

Deep Learning and its application in image processing, Natural Language Processing and Text Mining

BITSZG628T: Dissertation

By

Debanjan Chaudhuri

2014HT12387

Dissertation work carried out at
Atos India Pvt. Ltd., Kolkata



**BIRLA INSTITUTE OF TECHNOLOGY &
SCIENCE PILANI (RAJASTHAN)**

April 2016

Deep Learning and its application in image processing, Natural Language Processing and Text Mining

BITSZG628T: Dissertation

By

Debanjan Chaudhuri

2014HT12387

**Dissertation work carried out at
Atos India Pvt. Ltd., Kolkata**

**Submitted in partial fulfillment of M.Tech. Software Systems
degree programme**

**Under the Supervision of
Mr. Sanjay Singal
Technical Architect, Atos Pune**

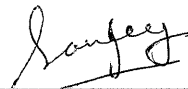


**BIRLA INSTITUTE OF TECHNOLOGY &
SCIENCE PILANI (RAJASTHAN)**

April 2016

CERTIFICATE

This is to certify that the Dissertation Deep Learning and its application in image processing, Natural Language Processing and Text Mining submitted by Debanjan Chaudhuri having ID-No. 2014HT12387 for the partial fulfillment of the requirements of M.Tech. Software Systems degree of BITS, embodies the bonafide work done by him under my supervision.



Signature of the Supervisor

Place: Pune
Date: 04th April 2016

Sanjay Singal
Technical Architect
Atos India Pvt. Ltd
Pune

III

Birla Institute of Technology & Science, Pilani

Work-Integrated Learning Programs Division

Second Semester 2015-2016

BITS ZG628T: Dissertation

ABSTRACT

BITS ID No. : 2014HT12387

NAME OF THE STUDENT: Debanjan Chaudhuri

EMAIL ADDRESS: deba.kgec@gmail.com

**STUDENT'S EMPLOYING:
ORGANIZATION & LOCATION** Atos, Kolkata

SUPERVISOR'S NAME: Sanjay Singal

**SUPERVISOR'S EMPLOYING:
ORGANIZATION & LOCATION** Atos, Pune

**SUPERVISOR'S
EMAIL ADDRESS:** sanjay.singal@atos.net

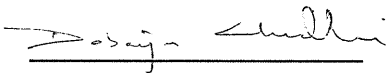
DISSERTATION TITLE: Deep Learning for image processing, Natural Language Processing and Text Mining

Abstract

This dissertation provides a review of all the Deep Learning techniques available for Image processing, Natural Language processing and Text mining and how they can be used in existing applications for state-of-art performances. Also a review of all the problems faced during training this Deep Architectures and the support needed from hardware perspective. We have successfully implemented a convolution neural network model both for image and text classification and it performs far better than traditional machine learning algorithms which are generally used for those tasks.

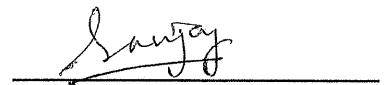
Broad Academic Area of Work: Artificial Intelligence

Keywords: machine learning, deep learning, Neural networks



Signature of the Student

Debanjan Chaudhuri



Signature of the Supervisor

Sanjay Singal

Place: Kolkata

Date: 4th April 2016

Table of Contents

Introduction	1
Understanding Convolution Neural Networks	2
From Neural Nets to Convolution nets	2
Local receptive Fields	2
Shared weights and biases	3
Pooling Layers	4
Emotion Detection using conv nets	4
The problem statement	4
The Data	5
The Model	5
The Application	5
Future Work	6
Vector Representation of Words.....	7
WORD2VEC	7
Domain specific vector representation of words	8
Conv nets for Query Classification.....	9
The Classifier	9
The Data	10
Results	10
Future Work	11
Natural Language Processing	12
Summary	12
Software used for implementations	13
References	13
Checklist of items for final dissertation report.....	14

Background:

The requirement for understanding of natural languages by machines is a very exciting and broad area of Artificial Intelligence. Previously many hand coded rules and statistical methods are implemented for Natural Language processing tasks like Part of Speech Tagging, Chunking, Named entity recognition and many sentiment analysis tasks. With the increase in hardware and computational power of machines we can now train some Deep Models which understand different tasks at a distributed level much like that in the human brain, which understands images/words at different levels of abstractions.

Objectives:

The main objective of our task is to improve human machine interfaces for building a virtual assistant product in retail domain. Humans and machines can interact at different levels, including audio- visual to EEG scans, in our project we have used Deep Learning both for image and text processing, below are a detailed list of tasks performed:

- Emotion Detection: Use a convolution neural network for predicting human emotion through facial expression detection
- Vector representation of words: Train shallow neural network models for understanding semantic relationship between words in our respective domain.
- Text Classification: Question type detection for proper understanding of the user uttered question type and act accordingly.

1. Introduction

Deep Learning is a very exciting and new field in Machine Learning where we train large, deep Neural Network models for different tasks from image processing to speech and language modeling. Deep Learning has proven to beat all the benchmarks on different machine learning tasks over shallow neural networks or shallow SVM models.

Previously there were efforts to train deep neural network models for different speech, image and natural language processing tasks but those attempts have failed for lack of huge processing capabilities and less availability of GPUs. Also there are issues like the vanishing gradient problems. Recently the term Deep Learning again gained traction in mid-2000's Geoffrey Hinton and Salakhutdinov showed how a many-layered feed-forward neural network could be effectively pre-trained one layer at a time treating each layer at a time as an unsupervised RBN's. Since its resurgence Deep Learning has been the state of the art in many disciplines mainly for image recognition/computer vision and Automatic speech recognition.

The main resurgence for Deep Learning is because of the following reasons:

Learning Representations:

For Machine Learning tasks, the inputs to the systems are mainly handcrafted features extracted from the original image/text. Automatic learning of the features is one of the most important aspects of Deep Architectures.

Distributed Representations:

Many NLP models takes in count the count of words or bag of word approach for classification or retrieval, but that will generalize the new test data and the representations are very sparse too. Recently the word2vec approach using a shallow neural network and skip n gram has shown very good results in understanding semantic relationship between words. Deep learning models can be fed as input this vector representation of words for different NLP tasks. Different Deep Learning Algorithms that are state of the art for different Machine Learning tasks are as follows:

- a) Convolution Neural Network
- b) Recurrent Neural Network
- c) Auto Encoders
- d) Deep Belief Networks etc.

2. Understanding Convolution Neural Networks

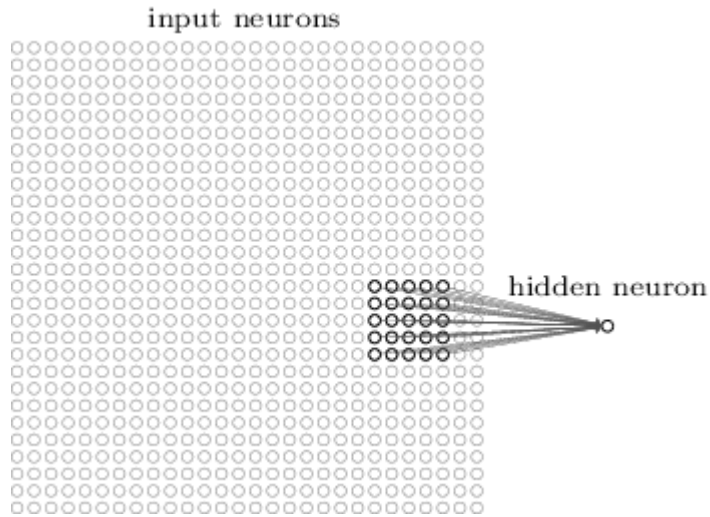
This chapter is focus on understanding convolution neural networks, the architecture and mathematical know-hows.

2.1 From Neural Net to convolution nets:

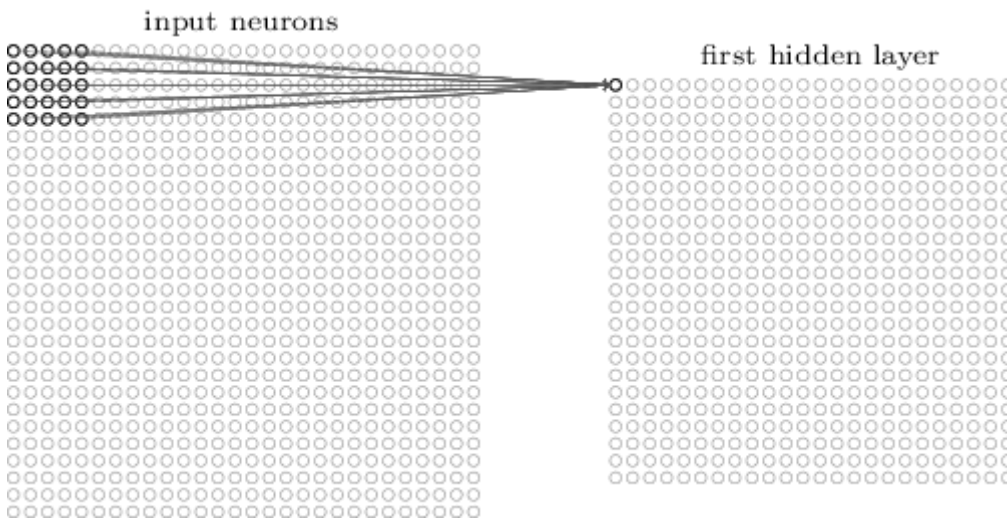
It can be mathematically hypothesized that Neural Networks can approximate any real valued functions and can be trained to understand the relation between the data and the output, but to understand such complex relations shallow neural nets are not sufficient, we have to go deeper. By “Deep” we mean large number of hidden layers at least more than 2. But to train such deep architectures is not feasible because of the vanishing gradient problem. While back propagating the error in order to learn the weights of a neural net the first or early hidden layers learns more slowly than the later hidden layers and henceforth the gradient tends to get smaller as we move backward, this phenomenon is known as vanishing gradient problem, which makes training the deep neural net architectures really hard.

To get rid of this above problem convolution neural networks were implemented which consist of shared weights in the initial hidden layers and unlike MLP's all layers are not connected, which makes the weights easier to train. Neural nets were specially designed for image recognition tasks, where a filter will convolve around the image pixels to understand it and perform certain classification tasks through various layers of extractions. The basic ideas of convolution neural networks are as follows

2.1.1 Local receptive fields: To understand this let us consider an image of dimension 28×28 pixel intensities. In case of traditional fully connected MLP's, for various image classification task, the whole $28 \times 28 = 784$ pixels are fed in as input. But, in case of convolution neural nets we will connect only localized regions of inputs to the next hidden layer, instead only a certain patch of pixels will be connected to the next layer as shown in the picture below:



This region is called the local receptive field. We slide this receptive field through the whole image and feed in to the next hidden layer as shown in the image below:

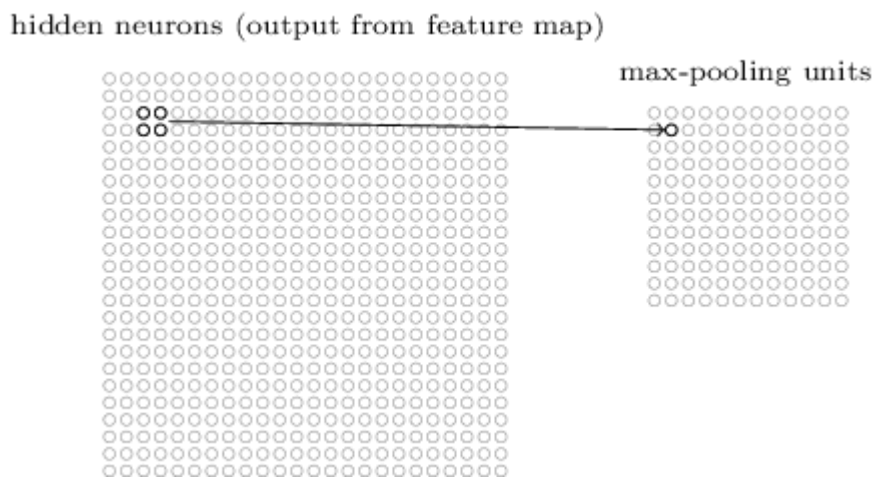


2.1.2 Shared weights and biases: As mentioned before, unlike conventional neural nets, all the weights and biases from the local receptive fields to the hidden layer are same, mathematically the j, k th hidden neuron the output is given by:

$$\sigma \left(b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l,k+m} \right)$$

Where σ is the non-linear activation function, can be sigmoid, tanh or relu, b is the shared bias and w_{lm} is the array of shared weights, which works as a filter. The map from the input feature to the next hidden layer is called a feature map and we can construct several feature maps for a particular task.

2.1.3 Pooling Layers: The feature maps learned from the input by the filter/kernels are simplified by this layer; it directly follows the hidden layer and is used mainly for effective computational purposes. Below is an example of a max pooling layer from a hidden layer:



Here from a 2 x 2 hidden unit the max number is chosen, it is called the max pooling unit. Another variation is the L2 pooling unit, which is the square root of the sum of the unit weights.

3. Emotion Detection using conv nets

An essential feature of the man-machine interface to work well is emotion understandings. Artificial intelligence one of the main objects is that the machine must understand the emotional content of the human in order to have a human like conversation.

For our virtual assistant system we will implement an emotion detection system both from voice and facial expression. In this project I have implemented an emotion detector based on a convolution neural net architecture.

3.1 The problem statement:

Human emotion is a very complex phenomenon of the human psychology, it can change and alter every second depending on surroundings and many other factors. While a machine having a human-like conversation, it must understand the emotion that the human is going through.

Emotions can be predicted both from voice and facial expression, we have used facial expression to detect the emotion of the human, the one it is conversing with

3.2 The Data:

The data consists of a 48 X 48 pixel images which contains different emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

The data is from kaggle competition and an open source data.

3.3 The Model:

We have trained a convolution neural network model on the given data, the layers of the conv nets are:

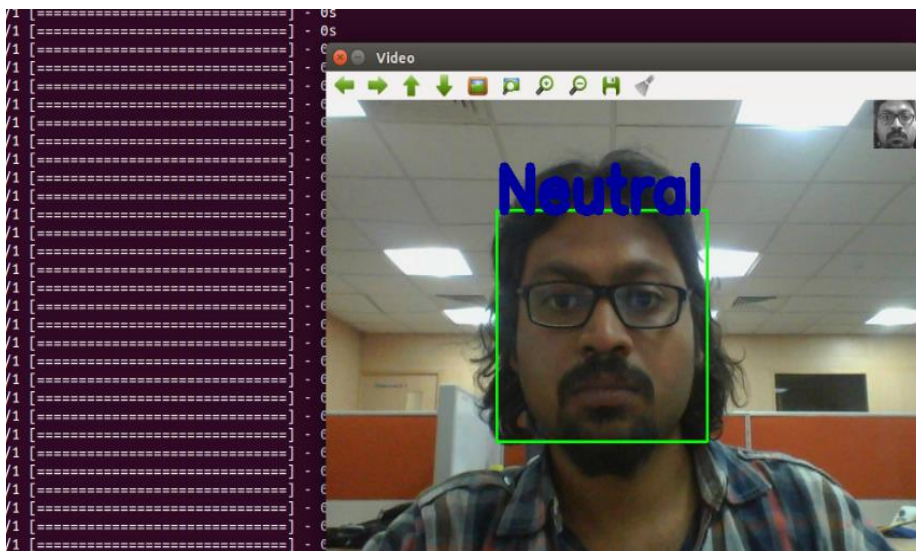
- Convolution 2D, which takes a 48 X 48 input and maps it into 16 feature maps with 3 X 3 filters/kernels
- The layer is followed by a non-linear relu layer
- Followed by a max pooling layer on 2 X 2 feature matrix
- Again followed by a relu layer
- The next layer is a fully connected layer with 128 neurons
- The last layer is a softmax layer with 7 neurons

The network is trained using SGD and nestelrov momentum, on a batch size of 128 and 10 epochs

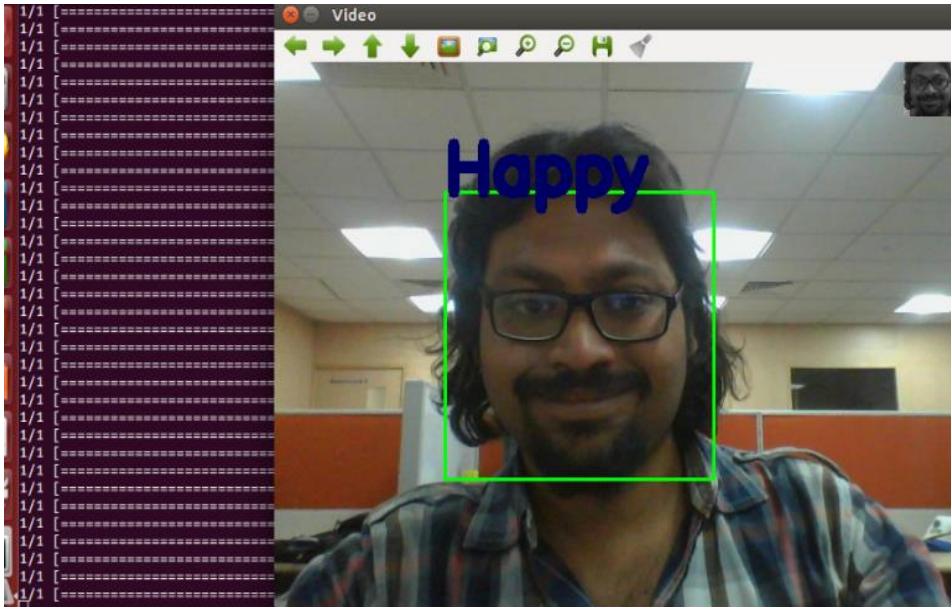
3.4 The Application:

The application is a simple one, where we use open-cv tool kit to capture a 48 X 48 image from the webcam, feed it to the trained conv net classifier and predict the emotion.

Here are few screenshots from the same:



Prediction: Neutral



Prediction: Happy

Tools used: open-cv, keras(python library for deep learning based on top of theano).

3.5 Future work:

A problem with our architecture is our model is overfitting at times, we haven't implemented any L2 regularization or dropouts in the layers, our future work is to avoid overfitting, also the data is only for research purposes, we are gathering our own data to implement in our application, for training the convolution neural net architecture. As this is a prototype model only, we haven't implemented proper evaluation criterions. We will train deeper conv net models and analyze the performances on the emotion recognition task, to implement in our final product.

4. Vector representation of words

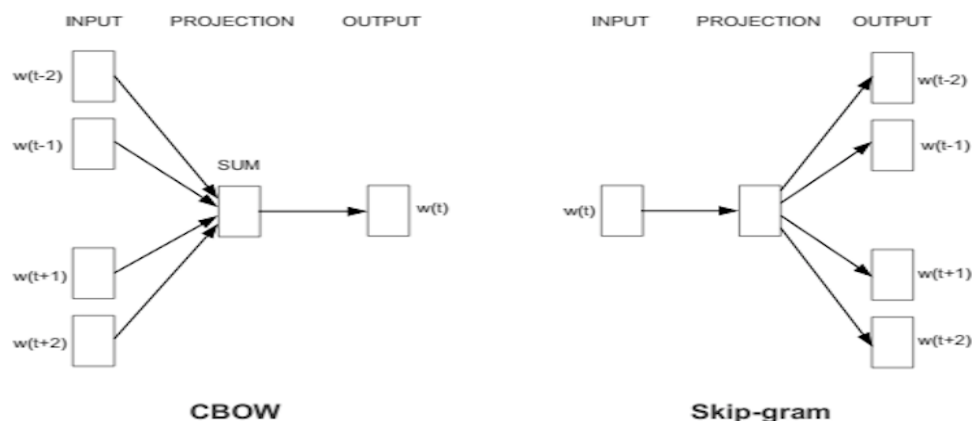
A challenge in language modelling tasks is to find efficient features which will be provided as input to the machine learning algorithms to predict a certain output from it.

This manual feature recognition is generally hand coded and there are no efficient way for choosing an optimum one. Moreover, the words are also fed into the models for classification tasks as a term-document incident matrix, where rows are the documents and columns represent terms. Though many state of the art classifiers use this representation as input but the matrix are very sparse and need compressions using SVD for efficient representations.

In order to get rid of this problem, Mikolov et. al. proposed a shallow neural network architecture to represent words as vectors. The proposed model gave excellent accuracies on understanding semantic relationships between different words. The model is generally known as word2vec

4.1 WORD2VEC

A particular word's meaning depends on the context of the different words that are accompanying and following it, i.e. the context of that particular word. Mikolov et. al. used this concept to train a shallow neural network model, where the input is a context of an word, and the output is a particular word whose representation we are learning. This model is the CBOW model, another very interesting model is the skip-n-gram model, which has the input a particular word, and its output is the context of the same based on a particular window size. The later gave better performance in understanding different semantic relationships between different words. Below is a pictorial representations of the models:



4.2 Domain specific vector representation of words

To understand the semantic relationship between words, we need to train the proposed word2vec models with domain related data. We have done that from retail domain. After training the word2vec models, we found very interesting relationships between different words which are semantically similar

Here are the learned representation of words and their closer meaning groups:

```
In [8]: ModelW2VGSMARena.most_similar('tablets')
```

```
Out[8]:
```

```
[('slates', 0.7113437056541443),
 ('phones', 0.6868905425071716),
 ('smartphones', 0.6756247878074646),
 ('handsets', 0.6535323262214661),
 ('phablets', 0.6455134749412537),
 ('devices', 0.6288411617279053),
 ('smartwatches', 0.5922717452049255),
 ('flagships', 0.5876374244689941),
 ('members', 0.568874180316925),
 ('platforms', 0.5600169897079468)]
```

```
In [9]: ModelW2VGSMARena.most_similar('motorola')
```

```
Out[9]:
```

```
[('google', 0.6531835794448853),
 ('maxx', 0.6033645272254944),
 ('i', 0.5994488596916199),
 ('moto', 0.595329761505127),
 ('razr', 0.5951530933380127),
 ('turbo', 0.5853574275970459),
 ('droid', 0.5852736234664917),
 ('x+1', 0.568969190120697),
 ('360', 0.5533811450004578),
 ('e', 0.5444245338439941)]
```

```
In [11]: ModelW2VGSMARena.most_similar('cpu')
```

```
Out[11]:
```

```
[('processor', 0.938488781452179),
 ('cortex-a53', 0.8411268591880798),
 ('2ghz', 0.8148941397666931),
 ('cores', 0.8137502074241638),
 ('cortex-a7', 0.8126228451728821),
 ('krait', 0.7954966425895691),
 ('gpu', 0.7847505211830139),
 ('3ghz', 0.7836884260177612),
 ('architecture', 0.7830139994621277),
 ('1ghz', 0.7815449237823486)]
```

```
In [12]: ModelW2VGSMARena.most_similar('memory')
```

```
Out[12]:
```

```
[('storage', 0.8858603835105896),
 ('internal', 0.8275272846221924),
 ('built-in', 0.7967686653137207),
 ('8gb', 0.7955197691917419),
 ('expandable', 0.7918469309806824),
 ('onboard', 0.786796510219574),
 ('on-board', 0.7787306308746338),
 ('non-expandable', 0.7786121964454651),
 ('inbuilt', 0.7003437280654907),
 ('128gb', 0.690030038356781)]
```



```

In [13]: ModelW2VGSMarena.most_similar('gsm')
Out[13]:
[('td-scdma', 0.9163823127746582),
 ('3g', 0.8785057663917542),
 ('cdma', 0.8704987168312073),
 ('td-lte', 0.8678370118141174),
 ('wcdma', 0.8640995621681213),
 ('hspa', 0.8529325723648071),
 ('quad-band', 0.8465782999992371),
 ('hspa+', 0.8455212116241455),
 ('2g', 0.8265511989593506),
 ('fdd-lte', 0.8263927698135376)]

```

As seen from the representation, gsm is semantically similar to td-scdma. We will use this learned vectors as inputs to convolution neural net classifier for query classification.

5.Conv nets for query classification

We have seen before that conv nets can be used to classify images with image pixel intensities as inputs. Similarly Yoon Kim has trained convolution models on different query classification tasks and they have performed better than the state-of-art models.

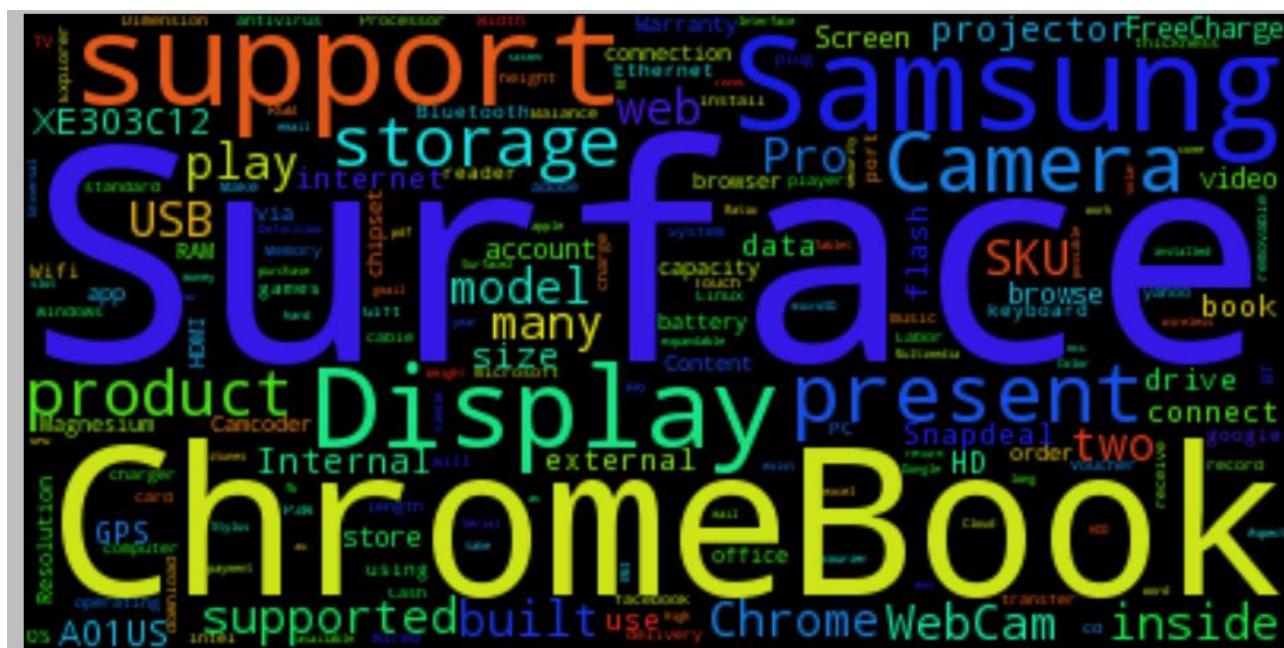
In our virtual assistant application, we have a query classifier which predicts the user query as FAQ, Specification or Affirmative. Based on this, the data is retrieved from the QA system's database using specific ontologies.

5.1 The classifier:

The vector representation of words can have different dimensions, generally from 300 – 500. This is used as input to conv nets model whose input layers are padded words from a sentence, and the next hidden layer or CNN layer has a kernel width of the vector size and varying height as proposed by Yoon Kim. The architecture for sentence classification as shown in the paper is as below:

A lookup table is initially created, which contains the words and their individual vector representation. From that table we form the inputs to the CNN model.

We have used data from a site and manually assigned labels to it, to understand the data the frequent words, below is a word cloud generated from the data:



We have compared the results of query classification into three classes with different traditional machine learning models and CNN classifier. We have also experimented with the inputs to the CNN classifier and compared/contrasted the

results. The training set consists of queries from retail domain with 1800 classified queries. Here are the observed results on 10-fold cross validation sets:

Model	Observed Accuracy on 10-fold CV
Naives Bayes classifier	0.75040000000000
Random Forest(document term matrix with number of terms at each node = 2)	0.82294110000000
Random Forest(document term matrix with number of term at each node = 50)	0.95418840000000
Random Forest(document term matrix with number of term at each node = 98)	0.95308650000000
CNN on domain data and variable kernel size {3, 4, 5}	0.96748218104063
CNN on Google's pre-trained word-vectors and variable kernel size {3, 4, 5}	0.98118147046324
CNN on Google's pre-trained word-vectors and fixed kernel size = 3	0.97842511080080
CNN on randomly Initialized vectors	0.93659776669043

As seen from the results above, CNN on Google's pre-trained vector gave the best performance on our query classification task.

5.4 Future work:

We still haven't deployed the CNN model in our product yet because we are still looking for more labelled data in the domain and we need to be sure that the performance is really good. Currently we are continuing with the random forest model only.

We have also tried to train an LSTM model for this query classification task, but our CPU's doesn't have enough computational efficiency for this task. We will try to train such models on GPU's and compare/contrast the performances with various other models, both considering efficiency and time complexities.

6.Natural Language Processing

The AI singularity is near the horizon; currently researchers are approaching it in bits and pieces. Natural Language processing is one such AI dream in which our machines will be able to communicate with us like humans. Currently we are focusing on small individual tasks to make language understandable to the machines both syntactically and semantically.

The core NLP tasks are:

- a) Part-of-Speech Tagging
- b) Chunking
- c) Named Entity Recognition
- d) Semantic Role Labelling

Collobert et. al. has implemented many neural network architectures for the above tasks which gave near to state of the art performances. We have tried the SENNA model on our product but it has license limitations and also don't do much well on domain specific labels. We will train our own neural network models using word embedding on different natural language processing tasks.

Summary

This dissertation is a review on Deep learning techniques which are currently state-of-art for various tasks, from image recognition, speech recognition, sentiment analysis, topic modelling and various other tasks. We have used convolution neural network for image recognition and query type classification. More complex architectures, like recurrent neural networks and long short-term memory (LSTM) models have also produced great results in query understanding and different machine learning tasks, but due to hardware limitations, we were not able to train any such architectures.

In future we will implement such models which understands the relation between inputs for different language modelling tasks, we will also use some sparse autoencoders for different information retrieval tasks.

Software used for Implementation

- Operating System: Linux (Ubuntu 15.04)
- Languages: Python/Java/Torch
- Library: Gensim/Keras/Theano/open-cv

References

- [1] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa Natural Language Processing (almost) from Scratch
- [2] SiweiLai,LihengXu,KangLiu,JunZhao Recurrent Convolutional Neural Networks for Text Classification.
- [3] Richard Socher, Recursive Deep Learning for Natural Language Processing and Computer Vision.
- [4] Yoon Kim. Convolution Neural Network for Sentence Classification
- [5] Michael Nielsen, Neural Network and Deep Learning book (<http://neuralnetworksanddeeplearning.com/index.html>)
- [6] Deep Learning Net (<http://deeplearning.net/tutorial/>)
- [7] Gensim tutorial (<https://radimrehurek.com/gensim/models/word2vec.html>)
- [8] Deep learning book by Yoshua Bengio and Aaron Courville (<http://www.deeplearningbook.org/>)

Checklist of items for the Final Dissertation Report

This checklist is to be duly completed, verified and signed by the student.

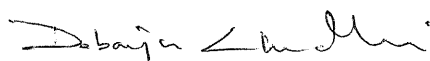
1.	Is the final report neatly formatted with all the elements required for a technical Report?	Yes
2.	Is the Cover page in proper format as given in Annexure A?	Yes
3.	Is the Title page (Inner cover page) in proper format?	Yes
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	Yes Yes
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	Yes Yes
6.	Is the title of your report appropriate? The title should be adequately descriptive, precise and must reflect scope of the actual work done. Uncommon abbreviations / Acronyms should not be used in the title	Yes
7.	Have you included the List of abbreviations / Acronyms?	No
8.	Does the Report contain a summary of the literature survey?	Yes
9.	Does the Table of Contents include page numbers? (i). Are the Pages numbered properly? (Ch. 1 should start on Page # 1) (ii). Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) (iii). Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) (iv). Are the Captions for the Figures and Tables proper? (v). Are the Appendices numbered properly? Are their titles appropriate	Yes Yes Yes Yes Yes Yes
10.	Is the conclusion of the Report based on discussion of the work?	Yes
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	Yes Yes No
12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	Yes

Declaration by Student:

I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.

Place: Kolkata

Date: 4th April, 2016



Signature of the Student

**Debanjan Chaudhuri
ID No.: 2014HT12387**