Learning Algorithm for Threshold Softmax Layer to Handle Unknown Class Problem

Gaurav Jaiswal

Abstract Neural network are mostly trained with predefined class training data in supervised learning. But, when unknown test data (other than predefined class) are classified by a trained neural network, they are always misclassified into predefined classes, thus misclassification rate of trained neural network increases. To tackle these problems, Threshold Softmax Layer (TSM) and learning algorithm is proposed. In which, a normalized probability of each output class of the neural network is calculated and a threshold value is updated for each class during threshold learning process. If the maximum normalized probability of test data does not cross threshold value of the corresponding class, we will classify test data into unknown class. This TSM layer with neural network is evaluated on three UCI benchmark dataset (Glass, Yeast and Wine quality) and successfully handles the unknown class problem with reduced misclassification error.

Keywords Threshold softmax layer · Unknown class · Open set classification · Neural network · Softmax function · Misclassification error

1 Introduction

Neural network-based classification models such as MLP, BPNN, CNN and ELM are widely used in various fields [26, 37], i.e. computer vision [7, 21, 29], health sector [1, 24], industrial sector [23, 33], OCR [8, 17], biometric [15], financial sector [27, 34], etc. for classification of real-world data. These models are generally trained on specific predefined class data for specific tasks. One of the main limitations of these classification models is that it always classifies any unknown input data into predefined output class, whether input data belongs to the predefined class or not [26, 37]. This always increases the misclassification rate of classification model.

G. Jaiswal (✉)
ICT Research Lab, Department of Computer Science, University of Lucknow, Lucknow, India
e-mail: gauravjais88@gmail.com

M. D. Borah et al. (eds.), *Big Data, Machine Learning, and Applications*, Lecture Notes in Electrical Engineering 1053, https://doi.org/10.1007/978-981-99-3481-2_13

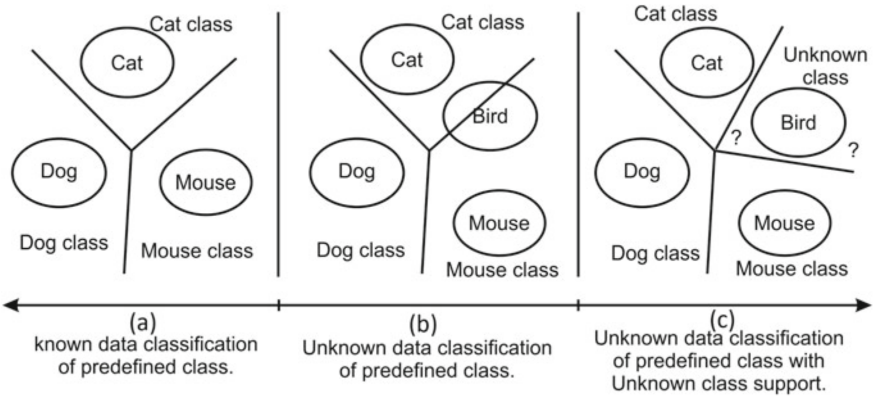

Fig. 1 Unknown class classification problem

To understand classification problem, suppose a classifier is trained with three predefined class training data (dog, cat and mouse). When the input domain of classifier is closed to the predefined class, classifier performs well for closed input domain. If input data (bird), other than predefined class, is given to this trained classifier, bird data will be classified into one of the predefined classes (dog, cat and mouse) while bird input data does not belong to any of predefined class. Therefore, misclassification error of classifier is increased. This problem is depicted in Fig. 1.

Traditional and regular classifiers only work in closed class environments to avoid these misclassification errors for unknown class data. For generalization of classifiers for open set environments, various researchers explored the open set classification domain and developed various models to handle the unknown class problem [2, 3, 18, 30, 31]. In this paper, threshold softmax layer is introduced and threshold learning algorithm is proposed for handling unknown class problem in classification.

Proposed threshold softmax layer (TSM Layer) and learning algorithm reduce misclassification rate of neural network. This layer is based on threshold of normalized probability of each class. Threshold value works as decision criteria for classification to unknown class. This layer is deployed in the neural network to improve unknown class classification capability of the neural network.

This paper is organized as follows: Sect. 2 reviews the related works in literature. Section 3 describes the concepts of threshold softmax layer with application to handle unknown test data. TSM layer is implemented and improved neural network is evaluated on three UCI benchmark classification dataset in Sect. 4. Finally, paper is concluded in Sect. 5. Symbols used in this paper are given in Table 1.

Table 1 Symbols and Descriptions

Sr. No.	Symbols	Descriptions
1	$Y = [y_1, y_2, \ldots, y_k]$	Output vector
2	$\sigma(Y)$	Softmax function
3	e^{y_i}	Un-normalized probability of of ith value (y_i) of Output vector Y
4	$[X_n, Y_n]$	n Training Samples
5	T_k	Threshold vector of k length
6	T_m	mth value of Threshold vector
7	O	Output score of Neural Network
8	p	Confidence value
9	i, j, k, m	Index of vectors

2 Related Work

Various different classification models have been proposed to handle unknown class for open set classification. In literature, Gorte et al. [11] first considered this problem and proposed non-parametric classification algorithm using posteriori probability vector. Gupta et al. [13] introduced a binary tree structure called class discovery tree for dealing with unknown class. Scheirer et al. [31] formalized open set recognition problem, open space risk and openness of classification and proposed 1-vs-set machine. Afterwards, Scheirer et al. [30] proposed a novel Weilbull-calibrated SVM (W-SVM) which is based on compact abating probability and binary SVM. Rattani et al. [28] used binary and multiclass W-SVM for fingerprint spoof detection. Costa et al. [4] extended SVM using decision boundary carving algorithm for source camera attribution and device linking. Jain et al. [16] proposed Pi-SVM which was introduced for estimating the un-normalized posterior probability of class inclusion. Li et al. [22] extended nearest neighbour using transduction confidence while Júnior et al. [20] using distance ratio for open set classification.

Bendale et al. proposed nearest non-outlier algorithm [2] and OpenMax layer [3]. Ge et al. [9] extended OpenMax by applying generative adversarial networks (GANs). GANs are used to generate fake unknown data [19, 36] for training of classifier. Zhang et al. [38] simplified open set recognition problem into a set of hypothesis testing problem and proposed sparse representation-based model using extreme value theory. Gunther et al. [12] suggested thresholding extreme value machine probabilities to handle open set face recognition problem. Shu et al. [32] proposed joint open classification model with a sub-model for classifying known and unknown class. Neira et al. [25] proposed a newly designed open set graph-based optimum path forest classifier using genetic programming and majority voting fusion techniques. Xiao et al. [35] gave the idea of compact binary feature generated by ensemble binary classifier. Geng et al. [10] introduced hierarchical Dirichlet process-based model which does not overly depend on training sample. Hassen et al. [14] proposed a loss function-based neural network representation for open set recognition. Different from existing models, our proposed threshold softmax layer is an extension model of neural network which enables simple neural network to classify unknown class.

3 Threshold Softmax Layer

TSM layer combines two processes, i.e. softmax function [6] and threshold learning. Architecture of neural network with TSM layer is shown in Fig. 2. Description for Softmax function [6], Threshold learning and handling unknown class problem using TSM layer is given in Sects. 3.1, 3.2 and 3.3 respectively.

3.1 Softmax Function

Softmax function is a neural transfer function, which is used for calculating normalized probability of each output class for given an input sample [6]. Softmax function is defined as follows:

Let output vector $Y = [y_1, y_2, ..., y_k]$ of k node of the output layer of neural network, Softmax function $\sigma(Y)$ calculates predicted normalized probability of each class.

$$\sigma : Y = [y_1, y_2, \ldots, y_k] \rightarrow [0, 1]^k \tag{1}$$

Un-normalized Exponential Probability is calculated by taking power of each y_i to exponent

$$prob_i = e^{y_i}, \ \ for\ i = 1, 2, \ldots, k \tag{2}$$

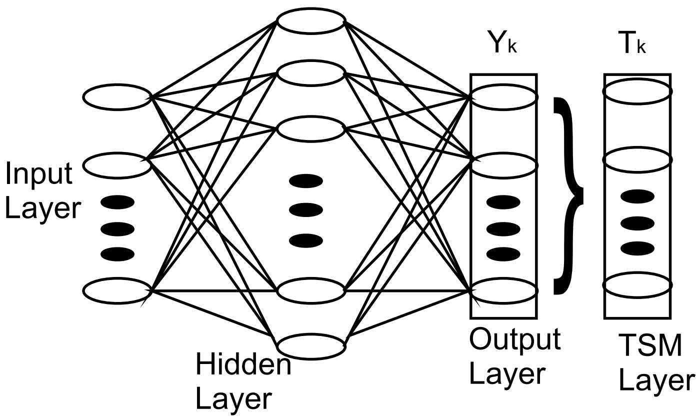

Fig. 2 Architecture of Neural network with TSM layer

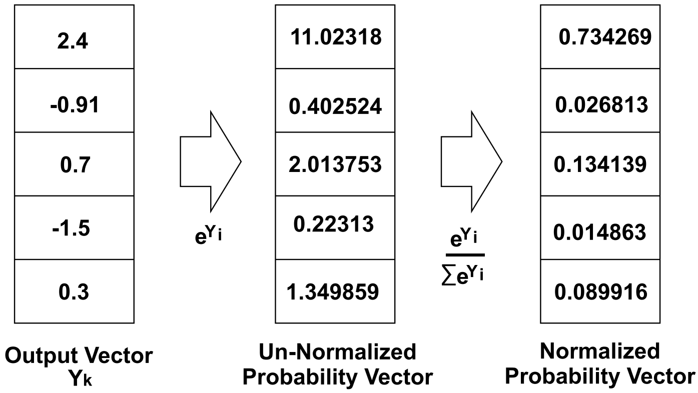

Fig. 3 Illustration of softmax function

Normalized Probability of each class is calculated by dividing each un-normalized probability by the sum of all classes' un-normalized probability

$$\sigma(Y)_i = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}}, \ for\ i = 1, 2, \ldots, k \tag{3}$$

Softmax function can be understood as it computes the class probability of each input data. Working of softmax function is illustrated in Fig. 3.

3.2 Threshold Learning Algorithm

TSM layer of neural network learns threshold values for each class during threshold learning process. When softmax function compute the normalized probability $\sigma(Y_k)$ of each class for given output layer vector Y_k, threshold learning process updates threshold value in threshold vector T_k for each class by taking the minimum of the normalized probability of actual class. A confidence value p (range 0.0–1.0) is taken from the user which is used for restricting threshold value approaching to 0 when noisy or outliers data occurs. The final threshold value is determined by taking maximum between confidence value (p) and threshold value. Working process of threshold learning is given in Algorithm 1.

3.3 Handling Unknown Class Problem Using TSM Layer

Threshold vector (Tk) is obtained by threshold learning process which contains threshold value of each predefined class. Threshold value of each class works as decision criteria for classification. If any unknown sample is given to trained improved neural network, output layer predicts the output score. In TSM layer, the normalized

Algorithm 1 Algorithm for Threshold learning

Input Training sample $[X_n, Y_n]$ with k class, Confidence Value (p)
Output Threshold Vector T_k

```
1: Initialize Threshold Vector [T_k] to [1]^k
2: for each sample [X_i, Y_i] in [X_n, Y_n] do
3:     Calculate feed forward output score [O_i] of trained Neural Network
4:     Calculate max of normalized probability σ([O_i]) using softmax function
5:     O_max = max(σ([O_i]))
6:     Update T_m in T_k for m = class value of Y_i
7:     T_m = min(O_max, T_m(prev))
8: end for
9: for each T_m in T_k do
10:    Update T_m = max(T_m, p)
11: end for
```

class probability is calculated using softmax function [6]. If max normalized probability is greater than the corresponding threshold value, then corresponding class value will assign and if not, then unknown class value will assign. The procedure of handling unknown test data is given in Algorithm 2.

For example, for any given sample, the normalized probability vector is calculated as [0.0345, 0.5293, 0.2542, 0.1820] and learned threshold vector for given confidence value (p) 0.5 is [0.5, 0.7823, 0.8246, 0.8185]. Max normalized class probability 0.5293 is less than corresponding threshold value 0.7823. So 'unknown class' is assigned for given sample.

Algorithm 2 Algorithm for Handling Unknown Class

Input Unknown sample $[X]$, Threshold Vector $[T_k]$ of predefined k class
Output Class_value

```
1: Calculate feed forward output score [O] of trained Neural Network for input [X]
2: Calculate normalized probability σ([O]) using softmax function
3: Find σ([O]_i = max(σ[O])) for i = 1, 2, ···, k
4: if σ([O]_i ≥ T_i then
5:     Assign i to Class_value
6: else
7:     Assign 'Unknown Class' to Class_value
8: end if
```

4 Experiment and Results

4.1 Experimental Setup

Neural Network with TSM layer implemented using Python and evaluated on three standard UCI machine learning dataset: Glass, Yeast and Wine quality (white) [5]. Each dataset is divided into two ratios of train data and test data in dataset A (ratio

Table 2 Description of benchmark dataset and their experimental configuration

Dataset/Properties		Glass [5]	Yeast [5]	Wine quality [5]
No. of attributes		9	8	11
No. of classes		6	10	11
No. of samples		214	1484	4898
DS A	Train (80%)	5 class data were taken	8 class data were taken	8 class data were taken
	Test (20%)	all class data were taken	all class data were taken	all class data were taken
DS B	Train (70%)	5 class data were taken	8 class data were taken	8 class data were taken
	Test (30%)	all class data were taken	all class data were taken	all class data were taken

80:20) and dataset B (ratio 70:30). Train sample of each dataset is modified by removing samples of some particular classes, so that samples of train data of that particular class can be treated as unknown class data. Training samples of five classes out of six classes are taken in Glass dataset. In Yeast dataset, training samples of eight classes out of ten classes are taken. Training samples of eight classes out of eleven classes are taken in Wine quality (white) dataset. In test data, all classes' samples are taken. Table 2 shows properties of each dataset and experimental configuration of training and test dataset.

4.2 Experimental Results

We first train a neural network with train data for each dataset. The neural network is evaluated on test data of each dataset which also contains the unknown samples. The model predicts the class value of predefined class for unknown class samples. This experiment shows that misclassification errors are higher due to unknown class sample in test data.

Therefore, the neural network with TSM layer (Same neural network configuration) train with train data for each dataset with confidence value (p) 0.5. Here, determination of confidence value (p) parameter is critical. If confidence value 0 is chosen, then most of the unknown class samples are classified into predefined class. If confidence value is chosen as 1, then all samples are classified into unknown class. So default value 0.5 is chosen. Now, neural network with TSM layer is evaluated on the test dataset. Details of experimental results and improvement in reduction of misclassification error for each dataset are tabulated in Table 3. The comparison of performance based on misclassification error of neural network (NN) and neural network with TSM layer (NN+TSM) is shown with graph in Fig. 4. It clearly shows that NN+TSM reduces the misclassification error.

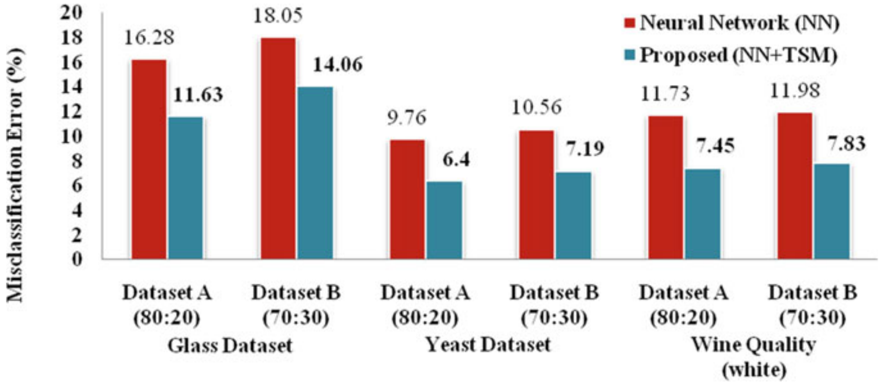

Fig. 4 Comparison of performance of NN and NN+TSM methods

Table 3 Details of experimental result of all datasets

Dataset		Misclassification error (%)		Improvement (%)	Average improvement (%)
		Neural network (NN)	Proposed (NN+TSM)		
Glass	DS A (80:20)	16.28	11.63	28.56	27.27
	DS B (70:30)	18.05	14.06	25.98	
Yeast	DS A (80:20)	9.76	6.40	34.43	33.17
	DS B (70:30)	10.56	7.19	31.91	
Wine	DS A (80:20)	11.73	7.45	36.49	36.57
	DS B (70:30)	11.98	7.83	36.64	

Table 3 shows that average improvement for Glass dataset, Yeast dataset and Wine quality dataset are 27.27, 33.17 and 36.57% respectively. These improvements show that test data included Unknown data samples are successfully classified, and misclassification errors are reduced.

5 Conclusion

The proposed Threshold learning algorithm and Threshold Softmax layer (TSM Layer) are implemented and illustrated its application to handle the unknown class problem. The experimental result shows that TSM layer reduces misclassification error of neural network by 27.27, 33.17 and 36.57% for Glass dataset, Yeast dataset and Wine quality dataset respectively. The threshold learning of this layer successfully handles the unknown class problem. This proposed work leads the classification models to perform in its specific input domain to open set domain.

References

1. Agatonovic-Kustrin S, Beresford R (2000) Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. J Pharm Biomed Anal 22(5):717–727
2. Bendale A, Boult T (2015) Towards open world recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1893–1902
3. Bendale A, Boult TE (2016) Towards open set deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1563–1572
4. Costa FDO, Silva E, Eckmann M, Scheirer WJ, Rocha A (2014) Open set source camera attribution and device linking. Pattern Recognit Lett 39:92–101
5. Dheeru D, Karra Taniskidou E (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
6. Duch W, Jankowski N (1999) Survey of neural transfer functions. Neural Comput Surv 2(1):163–212
7. Egmont-Petersen M, de Ridder D, Handels H (2002) Image processing with neural networks-a review. Pattern Recognit 35(10):2279–2301
8. Ganis M, Wilson CL, Blue JL (1998) Neural network-based systems for handprint ocr applications. IEEE Trans Image Process 7(8):1097–1112
9. Ge Z, Demyanov S, Chen Z, Garnavi R (2017) Generative openmax for multi-class open set classification. In: British machine vision conference 2017. British machine vision association and society for pattern recognition
10. Geng C, Chen S (2018) Hierarchical dirichlet process-based open set recognition. arXiv:1806.11258
11. Gorte B, Gorte-Kroupnova N (1995) Non-parametric classification algorithm with an unknown class. In: International symposium on computer vision, 1995. Proceedings. IEEE, pp 443–448
12. Günther M, Cruz S, Rudd EM, Boult TE (2017) Toward open-set face recognition. In: Conference on computer vision and pattern recognition (CVPR) workshops. IEEE
13. Gupta C, Wang S, Dayal U, Mehta A (2009) Classification with unknown classes. In: International conference on scientific and statistical database management. Springer, pp 479–496
14. Hassen M, Chan PK (2020) Learning a neural-network-based representation for open set recognition. In: Proceedings of the 2020 SIAM international conference on data mining. SIAM, pp 154–162
15. Jain LC, Halici U, Hayashi I, Lee S, Tsutsui S (1999) Intelligent biometric techniques in fingerprint and face recognition, vol 10. CRC Press
16. Jain LP, Scheirer WJ, Boult TE (2014) Multi-class open set recognition using probability of inclusion. In: European conference on computer vision. Springer, pp 393–409
17. Jaiswal G (2014) Handwritten devanagari character recognition model using neural network. Int J Eng Dev Res 901–906
18. Jaiswal G (2021) Performance analysis of incremental learning strategy in image classification. In: 2021 11th international conference on cloud computing, data science and engineering (Confluence). IEEE, pp 427–432
19. Jo I, Kim J, Kang H, Kim YD, Choi S (2018) Open set recognition by regularising classifier with fake data generated by generative adversarial networks. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2686–2690
20. Júnior PRM, de Souza RM, Werneck RDO, Stein BV, Pazinato DV, de Almeida, WR, Penatti OA, Torres RDS, Rocha A (2017) Nearest neighbors distance ratio open-set classifier. Mach Learn 106(3):359–386

21. Kannojia SP, Jaiswal G (2018) Ensemble of hybrid cnn-elm model for image classification. In: 2018 5th international conference on signal processing and integrated networks (SPIN). IEEE, pp 538–541
22. Li F, Wechsler H (2005) Open set face recognition using transduction. IEEE Trans Pattern Anal Mach Intell 27(11):1686–1697
23. Lu CH, Tsai CC (2008) Adaptive predictive control with recurrent neural network for industrial processes: an application to temperature control of a variable-frequency oil-cooling machine. IEEE Trans Ind Electron 55(3):1366–1375
24. Miller A, Blott B et al (1992) Review of neural network applications in medical imaging and signal processing. Med Biol Eng Comput 30(5):449–464
25. Neira MAC, Júnior PRM, Rocha A, Torres RDS (2018) Data-fusion techniques for open-set recognition problems. IEEE Access 6:21242–21265
26. Paliwal M, Kumar UA (2009) Neural networks and statistical techniques: a review of applications. Expert Syst Appl 36(1):2–17
27. Raghupathi W, Schkade LL, Raju BS (1991) A neural network application for bankruptcy prediction. In: Proceedings of the twenty-fourth annual hawaii international conference on system sciences, vol 4. IEEE, pp 147–155
28. Rattani A, Scheirer WJ, Ross A (2015) Open set fingerprint spoof detection across novel fabrication materials. IEEE Trans Inf Forensics Secur 10(11):2447–2460
29. Schalkoff RJ (1989) Digital image processing and computer vision, vol 286. Wiley, New York
30. Scheirer WJ, Jain LP, Boult TE (2014) Probability models for open set recognition. IEEE Trans Pattern Anal Mach Intell 36(11):2317–2324
31. Scheirer WJ, de Rezende Rocha A, Sapkota A, Boult TE (2013) Toward open set recognition. IEEE Trans Pattern Anal Mach Intell 35(7):1757–1772
32. Shu L, Xu H, Liu B (2018) Unseen class discovery in open-world classification. arXiv:1801.05609
33. Widrow B, Rumelhart DE, Lehr MA (1994) Neural networks: applications in industry, business and science. Commun ACM 37(3):93–106
34. Wong BK, Selvi Y (1998) Neural network applications in finance: a review and analysis of literature (1990–1996). Inf Manag 34(3):129–139
35. Xiao H, Sun J, Yu X, Wang L (2018) Compact binary feature for open set recognition. In: 2018 13th IAPR international workshop on document analysis systems (DAS). IEEE, pp 235–238
36. Yu X, Sun J, Naoi S (2018) Generative adversarial networks for open set historical chinese character recognition. Electron Imaging 2018(2):1–5
37. Zhang GP (2000) Neural networks for classification: a survey. IEEE Trans Syst, Man, Cybern, Part C (Applications and Reviews) 30(4):451–462
38. Zhang H, Patel VM (2017) Sparse representation-based open set recognition. IEEE Trans Pattern Anal Mach Intell 39(8):1690–1696