a company's performance and predicting its potential bankruptcy rate is necessary, since it would allow people to notice and decide whether they will invest in or quit the company.

Several models have already been established to predict the rate of bankruptcy, but many of them rely on a large amount of databases and perform poorly. In another scenario, these models use more complicated algorithms for simple calculations. For example, the neural networks, which have been increasingly popular in the recent decade, perform a high accuracy, especially with lots of hidden layers and the help of dropouts (to randomly ignore some nodes in the hidden layer). However, when it comes to relatively simple tasks, overfitting occurs constantly because of such accuracy. Therefore, K Nearest Neighbor along with Random Forest and Neural Networks are tested in the paper to find out a better algorithm that can avoid such problem.

3. Model Description

3.1. K Nearest Neighbor

In this algorithm, let the point labeled be x, and label the point closest ($k = 1$) or numerous points closest ($k = n$) to x be y. When there is a vast amount of data x, y will possibly be the same. For example, when a baised coin is tossed (the chance of getting one result is higher than the other) one million times, the result is nine hundred thousand times heads. Predictably, the next toss is very likely to be head. In this case, KNN uses a similar method.

$$C(x) = Y(1)$$

In Figure 1, K nearest neighbor classifier classifies this dataset into three groups of clusters represented by three different colors. The arrows are pointing to X's nearest neighbors. In this case, X is classified as orange because it has the most nearest neighbors in the K value set ($K = 5$ in the figure).

The classifier C assigns x to its closest neighbor Y depending on the value of k (in this case $k = 1$). As the size of data set increases, the error rate would be guaranteed to be less than twice of the Bayes error rate (The minimum error rate based on the data's distribution).

$$P^* \le P \le P^*\left(2 - \left(C/(C-1)\right)P^*\right)$$

Above is the formula for obtaining the tight error boundary. P^* is the Bayes error rate, C stands for the number of classes, and P is the nearest neighbor method's error rate. For example, when there is a large amount of data, Y is the nearest neighbor, and X needs to be classified, it is very likely for X to be classified with Y. The chance of getting an error from this classification will be greater than the minimum rate based on the population and less than two times that error rate, which is shown in the formula above.

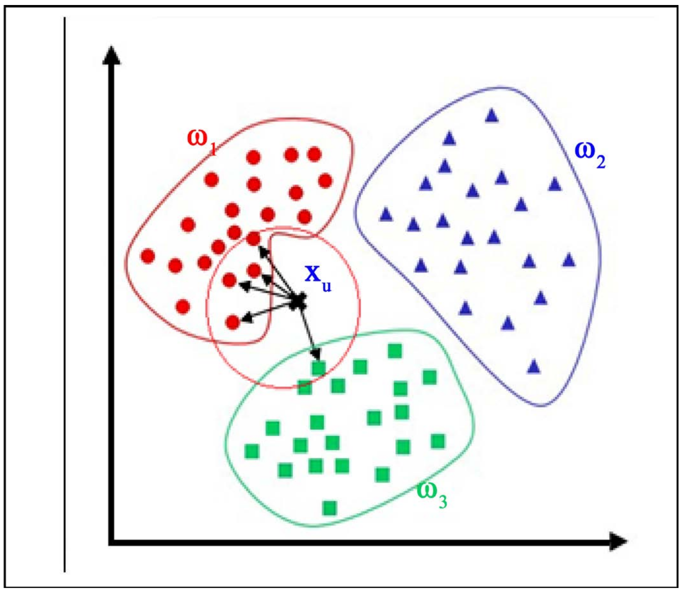

Figure 1. An illustration of K nearest neighbor model.

3.2. Random Forest

Random Forest can obtain a very accurate and precise result by planting huge amount of decision trees.

Training sets would be selected at random and be randomly distributed along X and Y vectors.

Then, tree bagging method (bootstrap aggregating) will choose a sample with replacement at random.

In Figure 2, random forest classifier plants several trees to form a forest. By calculating the mean of the results from the forest, the classifier generates a majority voting outcome, which is the final result of the classification.

For example, when selecting for B times, tree bagging would have B training samples from X and Y called X_b and Y_b. Next, it will make a decision tree fb based on X_b and Y_b. Therefore, after these trainings, the sample could be predicted from the mean number of the trees, with the formula:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b\left(x'\right)$$

The majority vote would be the result of the classification. This method strengthens the performance of the model by limiting the variances without increasing the degree of bias. In the other hand, in the case of training with few and noisy datasets, tree bagging the model would perform better by decreasing the correlation between the trees.

Random Forest differs in the learning process for each individual tree. It uses a modified learning process called feature bagging. Different from tree bagging, feature bagging allows each decision tree to have a random subset of features. By doing so, it is able to intentionally prevent some significant predictor features and therefore make them correlated.

3.3. Neural Network

A basic neural network consists of three parts, as shown in Figure 3: the input layer (layer L_1), the hidden layers (layer L_2, L_3), and the output layer (layer L_4).

We can view the model as $f\left(x\right): x \to y$

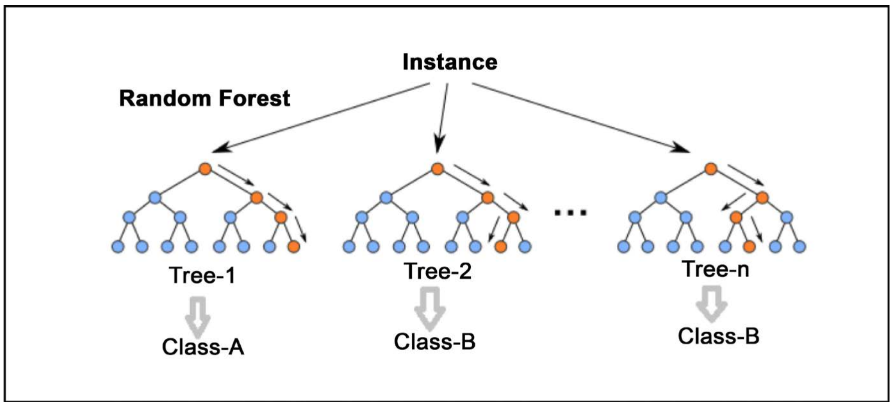

Figure 2. An illustration of random forest model.

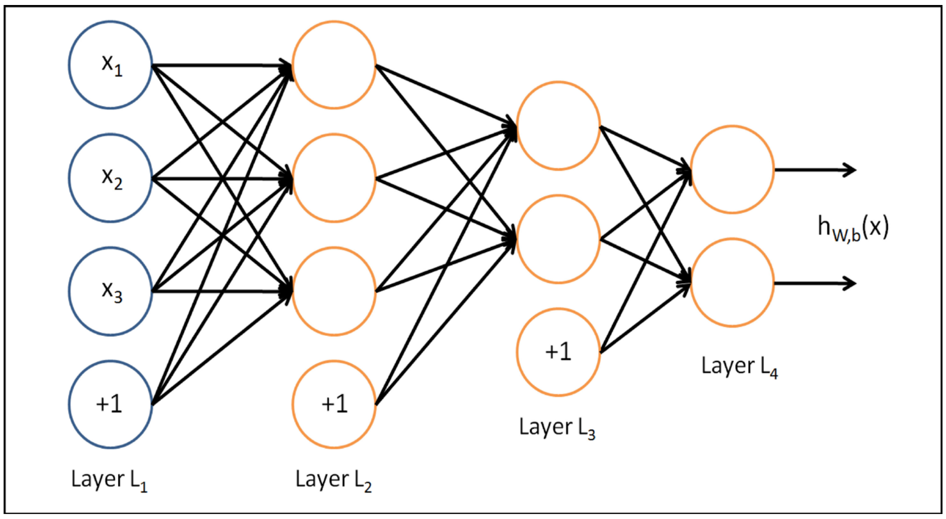

Figure 3. An illustration of neural network model.

$$f(x) = K\left(\Sigma W_i G_i(x)\right)$$

K in this equation is also known as the activation function, which is usually a defined function, such as the hyperbolic tangent function. The activation function helps address the values as the input layer change.

Each layer has its weight and the weight changes as the model goes through the hidden layers until it reaches the output layer. The components of each layer are completely independent of each other. As Figure 3 shows, more hidden layers would result in different outputs.

3.4. Dropouts

In another task, researchers from University of Toronto including Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov introduced the dropout method to neural networks. Normally, neural networks contain many hidden layers to express complicated relationships. However, when it is applied to test data with noisy training data, the models don't perform as well as that for the training datasets. Dropout is one of the methods designed mainly for reducing the amount of computation and decreasing the chance of overfitting. It is proved that dropouts can effectively help a large model to achieve these goals by intentionally ignoring random units of random layers (Srivastava et al., 2014).

As shown in Figure 4, Dropout method can avoid overfitting and provide a way of combing exponential architectures for neural networks. This method temporarily removes randomly hidden or visible units with their income and outcome connections. When applying dropouts, the training dataset can be viewed as a smaller version of the testing dataset. For example, n units of neural networks can be viewed as a collection of 2^n quantities of potential neural networks, yet, these networks still share the same weights so the number of total parameters is still the same, n^2.

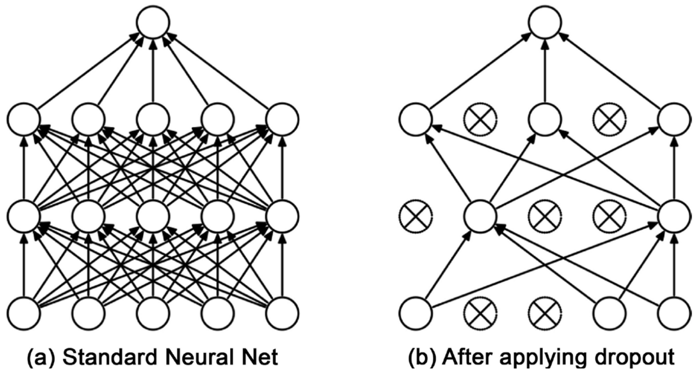

Figure 4. An illustration of neural network with and without dropout.

4. Related Works

Bankruptcy prediction has become very popular during the last decade. Myoung-Jong Kim and Ingoo Han have done a research on the same topic with different models. They used the same database with three different approaches for bankruptcy prediction, and focused especially on the efficiency of quantitive data mining for representing the experts' precision. The first approach is to establish quantitative models for a data mining understanding. This method specifically uses classifiers made up of a set of weights within the economic variables, such as discrimination analysis and neural networks. The second approach is to automatically screen bankruptcy prediction which rules out of the vast amount of population. The last approach is to use subjective models for data mining. This represents the experts deciding subjectively because they weight things differently and they take subject things into account. According to Kim and Han's work, Neural Networks is capable of extracting experts' decision rules out of their quantitative bankruptcy decisions (Kim & Han, 2003). In this paper, we aim at getting a better bankruptcy prediction through a quantitative approach. Different models such as KNN and RF are used to compare with the neural network model used in past papers.

5. Data Output

5.1. Test Datasets

Table 1 shows data published by Sebastian Tomczak, a researcher at University of Science and Technology, Poland. This dataset predicts bankruptcy of Polish companies. To get the data, the already bankrupted companies were evaluated in the period of 2000-2012, and the other companies were studied from 2007-2013.

5.2. Related Works

As shown in Table 2, truncate is imported for this task, which would only use the first quantity of the number previously set. Each algorithm would use truncate equal to 50 and 150, which means it would cut off the first 50 and 150 samples and later be used for accuracy calculation.

Table 1. UCI machine learning repository, Polish companies bankruptcy data set.

Data Set	Data Size	Training set percentage	Test set percentage	Validation set percentage
Bankruptcy Rates	250	80%	10%	10%

Table 2. The discovery of experts' decision rules from qualitative bankruptcy data using genetic Algorithms, Myoung-Jong Kim & Ingoo Han.

Technique	Data Size	Rules Extracted	Overall Accuracy
Genetic Algorithms	232	11	0.940
Inductive Learning	232	16	0.897
Neural Networks	232	12	0.903

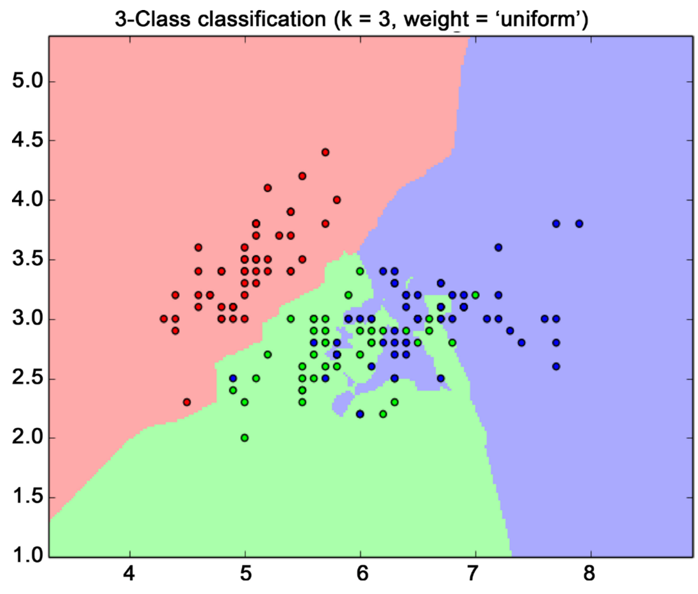

Figure 5. Visualization of KNN model with K value set to be 3.

5.3. K Nearest Neighbors

K value in Figure 5 is set to be 3. According to the figure, 3 appears to be a good k value because the classifier is able to classify many datasets outside of the majority neighbors.

K value in Figure 6 is set to be 5. According to the figure, 5 is a relatively alternative value for k. Although it is able to classify datasets, it lacks precision comparing to k value of 3.

K value in Figure 7 is set to be 7. According to the figure, 7 is the least alternate value for k among three experiments, because it is able to classify the least dataset that is outside of the majority neighbors.

Table 3 shows the result of accuracy rate based on the KNN calculation.

5.4. Random Forest

As shown in Table 4, a truncate of 50 results in an accuracy rate of 0.975, and that of 150 results in an ac-curacy rate of 0.99, which is relatively high.

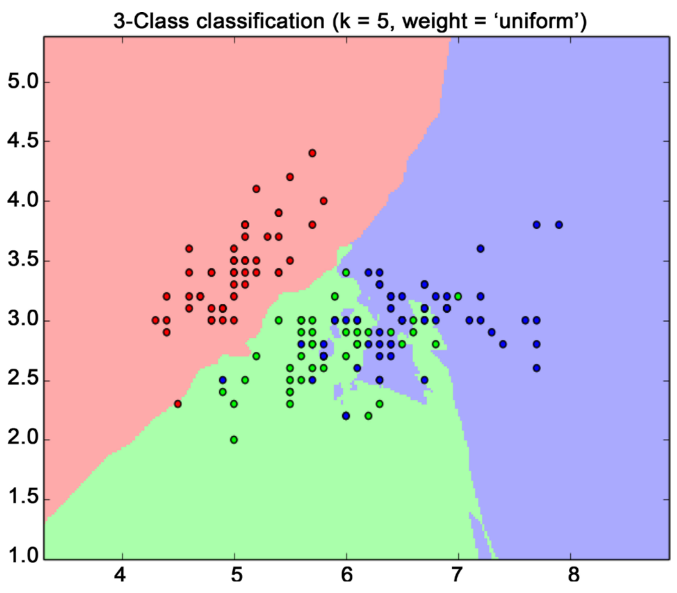

Figure 6. Visualization of KNN model with K value set to be 5.

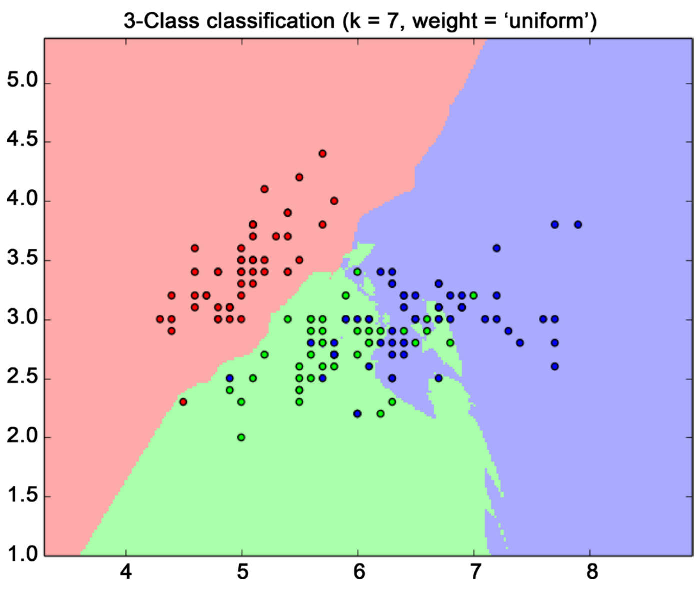

Figure 7. Visualization of KNN model with K value set to be 7.

Table 3. KNN classification result KNN classification results.

K	Truncate	Accuracy
3	50	0.995
3	150	0.98
5	50	0.985
5	150	0.97
7	50	0.98
7	150	0.94

Table 4. Random Forest classification results.

Truncate	Accuracy
50	0.975
150	0.99

Table 5. Neural Networks without dropout results.

Truncate	Accuracy	Loss
50	0.984	0.1015
150	0.98	0.0344

Table 6. Neural Networks with dropout results.

Dropout Rate	Truncate	Accuracy	Loss
0.5	50	0.955	0.1332
0.5	150	0.98	0.0312
0.3	50	0.995	0.0606
0.3	150	0.97	0.0366

5.5. Neural Networks

Table 5 shows the results of accuracy and loss rate using Neural Networks when truncate is 50 and 150 respectively.

Table 6 shows the results of accuracy and loss rate using Neural Networks when truncate is 50 and 150 respectively, with two dropout rates applied. The first dropout rate tested is 0.5, and the second dropout rate is 0.3. The results of Nerual Networks model are visualized in Figure 8 and Figure 9.

6. Conclusion

Overall, the three algorithms used (KNN, RF, and NN) perform a higher accuracy comparing to the three approaches used by Myoung-Jong Kim and Ingoo Han. Kim and Han focused too much on achieving different approaches to get an expert's decision. However, they neglected the problem of overfitting. As shown in Figures 5-7, KNN performs the best when K is 3 with truncate of 50 because the dataset is relatively small. Specifically, a smaller K value provides a more precise prediction. Similarly, RF performs a high accuracy too. As reflected in Table 4, RF obtains a 0.99 accuracy when truncate is 150. This is mainly because RF grows many trees. Essentially, the accuracy would increase as the training dataset increases. Lastly, as we can tell from Table 5 and Table 6, NN performs higher accuracy when dropout is applied. This method works the best with a 0.3 dropout rate with truncate equal to 150, which increases the accuracy and remains the loss value low. Figure 8 and Figure 9 visualize why the method works the best with a 0.3 dropout rate by drawing the results shown in Table 6. In conclusion, KNN, RF, and NN perform well when it comes to the accuracy

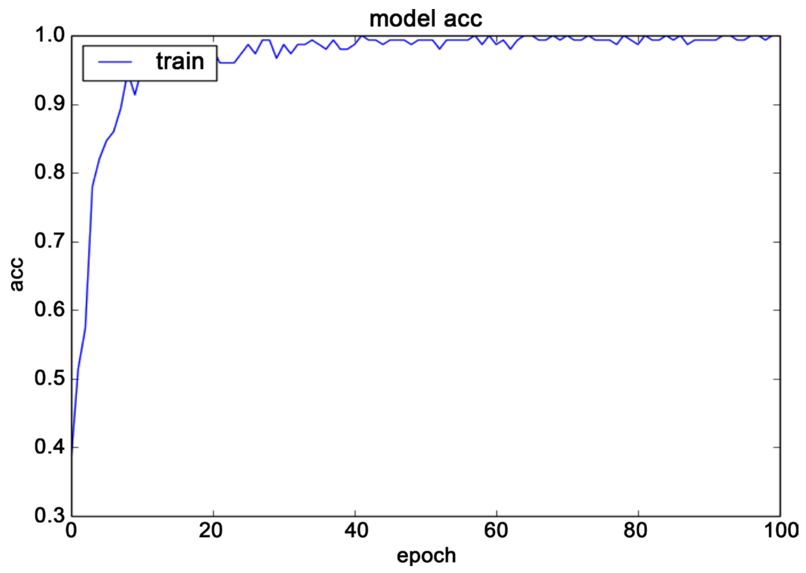

Figure 8. Visualization of neural network accuracy rate (Truncate = 150, dropout = 0.3).

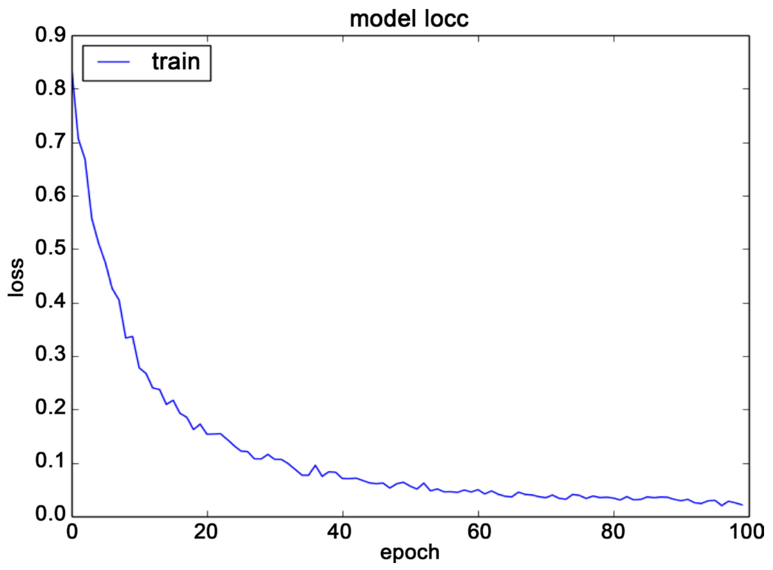

Figure 9. Visualization of neural network loss function rate (Truncate = 150, dropout = 0.3).

rate. However, more future research should be done to improve these models. For instance, autoencoders are methods that help reduce the dimensions of the dataset, which can further reduce the amount of calculation. It is especially useful for tasks that require large calculations, and it has the potential to be used widely for unsupervised learning and many other fields yet to be studied.

References

Beriman, L. (2001). *Random Forest*. Berkeley, CA: University of California.

Ho, T. K. (1995). Random Decision Forests. *Proceedings of the Third International Conference on Document Analysis and Recognition, 1,* 278-282. https://doi.org/10.1109/ICDAR.1995.598994

Jóźwik, A. (1983). A Learning Scheme for a Fuzzy-NN Rule. *Pattern Recognition Letters, 1,* 287-289. https://doi.org/10.1016/0167-8655(83)90064-8

Kim, M., & Han, I. (2003). The Discovery of Experts' Decision Rules from Quantitative Data Using Genetic Algorithms. *Expert Systems with Applications, 25,* 637-646. https://doi.org/10.1016/S0957-4174(03)00102-7

Nielsen, A. M. (2015). *Neural Networks and Deep Learning*. San Francisco, CA: Determination Press.

Saravanan, T. (2010). *A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm*. God. Your Book Is Great. https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research, 15,* 1929-1958.