NEURO-FUZZY TECHNIQUES

SENTIMENT ANALYSIS

TEAM MEMBERS:

KALYANI TAKALKAR

NEHA GADGIL

POOJA BIHANI

SHUBHAM PATIL

Project Mentor : **POONAM SHARMA**

What is sentiment analysis?

Sentiment analysis – otherwise known as opinion mining – is a much bandied about but often misunderstood term.

In essence, it is the process of determining the emotional tone behind a series of words, used to gain an understanding of the the attitudes, opinions and emotions expressed within an online mention.

The process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.

Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation, affective state (that is to say, the emotional state of the author when writing), or the intended emotional communication (that is to say, the emotional effect the author wishes to have on the reader).

Sentiment analysis uses:

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. Social media monitoring tools like Brandwatch Analytics make that process quicker and easier than ever before, thanks to real-time monitoring capabilities.

The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organisations across the world.

Shifts in sentiment on social media have been shown to correlate with shifts in the stock market.

The Obama administration used sentiment analysis to gauge public opinion to policy announcements and campaign messages ahead of 2012 presidential election.The ability to quickly understand consumer attitudes and react accordingly is something that Expedia Canada took advantage of when they noticed that there was a steady increase in negative feedback to the music used in one of their television adverts.

Sentiment analysis conducted by the brand revealed that the music played on the commercial had become incredibly irritating after multiple airings, and consumers were flocking to social media to vent their frustrations.

A couple of weeks after the advert first aired, over half of online conversation about the campaign was negative.

Rather than chalking up the advert as a failure, Expedia was able to address the negative sentiment in a playful and self-knowing way by airing a new version of the advert which featured the offending violin being smashed.

Contextual understanding and tone:

But that is not to say that sentiment analysis is a perfect science at all.

The human language is complex. Teaching a machine to analyse the various grammatical nuances, cultural variations, slang and misspellings that occur in online mentions is a difficult process.

Teaching a machine to understand how context can affect tone is even more difficult.

Humans are fairly intuitive when it comes to interpreting the tone of a piece of writing.

Consider the following sentence: “My flight’s been delayed. Brilliant!”

Most humans would be able to quickly interpret that the person was being sarcastic. We know that for most people having a delayed flight is not a good experience (unless there’s a free bar as recompense involved). By applying this contextual understanding to the sentence, we can easily identify the sentiment as negative.

Without contextual understanding, a machine looking at the sentence above might see the word “brilliant” and categorise it as positive.

What are neural networks?

A neural network is a powerful computational data model that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways:

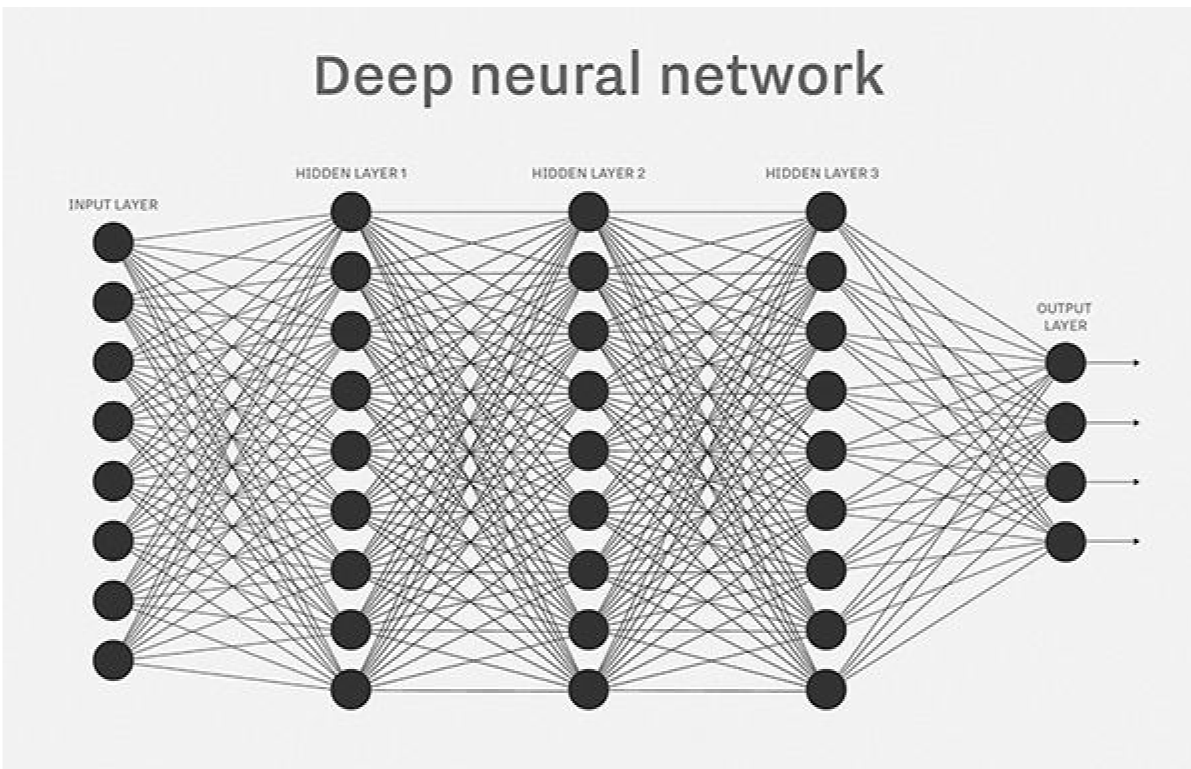
1. A neural network acquires knowledge through learning.
2. A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

A neural network usually involves a large number of processors operating in parallel and arranged in tiers. The first tier receives the raw input information -- analogous to optic nerves in human visual processing. Each successive tier receives the output from the tier preceding it, rather than from the raw input -- in the same way neurons further from the optic nerve receive signals from those closer to it. The last tier produces the output of the system.

Each processing node has its own small sphere of knowledge, including what it has seen and any rules it was originally programmed with or developed for itself. The tiers are highly interconnected, which means each node in tier *n* will be connected to many nodes in tier *n-1* -- its inputs -- and in tier *n+1,* which provides input for those nodes. There may be one or multiple nodes in the output layer, from which the answer it produces can be read.

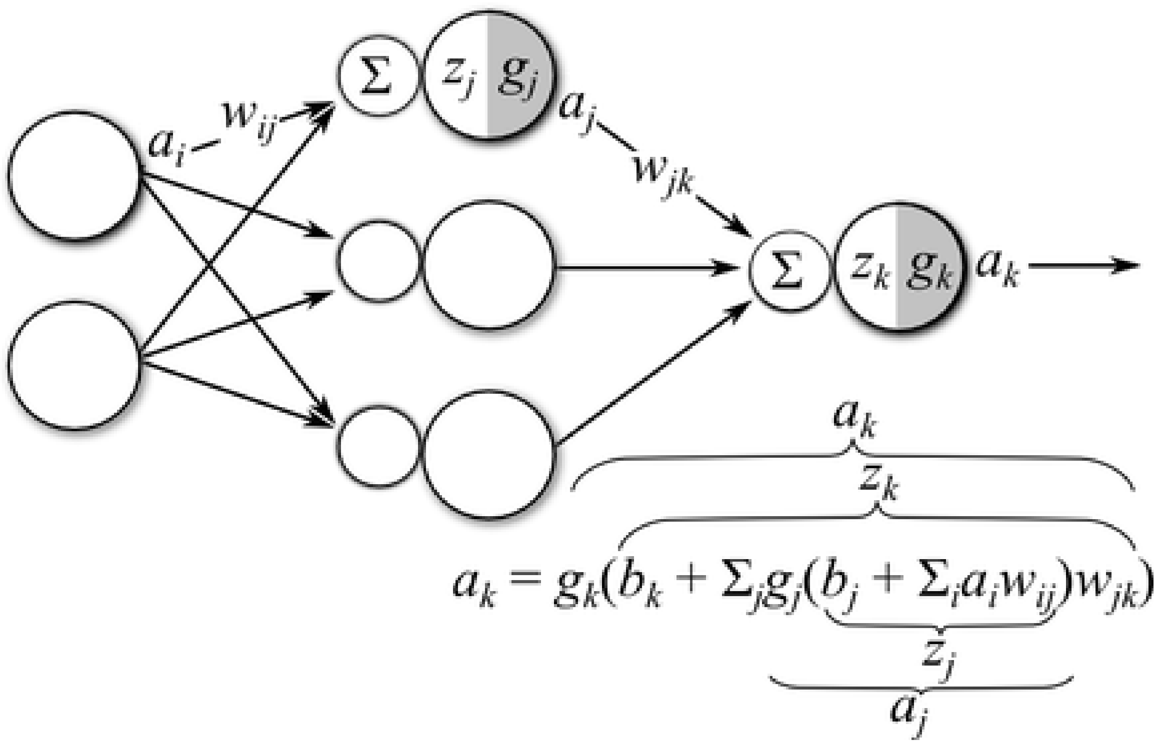
Neural networks are notable for being adaptive, which means they modify themselves as they learn from initial training and subsequent runs provide more information about the world. The most basic learning model is centered on weighting the input streams, which is how each node weights the importance of input from each of its predecessors. Inputs that contribute to getting right answers are weighted higher.

Typically, a neural network is initially trained, or fed large amounts of data. Training consists of providing input and telling the network what the output should be. Providing the answers allows the model to adjust its internal weightings to learn how to do its job better.



Neural networks are sometimes described in terms of their depth, including how many layers they have between input and output, or the model's so-called hidden layers. They can also be described by the number of hidden nodes the model has or in terms of how many inputs and outputs each node has. Variations on the classic neural-network design allow various forms of forward and backward propagation of information among tiers

Error Back propagation :



Motivation:

The goal of any supervised learning algorithm is to find a function that best maps a set of inputs to its correct output. An

example would be a classification task, where the input is an image of an animal, and the correct output would be the name of the animal. The goal and motivation for developing the backpropagation algorithm was to find a way to train a multi-layered neural network such that it can learn the appropriate internal representations to allow it to learn any arbitrary mapping of input to output.

The signals from the input layer a_i are multiplied by a set of fully-connected weights w_{ij} connecting the input layer to the hidden layer. These weighted signals are then summed and combined with a bias b_i (not displayed in the graphical model in Figure 1). This calculation forms the pre-activation signal $z_j = b_j + \sum_i a_i w_{ij}$ for the hidden layer. The pre-activation signal is then transformed by the hidden layer activation function g_j to form the feed-forward activation signals leaving leaving the hidden layer a_j. In a similar fashion, the hidden layer activation signals a_j are multiplied by the weights connecting the hidden layer to the output layer w_{jk}, a bias b_k is added, and the resulting signal is transformed by the output activation function g_k to form the network output a_k. The output is then compared to a desired target t_k and the error between the two is calculated.

Training a neural network involves determining the set of parameters $\theta = \{\mathbf{W}, \mathbf{b}\}$ that minimize the errors that the network makes. Often the choice for the error function is the sum of the squared difference between the target values t_k and the network output a_k

$$E = \frac{1}{2}\sum_{k\in K}(a_k - t_k)^2$$

$$\begin{aligned}
\frac{\partial E}{\partial w_{jk}} &= \frac{1}{2}\sum_{k\in K}(a_k - t_k)^2 \\
&= (a_k - t_k)\frac{\partial}{\partial w_{jk}}(a_k - t_k) \\
\frac{\partial E}{\partial w_{jk}} &= (a_k - t_k)\frac{\partial}{\partial w_{jk}}a_k \\
&= (a_k - t_k)\frac{\partial}{\partial w_{jk}}g_k(z_k) \\
&= (a_k - t_k)g_k'(z_k)\frac{\partial}{\partial w_{jk}}z_k, \\
\frac{\partial E}{\partial w_{jk}} &= (a_k - t_k)g_k'(z_k)a_j
\end{aligned}$$

e define δ_k to be all the terms that involve index k:

$$\delta_k = (a_k - t_k)g_k'(z_k)$$

we obtain the following expression for the derivative of the error with respect to the output weights w_{jk}:

$$\frac{\partial E}{\partial w_{jk}} = \delta_k a_j$$

$$\begin{aligned}
\frac{\partial E}{\partial w_{ij}} &= \frac{1}{2}\sum_{k\in K}(a_k - t_k)^2 \\
&= \sum_{k\in K}(a_k - t_k)\frac{\partial}{\partial w_{ij}}a_k
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial E}{\partial w_{ij}} &= \textstyle\sum_{k\in K}(a_k - t_k)\frac{\partial}{\partial w_{ij}} g_k(z_k) \\
&= \textstyle\sum_{k\in K}(a_k - t_k) g_k'(z_k)\frac{\partial}{\partial w_{ij}} z_k
\end{aligned}
$$

$$
\begin{aligned}
z_k &= b_k + \textstyle\sum_j a_j w_{jk} \\
&= b_k + \textstyle\sum_j g_j(z_j) w_{jk} \\
&= b_k + \textstyle\sum_j g_j(b_i + \sum_i z_i w_{ij}) w_{jk}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial z_k}{\partial w_{ij}} &= \frac{\partial z_k}{\partial a_j}\frac{\partial a_j}{\partial w_{ij}} \\
&= \frac{\partial}{\partial a_j} a_j w_{jk}\frac{\partial a_j}{\partial w_{ij}} \\
&= w_{jk}\frac{\partial a_j}{\partial w_{ij}} \\
&= w_{jk}\frac{\partial g_j(z_j)}{\partial w_{ij}} \\
&= w_{jk} g_j'(z_j)\frac{\partial z_j}{\partial w_{ij}} \\
&= w_{jk} g_j'(z_j)\frac{\partial}{\partial w_{ij}}(b_i + \textstyle\sum_i a_i w_{ij}) \\
&= w_{jk} g_j'(z_j) a_i
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial w_{ij}} &= \textstyle\sum_{k\in K}(a_k - t_k) g_k'(z_k) w_{jk} g_j'(z_j) a_i \\
&= g_j'(z_j) a_i \textstyle\sum_{k\in K}(a_k - t_k) g_k'(z_k) w_{jk} \\
&= a_i g_j'(z_j) \textstyle\sum_{k\in K}\delta_k w_{jk}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial z_k}{\partial b_i} &= w_{jk} g_j'(z_j)\frac{\partial z_j}{\partial b_i} \\
&= w_{jk} g_j'(z_j)\frac{\partial}{\partial b_i}(b_i + \textstyle\sum_i a_i w_{ij}) \\
&= w_{jk} g_j'(z_j)(1),
\end{aligned}
$$

giving

$$
\begin{aligned}
\frac{\partial E}{\partial b_i} &= g_j'(z_j)\textstyle\sum_{k\in K}\delta_k w_{jk} \\
&= \delta_j
\end{aligned}
$$

Algorithm:

1. Calculate the feed-forward signals from the input to the output.
2. Calculate output error E based on the predictions a_k and the target t_k
3. Backpropagate the error signals by weighting it by the weights in previous layers and the gradients of the associated activation functions
4. Calculating the gradients $\frac{\partial E}{\partial \theta}$ for the parameters based on the backpropagated error signal and the feedforward signals from the inputs.
5. Update the parameters using the calculated gradients $\theta \leftarrow \theta - \eta \frac{\partial E}{\partial \theta}$

Sentiment analysis using neural network:

Sentiment analysis involves prediction of the comment type based on outcomes of previously seen data set. This in a way is a feature of supervised learning networks: train neurons on previously seen outputs and predict the outcome of any unseen input.

Thus implementation of sentiment analysis using one such supervised learning algorithm is not something really unprecedented. The Error back propagation proves to be the best supervised training algorithm and hence the presented project attempts to implement the same.

Our Idea:

We have used Error Back Propagation to implement Sentiment Analysis.

We have treated every statement as a set of positive, negative and neutral words. The decision whether the entire statement tends towards positive or negative sentiment is taken based on the frequency of each category of words.

We have trained the network with some predefined statements along with their corresponding sentiment. We have maintained a dataset consisting of list of positive and negative words.

Algorithm:

Pre-processing the statements:

1. Lower Case - Convert the tweets to lower case.

2. URLs - I don't intend to follow the short urls and determine the content of the site, so we can eliminate all of these URLs via regular expression matching or replace with generic word URL.
3. @username - we can eliminate "@username" via regex matching or replace it with generic word AT_USER.
4. #hashtag - hash tags can give us some useful information, so it is useful to replace them with the exact same word without the hash. E.g. #nike replaced with 'nike'.
5. Punctuations and additional white spaces - remove punctuation at the start and ending of the tweets. E.g: ' the day is beautiful! ' replaced with 'the day is beautiful'. It is also helpful to replace multiple whitespaces with a single whitespace

Training the Neural Network:

After preprocessing each statement in the training set, we extract all the useful words from it and remove any punctuation from each word and look into the dataset for positive and negative words. We increment the counts accordingly.

We feed this input along with the already known output to train the Error Back Propagation.

Predicting the result:

Input statement is preprocessed and counts are calculated using the same method and input is fed to the network in order to get the underlying emotion.

Libraries used:

NUMPY:

NumPy (pronounced "Numb Pie" or sometimes "Numb pee" is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

RE:

Re library in Python is used for doing operations on regular expressions. There are some in-built functions that enable us to deal with a sentence in regex approach.

Uses in daily life:

In the context of marketing, sentiment analysis is commonly used for measuring social media performance. There are many tools out there like Radian6 that use a combination of text mining and their own algorithms to identify key indicators in the comments people are saying on social to determine what their sentiment is. The sentiment analysis is then used to determine effectiveness of messaging and how well it was perceived. It can also be used to measure customer satisfaction. For example, Company A may make an update to their product. By measuring reactions through sentiment analysis, the company can determine whether their customers took the update positively or negatively.

Predictions for the future of sentiment analysis:

While it’s difficult to speculate how a relatively immature system might evolve in the the future, there is a general assumption that sentiment analysis needs to move beyond a one-dimensional positive to negative scale.

In the same way that politics cannot always be reduced to a position on a left to right scale, there are other kinds of sentiment that cannot be placed on a simple barometer.

For the future, to truly understand and capture the broad range of emotions that humans express as written word, we need a more sophisticated multidimensional scale.

Can you measure skepticism, hope, anxiety, excitement or lack thereof? Until this happens, sentiment analysis is (literally) one-dimensional!

Organisations will certainly become more aware of the applications of sentiment analysis within their marketplace, fueling the growth of sector specific services and technology delivering sentiment specific use cases – for example, intelligence tools that aid decision-making for financial traders and analysts.

We will see a shift in perception of the reliability of sentiment analysis. Users will become more comfortable with the idea that the automatic analysis of individual text material is hard to match human performance.

The insight that can be gained from large datasets (millions of Tweets) will overshadow the concerns about reliability at a granular level (a single Tweet).

Instead, the focus will be on how to make results interpretable and actionable. In the meantime, we'll be ensuring we are working at making sentiment analysis as accurate and easy to understand as possible.