International JOURNAL OF CONTENTS Vol.18, No.4 December 2022

[2022. 12]

Probabilistic Target Encoding of Neural Networks with Softmax Output Nodes

Sang Hoon Oh[1,*]

[1]Mokwon University; Professor; ohsanghoon16@gmail.com
*Correspondence

The Korea Contents Association

ISSN:1738-6764, eISSN:2093-7504

Probabilistic Target Encoding of Neural Networks with Softmax Output Nodes

Sang Hoon Oh [1,*]

[1] Mokwon University; Professor; ohsanghoon16@gmail.com
* Correspondence

https://doi.org/10.5392/IJoC.2022.18.4.102

Manuscript Received 08 August 2022; Received 09 December 2022; Accepted 09 December 2022

Abstract: *When training neural networks with softmax outputs, one-hot target encoding is commonly used due to its simplicity. This strategy does not incorporate the probability that an input sample belongs to a certain class but adopts "one" or "zero" as the desired output values of neural networks. Instead of the most prevalent one-hot encoding, this paper proposes a probabilistic target encoding to prevent the overfitting of neural networks to training samples. This effect brings about the accuracy improvement of test samples. We demonstrated the effectiveness of the proposed target encoding through simulations of multi-layer perceptrons and convolutional neural networks for various classification problems such as handwritten-digit recognition, isolated-word recognition, image classification, and object recognition tasks. The simulation results show that the proposed probabilistic target encoding is superior to one-hot encoding as it prevents the overfitting of neural networks to training samples.*

Keywords: Target Encoding; One-Hot Encoding; Probabilistic Target; Softmax Function; Neural Networks

1. Introduction

When we apply machine learning models including feed-forward neural networks and deep neural networks for classification problems, there must be a target vector of output nodes where the class where an input sample originates from is coded [1]. A widespread target encoding strategy is the one-hot encoding of labels in the scheme of one-versus-all due to its simplicity [2]. However, the desired values "one" or "zero" in the one-hot encoding correspond to the extreme values of the softmax function. Therefore, training neural networks with softmax outputs in order to minimize a loss function between real and desired output values causes the overfitting of neural networks to training samples.

As an alternative to the one-hot encoding, the one-versus-all encoding scheme can be reduced to low-dimensional space for faster training convergence while preserving accuracy [2-4]. The representative method is ECOC(error-correcting output codes) based on coding and decoding steps that reveal the relationship between a low-dimensional codeword and a one-versus-all codeword assigned to each class [5]. Rodriguez et al. introduced a decoder between the weighted sums of the final layer corresponding to the low-dimensional codeword space and the softmax functions corresponding to the one-versus-all codeword space [2]. That is, the weighted sums of the final layer are decoded into the one-versus-all space and then fed into the softmax function. The softmax outputs indicate the predicted label. Whole parameters of neural networks are updated to decrease the cross-entropy loss function between the predicted label and the ground truth label. Although this strategy can decrease the dimension of the final layer, the desired values of softmax output nodes are still encoded with one-hot. Therefore, ECOC can not moderate the overfitting problem.

After successful training of neural networks, the outputs of neural networks can be interpreted as a posterior probability that an input sample belongs to a certain class [6-10]. If we incorporate the posterior probability in the target encoding strategy, we can expect prevention of overfitting to training samples and consequently attain performance improvement for test samples. From this point of view, this paper proposes a probabilistic encoding of targets in which desired values are related to a posterior probability.

In this paper, section 2 proposes the probabilistic target encoding scheme after a short introduction of the one-hot target encoding. In section 3, we conduct the training of multi-layer perceptrons and convolutional neural networks with various classification problems to verify the effectiveness of the proposed encoding scheme. Finally, section 4 concludes this paper.

2. Probabilistic Target Encoding

We train neural networks to minimize a loss function defined by a function of desired and real output nodes' values. In classifications, the representative loss function is the cross-entropy function defined by

$$E_{CE} = -\sum_{k=1}^{M} t_k \ log \ y_k, \tag{1}$$

where M is the number of output nodes, y_k is the k-th output node's value and t_k is its desired value [11]. We usually use the softmax activation function for output nodes of neural networks given by

$$y_k = \frac{e^{\hat{y}_k}}{\sum_{j=1}^{M} e^{\hat{y}_j}} \quad (k = 1,2, \dots, M) \tag{2}$$

where $\hat{y}_j$ is the weighted sum or net input to y_j [11].

Let $\boldsymbol{x}$ be an input vector and $\boldsymbol{t} = [\ t_1, t_2, \dots, t_M]^T$ be the target vector corresponding to an input $\boldsymbol{x}$. In the scheme of one-hot target encoding for classification problems, the target vector is coded as follows:

$$t_k = \begin{cases} 1, & \text{if } \boldsymbol{x} \text{ originates from class } k \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The desired values in (3) are "one" or "zero" which are the extreme values of the softmax function. Parameters of neural networks are continuously updated to decrease distances between the desired and real output nodes' values. Thus, minimizing the cross-entropy loss function given by (1) overfits neural networks to training samples and consequently degrades the classification performance to test samples [1].

In the limit that the number of training samples goes to infinity and neural networks are trained to minimize a loss function, we can interpret y_k as a posterior probability that an input vector belongs to the class k [6-10]. Therefore, it will be better to use a posterior probabilistic value as the desired value in the target encoding. This strategy can prevent the overfitting of neural networks to training samples and leads to an improvement in the classification performance for test samples [1,12,13]. In this sense, we propose a probabilistic encoding of the target vector as follows:

$$t_k = \begin{cases} Q_k(\boldsymbol{x}), & \text{if } \boldsymbol{x} \text{ originates from class } k \\ \dfrac{1 - Q_k(\boldsymbol{x})}{M - 1}, & \text{otherwise.} \end{cases} \tag{4}$$

Here, $Q_k(\boldsymbol{x})$ denotes the posterior probability that $\boldsymbol{x}$ originates from class k. (4) satisfies the constraint of $\sum_{k=1}^{M} t_k = 1$, which is essential for the cross-entropy loss function with softmax outputs. That is, the derivative of the cross-entropy loss function with respect to $\hat{y}_k$ is derived with the constraint $\sum_{k=1}^{M} t_k = 1$ and the derivative results for whole output nodes are backpropagated to update parameters of neural networks. Also, $Q_k(\boldsymbol{x}) = 1$ corresponds to the one-hot encoding. That is, the one-hot encoding is the extreme case of the proposed probabilistic target encoding.

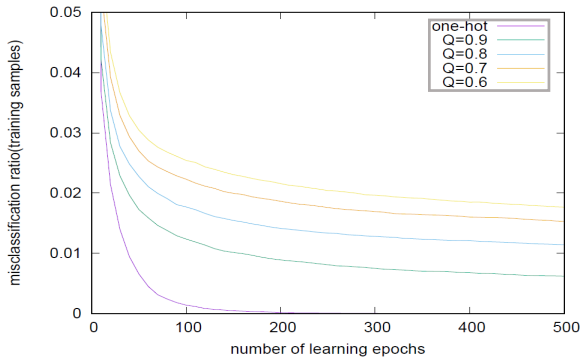
It is natural to anticipate better classification performance of test samples by using the probabilistic encoding of targets than by using the one-hot encoding with extreme values. However, estimating the posterior probability for whole training samples is a very difficult problem. Therefore, in simulations, we use 0.9, 0.8, 0.7, and 0.6 as $Q_k(\boldsymbol{x})$ for the probabilistic target encoding in (4) and the results are compared with the one-hot encoding.

There has been an n-th order extension of the binary cross-entropy loss function to prevent overfitting to training samples [1]. However, we can not use the extended cross-entropy loss function for neural networks with softmax output nodes since the extended cross-entropy loss function was proposed only for neural networks with sigmoid output nodes. Also, there are regularization methods such as dropout [14,15] and weight decay [16] to alleviate the overfitting of neural networks to training samples. In the training stage, the dropout technique randomly omits hidden nodes of neural networks to degrade the classification performance for training samples. In the test stage, we should estimate the equivalent value of each hidden node to that in the training stage using the dropout probability [14,15]. The weight decay technique is incorporated into the loss

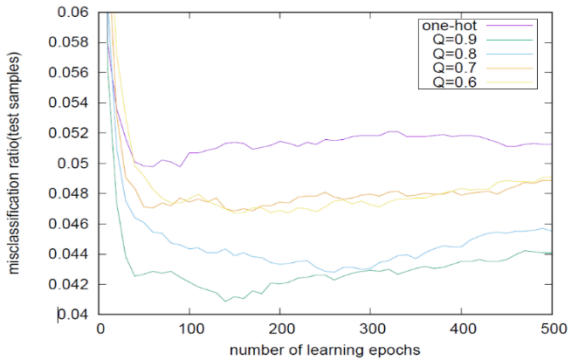
function in the form of weights norm [16]. Updating equations of weights to minimize the loss function have the term to minimize weights norm. Thus, the weight decay technique degrades the classification performance for training samples and anticipates the performance improvement for test samples. On the contrary, the proposed encoding method achieves the overfitting prevention effect with a simple target encoding scheme without any additional burden on the training algorithm of neural networks.

3. Simulations

To verify the effectiveness of the proposed probabilistic target encoding, we experiment with multilayer perceptrons (MLP's) and convolutional neural networks (CNN's) on various classification problems. In each experiment, we conduct nine times of simulations with different initial weights and the average is evaluated to draw a curve. We firstly experiment with an MLP on CEDAR handwritten digit recognition problem [17]. A digit image consists of 12×12 pixels and each pixel takes on integer values from zero to 15. The MLP consists of 144 inputs, 30 hidden nodes with sigmoid activation function, and ten softmax output nodes. A total of 18468 images are used for the training of MLP by the error back-propagation (EBP) algorithm with a fixed learning rate of 0.01 [18]. Figure 1(a) and 1(b) show the misclassification ratios for 18468 training samples and 2213 test samples, respectively. In Figures, "Q" denotes the proposed probabilistic target encoding with various $Q_k(\boldsymbol{x})$ values and "one-hot" denotes the prevalent one-hot target encoding. Here, we use the maximum rule for classification which means that the index of the maximum output node represents the classification results. Although the one-hot target encoding is the best for the training samples among simulation methods as shown in Figure 1(a), it is the worst for the test samples as shown in Figure 1(b). This is due to the overfitting of MLP to training samples. On the contrary, the proposed probabilistic encoding attains better performance for the test samples than that of the one-hot encoding. Thus, the probabilistic encoding alleviates the overfitting of MLP to training samples and improves the classification performance for test samples.



(a)



(b)

Figure 1. Simulation results of MLP's with CEDAR handwritten digit recognition task. "Q" denotes probabilistic target encoding with various $Q_k(\boldsymbol{x})$ values and "one-hot" denotes one-hot target encoding: (**a**) the misclassification ratio for the training samples; (**b**) the misclassification ratio for the test samples.

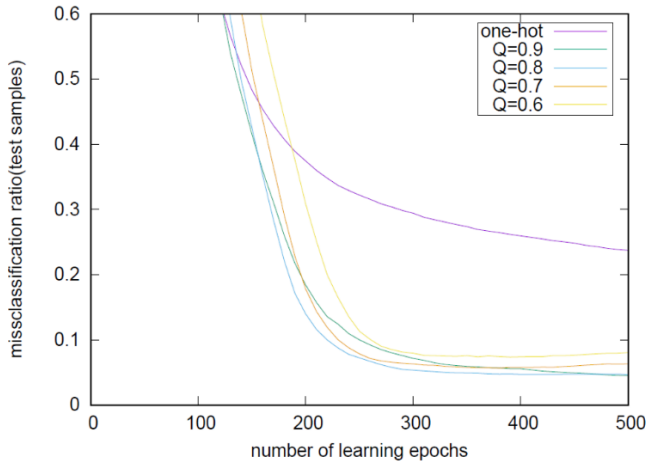


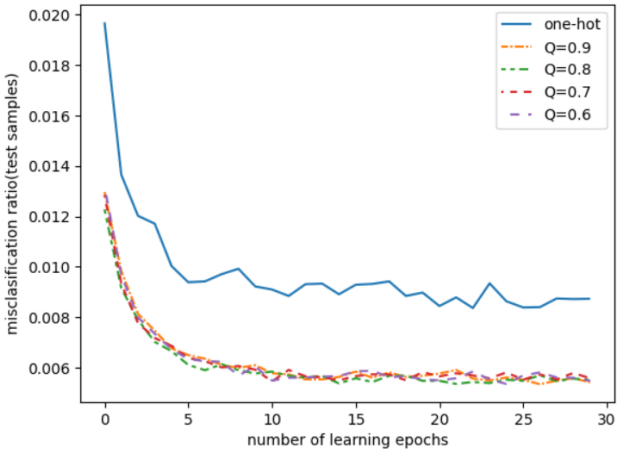
Figure 2. The misclassification ratio for test samples on the isolated-word recognition task.

The second problem of experiments is the isolated-word recognition task in which the vocabulary consists of 50 Korean words which seem to be necessary for the control of electric home appliances [19]. The utterances from 16 male speakers were sampled at 11.025kHz. The 900 tokens of nine speakers are used for training MLP after extracting the ZCPA (zero-crossings with peak amplitudes) feature of 1024 dimensions [19]. That is, the MLP with 1024 inputs, 50 sigmoid hidden nodes, and 50 softmax output nodes is trained via the EBP algorithm with a fixed learning rate of 0.01 [18]. 1050 tokens from the other speakers are used as test evaluation. As shown in Figure 2 which shows the misclassification ratio for the test samples, still the proposed target encoding attains much better performance than the one-hot encoding. This is consistent with the first experiment's results.

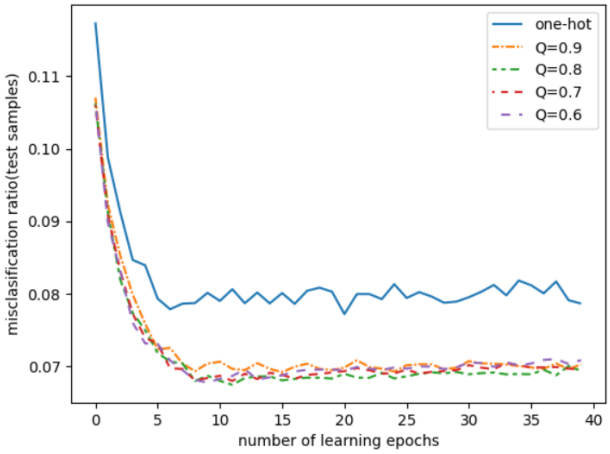
Subsequently, we simulate CNN's to verify the effectiveness of the proposed target encoding. In whole experiments of CNN's, we use the ReLU activation function for convolution layers and fully-connected hidden layers. CNN's with softmax outputs are trained via the Adam optimizer [20]. We experiment with LeNet-5 on MNIST handwritten digit dataset, in which a digit image consists of 28×28 pixels with a grayscale [21]. LeNet-5 is the CNN proposed by LeCun to recognize visual patterns directly from pixel images with minimal preprocessing [22]. We train LeNet-5 with 60000 training images and the misclassification ratios for 10000 test images during training are shown in Figure 3(a). The proposed target encoding is superior to the one-hot encoding. Table 1 summarizes the simulation results with CNN's, in which the misclassification ratios for the test images after the termination of 30 learning epochs and their improvement ratios are estimated. For MNIST dataset, the probabilistic target encoding attains above 36% improvement of the misclassification ratios over the one-hot encoding.

The next experiment is an image recognition task of CNN for the fashion MNIST dataset, which is a collection of fashion item images [23]. The data structure of fashion MNIST is the same as MNIST. We construct CNN with the architecture of C(32(3,3))-C(64(3,3))-P(2,2)-FC(128)-FC(10). Here, C($f(n,n)$), P(n,n), and FC(n) denote the convolution layer of f filters with $n \times n$ receptive fields, the pooling layer with a filter of size $n \times n$, and the fully-connected layer with n nodes, respectively. Figure 3(b) shows the misclassification ratio for the test images and the second row of Table 1 is the summary of simulation results after the termination of 30 learning epochs. In this experiment, the proposed target encoding is superior to the one-hot encoding and the improvement ratios are around 10%.

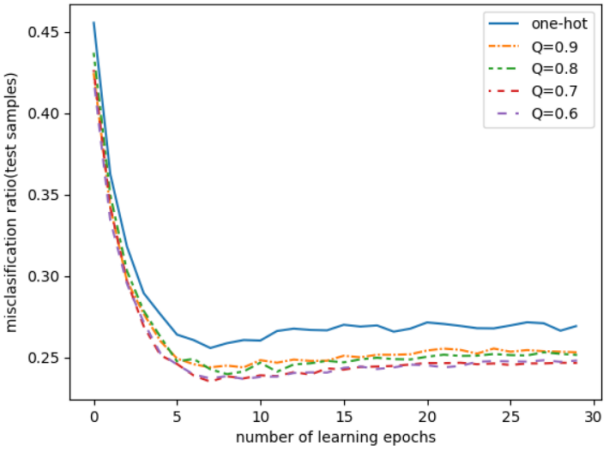
Finally, we simulate the object recognition task with CIFAR-10 which is an established computer-vision dataset used for object recognition. It consists of 60000 32×32 color images containing 10 object classes [24]. CIFAR-10 dataset is simulated in CNN's with the architecture of C(32(3,3))-C(32(3,3))-P(2,2)-C(64(3,3))-C(64(3,3))-P(2,2)-FC(512)-FC(10). Figure 3(c) shows the misclassification ratio for the test images and the third row of Table 1 is the summary of simulation results after the termination of 30 learning epochs. In this experiment, the proposed target encoding is superior to the one-hot encoding. The improvement ratios are above 6%. Although there are variations of improvement ratios in simulation problems, we can argue that the probabilistic encoding is better than the one-hot encoding due to the preventing effects on overfitting to training samples.



(a)



(b)



(c)

Figure 3. Simulation results of CNN's: the misclassification ratios for test samples. "Q" denotes probabilistic target encoding with various $Q_k(\boldsymbol{x})$ values and "one-hot" denotes one-hot target encoding: (**a**) MNIST task; (**b**) FashionMNIST task; (c) CIFAR-10 task.

Table 1. Summary of simulation results in CNN's: the misclassification ratios for test samples after the termination of 30 learning epochs and their improvement ratios. "Q" denotes probabilistic target encoding with various $Q_k(\boldsymbol{x})$ values and "one-hot" denotes one-hot target encoding. C, P, and FC denote the convolution layer, the pooling layer, and the fully-connected layer, respectively.

Problem	Architecture	Target Encoding	Misclass. Ratio(%)	Improv. Ratio(%)
MNIST	LeNet5	one-hot	0.873	0
		Q=0.9	0.542	37.92
		Q=0.8	0.551	36.88
		Q=0.7	0.558	36.08
		Q=0.6	0.548	37.22
Fashion MNIST	C-C-P-FC-FC	one-hot	7.866	0
		Q=0.9	7.028	10.65
		Q=0.8	6.937	11.81
		Q=0.7	6.956	11.57
		Q=0.6	7.086	9.91
CIFAR-10	C-C-P-C-C-P-FC-FC	one-hot	26.949	0
		Q=0.9	25.323	6.03
		Q=0.8	25.16	6.64
		Q=0.7	24.669	8.46
		Q=0.6	24.819	7.90

4. Conclusions

Instead of the one-hot encoding which adopts the extreme values of softmax functions, this paper proposed the probabilistic target encoding for improving the classification of test samples. Through simulations of MLP's with handwritten digit recognition and isolated-word recognition tasks, we validated that the one-hot encoding causes overfitting to training samples. Then, we verified that the proposed target encoding had the effects of overfitting prevention and classification improvement for test samples. Also, through simulations of CNN's with handwritten digit recognition, image recognition, and object recognition tasks, the proposed target encoding attained better classification performances for test samples than those of the one-hot encoding. Thus, we can argue that the proposed target encoding has the effect to improve the classification performance for test samples by alleviating the overfitting of neural networks to training samples.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S.-H. Oh, "Improving the Error Backpropagation Algorithm with a Modified Error Function," IEEE Trans. Neural Networks, vol. 8, no. 3, pp. 799-803, May 1997, doi: 10.1109/72.572117.

[2] P. Rodriguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond One-Hot Encoding: Lower Dimensional Target Encoding," Image and Vision Computing, vol. 75, pp. 21-31, July 2018, https://doi.org/10.1016/j.imavis.2018.04.004.

[3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-Embedding for Image Classification," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 38, no. 7, pp. 1425-1438, July 2016, doi: 10.1109/TPAMI.2015.2487986.

[4] S. Bengio, J. Weston, and D. Grangier, "Label Embedding Trees for Large Multi-Class Tasks," In Advances in Neural Information Processing, pp. 163-171, 2010.

[5] S. Escalera, O. Pujol, and P. Radeva, "On the Decoding Process in Ternary Error-Correcting Output Codes," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 120-134, Jan. 2010, doi: 10.1109/TPAMI.2008.266.

[6] H. White, "Learning in Artificial Neural Networks: a Statistical Perspective," Neural Computation, vol. 1, no. 4, pp. 425-464, Dec. 1989, doi: https://doi.org/10.1162/neco.1989.1.4.425.

[7] M. D. Richard and R. P. Lippmann, "Neural Network Classifiers EstomateBayesian a Posteriori Probabilities," Neural Computation, vol. 3, no. 4, pp. 461-483, April 1991, doi: 10.1162/neco.1991.3.4.461.

[8] S.-H. Oh, "A Statistical Perspective of Neural Networks for Imbalanced Data Problems," International Journal of Contents, vol. 7, no. 3, pp. 1-5, Sep. 2011, doi:10.5392/ijoc.2011.7.3.001.

[9] S.-H. Oh, "Statistical Analyses of Various Error Functions for Pattern Classifiers," *Proc. Convergence on Hybrid Information Technology*, Daejon, Korea, vol. 206, pp. 129-133, Sept. 22-24, 2011, https://doi.org/10.1007/978-3-642-24106-2_17.

[10] J. B. Hamshire II and B. Pearlmutter, "Equivalence Proofs for Multilayer Perceptron Classifier and the Bayesian Discriminant Function," *Proc. 1990 Connectionist Model Summer School*, pp. 159-172. 1990, https://doi.org/10.1016/B978-1-4832-1448-1.50023-8.

[11] S. Horiguchi, D. Ikami, and K. Aizawa, "Significance of Softmax-Based Features in Comparison to Distance Metric Learning-Based Features," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 42, no. 5, pp. 1279-1285, May 2020, doi: 10.1109/TPAMI.2019.2911075.

[12] J. B. Hampshire II and A. H. Waibel, "A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks," IEEE Trans. Neural Networks, vol. 1, pp. 216-228, June 1990, doi: 10.1109/72.80233.

[13] S.-H. Oh and Y. Lee, "A Modified Error Function to Improve the Error Backpropagation Algorithm for Multilayer Perceptrons," ETRI Journal, vol. 17, no. 1, pp.11-22, Apr. 1995, https://doi.org/10.4218/etrij.95.0195.0012.

[14] K. Baldi and P. Sadowski, "The Dropout Learning Algorithm," Artificial Intelligence, vol. 210, pp. 78-122, 2014, https://doi.org/10.1016/j.artint.2014.02.004.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning researches, vol. 15, pp. 1929-1958, 2014.

[16] G. Gnecco and M. Sanguineti, "The Weight-Decay Technique in Learning from Data: An Optimization Point of View," Computational Management Science, vol. 6 pp. 53-79, 2009, https://doi.org/10.1007/s10287-008-0072-5.

[17] J. J. Hull, "A Database for Handwritten Text Recognition Research," IEEE Trans. Pattern Anal. Machine Intell., vol. 16, no. 5, pp. 550-554, May 1994, doi: 10.1109/34.291440.

[18] D. E. Rumelhart and J. L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, MA, 1986.

[19] D.-S. Kim, S.-Y. Lee, and R. M. Kil, "Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environment," IEEE Trans. Speech Audio Processing, vol. 7, no. 1, pp. 55-69, Jan. 1999, doi: 10.1109/89.736331.

[20] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980[cs], 2014.

[21] L. Deng, "The MNIST database of Handwritten Digit Images for Machine Learning Research," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141-142, Oct. 2012, doi: 10.1109/MSP.2012.2211477.

[22] Y. LeCun, "LeNet-5, Convoltuional Neural Networks," http://yann.lecun.com/exdb/lenet/.

[23] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," arXiv: 1708.07747v2, 2017.

[24] Y. Abouelnaga, O. S. Ali, H. Rady, and M. Moustafa, "CIFAR-10: KNN-based Ensemble of Classifiers," *Int. Conf. Computational Science and Computational Intelligence*, Las Vegas, USA, Dec. 2016, pp. 1192-1195, doi:10.1109/CSCI.2016.0225.