2017 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017, India

Classifying multi-category images using Deep Learning : A Convolutional Neural Network Model

Ardhendu Bandhu,
School of Computer Science and Engineering,
VIT University,Vellore,
ardhendu.bandhu2015@vit.ac.in

Sanjiban Sekhar Roy,
School of Computer Science and Engineering,
VIT University,Vellore,
sanjibanroy09@gmail.com

Abstract— This paper presents an image classification model using a convolutional neural network with Tensor Flow. Tensor Flow is a popular open source library for machine learning and deep neural networks. A multi-category image dataset has been considered for the classification. Conventional back propagation neural network has an input layer, hidden layer, and an output layer but convolutional neural network, has a convolutional layer, and a max pooling layer. We train this proposed classifier to calculate the decision boundary of the image dataset. The data in the real world is mostly in the form of unlabeled and unstructured format. These unstructured data may be image, sound and text data. Useful information cannot be easily derived from neural networks which are shallow i.e. the ones which have less number of hidden layers. We propose deep neural network based CNN classifier which has a large number of hidden layers and can derive meaningful information from images.

Keywords— Image, Classification, Convolutional Neural Network, Tensor Flow, Deep Neural Network.

I. INTRODUCTION

Image classification refers to the function of classifying images from a multiclass image set. For classifying an image dataset into multiple classes or categories, there must be a good understanding between the dataset and the classes. Critical condition of accurate classification is based on adopted learning techniques and feature sets. In the recent past, machine learning techniques were dependent on using networks which were shallow and which do not have a large number of hidden layers.

The familiarity between these models is the elementary architecture in which a single layer is responsible for converting the basic input onto the problem precise feature space. Networks, which are shallow, are efficient in solving simple problems but in the case of the more complex real-world problems, they are not efficient enough. The essential task of a Neural Network is to take a set of inputs, execute steadily complicated computation and use this computation to work out the problem. A neural network consists of many interconnected points called as neurons, which produce output using a range of activation functions. A neuron which is interconnected through another neuron, gets activated with the help of weighted connections. The “Shallow Neural Network” has been around for a long time and has also been used widely in the past.

The artificial neural network research was the focal point for the origination of the concept of deep learning. The first and the simplest neural network was known as single layer perceptron, which do not have any hidden layer. Afterwards, multi-layer perceptron was introduced which had very few hidden layers. By increasing the number of hidden layers we have moved from a shallow network to deep networks.

A very common example of deep architecture is a Feed-forward neural networks constituting of many hidden layers. Back-propagation is widely used and a popular learning algorithm. When we train deep network, with backpropagation, we run into a fundamental problem called vanishing gradient. During this time training takes lot of time and the accuracy suffers. When we train a neural network, we actually calculate a cost value; the cost is the difference between the network’s predicted output and the actual output from the set of training data. The cost is lowered by making the adjustments to the weights and bias. This happens throughout the training process until the lowest possible value is obtained.

In this paper, we address the following,

1. We propose deep learning based Convolution Neural Network for classifying images.

2. The proposed model achieves high accuracy after we iterate for 10000 times, on our dataset containing 20,000 images of dogs and cats, which takes around 300 minutes to train and cross validate our dataset.

Deep learning is one of the sub domains of machine learning. in which pattern classification and representation learning is achieved by encompassing the different layers of data handling stages in hierarchical models. The extensive usage of deep learning these days is due to the factor that the capabilities of chip processing has increased many a times, the reduction in the price of computing hardware and the machine learnings has been advanced by manifold. The problem of optimization of deep learning has successfully been eliminated when a considerably effective unsupervised learning algorithm was initiated in 2006[1]. A new class of deep generative model was identified by these papers, called deep belief network (DBN), which is constituted by layer by layer stack of Restricted Boltzmann Machines (RBMs).

A Deep Boltzmann Machine(DBM) constitutes of many layers of hidden variables, and no variables has any kind of association between them within the same layer. In Deep Boltzmann Machine(DBM), if the number of hidden layers is reduced to one, we have Restricted Boltzmann Machine (RBM)[2].

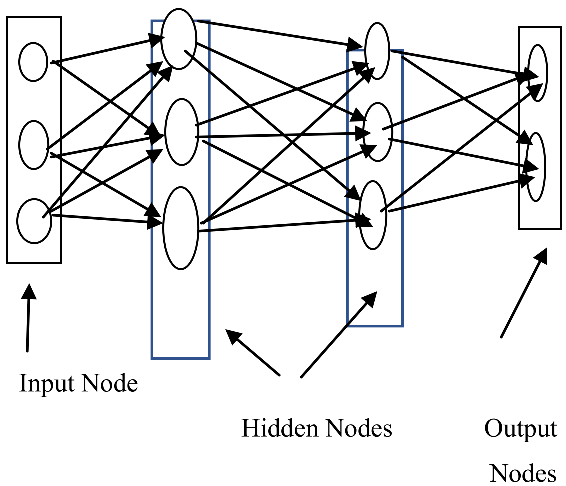

Figure.1. Deep Neural Network [1][2]

$$\text{Output } y_i \Rightarrow f(w_i^1 x_1, w_i^2 x_2 \ldots\ldots\ldots w_i^n x_n \qquad (1)$$

$$\Rightarrow f\left(\sum w_i^j x_j\right) \qquad (2)$$

CONVOLUTIONAL NEURAL NETWORK

We use a convolutional neural network which consists of a convolution layer, a RELU, a pooling layer and a Fully-Connected layer. Convolutional neural network is an automatic choice when it comes to image recognition using deep learning.

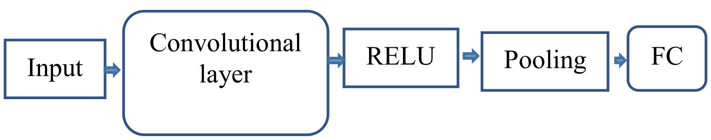

Figure 2. Convolutional Neural Network

For classification purpose the convolutional network can have the architecture as [INPUT - CONV - RELU - POOL - FC].

INPUT- It will contain the image raw pixel values.

CONV-The output from the first set of neurons will be contained In it.

RELU- The activation function will be applied by it.

POOL- It will do a down-sampling operation.

FC- It will compute the class score.

Forward propagation in convolutional neural network is represented by-

$$x^l{}_{ij} = \sum_{a=0}^{m-1} \quad w_{ab} y^{l-1}(i+a)(i+b) \qquad (3)$$

Backward Propagation in convolutional neural network is represented by-

$$\frac{\partial E}{\partial w_{a,b}} = \sum_{i=0}^{N-m} \quad \sum_{j=0}^{n-m} \quad \frac{\partial E}{\partial xl_{ij}} \frac{\partial xl_{ij}}{\partial w_{a,b}} \qquad (4)$$

By using chain rule, we obtain,

$$W_{ab} \frac{\partial E}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1} \quad \sum_{b=0}^{m-1} \quad \frac{\partial E}{\partial x^l_{(i-a)(j-b)}} \qquad (5)$$

Why Deep Learning with Tensor Flow is getting popularity? In Tensor Flow, all data are passed between operations in a computation graph, in the form of tensors, hence the name 'Tensor Flow'. Tensor Flow supports CPUs, GPUs and distributed processing as well. This is in turn very much favourable due to the increased processing capabilities. The Tensor structure helps us by giving the freedom to shape the dataset the way we want. Tensor Flow in comparison to other deep learning libraries compiles faster and it is easier to use. Tensor Flow structure is based on data flow graph. It has extensive built-in support for deep learning and neural networks. It has wide variety of tools to assemble neural networks and has mathematical functions for neural networks which will in handy for classifying of cat vs dog images. Tensor Flow has a versatility attached with it which is very useful for wide variety of machine learning problems and solving complex real world problem. Deep learning uses Linear Algebra, Statistics and Calculus. Tensor Flow cannot be just used for neural networks; it can be used to implement other regression problem like linear regression and other classifications. Tensor Flow has a wide variety machine learning algorithms which are contained inside it's estimator module.

i) Tensor Flow also hosts a numeric deep learning APIs such as Deep Neural Networks, Recurrent Neural Networks and algorithms for machine learning. A wide assortment of statistical distributions capacities is given by Tensor Flow, including Bernoulli, Beta, Chi2, Dirichlet, Gamma, Uniform etc. Tensor Flow has operations that generate the associated weights and bias required to build a deep architecture which is an added advantage for anybody trying to implement a deep architecture like us and also in layered operations having many hidden layers. Using this, we can easily implement functions such as batch normalization, dropout layer, and convolution layers. Optimizers are an essential feature to optimize our data and help in solving the problem of optimization for numerical analysis. This includes finding a better model by optimizing parameter space and also to optimize our inputs. Sometimes we can overfit our model by using large number of features in our model. Tensor Flow has regularizers which prevent our model from getting overfitted and by providing a penalty to large number of features used in the model. Tensor Flow provides a huge variety of initializers. Tensor Flow also provides a platform to converge the functionalities in our data into both continuous and categorical features. Machine learning algorithms need a loss function which is provided by Tensor Flow to provide optimizations.

ii) Feature engineering: It is the process of extracting useful patterns from the input data in order to help the predictive model understand the true nature of the problem. Finding the right features can improve an algorithm's accuracy and performance beyond what a machine learning model could do on its own. A feature learning algorithm will determine which features are useful for distinguishing between different classes. After this process is complete, the extracted features are ready to be used for classification or even regression.

II. RELATED WORK

The earliest work and the evolution of Deep learning was explained in[1] [2], why we needed deep architectures and deep learning and all the major aspects of Deep learning, the various types of deep neural network and deep architectures was also described in paper. These two papers have explained all the essential things and the basics of learning algorithms for deep architectures and also laid down the basics for constructing unsupervised learning models such as Restricted Boltzmann Machines which are the basic building block of Deep Belief Networks. Paper [4] describes useful concepts for deep supervised learning, unsupervised learning, reinforcement learning & evolutionary computation, and indirect search for short programs encoding deep and large networks. Paper [5] has formulated the idea of a spectral–spatial feature learning (SSFL) method to gather important features of hyperspectral images (HSIs). Paper[6] has enumerated the knowledge of Active deep learning classification of hyperspectral images. Selection is enhanced by selecting samples that maximize it. A deep network is perfectly trained in this paper[6] while selecting samples after each iteration. The illustrated algorithm in this paper[6] is useful for the process of classification of hyperspectral images and a comparison is made between other methods. A very down to earth approach for image classification using deep learning networks was provided by authors [7].Work of paper [8] plans a multistage profound learning system for picture characterization and applies it on body part acknowledgment. In particular, the proposed structure shows 1) How to find the nearby districts that are discriminative and non-instructive to the picture grouping issue, and 2) How can picture level classifier in view of these neighbourhood areas happens. Paper [9] has produced a convolutional neural network which is to detect a nucleus. CNN relapses the probability of a pixel being the focal point of a core, where high likelihood qualities are spatially obliged to situate in the region of the focuses of cores. Work[10] has composed an algorithm to transform the raw input into a representation, followed by a trainable classifier. They showed that by learning the hierarchies by themselves from huge datasets, deep convolutional networks have made an impressive gain in the field of computer vision and speech recognition [10]. Paper [11] has proposed a method to use a pre trained model to retrain it and a regression model has been used to classify the type of blur image and then to calculate its parameters**.** Deep Learning is a branch machine learning algorithms that mainly used for the processing of nonlinear type [12]. Moreover, in the literature many applications of other machine learning algorithms can be found in various engineering domains [13][14][15][16][17].

III. METHODOLOGY

We have used a dataset of cat vs dog images, containing a total of 20,000 images from Kaggle data base (https://www.kaggle.com/c/dogs-vs-cats/data)[3]. In kaggle database, there are total 25000 images available. We divide the images into training set and test set. We input 12,000 images to the training set and 8,000 images to the test set. The divided dataset of training set and test set helps in cross validation of data and keep a check on the errors; besides cross validation checks whether proposed classifier is accurately classifying the cat vs dog images or not. Kaggle is a data science community; since we need to classify images, the bigger the number of images in the dataset, the better we can test our classifier for it accuracy. Since Tensor Flow is an open source library, so it is easily available for download. We prefer python for implementation, as in python classification and implementing of deep learning libraries are relatively easy. We use a wide variety of packages and machine learning libraries like the Tensor Flow, Numpy and Scipy. Packages help us work with various complex functionalities, easily.

Building a Graph

Tensor Flow structure is based on data flow graph. We create a graph with the following computation units:

Nodes=Mathematical Operation

Edges=Multi-Dimensional Arrays(Tensor)

Tensor Flow defines computation as graphs and these are made with operation in short written as Ops. To execute these operations as computations, we must launch the graph into a session. The session translates and passes the operations represented into the graphs to the device you want to execute them on, be it a GPU or CPU. The Tensor Flow needs to initialize a session to run the source code. Sessions are, in a way, a context for creating a graph. We always need to create a session object to launch the graph. The following experimental setup is done on Spyder, which is a scientific python development environment.

i)First the required libraries such as Scipy, Numpy and Tensor Flow, Numpy are essentially used in python for scientific computing; generic data can also be contained inside Numpy which in terms help Numpy collaborate faster with a wide variety of databases.

ii) A start time, train path and a test path are fixed. We provide the image height and image width as 64 pixels. Then image dataset which contains 20,000 images is loaded. The images are loaded after iterating and it takes 5-10 minutes to load images because of its huge numbers.

iii) Now that data is fed to Tensor Flow. In Tensor Flow, all data is passed between operations in a computation graph. The tensors structure helps us giving the freedom to shape the dataset the way we want. Tensors are generic version of vectors and matrices. Vector is a one-dimensional array. Matrix is a two-dimensional array of numbers. We convert the input features and labels to input tensor. The features and labels should be in a matrix form so that Tensors can easily interpret it.

iv) Notice this is a declaration step where we instruct Tensor Flow how the prediction to be calculated. We have to characterize and run the session, however since we have made a "gap" in the model to pass the information, when we introduce the session we are committed to pass a contention with the information else we ought to get an error. In order to invoke the data inside the model, we initiate the session with an additional argument inside which all placeholders name with their corresponding data is placed. Data in Tensor Flow is passed in form of multidimensional array; we can pass any kind of Tensor through the placeholders to get the answer to the sample multiplication operation**.** Since data in Tensor Flow is passed in form of Variables need to be initialized before a graph can be run in a session. To update the value of a variable, we define an update function We can then run the operation. Next, we can then start the session and run the graph. After the variables are initialized, we print the initial value of the state variable, we run the update operation, and then we print the result after each update. Next, we feed Tensor Flow an array of example images. We also maintain a weight matrix which we keep updating throughout the training process. We also maintain bias values. The first step in calculating the prediction would be to multiply the input matrix by the weights matrix and then add the biases. After this it is the turn of the activation function, the choice of activation function has a big impact on network's behaviour. Activation function for a particular node is the output of that particular node provided an input or a set of inputs.

v) Next, we define the hyper-parameters which we will need to train our input features. More complex the Neural Network, more hyper-parameters we come across. Some of our hyper-parameters can be like the learning rate; learning rate defines how fast our model reaches convergence that is achieves accuracy. Another hyper-parameter is the number of iterations, which is the number of times we train our data. The next hyper-parameter is the batch size, which chooses the size of image set to be sent for classification at one time.

Since we are using a convolutional neural network for classification, we need to define the various functions for the different layers of the convolutional neural network like one for the convolutional layer, one for RELU, one for pooling layer and one for fully connected layer. We have used Softmax as our activation function. Softmax is an activation function normally used in classification problem. It generates the probabilities for the output instead of yes or no answer. A machine does not have all this certainty, so we want to know what is the best guess, but we also want to understand how sure it was and what was the second better option.

vi) Finally after all this, we initialize the Tensor Flow session, which makes Tensor Flow run, as without initializing a session a Tensor Flow would not work. After that our model will start the training process.

IV. RESULTS AND DISCUSSIONS

Image classification has been carried out in the past also but the accuracy of classifying images was not that much impressive. After the introduction of deep networks, accuracy for image classification enhanced very much. We used a convolutional neural network as our deep architecture and with that we applied the Tensor Flow deep learning library. The following experimental setup was done on Spyder, which is a scientific python development environment. We used 20,000 images and the batch size was fixed to 100. It is essential that the accuracy of models should be examined in terms of testing data rather than with the training data. For running the convolutional neural network using Tensor Flow we used windows 10 machine, with our hardware having a 8gb ram, which had CPU version of the Tensor Flow installed in it. Time for training the data, depends on the number of iterations we assigned to the model. Here in the current work, separate convolutional neural network was developed using the Tensor Flow model, to enhance the performance of the convolutional neural network. As the number of iterations is increased our training accuracy also increases but our training time also increases. Table 1 shows the plot of number of iterations with accuracy we achieved.

TABLE 1 NUMBER OF ITERATIONS VS ACCURACY

Number of iterations	**Accuracy**
100	0.546
500	0.755
1000	0.841
2000	0.932
3000	0.975
4000	0.998
5000	1.0
6000	1.0
7000	1.0
8000	1.0
9000	1.0
10000	1.0

In order to get the best possible image classification accuracy, we must use a liable number of iterations. The training time for 5000 iterations was approximately 180 minutes for this dataset and the training time for 1000 iterations was approximately 400 minutes. Training time can be drastically reduced if we use GPU support instead of CPU support though it would not affect the accuracy. After the 5000 iterations mark, our model was able to classify the images almost accurately with an accuracy of 1.0. The graph for accuracy vs number of iterations is given below

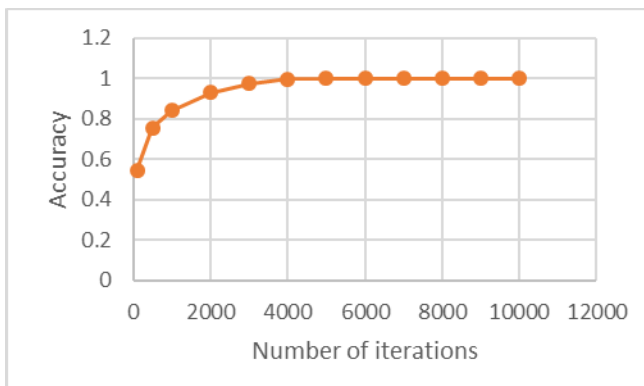

Figure 3 Number of iterations vs Accuracy

The graph is almost constant after a few thousand number of iterations. Different batch size may lead to different results, we used a batch size of 100 for the images.

CONCLUSION AND FUTURE SCOPE

In this paper, the proposed method has achieved a high accuracy in classifying the images. We have implemented the convolutional neural network using Tensor Flow. It has been shown that classifier performs well in terms of classification accuracy. However, we use a CPU based system, so experiment has taken extra training time, if we would have used a GPU based system, the training time would have been les. This model can be applied for solving complex image classification problem related to medical imaging and other fields.

REFERENCES

[1] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.

[2] Yoshua Bengio, "Learning Deep Architectures for AI", Dept. IRO, Universite de Montreal C.P. 6128, Montreal, Qc, H3C 3J7, Canada, Technical Report 1312

[3] Multi-category image Dataset from Kaggle.com

[4] Yann LeCun, Yoshua Bengio & Geoffrey Hinton, "Deep learning ", NATURE |vol 521 | 28 may 2015

[5] Jürgen Schmidhuber, "Deep learning in neural networks: An overview",neural networks,Elsevier,2014

[6] Yicong Zhou and Yantao Wei, "Learning Hierarchical Spectral–Spatial Features for Hyperspectral Image Classification",IEEE Transactions on cybernetics, Vol. 46, No.7, July 2016

[7] Peng Liu, Hui Zhang, and Kie B. Eom," Active Deep Learning for Classification of Hyperspectral Images",IEEE jounal of selectd topics in applied earth observation and remote sensing

[8] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma," PCANet: A Simple Deep Learning Baseline for Image Classification?", IEEE transactions on image processing, Vol.24, No..12, December 2015

[9] Zhennan Yan, Yiqiang Zhan, Zhigang Peng, Shu Liao, Yoshihisa Shinagawa, Shaoting Zhan, Dimitris N. Metaxas, and Xiang Sean Zhou, "Multi-Instance Deep Learning: Discover Discriminative Local Anatomies for Bodypart Recognition", IEEE transactions on medical imaging, Vol.35, No.5, may 2016

[10] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David R. J. Snead, Ian A. Cree, and Nasir M. Rajpoot, "Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images", IEEE transactions on medical imaging", Vol.35,No.5,May 2016

[11] Sizhe Chen, Haipeng Wang, Feng Xu, and Ya-Qiu Jin," Target Classification Using the Deep Convolutional Networks for SAR Images", IEEE transactions on geoscience and remote sensing, Vol. 54 , No.8, August 2016

[12] Roy, Sanjiban Sekhar, et al. "A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection." International Conference on Mathematics and Computing. Springer, Singapore, 2017.

[13] Roy, Sanjiban Sekhar, et al. "Applicability of rough set technique for data investigation and optimization of intrusion detection system." International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. Springer Berlin Heidelberg, 2013.

[14] Roy, Sanjiban Sekhar, et al. "Stock market forecasting using lasso linear regression model." Afro-European Conference for Industrial Advancement. Springer International Publishing, 2015.

[15] Basu, Avik, Sanjiban Sekhar Roy, and Ajith Abraham. "A Novel Diagnostic Approach Based on Support Vector Machine with Linear Kernel for Classifying the Erythemato-Squamous Disease." Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. IEEE, 2015.

[16] Roy, Sanjiban Sekhar, V. Madhu Viswanatham, and P. Venkata Krishna. "Spam detection using hybrid model of rough set and decorate ensemble." International Journal of Computational Systems Engineering 2.3 (2016): 139-147.

[17] Roy, Sanjiban Sekhar, and V. Madhu Viswanatham. "Classifying Spam Emails Using Artificial Intelligent Techniques." International Journal of Engineering Research in Africa 22 (2016).