IJARIIT
INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 4 - V7I4-1145)
Available online at: https://www.ijariit.com

Analysis of Dropout in ANN using MNIST Dataset

Swathi N.
swathinagaraj31@gmail.com
Atria Institute of Technology, Bengaluru, Karnataka

Shreeraksha
shreerakshapoojary@gmail.com
Atria Institute of Technology, Bengaluru, Karnataka

Shruti A. P.
shrutiap625@gmail.com
Atria Institute of Technology, Bengaluru, Karnataka

Sparsha S. G.
sparshasg5@gmail.com
Atria Institute of Technology, Bengaluru, Karnataka

Farhana Kausar
farhanakausar@atria.edu
Atria Institute of Technology, Bengaluru, Karnataka

ABSTRACT

The concept of Neural Networks is propelled by the neurons within the human brain and researchers needed a machine to imitate the same process. A Neural Network (NN) is a circuit of connected neurons, or in a present-day sense, an artificial neural network, composed of artificial neurons constructed for solving the artificial intelligence problems. In the deep neural network, Overfitting is a severe issue. This issue could be caused by unbalanced datasets and incorrect model parameter initialization, which causes the model to adhere too closely to the training data and reduces the model's generalization performance for unknown data. To overcome such problems, Regularization techniques are used. This technique modifies the learning algorithm in a way that increases the model's generalization and performance. Dropout is one such regularization technique for addressing the overfitting problem. During the training it randomly drops the hidden units or neurons to prevent the units or neurons from the co-adaption. This method significantly reduces the overfitting and improves the performance of the neural network model. Dropout is preferred for the large neural networks in order to have more randomness.

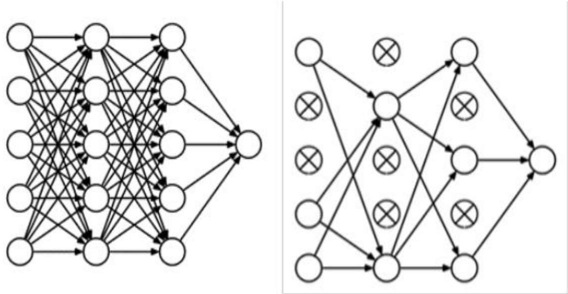
Keywords: *Neural Network, Overfitting, Regularization, and Dropout.*

1. INTRODUCTION

Machine-learning architecture is built on the foundation of neural networks. They are artificial neural networks that were inspired by biological ones. Without any task-specific rules, these systems learn to do tasks by being exposed to a variety of datasets and examples. One input layer, one or more hidden layers, and an output layer make up these layers. Each layer of a neural network contains neurons that execute the same function. They simply add the bias and perform an activation function after calculating the weighted sum of inputs and weights. The input layer is in charge of receiving the data. These inputs can come from a variety of places, including a web service or a csv file. The output layer is in charge of generating the final product. In a neural network, there must always be one output layer. Hidden layers are a layer that sits between the input and output layers and is where all of the calculation takes place.

When the network tries to understand too much or too many information as well as the noise from the training data, it performs poorly on unknown or test datasets. Overfitting occurs as a result of this. In such networks, overfitting is a serious problem. This issue could be caused by unbalanced datasets and incorrect model parameter initialization, which causes the model to adhere too closely to the training data and reduces the model's generalization performance for unknown data. When the error on training data reduces to a very tiny value, but the error on new data or test data climbs to a big value, this is an indication of overfitting during training.

To reduce overfitting, a variety of regularization strategies have been devised, including early stopping, adding weight penalties to the cost function of networks like L1 and L2, and dropout. This method alters the learning algorithm in such a way that the model's generalization and performance improve. Dropout is one such regularization approach for dealing with the problem of overfitting. It prevents overfitting and allows for the efficient combination of exponentially many distinct neural network topologies. The concept "dropout" denotes to dropping a unit out, means removing it from the network, together with all of its incoming and outgoing connections, for the time being. The units to be dropped are chosen at random. It is a powerful regularization approach for neural network training. During training, a random set of units and their accompanying connections are removed from the network, and all units are used during inference (test). This strategy decreases the number of parameters to tune in each training iteration while also preventing units from over-co- adapting. A n-unit neural network can be thought of as a collection of 2n thinned neural networks. Because the weights in these networks are shared, the total number of parameters is still O (n2) or fewer. A new thinning network is chosen and trained for each display of each training case. As a result, training a neural network with dropout can be thought of as training a set of 2n thinned networks with significant weight sharing, in which each thinned network is only trained once or twice.



(a) **(b)**

Figure 1 : (a) Neural Network without dropout. (b)Neural Network with dropout.

2. LITERATURE SURVEY

[1] Dropout is a comparatively new neural network training approach that relies on randomly "dropping out" neurons for training in order to prevent neurons from dropping out during co-adaptation of detectors with features. It is assessed by using averaging and regularizing aspects of drop- out in both linear and non-linear networks to construct the generic formalism for the analysis of drop-out on either blocks or links, with variable probability values. The average properties of dropout for deep neural networks are defined by three recursive equations, along with the approach of expectations by normalized weighted geometric means. For these approximations, simulations are used to approximate, limit, and validate the results. We also show how, among other things, dropout results descend randomly on a regularized error function.

[2] The top state-of-the-art contributions for handwritten digit recognition reported on the MNIST dataset are summarized in this study. The MNIST database of handwritten digits has been widely used to evaluate computer vision algorithms since its inception over two decades ago. LeCun et al introduced MNIST in 1998. This method has grown over the last two decades, and with the advancement of graphics hardware, it is quickly becoming the industry standard for solving problems. A wide range of computer vision difficulties, as well as difficulties in other fields. Although MNIST accuracy is near to 100 percent and is unlikely to improve, these new advances may hold the key to solving more complicated computer vision challenges.

[3] Dropout is a strategy for decreasing overfitting in neural networks. The basic concept is that Co-adaptation of concealed units should be avoided. Dropout increases the performance of neural networks in a lot of instances like object categorization, digit recognition, speech recognition, and document recognition etc. This shows that dropping out as a strategy is extremely effective generic term that does not refer to a specific domain. Dropouts' main idea is to take a larger model that readily overfits and train small sub-models from it. This method trains the main model, which can then be used at test time, because all of the sub-models share parameters with it.

[4] The ANN model was proposed using the backpropagation (BP) technique. In a standard BPANN (Backpropagation Artificial Neural Network) topology, there are three layers: input layer, where data is provided to the network; hidden layer, where data is being processed; and output layer, where the results of the provided input are generated. Back-propagation algorithm is a plain and easy iterative technique that usually works well, also with complex data. It has good computing features when working with largescale data, unlike other learning methods (such as Bayesian learning). The feedforward of the input training pattern, the computation and backpropagation of error, and the adjustment of synapses' weights are all part of the backpropagation training process.

IN [5] the emphasis is on the motivations and intuitions behind the design of the supervised learning algorithm are to prevent overfitting. When there are too many parameters to alter in a learning model, overlap occurs. A popular solution to this problem is to incorporate a penalty term in an objective feature known as regularization, which prevents network parameters from becoming too large. Dropout is another suggested strategy for avoiding co-adaptation, which involves randomly dropping specific units out in the training phase on every training data set. The MNIST dataset is used in this article to look at several L2 and drop-out regularization strategies inside a single hidden neural layer network.

3. DROPOUTS

Dropout is a powerful technique for re-shaping neural networks. Over-specific co-adaptations of feature detectors are discouraged by stochastically "dropping out" units with a certain probability, preventing overfitting and boosting network generalization. Dropout can also be thought of as an approximate model aggregation strategy, in which a large number of smaller networks are aggregated to create a more effective ensemble.

Regularization was a prominent study topic prior to Dropout. Regularization approaches for neural networks, such as L1 and L2 weight penalties, were first introduced in the early 2000s. These regularizations, however, did not totally eliminate the overfitting problem. The reason was Co-adaptation. Co-adaptation is a critical difficulty when learning huge networks. If all of the weights are learned at the same time in such a network, it's likely that some connections will be more predictive than others.

As the network is trained repeatedly in this scenario, the stronger connections are learned more, whereas the weaker ones are avoided. Only a percentage of the node connections get taught after many repetitions. The rest, on the other hand, stop taking part. Co-adaptation is the term for this phenomenon. Traditional regularization, such as the L1 and L2, could not prevent this. The reason for this is that they also regularize based on the connections' prediction abilities. As a result, they approach determinism in selecting and rejecting weights. As a result, the strong become powerful and the week become weaker.

One important implication was that increasing the size of the neural network would not assist. As a result, the size and accuracy of neural networks were limited. Then came Dropout. A novel regularization strategy. It was able to put an end to the co-adaptation. We might now create deeper and broader networks. And make use of all of it's predictive power. Dropout is a reasonably cheap and extremely effective regularization strategy for deep neural networks of all types that reduces overfitting and improves generalization error.

As a result, dropout is a regularization approach for avoiding complex coadaptation's to the training data in order to reduce artificial neural networks' excessive fitness. It is an efficient method of averaging models with neural networks. Dropout has tweaked the principle of collecting all weights by only learning a subset of the network weights in each training session.

To see how far your model output differs from the actual output, we need to calculate error. Then we must determine whether or not the error has been minimized. This can be done using backpropagation algorithm. If the error is large, you should update the parameters (weights and biases). Check the error once again. Repeat the procedure until the error is reduced to a bare minimum. You can give some inputs to your model and it will produce the output once the error has decreased to a minimum.

4. BACKPROPOGATION

Backpropagation is a typical method of training artificial neural networks. It is short for "backward propagation of errors." This method is useful for calculating the gradient of a loss function with respect to all of the network's weights.

Backward propagation, which is utilized in the backpropagation method, is closely connected to dropout. Backpropagation employs an ordered alternating sequence of multiplications through the transpose of the forward weight matrices then by the derivatives of the activation functions to start from the errors at the output layer. As a result, backpropagation is essentially a kind of linear propagation in a reverse linear network multiplied by the derivatives of the activation functions at every node.

4.1 How Backpropagation Works

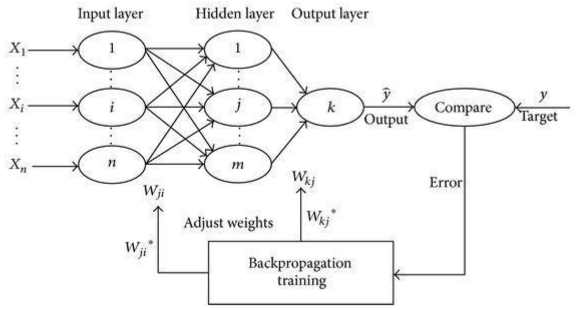


Figure 2 : Backpropagation

1. Inputs X, enter via a pre-connected pathway.
2. Real weights W are used to simulate the input. The weights are normally chosen at random.
3. First from input layer towards the hidden layers tothe output layer, calculate the output for each neuron.
4. Determine the amount of error in the outputs.

$$\textit{Error}_B = \textit{Actual Output} - \textit{Desired Output}$$

5. Return from the output layer to the hidden layer to change the weights in order to reduce the error. Repeat the process until the desired result is obtained. The purpose of back-propagation training is to reduce the squared error as much as possible. To do so, the error function's gradient must be determined.

Gradient is a calculus derivative having a value like +1.23 or -0.33. The gradient's sign indicates whether you should increase or decrease the weights and biases to minimize errors. The learning rate and the size of the gradient are used to calculate how much to raise or reduce the weights and biases.

As a result, backpropagation simplifies the network topology by deleting weighted links that have little impact on the trained network. It's especially valuable for deep neural networks working on tasks that are prone to errors, like image or speech recognition.

5. EXPERIMENTAL SETUP

The experimental technique will be described in this part, along with the outcome of the dropout rate comparison. The purpose of this experiment is to determine the accuracy and the loss of dropouts that varies on different dropout rates. MNIST dataset is used for this experiment.

5.1 MNIST

The MNIST database (Modified National Institute of Standards and Technology database) is a huge collection of handwritten digits which is widely used for image processing system training. In the field of machine learning, the database is also frequently utilized for training and testing. Each of the 60,000 training and 10,000 test samples in the MNIST data set represents a 28X28 digit picture. For validation, we set aside 10,000 random training pictures. On the validation set, hyperparameters were adjusted to give the best validation error after 1 million weight changes.

The validation set was then merged with the training set, and 1 million weight updates were performed. This net was used to assess the test set performance. We picked this method of using the validation set since it was simple to set up hyperparameters and no early halting was necessary. As a result, after the hyperparameters were determined, combining the validation and training sets and training for a long period made sense.

The experiment in this study is based on the MNIST dataset. The MNIST dataset is popular in introducing machine learning for various reasons. One of them might be difficult to categories owing to handwriting inconsistencies, but simple enough because there are no other inconsistencies. There is no noise surrounding the handwriting itself, and all numbers are aligned appropriately.

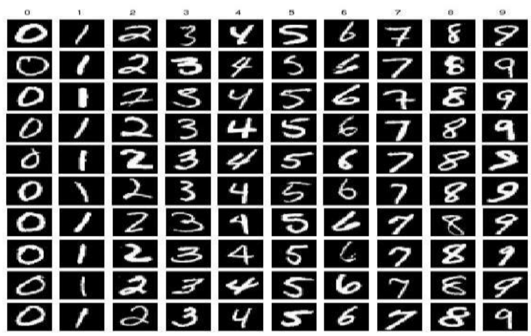


Figure 3 : MNIST Dataset

5.2 Effect of dropout rate

Dropout adds a new hyperparameter to the equation: The probability of preserving a unit p. The severity of dropout is controlled by this hyperparameter. The chance of training a particular node in a layer, where 1.0 indicates no dropout and 0.0 denotes no outputs from the layer, is the default meaning of the dropout hyperparameter.

For the hidden layers, we have used ReLu as an activation function and for the output layer, we have used Sigmoid function. Dropout was used in all layers and is increased from 0.1 to 0.9 with an increase in step size of 0.1. Below is a graph plotted for Accuracy Vs Dropout rate and Loss Vs Dropout rate for the Epoch values 0.1 and 0.9. We have got the highest accuracy value for 0.1 and the lowest accuracy value for 0.9.

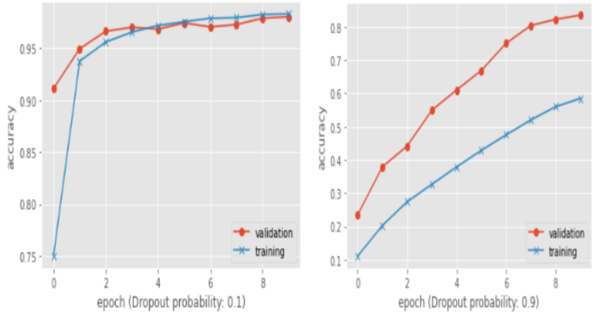


Figure 4: Comparison of training and validation accuracy value for 0.1 and 0.9.

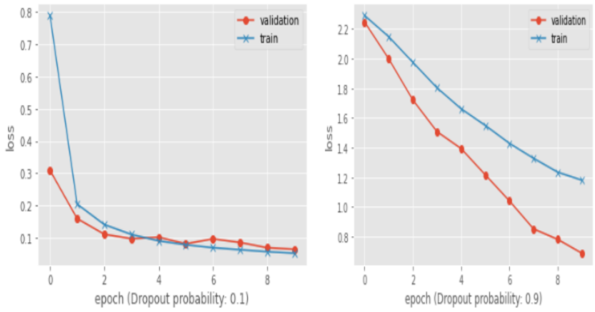
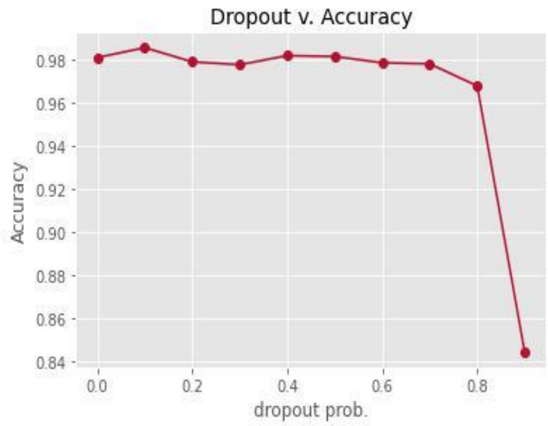


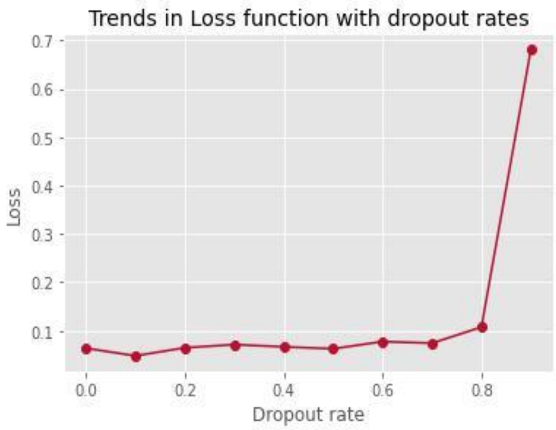
Figure 5: Comparison of training and validation loss value for 0.1 and 0.9 dropout probability.

6. RESULT

In this experiment, we have conducted a study to understand the effect of dropout. For the deep learning model, we have used multi-layered dense neural networks and activation functions to observe the performance of the model in both training the testing phase. The graphs of Dropout vs Accuracy and Dropout vs Loss function with dropout rate from the result obtained is shown below:



(a)



(b)

Figure 6 : Plot of accuracy score and loss over epochs on MNIST

Following are the outcomes:

Table 1 : Represents the dropout rate of increasing epochs with their loss and accuracy

Dropout Rate	Loss	Accuracy
0.0	0.0573	0.9806
0.1	0.0679	0.9771
0.2	0.0638	0.9775
0.3	0.0566	0.9811
0.4	0.0703	0.9770
0.5	0.0741	0.9755
0.6	0.0673	0.9782
0.7	0.0731	0.9783
0.8	0.1009	0.9698
0.9	0.3059	0.8921

7. CONCLUSION

Overfitting and a large number of parameters to optimize are two common problems with deep neural networks. Dropout is a strategy for decreasing overfitting in neural networks. Although it is a relatively new approach, it has already had a significant impact in this field. Dropout acts as a regularizer by killing neurons in hidden layers on a random basis. As a result, the network is forced to generalize even more. The experimental results demonstrate that Dropout network not only performs better, but consistently better. This could be interpreted as an indication of overfitting. Dropout, in essence, could help us to regularize our network and make it more resistant to overfitting. In short, Dropout training is regarded as one of the most reliable methods for improving the performance of a large neural network.

8. REFERENCES

[1].Pierre Baldi , Peter Sadowski, "Understanding Dropouts" University of California.

[2].Alejandro Baldominos , Yago Saez ,Pedro Isasi," A Survey of Handwritten Character Recognition with MNIST and EMNIST", Computer Science Department, Universidad Carlos III of Madrid, 28911 Leganés, Madrid, Spain,2019.

[3].S. Nitish. Improving Neural Networks with Dropout. PhD thesis, University of Toronto, Toronto, Canada, 2013.

[4].Sandy Putra Siregar, Anjar Wanto," Analysis Accuracy ofArtificial Neural Network Using Backpropagation Algorithm In Predicting Process (Forecasting)" International Journal Of Information System & Technology Vol. 1, No. 1, (2017), pp. 34-42.

[5]. Ekachai Phaisangittiasagul, "An Analysis of the regularization between L2 and dropout in the single hidden layer Neural Network", International Conference on Intelligent Systems, Modelling and Simulation 2016.