Deep Learning for Image Classification

Harish Ravishankar
Department of Computer Science
University of Florida
Gainesville, United States of America
harish22@ufl.edu

***Abstract*—In this study, we propose deep learning techniques to implement an image classifier to recognize whether the animal in any given image is a dog or a cat. Previous methods of detection were based on classification pertaining to a specific task which did not use the vast volume of training data available. For identification task convolutional neural network is used. Key performance metric used for evaluation of the network is accuracy. Caffe framework will be used for modelling the architecture.**

Keywords—image classification; dog or cat; convolutional neural network.

I. INTRODUCTION

Image classification is fast becoming one of the most significant applications of deep learning in the real world. It is used to identify shapes, label objects and distinguish between species. Wildlife conservation is a domain where image classification plays a big role in identifying various kinds of species and matching it to a particular sub-species rather the vast spectrum of Animal or plant life in the wild. It is also applicable in industry such as rangeland management to specify areas of land cover. By training a system over a large number of labelled images of a certain class, the system can then be used to classify or identify objects within new images without supervision. CAPTCHA (Completely automated Public Turing Test to tell computers and humans apart) are now modified to make subjects classify random images to enhance security and avoid image recognition attacks.

This is a tedious task because of variation of the common subject (cat or dog) and similarity of the training data set while classifying the animal type in the image that is chosen. We propose a deep learning approach for the image classification. One of the most efficient deep learning models for visual recognition is the Convolutional neural network (CNN). The deep learning framework used is Caffe Model. The programming language used is Python.

Training a CNN using Caffe involves: Data Preparation, Model definition, Solver Definition, Model Training.

II. DATA SET

The data set source chosen for this task is Kaggle. Kaggle is a popular competition host for deep learning and machine learning challenges. It was chosen for its clear problem statement and large image volume. The dataset consists of 25000 images for training and 12500 for test purposes. For the purpose of the competition, the images contain only either cats or dogs. The images in the testing data are unlabeled and are classified using our trained model.

Fig: Sample of Cats and Dogs images from Kaggle Dataset.

Not all, but few images contain more than 1 dog or cat in them. None of the images contain both a dog and a cat in them.

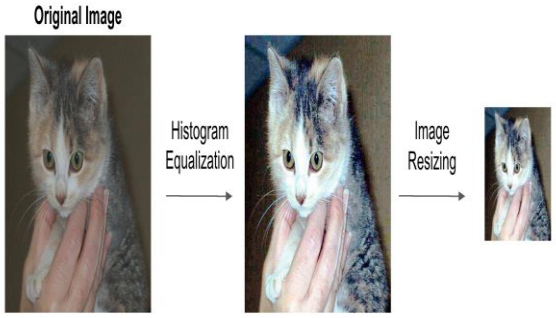
III. PREPROCESSING

First the training data was divided into training and for evaluation. 80% of the images are used for training and 20% is used in validation set to calculate accuracy of the model. Preprocessing on the data involved Dimensionality reduction, Histogram equalization.

On average the image dimension was **400x480.** One of the main issue encountered during the preprocessing phase was effective reduction of image dimensions without eliminating other features associated with the image. The images were resized to **227x227** in order to reduce the computational constraints that might arise during the dataset training phase.

Before resizing the image, Histogram equalization is applied on the original image. Histogram equalization is used to adjust

contrast in images. It is done on 3 color channels. Finally the image has reduced pixel resolution.



Example of image transformations applied to one training image

As stated earlier, after the data preprocessing task is done, we separate the databases images into training and test images.21000 images, which corresponds to 80% of it is used for training the algorithm and remaining 20% are used for testing purposes. The dimensions of the image are where 1 corresponds to the input picture channel with a picture size **120x120.** Images are segregated into classes from 0 to 9.

For the image classification task, we propose a learning mechanism of features from the images by the algorithm which is aided by usage of convolutional neural network layers. The input image is a histogram-equalized image. Features are derived from each image by convolutional layers. Global connectivity can be established for these individual images by using fully connected layers.

IV. Related work

Image classification using deep learning is usually done by neural networks. It is now being increasingly preferred over traditional machine learning classification. Traditional machine learning comprises of 2 phases: *Training and Prediction*. The training phase has 2 steps: *extracting the features and Training the model*.

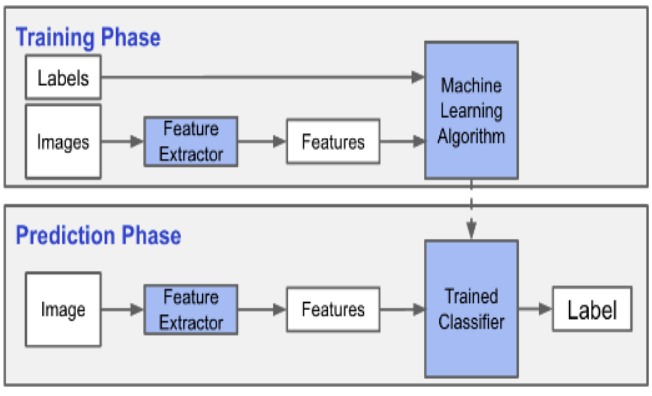


Fig: Machine learning phases(source: Wiki).

Machine learning involve tedious crafting of features. In Deep learning, the algorithms manage the feature engineering part.

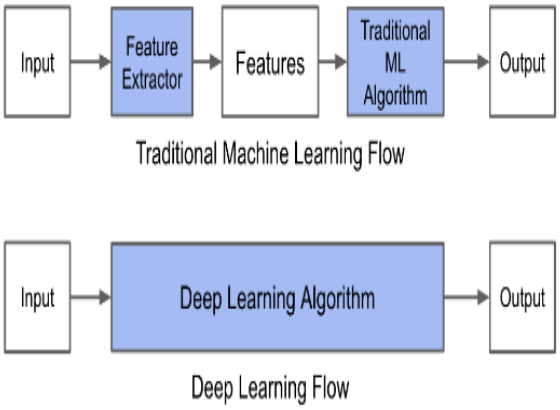
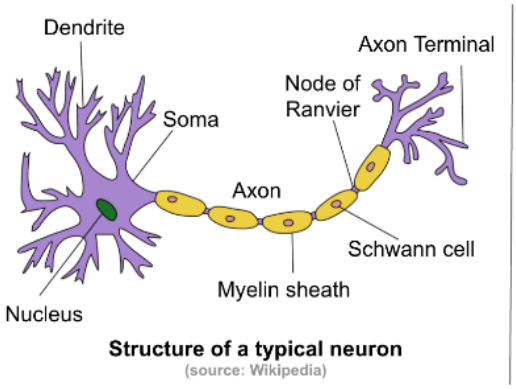
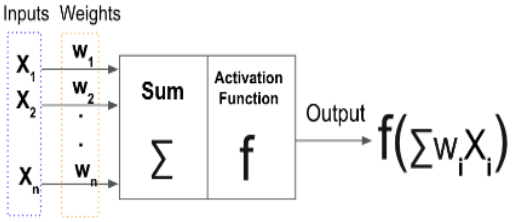


Fig: Deep learning Flow .

Artificial neural network (ANN) is a class of machine learning models based on biological neural networks. An artificial neuron consists of 2 things: A fixed number of inputs with weights, along with an activation function. Artificial neurons together form artificial neural networks. The output of the neuron is the outcome of the activation function applied on the total of weighted inputs.



Structure of a typical neuron
(source: Wikipedia)

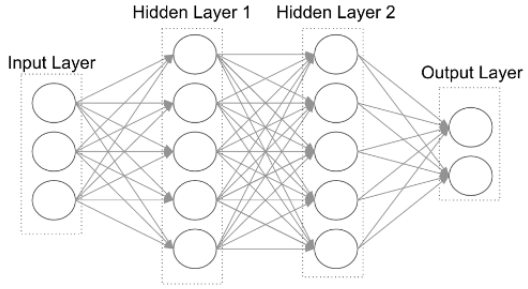


Structure of artificial neuron

The feedforward neural network is the first and simplest type of artificial neural network. The information moves in one direction, forward, from the input nodes, via the hidden nodes and to the output nodes. There is no cycles or loops in the network.

FFN have 3 layers namely Input, output and hidden. Data flows from the input layer through the hidden to the output nodes in these networks.

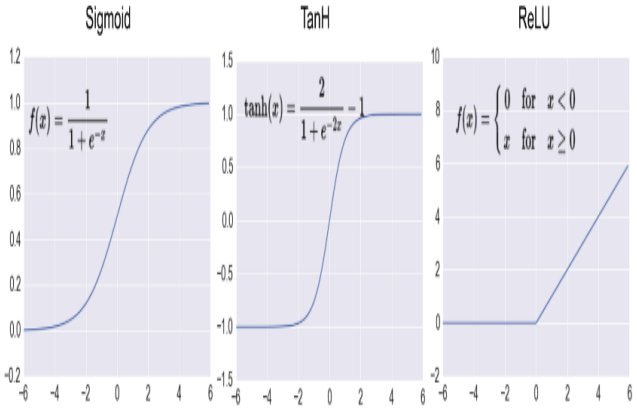
With a higher number of hidden layers, more complex patterns can be modelled. We are free to decide on the number of hidden layers and their size as parameters.



Feedforward neural network with 2 hidden layers

Fig : Feedforward neural network.

ReLU is a very popular activation function in deep neural networks, in addition to Tanh and Sigmoid.



Training data and a loss function are the 2 things required to model a neural network.

We do backpropagation, using the loss function and dataset, along with gradient descent to train the ANN

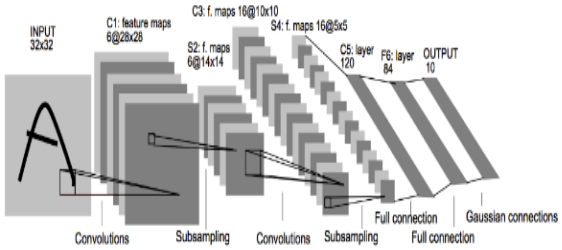
Currently used models have a general limitation in image classification which is their limitation in shallowness essential in predicting the features essential for the output.

Therefore, by utilizing a nonlinear convoluted neural network we optimized the subject detection process which showed better performance than all aforementioned techniques.

V. ARCHITECTURE

The goal of this project was to implement a CNN architecture to effectively predict the subject type in image (Dog or Cat). Convolutional neural networks are a type of FFN. They mimic the behavior of a visual cortex. CNNs have 2 layers primarily: convolutional layers and pooling layers. They help capture images properties CNNs perform highly on vision tasks.

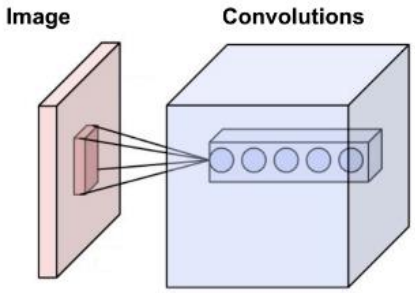
Convolutional Neural Network (CNN) was chose over than regular Artificial Neural Network (ANN) because CNNs are better in learning the localized features of images compared to ANN's.



CNN called LeNet by Yann LeCun (1998)

Convolutional Layer

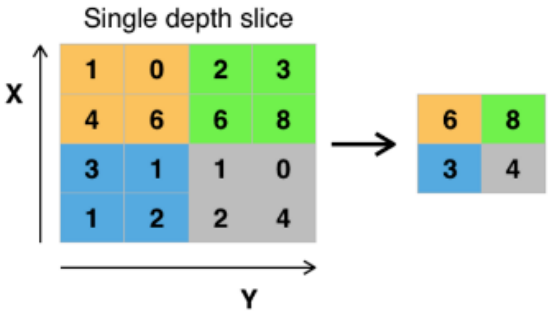
The convolutional layer comprises of a set of learnable filters that slide over the image, spatially computing dot products between the entries of the filter and the input image. These filters will activate when they see a particular structure in the images.



Neurons of a convolutional layer, connected to their receptive field
(source: Wikipedia)

Pooling Layer

The goal of the pooling layer is to reduce the spatial size of the representation and to control overfitting. Pooling is applied with filters of size 2x2 applied with a factor of 2 at every depth slice. A pooling layer of size 2x2 with stride of 2 shrinks the input image to a 1/4 of its original size. max pooling is one of the most common functions to implement pooling [2]

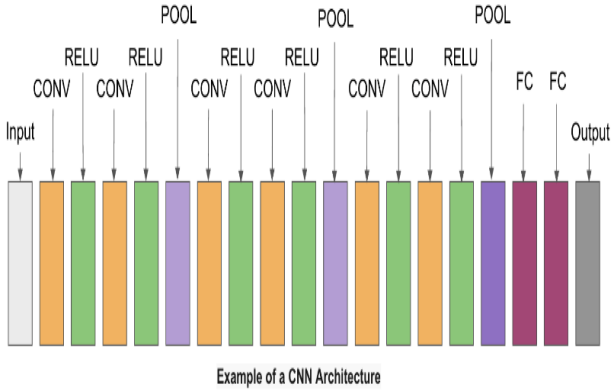


Max pooling with a 2x2 filter and stride = 2
(source: Wikipedia)

The convolutional layers are succeeded by ReLU activation functions layers.

The convolutional, pooling and ReLU layers are the feature extractors with learnability. The fully connected layers is the machine learning classifier.The early layers of the network capture generic patterns of the images, while later layers learn and encode the details patterns of the images.

Only the convolutional layers and fully-connected layers have weights. These weights were learned during training..



Example of a CNN Architecture

VI. IMPLEMENTATION DETAIL

A. Caffe Model

Caffe is an open source deep learning framework developed at the University of California, Berkley. It is written in C++ and has Matlab and Python capabilities.

B. Data Preparation

The key components of code used for data preparation below

transform_img(img,img_width=IMAGE_WIDTH, mg_height=IMAGE_HEIGHT):

transform_img accepts a colored image as input, performs the histogram equalization and resizes the image.

make_datum(img, label):
return caffe_pb2.Datum(
channels=3,
width=IMAGE_WIDTH,
height=IMAGE_HEIGHT,
label=label,
data=np.rollaxis(img, 2).tostring()).

make_datum takes an image and its label and return a Datum object that contains the image and its label.

We then generated the mean image of training set . Then subtracted the mean image from each input image ensures each pixel of feature has 0 mean.

C. Model Definition

Caffe offers choices of CNN models. For our implementation, we use bvlc_reference_caffenet model. We changed the number of outputs from 1000 to 2 for our implementation asbvlc_reference_caffenet is orginally a classification problem with 1000 categories.

D. Solver Description

The solver handles model performance. Using the validation set the solver computed the accuracy percentage of the model for every 1000 iterations. The optimization ran for a maximum of 40000 iterations and took a snapshot of the trained model every 5000 iterations.

We started with a learning rate of 0.001, and for every 2500 iterations. reduced the learning curve by a factor of 10.

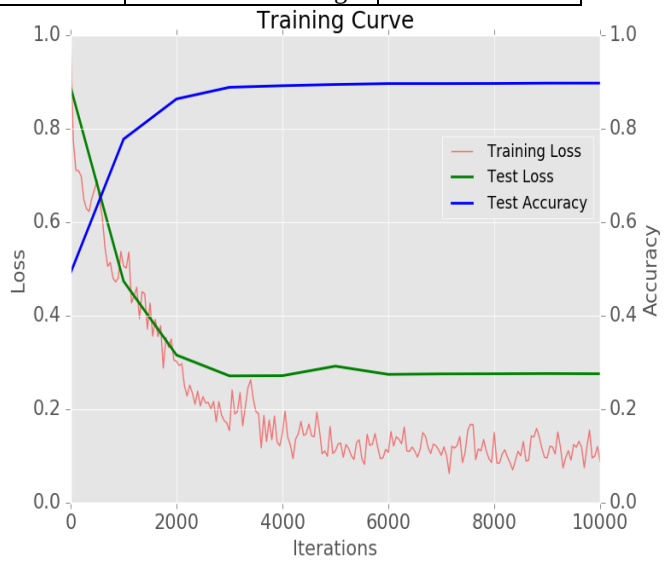
E. Model Training

During the training process, the loss and the model accuracy were monitored. The Caffe snapshots (after every 5000 iterations) had a. caffemodel extension.

VII. PERFORMANCE EVALUATION

We observed that the system had a correctness in prediction of 90% and after 3000 iterations the improvement stopped.

Architecture	Model Training Method	Accuracy Achieved
CNN	From Scratch	86.691
CNN	Transfer Learning	97.154



VIII. TRANSFER LEARNING PERFORMANCE

Convolutional neural networks need huge training data and a lot of time to train. Transfer learning is a handy method that aims to solve these issues. Transfer learning involves using a pre-trained model on a different dataset, and adjusts it to our particular problem. This is much more efficient than training a model from scratch.

Caffe offers a library for machine learning scientists to share and use trained models. This repository is called Modelzoo. This model was pre-trained on the ImageNet dataset which contains millions of animal images over 1000 categories.

To train the model with transfer learning we used bvlc_reference_caffenet. We observed that with just 1000 iterations the model's learning curve attained a score of 97%. This highlighted the efficiency of transfer learning. We could see a higher accuracy with a fewer number of runs.

IX. CONCLUSION

By building this classifier, we understood fundamental concepts of deep learning. We also compared the performances of convolutional neural networks that were trained from scratch against using a pre-trained model (Transfer learning). This gave a holistic review about deep learning techniques and its applications in image classification.

REFERENCES

[1] Kaggle DogVCat Competation: http://www.kaggle.com/c/dogs-vs-cats
[2] MSR Asirra: http://research.microsoft.com/en-us/um/redmond/projects/asirra/
[3] https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/
[4] Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Neural Networks. arXiv preprint arXiv:1311.2901.
[5] http://caffe.berkeleyvision.org/tutorial/solver.html
[6] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, Stefan Carlsson; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2014, pp. 806-813.
[7] J. Elson, J. Douceur, J. Howell and J. Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. Proc. of ACM CCS 2007, pp. 366-374.
[8] http://cs231n.github.io/transfer-learning/