ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Introduction To Deep Learning Algorithms and Sentiment Analysis

[1]Vishu Madaan, [1]Pawan Verma, [1,2]Prateek Agrawal
[1]Lovely Professional University, Phagwara
[2] University of Klagenfurt, Austria
{vishumadaan123, abes.pawan, prateek061186 }@gmail.com

Deep Learning is a powerful Machine Learning technique that includes multiple layers for representation of data and produces the prediction result. Deep Learning is inspired by structure and function of brain called artificial neural network. In late 1990, only shallow neural network was feasible because deep learning neural network was computationally expensive and complicated. From the last decade, deep leaning has been applied in the field of Computer Vision, Speech Recognition & information retrieval etc. Sentiment Analysis (SA) is one of the major applications of deep learning. In this chapter, we aim to provide the basics of deep learning and some overview about analysis of sentiments using deep learning neural network. From the last two decades, SA has been the interesting research area in the field of Natural Language Processing (NLP). It is also known as opinion mining. It is the process to obtain the overall polarity of the emotion or sentiment. Now-a-days, people are widely using social networking sites to express their feelings or sentiments or emotions towards any entity. It is also helpful to the people to take some decision while choosing those entities. For that purpose, it is required to specify the overall polarity about the entity. SA and classification is done primarily at 3 different levels:

Document Level, Sentence Level and Aspect Level. Earlier, the concept of Bag of Words (BoW) was used at document level sentiment classification but BoW model had some shortcomings. To overcome the limitations of BoW the concept of deep learning was applied along with the BoW model to extract the opinion about any entity. Sentence level sentiment classification aims to extract opinion expressed in a single given sentence. In early research, parse tree was used but lately Convolution Neural Networks (CNN) and Recurrent Neural Network (RNN) techniques were applied as they do not need parse tree to extract the classified word. Aspect level sentiment classification uses the concept of both document and sentence level classification and the target information. Aspect level sentiment analysis has three tasks:

a) To find out the context of target,

b) To generate a target, and

c) To identify the sentiment word for the specified target.

The problem in aspect level sentiment analysis was to group the same aspect expressions into one aspect category. This problem was sorted by applying deep learning algorithm as deep learning is quite good at learning complicated feature representation. In the chapter, we have explained various deep learning techniques/algorithms in sentiment analysis to extract the opinion about any entity.

History:

"Neural" is a predicate of "neuron" and "network" that is represented through a graph like structure. An Artificial Neuron Network (ANN), popularly known as Neural Network, is a computational model based on the structure and functions of biological network of neurons. In the year of 1871 and 1873, Joseph von Gerlach proposed that our nervous system is a single continuous network as opposed to a network of many discrete cells. This theory was known as the reticular theory. At the same time Camillo Golgi discovered that nervous system is just one single cell and not a network of discrete cells. So, he was proponent of reticular theory. Another scientist Santiago Cajal used the same technique which Golgi proposed, and he studied the same tissue and came up with the conclusion that this is not a single cell but is actually a collection of various discrete cells, which together forms a network. This was eventually come to be known as the neuron doctrine. Around in 1891, the term neuron was introduced by Heinrich

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Wilhelm Gottfried von Waldeyer-Hartz. And this author consolidated the Neuron Doctrine. Finally in 1950, electron microscopy finally confirmed that there is a gap between neurons, i.e. it is not a single cell, and it is a network of cells which is known as synapses.

In 1943 McCulloch (neuroscientist) Pitt's (logician) proposed a mathematical model of neuron [1]. These are artificial neurons not biological neurons. Around 1956 in a conference the term Artificial intelligence term was coined.

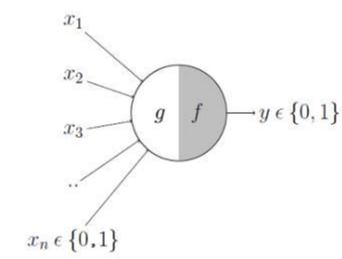

Fig 1: McCulloch-Pitts Neuron

$$g(x_1, x_2, x_3 \ldots x_n) = g(x) = \sum_{i=1}^{n} x_i$$

$$y = f(g(x)) = \begin{cases} 1 \; if \; g(x) \ge \theta \\ 0 \; if \; g(x) < \theta \end{cases}$$

Where θ is called threasholding parameter.

This is called Thresholding logic.

In 1958, Frank Rosenblatt came up with some modified model known as Perceptron model [2].

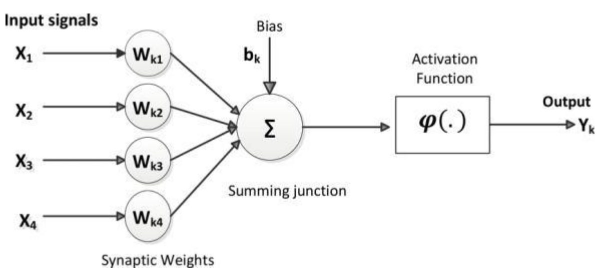

Fig 2: Perceptron model

$$y = \begin{cases} 1 \; if \; \sum_{i=1}^{n} w_i * x_i \ge \theta \\ 0 \; if \; \sum_{i=1}^{n} w_i * x_i < \theta \end{cases}$$

Rewriting the above

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

$$y = \begin{cases} 1 \ \ if \ \sum_{i=1}^{n} w_i * x_i - \theta \geq 0 \\ 0 \ \ if \ \sum_{i=1}^{n} w_i * x_i - \theta < 0 \end{cases}$$

A more accepted convention:

$$y = \begin{cases} 1 \ \ if \ \sum_{i=0}^{n} w_i * x_i \geq 0 \\ 0 \ \ if \ \sum_{i=0}^{n} w_i * x_i < 0 \end{cases}$$

Where $x_0 = 1$ and $w_0 = -\theta$, and w_0 is called the bias.

In this perceptron model, activation function takes the "weighted sum of input plus the bias" as the input to the function and decides whether it should be fired or not. There are different types of activation function like Sigmoid function, Threshold Function, ReLU Function, Hyperbolic Tangent Function etc. For different activation function is used for different situations. The perceptron model may able to learn, make decisions. There is a limitation with single perceptron, it can deal with a function which are linearly separable i.e. it always gives output in the form of zero or one. That's why we use multi-layer network of perceptrons for computation of complex operations.

Introduction to Neural Network:

Neural network is based on this biological model of brain. ANNs are the programs developed to extricate any problem by using the structure and the function of our nervous system. The smallest component of ANN is artificial neuron, sometimes called perceptron. In neural network neurons are connected to each other through various ways to form networks. ANN is defined as an artificial human nervous system that receives, processes, and transmits information in terms of Computer Science. Artificial Neural Network are also described as "neural nets", "artificial neural systems", "parallel distributed processing systems", "connectionist systems". Typically, Neural Network has three layers:

- Input Layer (This layer is used to feed inputs in the model)
- Hidden Layers (Hidden layer processes the information received from the input layers and also there can be more than one hidden layers)
- Output Layer (Output layer shows the information available after processing)

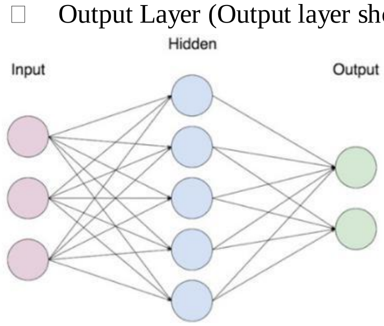

Fig 3: Neural Network

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Input Layer: The Input layer establishes the communication with the external environment that presents a pattern to the neural network. This layer is associated with all the input information only. These inputs are then transferred to the hidden layers, explained below. This layer also represents the condition for which the neural network has to be trained. Every neuron in the input should be assigned some independent variable that affects the results of the neural network

Hidden Layer The hidden layer is a mesial layer found in amidst of the input layer and the output layer. This layer collection of neurons which has activation function applied on it. It processes the information obtained from the previous layer and hence this layer is responsible for extracting the required features from the input data. There are many researches done to enumerate the neurons in the hidden layer but still none of them got success in finding the accurate result. Also there can one or more than one hidden layers in a Neural Network. So you must be thinking that how many hidden layers have to be used for which kind of problem. Suppose we have a data which can be linearly separated, then there will not be any requirement hidden layer that can apply the activation function but to solve the problem the activation function be applied directly into the input layer. But in case of problem which needs complex decisions, we can use 3 to 5 hidden layers based on the degree of complexity of the problem or the degree of accuracy required. It does not mean that if we keep on increasing the number of layers, the neural network will give high accuracy. A stage comes when the accuracy becomes constant or falls if we add an extra layer. Also, the number of neurons should be calculated in each network the number of neuron in the hidden layer is dependent upon the complexity of the problem. If we want to solve complicated problem, we will need the large number of neurons in the hidden layer but if we solve simple computation with large number of neurons in the hidden layer, over fitting problems occur. Till now we don’t have any technique by which we can compute the number of hidden layers and number of neurons in each hidden layer.

Output Layer The output layer of the neural network collects information from hidden layer and then transmits information accordingly for which it has been designed. Output layer gives the output patterns depending upon the input. We can also view the patterns of the activated neurons throughout the input to output layer. We can decide the number of neurons in the output layer on the basis of problem which we want to solve using this network.

ANNs are the simple mathematical models used to enhance existing data analysis technologies. The learning process of a Multi Layer Perceptron is called the Backpropagation algorithm. Backpropagation is a supervised learning algorithm that is used for training the Multi-layer Perceptrons (Artificial Neural Networks). A neural network is a machine learning algorithm based on the model of a human neuron. It is intricate to apply Machine Learning models or algorithms to the complex problems like handwriting recognition, object recognition, NLP (Natural Language Processing), etc that's why Deep Learning algorithms are needed.

Introduction to Deep Learning:

Deep Learning is based on the functioning of a human brain. Deep learning is a large number of artificial neurons connected to each other in layers and functioning towards achieving certain goal. Deep Learning is a subfield of Machine Learning that deals with algorithms inspired by the structure and function of the brain. The primary difference between machine learning and deep learning is that, deep learning uses neural networks and it is suitable for handling large amounts unstructured data and in machine learning the feature extraction is done by data scientists manually but in deep learning feature extraction is automatically using neural networks. The needs of Deep learning are:

1) Machine learning algorithms works with huge amount of structured data but Deep Learning algorithms can work with enormous amount of structured and unstructured data.

2) Machine Learning algorithms cannot deal with complex operations and for performing that there is the needed the Deep Learning algorithms.

3) As the amount of data increases, the performance of Machine Learning algorithms falls down, to make sure the performance of the models remains maintained Deep Learning algorithms are needed.

4) Machine Learning algorithms extract patterns based on labeled sample data, while Deep Learning algorithms are capable of taking large volumes of data as an input, process the input data to extract features out of an object and identify similar objects.

Deep learning is not a new term. In 1965-1968 Ivakhenkon et.al. proposed Multilayer Perceptrons model that is look likes modern deep learning model[3].

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

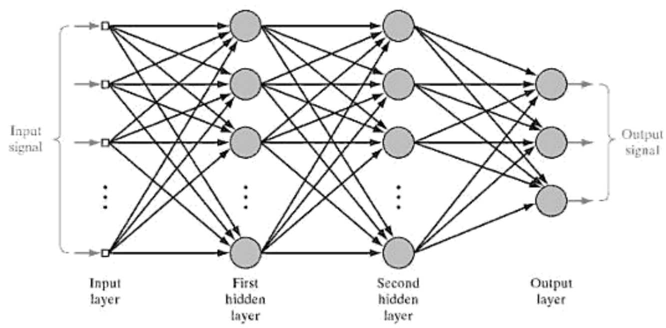

Fig 4: Multilayer Perceptrons model

But 1969 two author named Minsky and Papert defined the limitaion of perceptron. These authors told that Perceptron cannot handle some very simple functions. After the statment of Minsky and Papert people looses the interest in AI. After the long gap of 1969 to 1986, again in 1986 Backpropogation algorithm was popularized by the work of Rumelhart et. al.[4]. This algorithm actually enables to train a deep neural network.

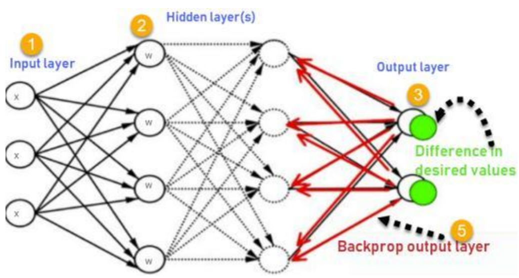

Fig 5: Backpropogation

After the discovery of Backpropogation algorithm, people cannot able to use deep neural networks for solving a problem beacuse of limitation of compute power of system at that time. At that time systems are not able to support deep neural network and we do not have enough amount of data for training deep neural network. Because of these reasons there is still not a real big hype in the field of deep neural network or artificial network and at time again a slump till 2006. In the period from 1989 to 2006, we knew about many algorithms for training deep neural networks and they can potentially be used for solving a wide range of problems but we were not able to use it because of limitation in computational requirements. But from 2006 to 2009 people again started using the deep neural network and deep neural network started becoming popular. From 2009 onwards there was a series of success in research the after using deep neural network. In the year of 2010 the concept of GPU comes in picture, people using GPU for training the deep neural network that drastically reduced the training and inference time. Application of deep learning are Atari Games[5], AlphaGo[6], DeepStack[7], defence of ancients etc. These games are based on deep reinforcement learning based agents have shown a lot of success in beating top professional players. Deep learning also used in other fields like Language modeling[8][9][10], Speech Recognition[11][12][13], machine translation[14][15][16], Conversation Modeling[17][18][19], Question Answering[20][21][22], Object Detection/ Recognition [23][24][25] etc.

Different neural networks are used for some special types of problem, like Feed Forward Neural Network is used in general Regression and Classification problems, Convolutional Neural Network is used for Image Processing, Deep Neural Network is used for acoustic modeling, Deep Belief Network is used for cancer detection, Recurrent Neural Network is used for Speech Recognition.

Feed Forward Neural Network:

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

We have already discussed about MP neurons but it has some problems, it can handle only Boolean input and Boolean output. So we moved to perceptrons which allowed for real inputs and binary outputs. Single perceptrons has some limitations; it can only deal with functions which are linearly separable. So we went for multi layer network of perceptrons. In multilayer perceptrons we need large number or neurons in the hidden layer. Now we discuss first multilayer network i.e. Feed Forward network. The input to the feed forward neural network is an n-dimensional vector, network contains L-1 hidden layer and it has one output layer containing k neurons. All the neurons in the hidden layer are divided into two parts, first part is called pre activation (a_i) and second part is the activation part(h_i). The input layer is also called 0^{th} layer. We have weights between the different layer. The feed forward neural network is show below.

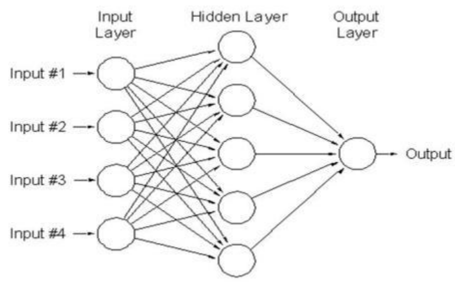

Fig 6: Feed Forward Neural Network

We can give the mathematical formulation for output layer.

- The pre-activation at layer i is given by

$$a_i(x) = b_i + W_i h_{i-1}(x)$$

Where b is the bias.

- The activation at layer i is given by

$$h_i(x) = g(a_i(x))$$

Where g is called the activation function (for example: logistic, tanh, linear etc)

- The activation at layer L is by

$$f(x) = h_L = O(a_L(x))$$

Where O is the output activation function (for example: softmax, linear etc.) We can compute the value of b and W using Gradient Descent Algorithm.

Autoencoder Network:

An autoencoder is a special type of feed forward neural network which can perform encoding and decoding. It encodes input to the hidden representation for that purpose it uses encoded function.

$$h = g(W x_i + b)$$

where g is the sigmoid function or logistic or tanh and so on. After the construction of hidden layer using encoder function the next half of the network work as decoder. And try to reconstruct x again from it.

$$\hat{x} = f(W^* h + c)$$

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

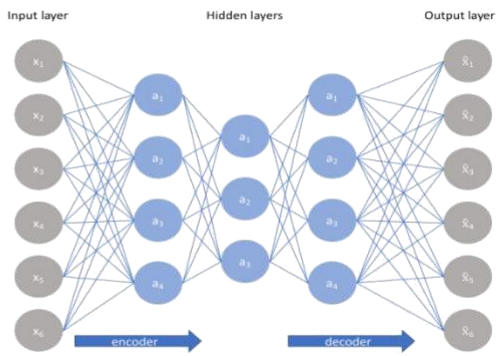

Fig 6: Autoencoder Network

With this network we took the input x_i and compute the hidden representation by hidden layer by doing some non-linear and linear transform and then again trying reconstruct x_i. We are doing this because we want to learn most important characteristics of input data x_i. So, if we compute a hidden representation h, which is presumably smaller than your original input data and from that hidden representation if we are able to reconstruct x_i. Then that would mean that this hidden representation captures everything that is required to reconstruct x_i from the original input. In autoencoder we have two cases:

1. Autoencoder where the dimension of the hidden representation is less than the dimension of your original input is known as an under complete autoencoder.
   $\dim(h) < \dim(x_i)$
2. Autoencoder with more hidden representation of neurons as compared to original input is called over autoencoder.
   $\dim(h) > \dim(x_i)$

Convolutional Neural Network

Convolutional Neural Network (CNN or ConvNet) is a kind of feed forward artificial neural network in which the organization of the animal visual cortex activates the neuron and then decides the connectivity pattern between its neurons. The visual cortex has small region of cells that is sensitive to specific regions of the visual field. In CNN the neurons get activated (gives responses or fires) only in the presence of certain patterned edges. For example, some neurons activates when exposed to the vertical edges and some when activates to the horizontal or diagonal edges. CNN is generally used for image recognition. CNN compares the images piece by piece.CNN contains four layers: Convolution layer, ReLU layer, Pooling Layer and last Fully Connected Layer.

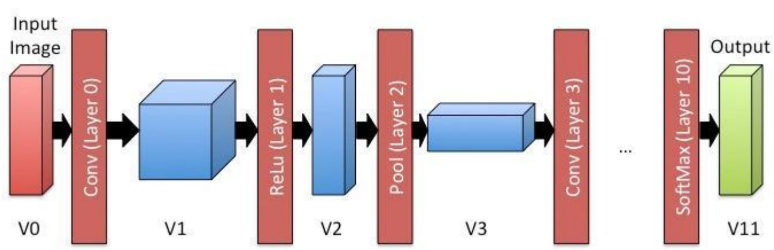

Fig 7: Convolutional Neural Network

In Convolution layer, first we consider different features of the images and one by one we'll take one feature and move filter throughout the entire image and when we are moving it at that time we are multiplying the pixel value of the image with that of the corresponding pixel value of the filter adding them up and dividing by the total number of pixels to get the output. The output of the convolution layer is then transferred to the next layer i.e. ReLU layer.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

ReLU function is an activation function. The output of this function is dependent upon the input. If the input value is less than zero, the function returns zero and it activates the specific node if input is greater the certain value. If the input value is greater than threshold value, it gives the linear relationship for any variable. We are using ReLU function for removing all the negative values from our output that we got through convolution layer. We perform ReLU function for all different features which we get from Convolution Layer. After performing the function for all different features we transfer output to the Pooling Layer. In Pooling Layer we are trying to shrink our image. For that purpose we took the window (usually 2 or 3) and walk that window across the filtered image. For each window, take the maximum value and we take it in a different matrix, so that we get a shrink image. After that we add one more layer of convolution, ReLU and pooling we again shrink the image. After that we use fully connected layer, in fully connected layer actual classification happens. In this layer we take the shrink images and put it into a single list or vector. When we feed in, image, there will be some high element will be there in the vector. After that if we have an input image which has same vector value high as image vector then we can classify that image as same image.

Recurrent Neural Networks

In feedforward network new output has no relation with the previous output or the output at time T is devoid of output at time T-1. We can say that outputs are independent to each other. But in some cases we actually need the previous output to get the new output. For example when we read a book, we will understand that book only on the understanding of previous words. So if we use a feed-forward network and try to predict next word in a sentence we cannot do that because my output will actually depend on the previous output but in the feed-forward network new output is independent of the previous output. So we cannot use feedforward network for predicting the next word in a sentence. We can solve this problem with the help of RNN. Recurrent Network is a type of an artificial neural network contemplated to diagnose patterns in series of data such as the spoken word, text, handwriting, genomes.

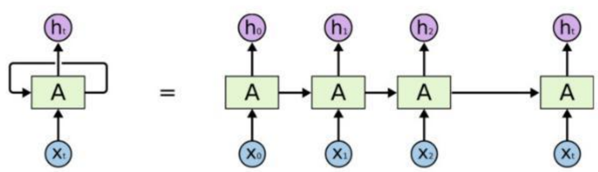

Fig 7: Recurrent Neural Network

In this network we have new information along with the information received from the previous time stamp the output that we have got in the previous time step and use certain information from that will feed into our network as inputs and that will help us to get the new output similarly this new output that we have got will take some information from that feed in as an input to our network along with the new information to get the new prediction and this process keeps on repeating. The mathematical equation of RNN is follows:

$$h^{(t)} = g_h\left(w_1 x^{(t)} + w_2 h^{(t-1)} + b_h\right)$$
$$y^{(t)} = g_y\left(w_y h^{(t)} + b_y\right)$$

RNN uses Backpropagation algorithm, but applied for every time stamp and it is generally known as Backpropagation Through Time (BTT). With Backpropagation there are certain issues namely Vanishing Gradient and Exploding Gradient. Exploding gradients problem can be solved with the help of Truncated BTT or Clip gradients at threshold or RMSprop to adjust learning rate. For vanishing gradients problem we can use ReLU activation function or LSTM and GRUs.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Long Short Term Memory Networks

Long Short Term Memory Networks (LSTMs) are a special kind of RNN and are capable of learning long term dependencies. LSTM also have a chain like structure like RNN. All RNN have the form of a chain of repeating modules of neural network. In standard RNN the repeating modules will have very simple structure. LSTMs have a chain like structure but the repeating module has a different structures. Instead of having a single neural network layer there are four functions interacting in a special way.

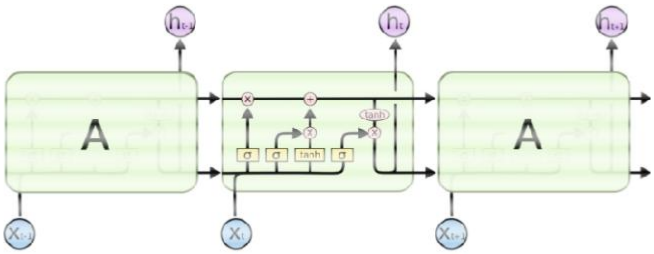

Fig 8: Long Short Term Memory Networks

The processing in the LSTM is as follows:

- The first step involved in the processing of the LSTM is to find out the unrequired information and will be unfettered from the cell state and this decision is taken by a sigmoid layer which is also known as forget gate layer.

  $f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$

  $where\ w_f = Weight$

  $h_{t-1} = Output\ from\ the\ previous\ time\ stamp$

  $x_t = New\ input$

  $b_f = Bias$

- The next step makes the decision of the new information which is going to be stored in the cell state. This whole process comprises of the following steps: A sigmoid layer, also called the "input gate layer", that decides which values will be updated; a tanh layer that creates a vector of new candidate values, which could be added to the state.

  $i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$

  $c_t^{\sim} = \tanh(w_c[h_{t-1}, x_t] + b_c)$

- Now, the old cell state, C_{t-1} will be updated into the new cell state C_t. First, we multiply the old state (C_{t-1}) by f_t, forgetting the things we decided to forget earlier and then add $i_t * \tilde{c}_t$. This is the new candidate value, scaled by how much we decided to update each state value.

  $c_t = f_t * c_{t-1} + i_t * c_t^{\sim}$

- We will execute a sigmoid layer which decides the parts of the cell state that we are going to output. Then, we put the cell state through tanh (push the values to be between -1 and 1) multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

  $o_t = \sigma(w_0[h_{t-1}, x_t] + b_0)$

  $h_t = o_t * \tanh(c_t)$

After processing these above steps we can predict next word in a sentence.

Introduction to TensorFlow

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

TensorFlow is an open software source library developed by the Google Brain Team and it was initially created to perform tasks that require heavy numerical expressions. Due to this reason only TensorFlow was involved with the problems of machine learning and deep neural networks. Since TensorFlow has the C/C++ backend, it runs faster in comparison to the pure Python code. The TensorFlow application uses the data flow graph and it offers several advantages for an application. It provides many API (Application Programming Interfaces) such as Python API and C++ API but the python API is more complete and it's comparatively easy to use. In deep learning we have several libraries but TensorFlow library is most popular because of its better compilation time as compare other library. TensorFlow library can be executed with the help of CPUs, GPUs (graphics process units) and even we can execute with the help of distributed system. With the help of TensorFlow libraries we can executes complex numerical computation using data flow graph. Data flow graph consists of nodes and edges. Node represents the complex mathematical operations and edges represents multidimensional array, these multidimensional arrays are also called tensors. Tensors have the flexible architecture that allows deployment on several CPUs, androids or GPUs and all of these can be done using single API.

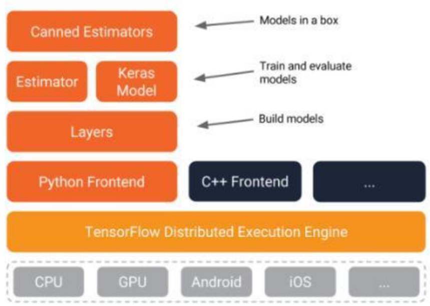

Fig 8: TensorFlow Architecture

TensorFlow has inbuilt support feature, with the help these features we can develop neural network as well as deep learning network. With the help of built-in feature we can train our model with the help of training process. And TensorFlow's auto-differentiation and suite first-rate optimizers benefit any gradient-based machine learning algorithm. With the help of TensorFlow we can implement several machine learning algorithms or models as it has a big collection of flexible tools.

A neural network is collection of artificial neurons that acts as a brain. In neural network we have three layers named as input layer, hidden layer and output layer. Data comes from an input layer as an information and passes through different hidden layers till it reaches to the output layer and output layer gives the final decision. In neural network we can use one or more hidden layers depending upon the complexity of the problem. The basic neural network is also called multi layer perceptron or shallow neural network. We can move from shallow neural network to deep neural network by increasing the number of hidden layers between the input layer and the output layer. Each neuron works as an activation function depending upon the requirement of the model. With the help of TensorFlow we can easily implement or process these activation functions because of its flexible architecture. We can also use TensorFlow in the field of speech recognition, natural language processing NLP (sentiment analysis), video detection, image recognition.

Sentiments Analysis

Internet plays a pivotal role in communication as it contains many micro-blogging websites which provide the facility to share user’s views and thoughts freely all over the world. Micro-blogging websites are platforms where the views are shared in few words only and also provide instant messaging facility. Facebook, Google+ and twitter are some popular micro-blogging websites. Twitter is a most used micro-blogging website throughout the world. Twitter allows the user to update their status as a message of maximum 280 characters named as tweets. Because of

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

this limitation users are using acronyms and emoticons for expressing their views. It is widely used micro-blogging website since it is free from any format and also provides an easy platform to share the ideas, views, thoughts and information. With this reason the twitter data is rapidly increasing. Many users share their personal, religious and political views about any particular topic such as movie, product, item or person. Therefore tweets are important source of sentiment and opinions of the users. We can use this data for decision making or to forecast impact /effect of any particular product or person. For example if a person wants to go for a movie, first checks the reviews of the movie on the micro-blogging website and then decides either to go or not. But micro-blogging website contains huge amount of data for a single topic, therefore we need some automated system to analyze data. For this purpose we have several types of sentiment analysis techniques.

Sentiment analysis is also known as opinion mining. Sentiment analysis is a process that extracts data about any particular topic and defines the overall opinion or feeling whether it has positive influence, negative influence or neutral influence [26]. Sentiment analysis process uses the concept of Natural Language processing and document analysis and computational technique to develop the automated system for extracting and classifying the overall opinion from different resources [27]. This might be helpful for those who are concerned about this opinion. Sentiment analysis process can also be stated as classification process of sentiments. In other words we can say that sentiment analysis is classification process and it is done at three levels

1. Document level [28]
2. Sentence level [29]
3. Aspect level [30][31].

On the basis of sentiment analysis we can generalize the overall impact of a product on our society or reviews of a particular topic/product/item or customer satisfaction level. It also helps in some decision making process such as to elect a member for election or to which location a product should be launch. Sentiment analysis process can be illustrated in figure.

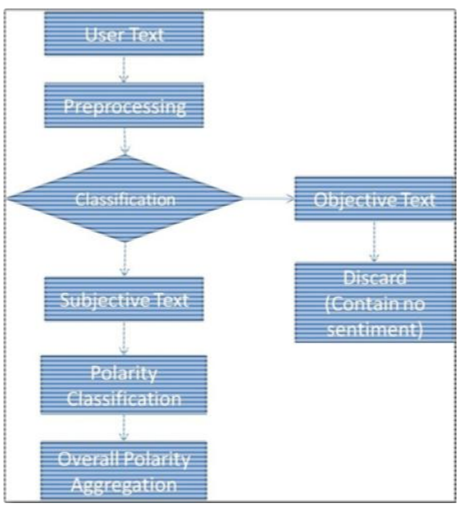

Fig 9: Basic Sentiment analysis process

Following are the steps included in Sentiment Analysis process-

A. Text

Text is defined as a piece of writing that contains someone's views or thoughts or some ideas. It is the data that has been extracted from some websites. Text can be in one or two words or it can be in descriptive format. These text

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

contains simple words but in some cases it also contain some short terms named as internet acronyms and also it can contain some emoticons.

B. Preprocessing

Since the text has been extracted as it is, it is highly required to preprocess the text before classification process. Preprocessing is a process that includes the following tasks.

a. Removal of punctuation marks like '?', '!' etc.
b. Removal of stop words like 'is', 'are', 'am', 'of' etc.
c. Removal of special characters like '$', '&', '@' etc.
d. Checking of spellings and correcting them.
e. Perform stemming i.e. obtain the root word.
f. Removal of question words like 'how', 'who', 'what' etc.
g. POS tagging (also known as Part Of Speech Tagging) defines each word if it is noun, pronoun, adjective, verb or adverb.

C. Preprocessing

Here the classification is the process that defines if the text is subjective or objective. Subjective text is defined as a text that contains some opinion words and objective text is text that does not contain any opinion words. For example-

a. “The story of movie revolves around the lifestyle of that emperor.”

   The above sentence is about the movie but it does not give any sentiment about the movie. Hence it cannot be considered in polarity classification. And it is an objective sentence and also discarded as it does not play any role in sentiment analysis process.

b. “This movie is quite motivational and entertaining.”

   The above example is subjective sentence as it is giving some sentiment about the movie that it is motivational and entertaining.

D. Polarity Classification

After classifying whether the text is subjective or objective, it is required to find out the polarity of anything or person. In this process we classify something or person if it gives positive sentiment or negative sentiment about that. The occurrence frequency of the word in a large text is responsible to identify the polarity [32].

E. Overall polarity aggregation

All the extracted text is now identified if it has a positive polarity or negative polarity. Aggregation means here to find out the overall sentiment about the topic. In the previous step we have found the sentiment of each tweet but in this step we have aggregated all the sentiments to find obtain the overall opinion about the topic.

Sentiment Analysis using Deep learning

The task of sentiment analysis is to train a machine on algorithm to determine the expressed opinion in the document. This can be positive or negative but it could be also some other classes like really positive neutral and really negative. Sentiment analysis is often also called opinion mining or emotion AI. If we talk about emotion AI the classes could be different, it could be things like happy or unhappy. We have two types of approached for sentiment analysis; first one is lexicon based approach and other one is machine learning based approach. In lexicon based approach the idea is that we use some manual defined rules to classify whether a review is positive or negative

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

and in machine learning based approach we want to use some supervised learning algorithm that we can apply after some text preprocessing.

1. Lexicon/Rules Based Approach

   In the lexicon based approach we have the dictionary that has the negative and positive words stored in. In this approach full text is read out where we have to find the sentiment score. This text is then preprocessed i.e. it extracts the keywords that contains some emotions or sentiments. These keywords are then mapped to the dictionary and we find out the total number of positive words and negative words. On the basis of these positive words and negative words sentiment score if obtained using the following formula:

$$sentiment\ score = \frac{\#positive\ words - \#negative\ words}{\#total\ words}$$

$$if\ sentiment\ score > 0\ then\ positive\ class$$

$$if\ sentiment\ score \leq 0\ then\ negative\ class$$

2. Machine learning Approach

   When we discuss about machine learning based approach, we have two phases one is training phase and other one is deployment phase. In training phase, first we read all labeled text and then we extract features from the labeled data. Then after the feature extraction we train a model where we create training and test set so that we can evaluate the model. Once the model works efficiently then we go to deployment phase. In the deployment phase we give a new text or review where the label is unknown, we do the same feature extraction like in the training workflow then we apply the model and then we get some labels. Depending on the model we can use different feature extraction technique. For example, if we use deep learning model then we use different feature extraction technique. When we use sentiment analysis in case of deep learning, RNN and LSTM is used.

In the above approaches, Lexicon approach is easier but machine learning approach is more accurate. For machine learning approach we use deep learning approach because deep neural nets can understand subtleties. This type of nets does not analyze text at face value; it creates abstract representations of what it learned. We can perform following task for the sentiment analysis: Deep learning approach works on word vector that is why we have to convert training words in the word vector using different tools such as Word2Vec. in the next step the ID's matrix is created. After that for processing the data we can use RNN (with LSTM units) to generate the graphs. And in the next step we train and test the model.

Conclusion

As compared to traditional machine learning approaches, deep learning approach gives better result in the field of sentiment analysis because of its multi layer architecture. Because of its better result it becomes a most popular topic for NLP. Since a lot of work is going on in deep learning and continuously the progress is made in deep learning research and application, it has much feasibility in future that deep learning will be applied for better efficiency in sentiment analysis.

References

1. Warren S. McCulloch and Walter H. Pitts, "A logical Calculus of the ideas immanent in nervous activity", published in Bulletin of Mathematical Biophysics, Vol. 5, 1943, p 115-133.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

2. F. Rosenblatt, "The perceptron: A probabilistic model for informarion storage ans organization in the brain", published in Psychological Review, Vol 65, No 6, 1958, p 386-408.
3. Ivakhnenko, A. G. and Lapa, V. G. (1965). Cybernetic Predicting Devices. CCM Information Corporation.
4. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propogation. In D.E. Rumelhart and J.L. McClelland, editors, Parallel Distributed Processing, Vol 1, p 318-362, MIT Press, 1986
5. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg & Demis Hassabis, "Human-level control through deep reinforcement learning", Nature volume 518, pages 529–533 (26 February 2015)
6. Scott R. Granter, Andrew H. Beck, and David J. Papke Jr (2017) AlphaGo, Deep Learning, and the Future of the Human Microscopist. Archives of Pathology & Laboratory Medicine: May 2017, Vol. 141, No. 5, pp. 619-621.
7. Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, Michael Bowling, "DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker", Mar 2017.
8. T. Mikolov, M. Karafi´at, L. Burget, J. Cernock`y, and S. Khudanpur. Recurrent neural network based language model. In INTERSPEECH, pages 1045–1048, 2010.
9. R. Kiros, R. Salakhuditnov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539, 2014.
10. Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-Aware Neural Language Models. CoRR, abs/1508.06615.
11. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing coadaptation of feature detectors. arXiv:1207.0580, 2012.
12. Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 6645–6649. IEEE, 2013.
13. Jan Chorowski , Dzmitry Bahdanau , Dmitriy Serdyuk , Kyunghyun Cho , Yoshua Bengio, Attention-based models for speech recognition, Proceedings of the 28th International Conference on Neural Information Processing Systems, p.577-585, December 07-12, 2015, Montreal, Canada.
14. Kalchbrenner, Nal and Blunsom, Phil. Recurrent continuous translation models. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013.
15. Sutskever, I. Vinyals, O. & Le. Q. V. Sequence to sequence learning with neural networks. In Proc. Advances in Neural Information Processing Systems 27 3104–3112 (2014).
16. Rongxiang Weng, Shujian Huang, Zaixiang Zheng, Xinyu Dai, and Jiajun Chen. Neural Machine Translation with Word Predictions. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pages 136–145, 2017.
17. Shang, L., Lu, Z., and Li, H. Neural responding machine for short-text conversation. In Proceedings of ACL, 2015
18. Iulian Vlad Serban, Alberto Garc´ıaDuran, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. arXiv preprint arXiv:1603.06807, 2016.
19. Chia-Wei Liu, Ryan Lowe, Iulian V Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023, 2016.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

20. Karl Moritz Hermann, Toma´s Ko ˇ cisk ˇ y, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman and Phil Blunsom. 2015. Teaching machines to read and comprehend. arXiv preprint arXiv:1506.03340.
21. Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In ICLR.
22. Bhuwan Dhingra, Kathryn Mazaitis, and William WCohen. 2017. Quasar: Datasets for Question Answering by Search and Reading. arXiv preprint arXiv:1707.03904.
23. Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. CoRR, abs/1411.4038, 2014.
24. J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017.
25. S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers and L. Van Gool, “One-shot video object segmentation,” in CVPR, 2017.
26. Tang, H., Tan, S., Cheng, X.: A survey on sentiment detection of reviews. Expert Systems with Applications 36(7) (2009) 10760-10773.
27. Agarwal, B., Mittal, N., Bansal, P., Garg, S.: Sentiment analysis using commonsense and context information. Computational intelligence and neuroscience 2015 (2015) 30
28. Yessenalina, A., Yue, Y., Cardie, C.: Multi-level structured models for document-level sentiment classification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2010) 1046-1056.
29. Farra, N., Challita, E., Assi, R.A., Hajj, H.: Sentence-level and document-level sentiment mining for arabic texts. In: Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, IEEE (2010)1114-1119.
30. Engonopoulos, N., Lazaridou, A., Paliouras, G., Chandrinos, K.: Els: a word-level method for entity-level sentiment analysis. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics, ACM (2011) 12
31. Zhou, H., Song, F.: Aspect-level sentiment analysis based on a generalized probabilistic topic and syntax model. In: FLAIRS Conference. (2015) 241-245.
32. Read, J., Carroll, J.: Weakly supervised techniques for domain-independent sentiment classification. In: Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, ACM (2009) 45-52.