**Illinois Institute of Technology**

Natural Language Processing with Deep Neural Networks

**Submitted by: Bharath Elangovan**

**CWID: A20344918**

**Date: 12/10/2015**

## Introduction

In Natural Language Processing many challenges involve understanding the language at hand, enabling computers to derive meaning from human or natural language input, and others involve natural language generation. Modern NLP algorithms are based on machine learning, especially statistical machine learning. The paradigm of machine learning is different from that of most prior attempts at language processing. Prior implementations of language-processing tasks typically involved the direct hand coding of large sets of rules. The machine-learning paradigm calls for using general learning algorithms — often, although not always, grounded in statistical inference — to automatically learn such rules through the analysis of large corpora of typical real-world examples. Current NLP systems are incredibly fragile because of their atomic symbol representations. Most current machine learning works well because of representations and input features designed by humans. Representation learning attempts to automatically learn good features or representations.

Handcrafting features is time consuming. The features are often both over specified and incomplete. The work has to be done again for each task/domain. We must move beyond handcrafted features and simple ML Human-developed representations for learning and reasoning. Learning features that are not mutually exclusive can be exponentially more efficient than nearest neighbor like or clustering like models. New methods for unsupervised pre-‐training have been developed and more efficient parameter estimation methods have better understanding of model regularization. With Deep Learning we has achieved better and impressive results we will discuss on performing NLP using DL.

## Deep Learning

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers with complex structures, or otherwise composed of multiple non-linear transformations. Deep learning is part of a broader family of machine learning methods based on learning representations of data. An observation can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc. Some representations make it easier to learn tasks from examples. One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction. Research in this area attempts to make better representations and create models to learn these representations from large-scale unlabeled data. Some of the representations are inspired by advances in neuroscience and are loosely based on interpretation of information processing and communication patterns in a nervous system, such as neural coding which attempts to define a relationship between various stimuli and associated neuronal responses in the brain. Various deep learning architectures such as deep neural networks, convolutional deep neural networks, deep belief networks and recurrent neural networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks.

Neural networks have been used for implementing language models since the early 2000s. Key techniques in this field are negative sampling and word embedding. A word embedding, such as word2vec, can be thought of as a representational layer in a deep learning architecture transforming an atomic word into a positional representation of the word relative to other words in the dataset; the position is represented as a point in a vector space. Using a word embedding as an input layer to a recursive neural network (RNN) allows for the training of the network to parse sentences and phrases using an effective compositional vector grammar. A compositional vector grammar can be thought of as probabilistic context free grammar (PCFG) implemented by a recursive neural network. Recursive auto-encoders built atop word embeddings have been trained to assess sentence similarity and detect paraphrasing. Deep neural architectures have achieved state-of-the-art results in many tasks in natural language processing, such as constituency parsing, sentiment analysis, information retrieval, machine translation, contextual entity linking, and other areas of NLP.

## Shortcomings of NLP with Standard Machine Learning

In NLP and machine learning, we often develop an algorithm and then force the data into a format that is compatible with this algorithm. For instance, a common first step in text classification or clustering is to ignore word order and grammatical structure and represent texts in terms of unordered lists of words, so called bag of words. This leads to obvious problems when trying to understand a sentence. Another common simplification for labeling words with, for example, their part of speech tag is to consider only the previous word’s tag or a fixed sized neighborhood around each word.

While a lot of time is spent on models and inference, a well-known secret is that the performance of most learning systems depends crucially on the feature representations of the input. For instance, instead of relying only on word counts to classify a text, state of the art systems use part-of-speech tags, special labels for each location, person or organization (so called named entities), parse tree features or the relationship of words in a large taxonomy such as WordNet. Each of these features has taken a long time to develop and integrating them for each new task slows down both the development and runtime of the final algorithm.

## NLP with Deep Learning

The majority of deep learning work is focused on pure classification of fixed-sized flat inputs such as images. In contrast, the recursive deep models presented in this thesis can predict an underlying hierarchical structure, learn similarity spaces for linguistic units of any length and classify phrase labels and relations between inputs. This constitutes an important generalization of deep learning to structured prediction and makes these models suitable for natural language processing.

The syntactic rules of natural language are known to be recursive, with noun phrases containing relative clauses that themselves contain noun phrases. This can be done by introducing a max-margin, structure prediction framework based on Recursive Neural Networks (RNNs) for finding hierarchical structure in multiple modalities. Recursion in this case pertains to the idea that the same neural network is applied repeatedly on different components of a sentence. Since this model would show much promise for both language and image understanding, we could investigate the space of recursive deep learning models.

## Deep Neural Networks Language Processing

Deep learning—neural networks that have several stacked layers of neurons, usually accelerated in computation using GPUs—has seen huge success recently in many fields such as computer vision, speech recognition, and natural language processing, beating the previous state-of-the-art results on a variety of tasks and domains such as language modeling, translation, speech recognition, and object recognition in images.

Within neural networks, there are certain kinds of neural networks that are more popular and well-suited than others to a variety of problems. Continuing on the topic of word embeddings, let’s discuss word-level networks, where each word in the sentence is translated into a set of numbers before being fed into the neural network. These numbers change over time while the neural net trains itself, encoding unique properties such as the semantics and contextual information for each word.

Word embeddings are not unique to neural networks; they are common to all word-level neural language models. Embeddings are stored in a simple lookup table (or hash table), that given a word, returns the embedding (which is an array of numbers). Word embeddings are usually initialized to random numbers (and learned during the training phase of the neural network), or initialized from previously trained models over large texts like Wikipedia.

## Convolutional Neural Networks

Convolutional Neural Networks (ConvNets), have enjoyed wide success in the last few years in several domains including images, video, audio and natural language processing.

When applied to images, ConvNets usually take raw image pixels as input, interleaving convolution layers along with pooling layers with non-linear functions in between, followed by fully connected layers. Similarly, for language processing, ConvNets take the outputs of word embeddings as input, and then apply interleaved convolution and pooling operations, followed by fully connected layers. Convolutional Neural Networks—and more generally, feed-forward neural networks—do not traditionally have a notion of time or experience unless you explicitly pass samples from the past as input. After they are trained, given an input, they treat it no differently when shown the input the first time or the 100th time. But to tackle some problems, you need to look at past experiences and give a different answer.

If you send sentences word-by-word into a feed-forward network, asking it to predict the next word, it will do so, but without any notion of the current context. Clearly, without context, you can produce sentences that make no sense. You can have context in feed-forward networks, but it is much more natural to add a recurrent connection.

## Conclusion

A Recurrent neural network has the capability to give itself feedback from past experiences. Apart from all the neurons in the network, it maintains a hidden state that changes as it sees different inputs. This hidden state is analogous to short-term memory. It remembers past experiences and bases its current answer on both the current input as well as past experiences.

## References

<http://devblogs.nvidia.com/parallelforall/understanding-natural-language-deep-neural-networks-using-torch/>

<https://en.wikipedia.org/wiki/Deep_learning>

<http://nlp.stanford.edu/courses/NAACL2013/>

<http://nlp.stanford.edu/~socherr/thesis.pdf>