Demystifying a Classical Image Classifier and Leveraging its Accuracy

Saurabh Khanolkar[1] Nimisha Bhide[2]
[1,2]Vidyalankar Institute of Technology, Mumbai, India

Abstract— Deep learning can be called as a type of machine learning which is concerned with computer performing activities that are generally performed by humans. Deep learning finds its use behind driverless cars, helping them understand and distinguish a stop sign, or to tell apart a pedestrian from another car. It is highly used in IoT (Internet of Things) like in Television, speaker, kitchen appliances and many more. Deep learning is receiving a lot of attention lately and for good reason. It's achieving results that were not possible before. In deep learning, the learning process takes place from the information gathered from images, audio and other sources and then the computational model performs the classification. Deep learning models can achieve the accuracy, which sometimes exceeding human-level performance. In this, initially, a model is trained by using a labelled data, which is mostly verified and neural network architectures that mainly consist many layers. In this paper we mainly focus on the optimization of different parameters of convolutional neural network of deep learning for classifying 8000 labelled natural images of cat and dog. Various level of optimization have been done to improve the performance level of the network and finally, we achieved the best classification accuracy of 93.10% achieved the best classification accuracy of 93.10%.

Keywords: CNN, Deep learning, Classical Image Classifier and Leveraging

I. INTRODUCTION

Artificial Intelligence is a type of intelligence which is given to the machines by humans, so that the working of machines would require lesser intervening of the humans. Deep learning is a part of machine learning in artificial intelligence which is responsible for decision making. It imitates the human brain and takes decisions by processing data and creating and evaluating patterns. The word "deep" in deep learning refers to how the machine has to think deeply to come to a conclusion like a human. It has the provision of unsupervised learning in which it learns from unstructured and unlabelled data. In the project, we use convolutional layers for image recognition and classification. Convolutional Neural Networks help to distinguish between two images and then put them into two different categories. In fact, deep learning was further bolstered by introduction of a first ever convolutional neural network which was namedLeNet5. It was designed by Yann LeCun in 1988. At that time, it had very menial jobs to perform.[6]

Convolutional Neural Networks work on four basic operations. They are Convolution, Non Linearity (ReLU), Pooling and lastly Classification. These operations are extremely crucial for understanding the CNN. CNN has got its name from its first step, that is, the convolution step. It is used to extract features from an image. It stores the spatial relationship of pixels by learning features of the image. The features of Convolution are controlled by depth, stride and zero-padding. Depth refers to how many filters were used for a convolution operation. Stride talks of by how many pixels we move the filter matrix on the input matrix. Zero-padding helps to control the size of feature maps.[7] Since most real-life operations are non-linear, we introduce non-linearity in our CNN. It is generally applied pixel-by-pixel. It is primarily used to convert negative pixel values in feature map by zero. The pooling step is also sometimes referred to as spatial pooling. It mainly works for reducing the dimensionality while preserving the important data from the feature map. It has various types, namely, sum, max, min, average, etc.[4]
Performance measures are extremely crucial when dealing with deep learning models. When a classifier is used for classification of images into categories, it does not provide a great accuracy. Additionally it becomes very difficult when one is dealing with a huge amount of data to be classified.[5] In this, we make use many different images to be classifies. While we put some images in the training set and train our model, and then use it for classification of the test set. We expect it to classify the images into two parts and discard the images which do not fit into either of the classifications. We also aim at optimizing the accuracy at 93.1%.

In this paper, we attempt to increase the efficiency of a model which can give a 25% accuracy when only one convolution layer is being used. But, when the number of convolution layers used was doubled, an accuracy of 93.1% was being observed.

II. MATERIALS AND METHODS

A. Dataset:

The dataset is a set of 10,000 images of two categories (Cat and Dog) collected from "kaggle" database, which is the world's largest community providing data platform for the machine learners to use.
Structure of the dataset:

	Dogs	Cats
Training set	4000 images	4000 images
Test set	1000 images	1000 images

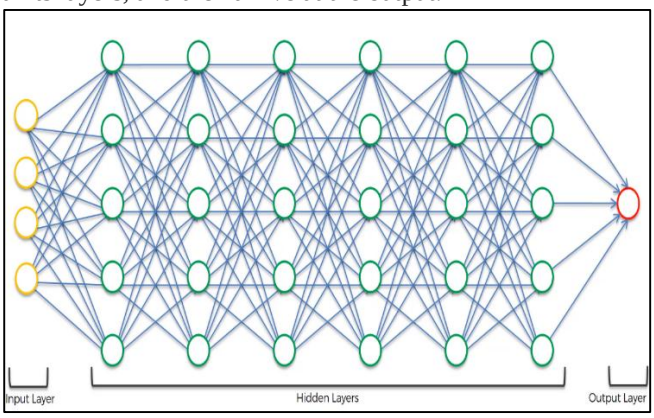
B. Model Description:

There are a lot of deep learning models accessible at present. The typical models include Autoencoder (AE), Deep Belief Network (DBN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

But first let us have a look at what a neural network is: A typical neural network has anything from a few dozen to hundreds, thousands, or even millions of artificial neurons called units arranged in a series of layers, each of which connects to the layers on either side. The input units are known to receive a variety of information from the world that has to be learned by our model, in order to recognize or to be processed.[9] The output unit is on the opposite side of the network and learns how the network and signal respond to certain things. In between the input units and output units are one or more layers of hidden units, which, together, form the majority of the artificial brain. [8]

The flow of information in a neural network takes place mainly in the two possible ways. When the model is

being trained, the input of the network is being loaded with patterns gathered, which are used for triggering the hidden units layers, and then arrive at the output.

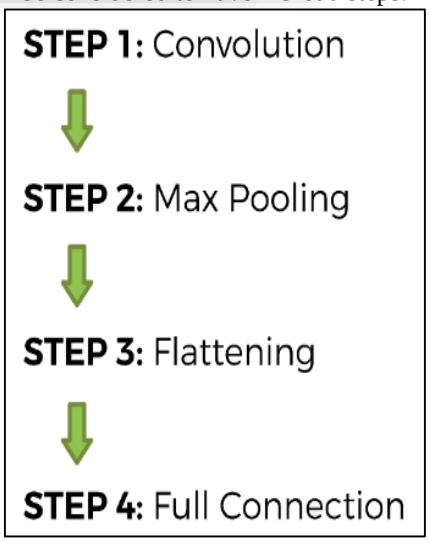


In this section we mainly acquaint with the CNN model.

Convolutional neural network (CNN) was at first put forward by LeCun in the early1980s. It was additionally changed by LeCun and his associates in 1990s[4]. CNN is an exceptional kind of deep, feed-forward artificial neural system that are essentially connected in visual patterns. [9]The CNN does the task of machine learning in the first part like feature extraction and learning and thereafter pattern grouping CNN model mainly depends on three key engineering thoughts: local receptive fields, weight sharing, and subsampling in the spatial space. The CNN is planned predominantly for the acknowledgment of 2-D visual patterns. It is widely accepted that CNN works perfectly for image problems and out-performed most of the other methods in image classification tasks.[1]

Convolutional neural systems are intended to accept 2-D pictures. A CNN comprises of three primary kinds of layers: (i) convolution layers, (ii) subsampling layers, (iii) an output layer.

A CNN can be considered to have 4 broad steps:



1) Step 1: Convolution

An Image is an array of pixels. The convolution operation makes the use of feature detectors to detect and learn the various features of the image. Multiple feature detectors are used to produce multiple feature maps that learn the different features of the image.

These feature maps are given as inputs to the neural network.

2) Step 2: Max pooling

In this, we pool the feature maps i.e. we retain the most useful feature information and discard the redundant features in each feature map, thus reducing its size.

The benefits of max pooling are:

1) Preserving the features
2) Introducing spatial invariance
3) Reducing the size by 75%(helps a lot in terms of processing)
4) Prevents over-fitting

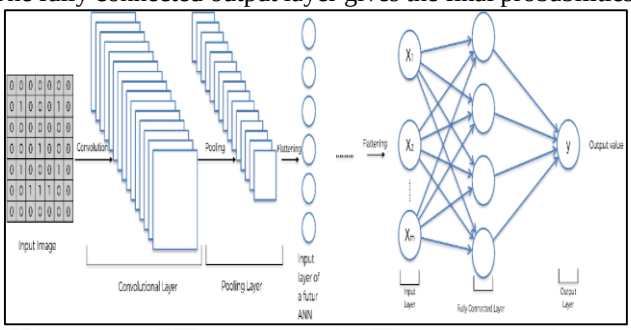
3) Step 3: Flattening

The pooled feature maps are flattened into 1D vectors to give as inputs to the neural network.

4) Step 4: Fully connected layers

They take input from the feature analysis and apply weights to predict the correct label.

The fully connected output layer gives the final probabilities.



Structure of a CNN

C. Deep Learning Frameworks Used:

Packages such as tensorflow, keras, pytorch, Theano, Caffe etc. can be used for deeplearning. TensorFlow is known for being able to work on images along with sequence-based data. As compared to TensorFlow, PyTorch is more intuitive. One which works excellently when bulding models of deep learning from data gathered from images is concerned is Caffe. But when it comes to recurrent neural networks and language models, Caffe lags behind the other frameworks.

Theano and Tensor Flow are the two best mathematical platforms in python that give the premise to deep learning research and development. However, these two powerful libraries are complicated to get started with.

In our paper, for this purpose, we make use of an open source library, which is mainly used for deep learning and written using python is the Keras library. It has the ability of running on top of tensorflow. Keras python library gives a perfect and advantageous approach to make a scope of deep learning models over Theano and Tensor Flow. Hence we have used Keras in our research.[9]

III. RESULT ANALYSIS

While compiling the CNN classifier, we use the 'adam' algorithm as the stochastic gradient descent algorithm, we use the 'binary cross entropy function' as the loss function (as we have a binary outcome i.e. cat or dog), we use 'accuracy' as the accuracy metrics.

To avoid over-fitting, we used the process of Image Augmentation. Over-fitting is the condition where we get a huge accuracy on the training set and a very low accuracy on the test set; this should be avoided.

We first implement the CNN with only one layer of convolution. We get an accuracy of 84.52% on the training set and 75.10% on the test set. There seems to be a large difference between the accuracies on the training set and the test set, which would not be classified as over-fitting, but is not desirable.

A. Challenges:

1) Increase the accuracy of the test set
2) Decrease the difference between the accuracies of the training set and the test set

1) Solution 1 Proposed:

This solution lies in the very essence of Deep learning; make a deeper neural network Therefore we add another layer of convolution after the first layer. This layer will be applied after the first layer, which means the input to this layer would not be images but they will be already pooled feature maps coming from the previous layer.

Here we get an accuracy of 85.16% on the training set and 81.80% on the test set.

Therefore, our goals are achieved:

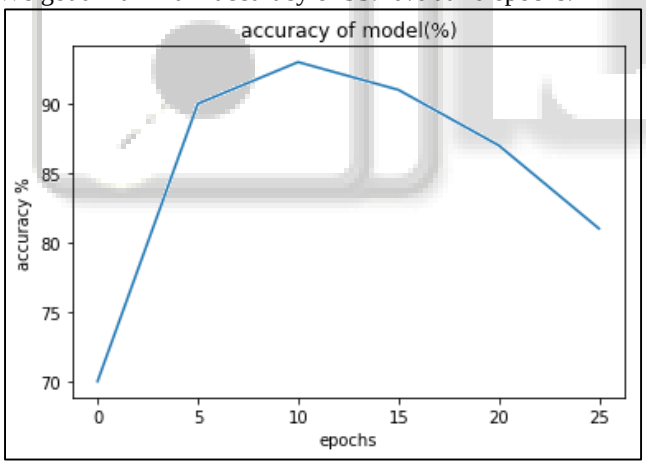
1) Increasing the accuracy of the test set
2) Reducing difference between the accuracy of the training and the test set.

2) Solution 2 proposed (To further increase the accuracy):

Here we varied the number of epochs that we ran the model on (with 2 layers of convolution).

The plot shows the dependence of the evaluation accuracy on the number of epochs.

We got a maximum accuracy of 93.10% at 10 epochs.



Finally, we managed to increase the accuracy of the model on the test set from 75.10 % on the test set (using 1 layer of convolution) to 81.80% (using 2 convolution layers) and to a maximum accuracy of 93.10% limiting the number of epochs to 10.

IV. Conclusions

We observed that one convolution layer was not able to achieve the desired accuracy and it was not giving justice to the motive. Hence, we made a shift to two convolution layers method, it was observed that it took the same number of epochs as one convolution layer and still did not achieve the desired accuracy. Then a new method was devised in which, the number of convolution layers was kept constant, the accuracy after each 5 epochs was noted. It was observed that, after reaching the maxima of accuracies, the accuracy kept on decreasing with each incrementing epoch.

References

[1] Harrison R. Continuous restricted Boltzmann machines. Wireless Networks. 2018;.
[2] Hart W, Watson J, Woodruff D. Pyomo: modeling and solving mathematical programs in Python. Mathematical Programming Computation. 2011;3(3):219-260.
[3] Liu M, Grana D. Accelerating geostatistical seismic inversion using TensorFlow: A heterogeneous distributed deep learning framework. Computers & Geosciences. 2019;124:37-45.
[4] LeCun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W et al. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation. 1989;1(4):541-551.
[5] Hinz T, Navarro-Guerrero N, Magg S, Wermter S. Speeding up the Hyperparameter Optimization of Deep Convolutional Neural Networks. International Journal of Computational Intelligence and Applications. 2018;17 (02):1850008.
[6] Tiwari T, Tiwari T, Tiwari S. How Artificial Intelligence, Machine Learning and Deep Learning are Radically Different?. International Journal of Advanced Research in Computer Science and Software Engineering. 2018;8(2):1.
[7] Barat C, Ducottet C. String representations and distances in deep Convolutional Neural Networks for image classification. Pattern Recognition. 2016;54:104-115..
[8] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review. 1958;65(6):386-408.
[9] Kaur A, Kaur G. A review on image enhancement with deep learning approach. ACCENTS Transactions on Image Processing and Computer Vision. 2018;4(11):16-20.

