ISSN 2278-3091
Volume 10, No.2, March - April 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse281022021.pdf
https://doi.org/10.30534/ijatcse/2021/281022021

A Study of incremental Learning model using deep neural network

Dr V.Goutham[1], SaiKrishna[2],Shruthi Kovela[3], V. Shreya Reddy[4],J Prashanth[5]
[1] Professor in Sreyas institute of engineering and Technology,JNTUH,India,v.goutham@gmail.com
[2] B.Tech in Sreyas Engineering college, 17VE1A0515
[3] B.Tech in Sreyas Engineering college , 17VE1A0534
[4] B.Tech in Sreyas Engineering college , 17VE1A0560
[5]B.Tech in Sreyas Engineering college, 17VE1A0528

ABSTRACT
Deep learning has arrived with a great number of advances in the research of machine learning and its models. Due to the advancements recently in the field of deep learning and its models especially in the fields like NLP and Computer Vision in supervised learning for which we have to pre-definably decide a dataset and train our model completely on it and make predictions but in case if we have any new samples of data on which we want our model to be predicted then we have to completely retrain the model, which is computationally costly therefore to avoid re-training the model, we add the new samples on the previously learnt features from the pre- trained model called Incremental Learning. In the paper we proposed the system to overcome the process of catastrophic forgetting we introduced the concept of building on pre-trained model.

Key words : Natural language processing (NLP), Comma-separated values(CSV),Convolution neural network (CNN), Stochastic gradient descent(SGD), Adaptive Moment Estimation(ADAM),Canadian Institute For Advanced Research(CIFAR-10) Modified National Institute of Standards and Technology database (MNIST) .

1. INTRODUCTION

The main thought is to reduce the computation cost of re-training the model for the sake of new samples which is very costly in terms of computation. So, the main objective is to improve accuracy and reduce the computation cost along with less time consumption, which is not possible in case of traditional machine learning models for two main reasons, first one is “Catastrophic Forgetting” and second is, it becomes very difficult to identify position and quantities of new units. Catastrophic Forgetting is the phenomenon of forgetting previously learnt feature after the arrival of new samples. To overcome these we train a deep model with dynamic connections in the incremental stages. So, that our model can accept any samples in the future with better accuracy, less computation cost, and less time consumption[1].

The main motive of the project is to overcome the drawbacks of the traditional machine learning process which is over computation cost, which is because of the static dataset we upload pre-definably, train on it, and predict on it, but which cannot handle any new samples in the future. But in our model we do not need to build a model from scratch every time we encounter new samples, rather build the new samples on the features that are already extracted from the pre-trained model. It helps us to boost the performance as we can use the model that is trained on a dataset for example,(ImageNet) and extract features from it and use it as an initialization to related classification tasks such as CIFAR10, MNIST. These are the datasets that we train in our paper.. In the case of ANN, such as MLP, when input new data, the network forgets the previous knowledge to give way to the learning of the new data. Therefore incremental training of a deep convolution neural network (CNN) model as new classes are added to the existing data [2].

2. LITERATURE SURVEY

It is the most significant step within the code development method. Before building the tool, it's necessary to see the time complexness, economy, and company strength. Once this stuff square measure glad, then future steps square measure to determine that package and language is used for developing the tool the programmers begin building the tool the programmers would like external support. Before building the system, the on top of thought is taken into account for developing the planned system.
Python could be a high-level, interactive, and object-oriented scripting language. Python is meant to be extremely powerful. Python is a high-level, interpreted and interactive and object- oriented scripting language. Python is designed to be highly readable [5].

Transfer Learning
The fig 1 shows to remember here is that, transfer learning, is not a new concept which is very specific to

deep learning. There is a stark difference between the traditional approach of building and training machine learning models, and using a methodology following transfer learning principles.

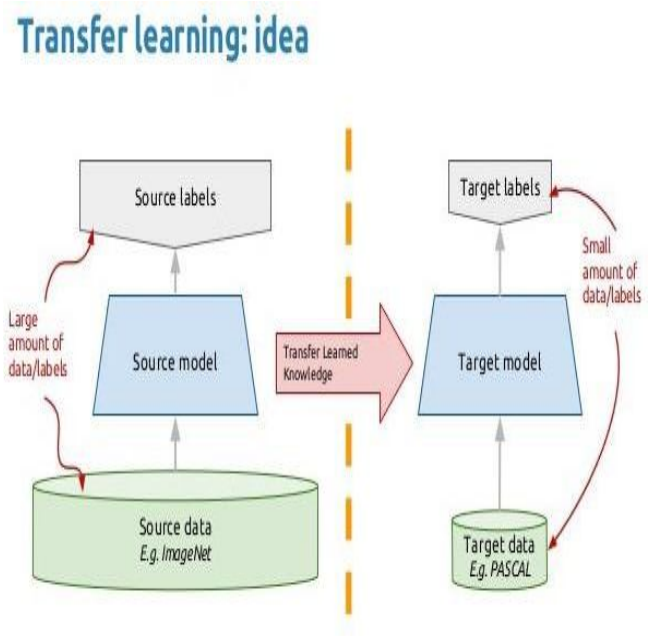
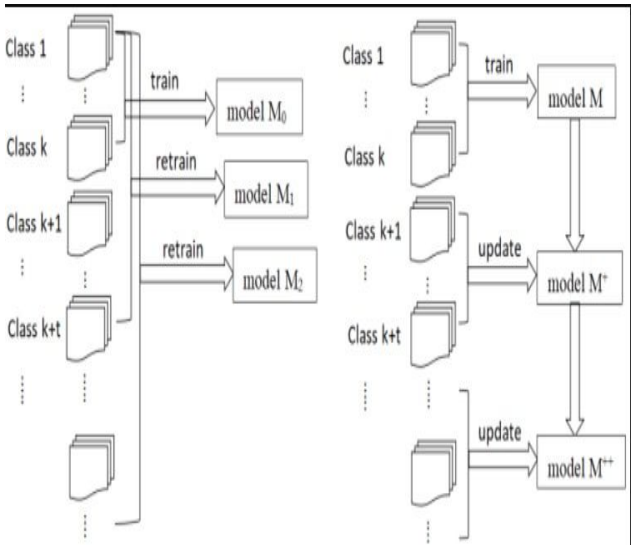


Fig. 1 Transfer Learning

The fig 2 shows difference between transfer and incremental learning is the ability add the new samples dynamically [9].



Traditional Machine Learning Incremental Machine Learning

Fig. 2 Traditional Vs Incremental Learning

3. METHODOLOGY

1. Importing the dataset, required modules and Libraries

We first import the dataset (MNIST, CIFAR10) from the client/user. Here the dataset is in the form of (CSV). We read this file by importing pandas module using read_csv() method.

2. Pre-Processing and extracting features from data.

We pre-process the data by removing missing values or redundant values and extract important features from the data

3. Split the data

We split the data into training, testing, and validation splits to perform training, testing, and validation respectively.

4. Model Building

Data augmentation is the method that is used to avoid over fitting of the data, the image is rotated into different angles and trained to the model. It helps to increase the diversity and uniqueness of the data that is available for training models. Some of the techniques are cropping, padding, and horizontal flipping. Build model can be divided into 3 parts. They are the Base Model, Flatten, and Dense.

a) Base Model

The base model or pre-trained model on which we want to train our machine and extract the features from them. In our case we use the weights VGG19 and ResNet50 which are built on the ImageNet. These are used as the base model and our model is built on this base model with some custom layers.

b) Flatten

We flatten the output of CNN to create a single feature vector, this is done by converting data into a 1-dimensional array as input to the next layer.

c) Dense

Dense Layer is similar to a regular layer of neurons in the neural network. Each neuron receives input from all the neurons in the previous layer, thus it is known as densely connected.

Any neural network without activation function is just a linear model. The activation function helps us to do the non- linear transformation to the input which helps us to learn complex tasks. We use 'ReLU' activation functions for all the hidden layers and 'Softmax' activation for the last fully connected layer.

5. Evaluation of metrics

Metrics evaluation to identify the performance of the model in terms of different metrics. For example precision-recall, accuracy, F1 score, ROC curve [2][4].

4. IMPLEMENTATION

Transfer/Incremental Learning is the process of building a model that is trained on a dataset and extract the features from it and use it as an initialization to the problem of similar task. This process of using another trained model for initialization is called as a pre-trained model. The Fig 3 shows pre-trained models are usually trained on benchmark datasets to solve a problem that is similar to ours. The pre- trained model we

used VGG19 and ResNet50. This VGG19 and ResNet50 both are trained Convolution Neural Network (CNN) where 19, 50 represents layers in the ConvNet[8].

To implement the transfer learning we follow 2 major steps: Chose the pre-trained model accordingly:

We need to choose which pre-trained model best suits our model. The problems must belong to a similar kind. For example for Image classification we use ImageNet trained VGG19 which has several images [3].

Restructure your CNN,after choosing the pre-trained model we freeze all the layers and remove the last fully connected layer to add custom layers related to our problem. We extract the necessary bottleneck features for our mode. It helps us build more accurate models while saving time. For optimizing w have chosen SGD optimizer and adam optimizer. SGD maintains the same learning rate throughout the training for all the parameters, adam (Adaptive Moment Estimation) updates the learning rates accordingly [6].

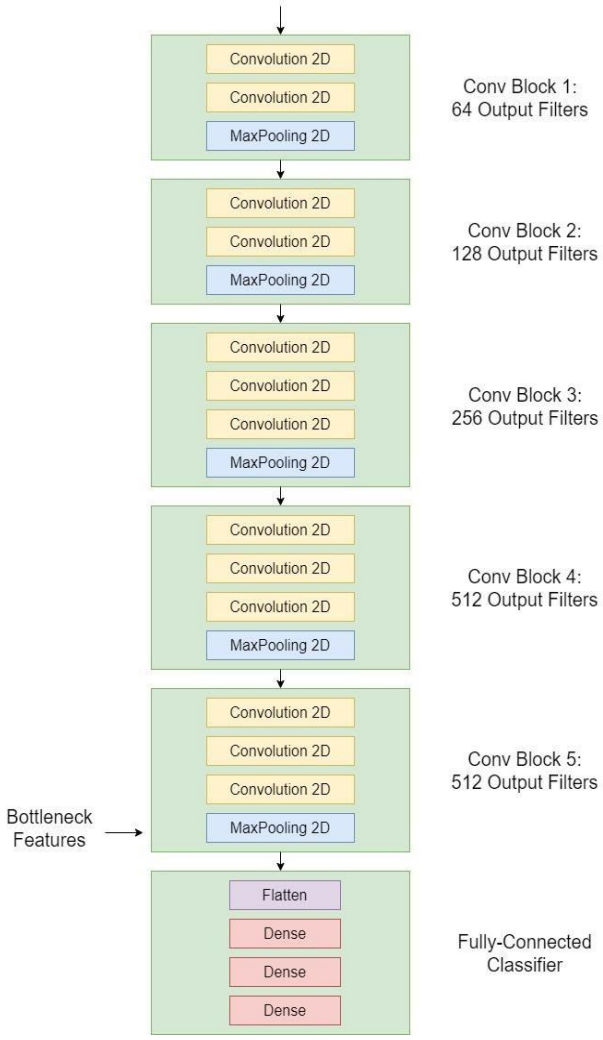


Fig 3. Architecture of VGG19

A Each CNN is divided into a convolution base and a classifier. Whenever we what to use this as the pre-trained model we just have to remove the last fully-connected classifier and build our own classifier.

5. SYSTEM ARCHITECTURE

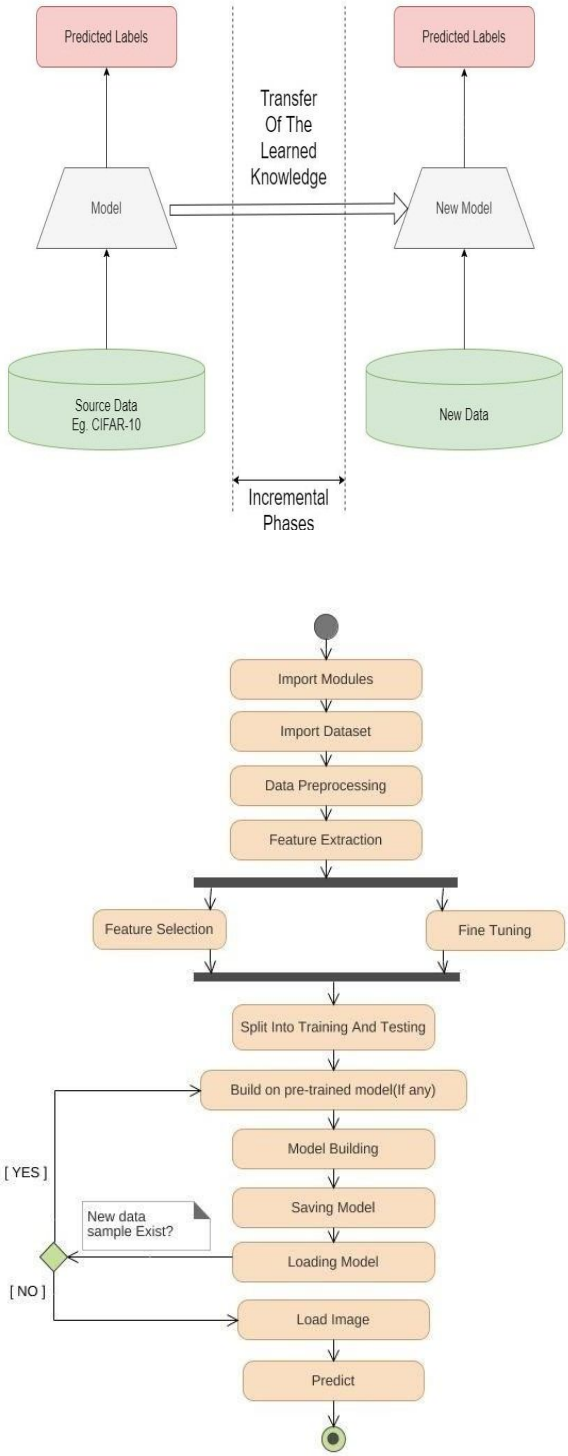


Fig 4: Flow diagram for Learning Automata based Incremental Learning.

A system architecture fig 4 diagrams would be used to show the relationship between different components. Usually they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them. However, it can also be created for web applications [7][8].

The Modules used

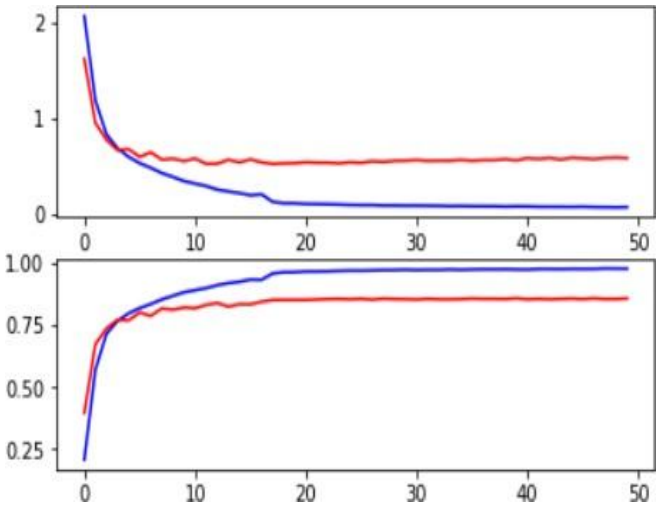
i)Pandas is an open-source, and has easy-to-use data structures and data analysis tools.

ii)NumPy is a general-purpose array-processing package. It provides tools for working with these arrays.

iii)MatPlotLib: It is a plotting library used for graphics in python. It can be used in python scripts, shell, web applications.

iv)Scikit-learn: It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports libraries like NumPy and SciPy.

6. DATA VISUALIZATION



This is a subplot in which the upper part depicts the plot between the training and validation loss and the lower part depicts the plot between the accuracy of training and validation accuracy [10].

7. RESULTS

```
In [30]: if result[0]==0:
             print("Aeroplane")
         elif result[0]==1:
             print('Automobile')
         elif result[0]==2:
             print('Bird')
         elif result[0]==3:
             print('Cat')
         elif result[0]==4:
             print('Deer')
         elif result[0]==5:
             print('Dog')
         elif result[0]==6:
             print('Frog')
         elif result[0]==7:
             print('Horse')
         elif result[0]==8:
             print('Ship')
         elif result[0]==9:
             print('Truck')
         else:
             print('Error')

         Bird
```

Fig 5: Image for prediction from user

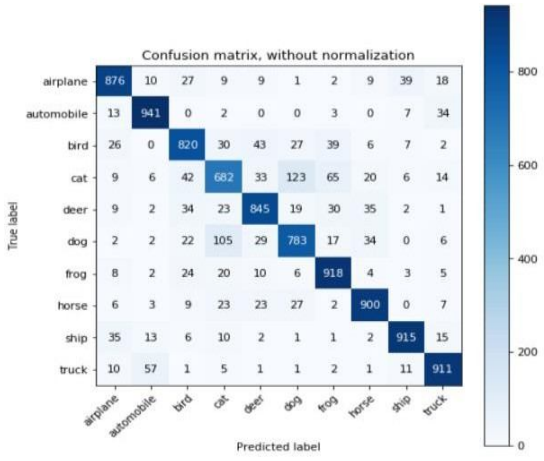


Fig 6: Confusion Matrix of the trained model without Normalization.

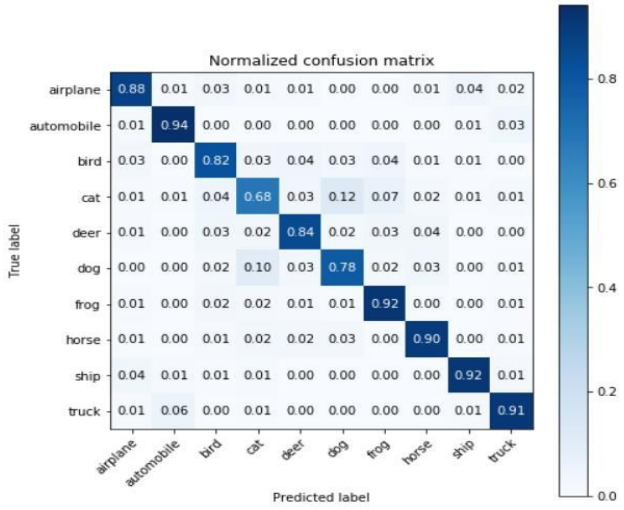


Fig 7: Confusion matrix with Normalization

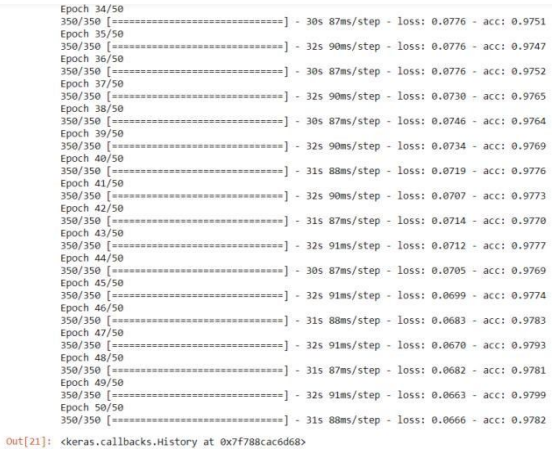


Fig 8: Accuracy and Loss Metrics of the Incremental Model

Fig 9: Accuracy and Loss Metrics for Non-Incremental

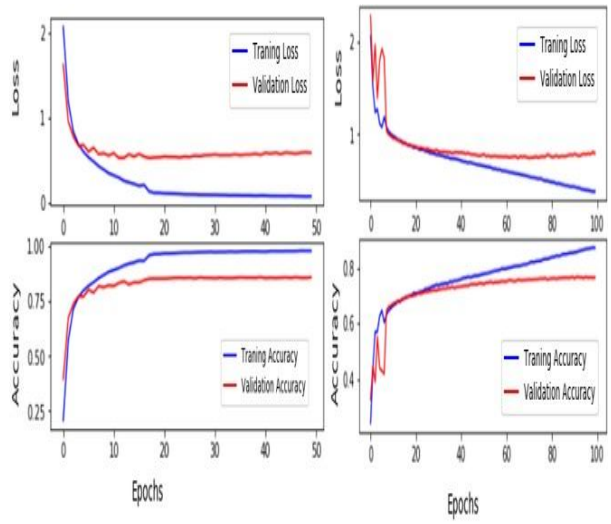


Fig. Data Visualization of VGG19 Model

Fig. Data Visualization of ResNet50 Model

Fig 10: Data Visualization VGG19 and ResNet50 Model

8. CONCLUSION

In this paper, we developed a neural network in an incremental method by accepting new samples. To overcome the phenomenon of catastrophic forgetting we proposed this method of dynamic connections. We used optimizers like SGD and ADAM to improve the performance. We were able to build the model on the pre- trained model without building it from scratch and improve the computational cost and accuracy. The experiments we conducted on MNIST and CIFAR10 helped us to build this incremental model. We can retain the previously learnt data and acquire additional data when needed. Incremental Learning is a method of machine learning in which it is continuously accepted even after training the model which is shown from fig 5 to fig 10. As future work, the proposed method could be applied to sample imbalance and even chunk learning.

ACKNOWLEDGEMENT

The authors wish to thank Sreyas institute of engineering & technology supported in providing infrastructure

REFERENCES

1. Christophe Giraud-Carrier, **"A Note on the Utility of Incremental Learning,** " Department of Computer Science, University of Bristol, Bristol BS8 1UB, U.K. AI Communications ISSN 0921-7126, IOS Press1998.

2. Pavel Laskov, Christian Gehl, Stefan Kr¨uger, **"Incremental Support Vector Learning: Analysis, Implementation and Application3s**, " Fraunhofer-FIRST.ID, Kekul´estrasse, 12489 Berlin, Germany, Journal of Machine Learning Research (2006).

3. Stephan K. and Chalup, **"Incremental Learning in Biological and Machine Learning Systems**", International Journal of Neural Systems, Vol. 12, No. 6 (2002), pp: 447{465} World Scientific.

4. Jeffrey C. Schlimmer, Richard H. Granger, Jr., **"Incremental Learning from Noisy Data**", Journal of Machine Learning 1, pp 317-354, 1986.

5. Tristan Ronald Ling**, "An Incremental Learning Method for Data Mining from Large Databases**," University of Tasmania 2006.

6. Prerana Gupta, Amit Thakkar, Amit Ganatra, **"Comprehensive study on techniques of Incremental learning with decision trees for streamed data**," International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-3, February 2012.

7. Schmidhuber**,''Deep learning in neural networks: An overview,** ''Neural Network., vol. 61, pp. 85–117, Jan. 2015.

8. A. Krizhevsky, I. Sutskever, and G. E. Hinton ''**ImageNet classification with deep convolutional neural networks**,'' in Proc. Int. Conf. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

9. SamanehKhoushrou, Jaime Cardoso, LiisTeixerira, **"Evolution of different Incremental Learning Methods for Video Surveillance scenarios**", Faculdade de Engenharia Universidade do Porto, 2010.

10. K. Simonyan and A. Zisserman. (2014). **''Very deep convolutional net- works for large-scale image recognition.''**