Significance Of Epochs On Training A Neural Network

Saahil Afaq, Dr. Smitha Rao

Abstract: Deep neural network (DNN), has shown an incredible success in the field of computer vision and in tasks such as classification, facial detection etc. But, accuracy of a model depends on a large number of parameters such as weights, bias, number of hidden layers, different kinds of activation function and hyperparameters. Epochs is a form of hyperparameter which plays an integral part in the training process of a model. The total number of epochs to be used help us decide whether the data is over trained or not. Recently, the performance of deep neural networks, have been improved by making use of pre-trained network architectures, and by the introduction of GPU-based computation and now recently we are even on the verge of training the models on TPU chips. However, there are many problems in the field of Deep Neural Network which concerns the training, back propagation, and customizing of the hyperparameters.

Keywords: Hyperparameters, epochs, nodes, hidden layers, Deep Neural Network (DNN), Architecture.

1. INTRODUCTION

In machine learning, there are parameters that cannot be directly learned from the regular training process. Hyperparameters are part of parameters which are usually defined before the actual training starts. Hyperparameters are variables that we need to set before applying a learning algorithm to a dataset. We have one more parameter called the learning parameter in neural networks. So, how is learning parameter different from hyperparameter. Basically, learning parameters consists of weights and biases. The values of which are initialized randomly before the training the starts. The values are adjusted towards values that have correct output. Epoch means one pass over the entire dataset. The weights initialized in the beginning will be subjected to changes when the next cycle of same dataset simulation called next epoch takes place. The epoch optimization mainly consists of two major problems namely Underfitting and Over-fitting. So, what is Overfitting and Underfitting. Overfitting is nothing but when a model is fed both the training data and noise in the training data to the extent that it impacts negatively to the performance of the model on any given data. Noise is nothing but the irrelevant data or the information. What this means is that both training data and noise is picked up by the model which would result in overfitting thus hindering the accuracy of the model. Underfitting on the other hand refers to a model that cannot learn the training data nor can it generalize any new data given to it. As it is obvious that it does not learn well on the training data; it will have poor performance on any data given to it. As you can see determining appropriate hyperparameters play an integral role in the success of our deep neural network architecture. In our case if we use small number of epochs the model might not learn completely and may cause underfitting. If we train the model using many epochs it may lead to overfitting where we the model is learning even the unwanted parts like the noise. So, how do we know the correct number of epochs to use? When do we stop? It all depends on the training loss and the validation loss. We shall dive deeper into that in our paper in the following portions. In our paper we are going to take a deeper look into how epochs play a role in model building. What actually is training loss and validation loss? When to stop the training process? First the paper is segmented into 4 sections. We will be taking a look into the already published papers in the field of epochs in our literature review sectionNext, we get into the experimental setup of our paper this helps us to know which datasets we will be using and which architecture we have opted for. Next, we dip into the results section wherein which we will arrive at a conclusion in the last section of this paper.

2. LITERATURE REVIEW

Many researchers went through different ideas and experiments about epochs. The precise number to use to build a model. When to stop the training process. Ochin Sharma in his paper [1] helps us to learn about what actually deep learning is and how a good deep learning model makes use of all the hyper parameters. Also, in his paper he talks about how cloud computing, parallel computing is useful in deep learning. [2] is a text book published by the MIT press with the authors Ian Goodfellow and Yoshua Bengio and Aaron Courville in their volume 1 textbook where they have made an effort for even a beginner to understand about deep learning concepts. The authors in their 2009 paper [3] stated how to determine the epochs for neural network by making use of self-organized maps (SOM). Self-organizing maps to are used to determine the optimum number of epochs to be used in a model. In [4] yasusi kanadas research paper stated how optimizing learning rate of a neural network by using a genetic algorithm with per-epoch mutations of 2016. He proposes a new learning method wherein he combines back propagation with a genetic algorithm. [5] Timothy Roberts and Kuldip K. Paliwal let us know how the new proposed method, ESOLA uses cross-correlation to align epochs for the training process. Eldem A., Eldem, H., & Ustun, D. in their 2018 paper [6] experiments are made on the iris dataset by using different hyperparameters and experimenting with different epoch values for training the model. In [7] Dhande, G., & Shaikh, Z. in their 2019 paper try to create a speech recognition system. They experiment with the number of epochs to train their model and check to see how changing the number of epochs result in a better speech recognition system. How hyper parameters help to formulate the cloud the stream videos are helpful in doing so is illustrated in their paper [8]. In a 2009 paper [9] have stated a new Algorithm for Improving Neural Network by the use of early stopping. they have clearly stated how we can avoid the overfitting problem by the usage of early stopping. The algorithm of early stopping is based on fuzzy clustering which was created to solve the overfitting problem. [10] in this paper we discover novel ensemble of deep learning architectures which are used for automatic feature extraction. The existing ensembles gives us a good accuracy however the accuracy more or less depends heavily on the hyperparameters used. This paper was published on 2016 by Fatma Shaheen, Brijesh

K. Verma. Based on literature survey and information gathered from different sources. Experiments have been carried out on two different datasets namely Iris Species and MNIST handwritten digit recognition dataset. The computational speed along with accuracy of the models based on different epochs have been calculated and compared. The experiment has been carried out in three phases. Phase [III] has the experimental setup, [IV] contain the result of the experiment, phase [V] carry the conclusion of the experiment.

3. EXPERIMENTAL SETUP

A. Dataset:

The first dataset on which we are going to do our experiments is the Iris Species dataset. The second dataset MNIST dataset we will be experimenting using both csv and image file. The Iris Species dataset is a .csv file. Both the datasets are clean dataset which means that there are no missing values in either of them. There are 150 records with 4 dependent variables and one independent variable in Iris Species dataset. MNIST images are black and white images of 28*28 size. In the csv file of MNIST the pixel values are given as data frame. The datasets are split into (80:10:10), in which we take 80 % as training data, 10% as validation data and another 10 % as our test data.

B. Architecture:

For the Iris Species dataset which is a .csv format dataset we make use of an Artificial Neural Network (ANN). So, what is an Artificial Neural Network? Artificial Neural Network is basically artificial depiction of the human brain. An Artificial Neural Network consist of an input layer, multiple hidden layers, and an output layer. Every layer consists of 'n' number of nodes, each and every node are interconnected to one another. We may change the network as one desires; we can add or reduce the number of hidden layers or even the nodes in the hidden layers. This again depends upon the problem which we are trying to solve. A given node takes weights and biases as its inputs, and is passed through a activation function. This gives us an output for a given node, which is then fed into the next node in the next layer. The flow is from left to right, and the output is obtained. When any input is given to its input node then it feedforwards the input through its hidden layer and through the activation function, we get our output. Then the loss is calculated by the loss functions like cross entropy, mean square error etc. loss is then backpropagated till the input node to adjust the weight and bias to get the desired output. This process continues until we get or required output. Next for the images dataset we use Convolutional Neural Network (CNN) same as Artificial Neural Network. A CNN takes in an input image, assign weights and biasesExcept for the pooling layer, rest of the part is similar as ANN. It also transfers the input data to the output layer through the hidden nodes and propagate back the error to adjust the weight and bias according to it. There are many hyperparameters that effect the accuracy and time cost for any given model. In this paper we are mostly going to uncover how epochs play an important role in getting a good accuracy for the model. Basically, we train the Neural Network on the training dataset only. The Network calculates the errors on both the training and validation set. We stop training when the validation error is the minimum. Basically, validation set is used to lower the overfitting as much as possible. When the accuracy in the training data increases, but the accuracy over the validation data remains the same or maybe even decrease, this means the model is getting overfitted this also means we should stop the training process. When the model learns only the training data it is called the training error. The validation error however tells you how well your learned model generalizes, which means how well it fits to data that it is not yet been trained on which is the validation data.

C. Loss function:

The Iris Species dataset is a multi-class classification problem. Which means it is the problem of how to classify values into two or more classes. Cross entropy loss plays a major factor in the determining how many epochs to be used. So, what is Cross entropy? It is a loss which measures the performance of a model more or less a classification model.

$$\text{Cross-Entropy Loss} = -\sum_{i=0}^{n} y_i \, log(y_i)$$

Where, y_i is the required output and $\tilde{(}y_i)$ is the predicted output.

4. RESULT

The experiments are run on Jupyter Notebook on a Laptop of i7 processor with 8GB DDR4 RAM, and Nvidia GeForce GTX 1650 graphics card. As we are trying to find the optimal or the correct number of epochs to train a neural network. We will be experimenting with various numbers of epochs and checking how the accuracy effects it as well as if there is any overfitting happening. Firstly, we do the training on the Iris Species dataset with 2 hidden layers one input and one output layer. In the first iteration we initialize epochs to 50.

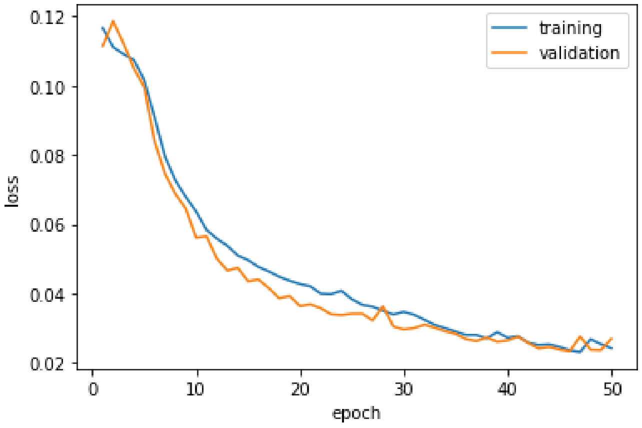

Fig 1.3 *Loss in Iris Species Dataset using 50 epochs*

At epochs = 50 we get an accuracy of 90.00%.

Next, we set the epochs to 100.

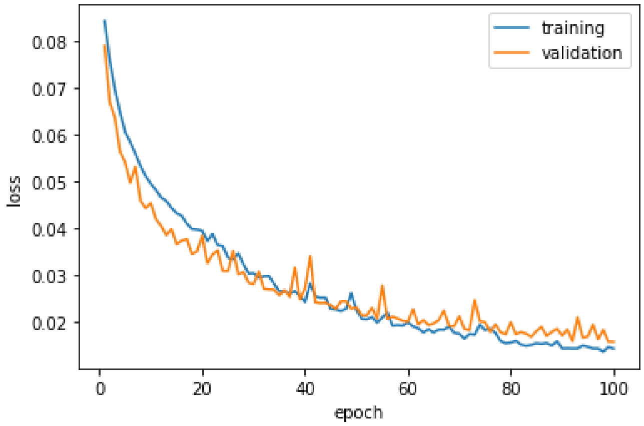

***Fig 1.4** Loss in Iris Species Dataset using 100 epochs*

At epochs =100 we get an accuracy of 96.67%.

When we initialize epochs to 600 we can see some changes in the cross-entropy loss

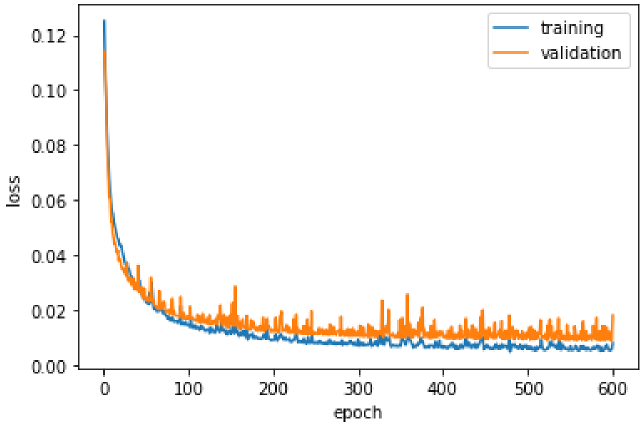

***Fig 1.5 Loss** in Iris Species using 600 epochs*

The accuracy of the model when we initialize the epochs to 600 is 90.00% which means the accuracy decreases when we have increased the epochs. But when do we know to stop. This is called early stopping. As we can see the validation loss increases when we increase the epochs. This is nothing but overfitting. So, we stop when we get better accuracy so at epochs 100 we can safely say that we can stop the training process. In the MNIST handwritten digits dataset we are going to build 2 model ANN as well as CNN models. First, we build the ANN model for the .csv file with 6 hidden layers one input and one output layer. In the first iteration we initialize the epochs to 50.

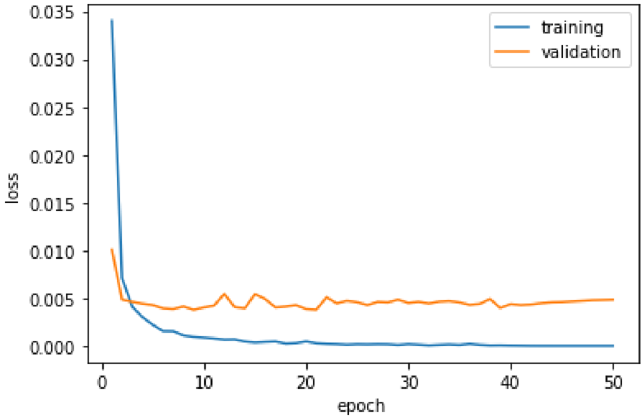

***Fig 1.6** Loss in MNIST digits (.csv) using 50 epochs*

This gives us a good accuracy of 97.65% but do we stop here? The answer is no, we reinitialize the epochs to 100 to see if we can better our accuracy without overfitting our model.

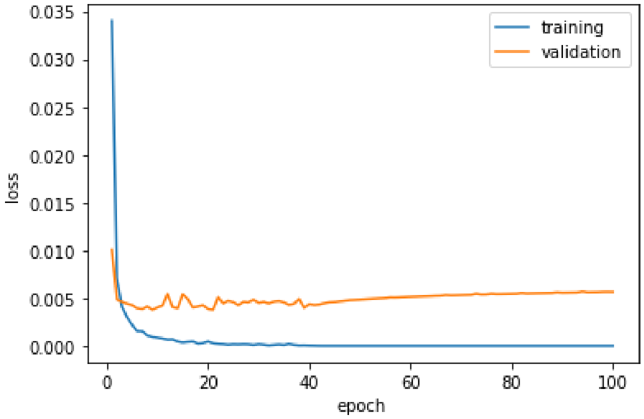

***Fig 1.7** Loss in MNIST digits (.csv) using 100 epochs*

From the above graph we can infer that the validation error increases. At epoch = 100 the accuracy may have increased to 97.67% but this is overfitting as the training error is an even 0. So from the ANN experiment we can infer that we can safely stop at epoch 40 this does hiner the accuracy of the network but it will ultimately mean that the model will just perform poorly on the test data. We can stop the training process early at around 40 epoch to get better permormance from the model. Next, we build a CNN model for the MNIST images with 2 convolutional layers and 3 hidden layer one input and output. We initially allocate 30 epochs to train the model. We get a very good accuracy of 99.21%.

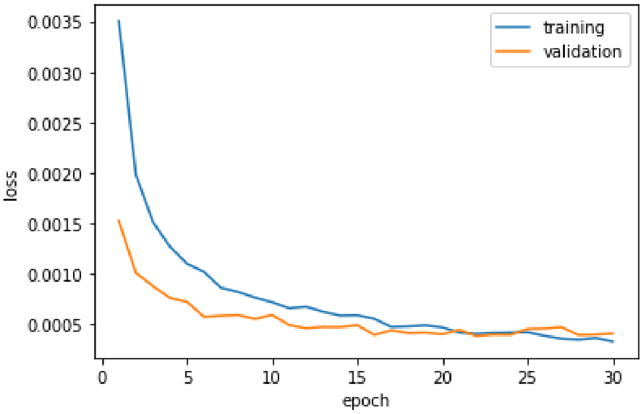

***Fig 1.8** Loss in MNIST digits (images) using 30 epochs*

As we can infer from the graph both training and validation loss are decreasing further. So, next we try to increase the number of epochs to 100. As we are training on images the training time is vastly more than ANN models.

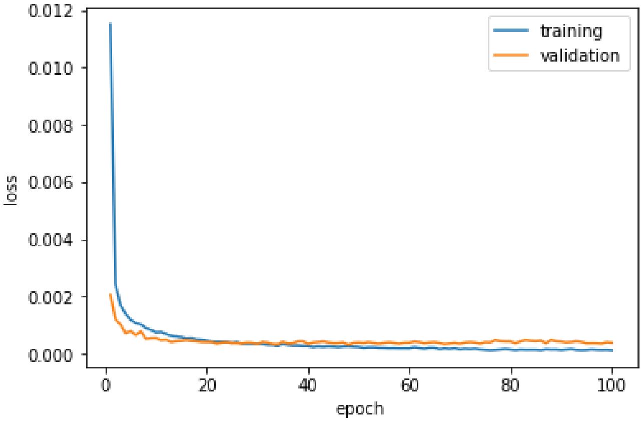

***Fig 1.9** Loss in MNIST digits (images) using 100 epochs*

***Table 1:** No of epochs on how it changes the accuaracy of the model*

Sl. No	Dataset	No.of epochs	Accuracy
1.	Iris Species	50	90.00%
		100	96.67%
		600	90.00%
2.	MNIST handwritten digit (.csv)	50	97.65%
		100	97.67%
3.	MNIST handwritten digit(images)	30	99.21%
		100	99.38%

From the above graphs and tables, we can infer the effect of epochs on model training and also on its accuracy.

VII.CONCLUSION

In this study it has been shown how best to train a neural network. How to actually avoid overfitting Through the experiments conducted we can safely say that there is no optimal number of epochs. Actually, the number of epochs differ from one dataset to other the main factor which comes into the picture is the training and validation error. Initially during the training process when both the training set and validation set are being learnt by the model. By using a graph, we can see that both the training and validation error keeps going down. This is a sign that the learning process is going well. But as soon as the validation error goes up that is when we should stop the training process. This process is called early stopping. As validation error goes up it means the model is getting overfitted which means that the model is by-hearting the data. This is very bad for the model as it will not be learning it.

REFERENCES

[1] Sharma, O. (2019). Deep Challenges Associated with Deep Learning. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)

[2] Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep learning (Vol. 1). Cambridge: MIT press

[3] Sinha, S., Singh, T. N., Singh, V. K., & Verma, A. K. (2009). Epoch determination for neural network by self-organized map (SOM). Computational Geosciences, 14(1)

[4] Kanada, Y. (2016). Optimizing neural-network learning rate by using a genetic algorithm with per-epoch mutations. 2016 International Joint Conference on Neural Networks (IJCNN).

[5] Roberts, T., & Paliwal, K. K. (2019). Time-Scale Modification Using Fuzzy Epoch-Synchronous Overlap-Add (FESOLA). 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).

[6] ELDEM, A., ELDEM, H., & USTUN, D. (2018). A Model of Deep Neural Network for Iris Classification with Different Activation Functions. 2018 International Conference on Artificial Intelligence and Data Processing

[7] Dhande, G., & Shaikh, Z. (2019). Analysis of Epochs in Environment based Neural Networks Speech Recognition System. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI).

[8] Yaseen, M. U., Anjum, A., Rana, O., & Antonopoulos, N. (2018). Deep Learning Hyper-Parameter Optimization for Video Analytics in Clouds. IEEE Transactions on Systems, Man, and Cybernetics: Systems

[9] Wu, X., & Liu, J. (2009). A New Early Stopping Algorithm for Improving Neural Network Generalization. 2009 Second International Conference on Intelligent Computation Technology and Automation. doi:10.1109/icicta.2009.11 Shaheen, F., & Verma, B. (2016). An ensemble of deep learning architectures for automatic feature extraction. 2016 IEEE Symposium Series on Computational Intelligence (SSCI)