International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Deep Learning Algorithms

Vidhi Sharma

Institute of Technology, Nirma University, Ahmedabad, Gujarat, India

Abstract: ***The paper discusses about the concepts of artificial intelligence and machine learning. The paper elucidates about the concepts of deep learning algorithms and the era of development in AI which follows deep learning algorithms closely. The algorithms discussed in the paper are Convolutional Neural Network and Autoencoders etc. The paper also includes about the advantages of using different algorithms in different scenes. Also, the paper includes the work already done by different researchers and different breakthroughs.***

Keywords: Artificial Intelligence (AI), Machine Learning (ML), Convolutional Neural Network (CNN), Auto-Encoders (AE)

1. Introduction

The computing has now shifted from hard coded programs to soft computing programs where the computer is only told what to do and how to do. But, the computer was not told about what would be the exact answer to the problem given. It had to predict the answer based on what it had learnt from the previous experiences. This was the time when AI was thought as the new generation of computing.

ML is used now-a-days in almost every field of interest including websites, smartphones, weather forecast, cameras and many more areas.

AI is closely related with the computing done in the human brain and people wanted to provide an artificial brain to computer. The smallest entity of a human brain is a Neuron, which motivated other software programmers to begin duplicate that into a program enabling the computer to have an artificial brain.

The algorithms for this started with simple perceptron model where the user would provide input to the network, which was a connection of neurons as in our brain, and the network would compute the answer based on what it was trained for. The answer generated by the network was then used to verify manually, whether the answer was correct or not. This manual checking lead to the correction of the network to increase the correctness in predicting the answer.

Currently, for not more than a decade, the computing has shifted from simple neural network to Deep Neural Network (DNN), where the connection between the neurons are highly entangled with each other and lead to computation of more complex data as compared to perceptron model. There are various algorithms which provide high computational power and these algorithms support image processing, video processing, speech processing and many more.

Deep Learning has been defined by Hinton and Bengio in their paper on "DEEP LEARNING" as – 'Deep-learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level.'[1]

In this paper, some of the algorithms has been explained and the work done by other researchers have been summarized to get to know which algorithm work best for which type of data.

Section II explains Convolutional Neural Network with an example. Section III covers Autoencoders and there variations along with the main application area.

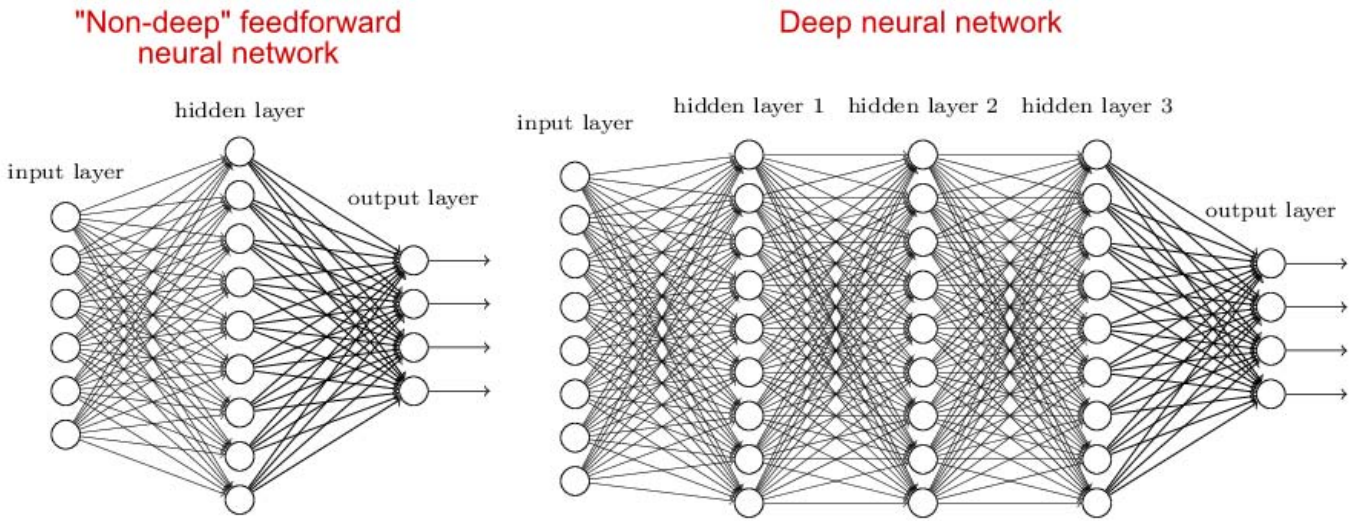

Figure 1: shows the difference between neural network and Deep Neural Network. [1]

2. Convolutional Neural Networks

The CNN is a variation of DNN, which is best suited for complex data such as images, speech, video and many more, where the dimension of data is more than just text or 1s and 0s.

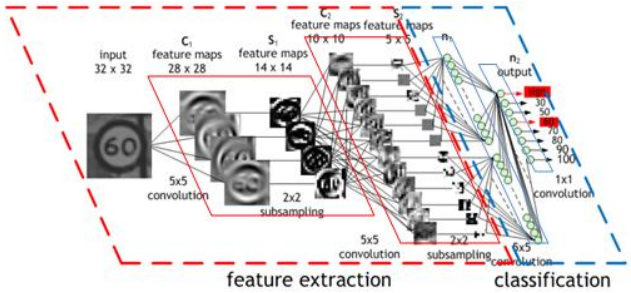

Figure 2: Typical structure of CNN.

The typical structure of CNNs have been depicted in the picture above. The structure is grouped into different layers. The starting layers being the 'convolution layer', where the input is divided into various sub-parts. The convolutional layer divides the images into various sub-parts which overlap partially to replicate the visual cortex feature of the Eye. The overlapping process helps in creating tiles of image, which are then used to dot product with the network's filters. The inputs that give positive answer are taken into account as they provide a valid feature for the next layer. [3]

The neurons of a particular layer are not connected to all the neurons of previous layer as they don't have the knowledge of all the data in the network. They are only locally connected to the previous layer's neurons which are their vicinity. This ensures that a particular feature is enhanced only once and not more times across all the layers.

CNNs exploit the fact that every image is made up high level features which in turn are made up of many low level features and CNNs help to derive those features so that at the end, the output of the network is just to compute the features thus derived from the image to the set of low-level features derived from training.

The next layer is the pooling layer, where the data collected by the convolutional layer is averaged out to be applicable to many set of features. For example, if the network is trained to identify an apple in an image, the apple may appear anywhere in the image and the CNN should be able to correctly identify the same. So, for that pooling layer is essential part of the layer.
The CNNs have been deployed by Microsoft for handwriting recognition. Also, in 1990s the convolutional network was used to identify the human parts of body and later face recognition.

Also, recently there have been great development in CNNs where the information of the image have been coded into the pixel values which could be used to identify what is happening in the image. This approach was complete by Google in 2014 which could correctly identify what was happening in the image.

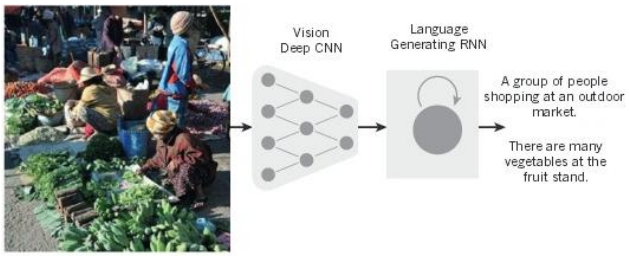

Figure 3: Depicts a CNN can be used to identify what is happening in the image.

The work done by Hinton, Krizhevsky and Sutskever have been summarized below. They implemented an image classification algorithm with CNN where they used 5-convultional layer and 3-pooling layer. The structure is depicted in the picture below. They used 2 GPUs of GTX 580 to get to the final answer as the dataset was large consisting only of 128 examples with momentum of 0.9 and weight decay of 0.0005. This weight decay was not only regularized the weight but also helped in reducing the error rate of the algorithms. The learning steps of the algorithms are depicted in Figure 4. [2]

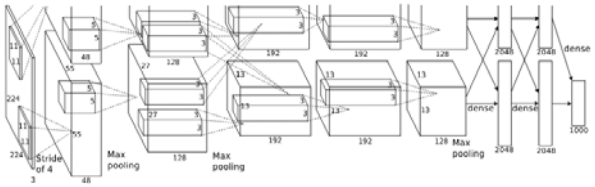

Figure 3: Depicting the structure of network used by Hinton, Krizhevsky and Sutskever in their classification experiment. [2]

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

Figure 4. Depicts the steps of training used in the experiment.[2]

The test set error recieves an error rate of 37.5% and 17.0% in their top-1 and top-5 error rates respectively. This was by far the lowest error rate at the time of the experiment was conducted. Also, the training set was ILSVRC-2010. They also compared the result they obtained using the CNN against other best approaches available at that time. The result showed that by using CNN, they could achieve highest accuracy rate than any other algorithms used on the same data set.[2]

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

Figure 5: The error rates in italics are the best error rates obtained on various models where the dataset was same.[2]

The above experiment proved that the CNNs are the best approach till date for dealing with dataset which involve images.

3. Auto Encoders

Auto encoders are a 3 layer neural network which encodes and decodes the input provided by the user. The output layer has a feedback to the input unit too to ensure multiple encoding of the user input. Also, one property of the network is that the number of hidden units are much less than the actual input and output units. The number of units are less to ensure the compression of the user input. The reason for training is to ensure is that the user data is correctly compressed which can be reconstructed by the output layer into the desired output.

The main purpose of this neural network is for the dimensionality reduction of the dataset provided. Mainly it is done in face recognition application where the different facial expression pose a problem for verifying the image.[5]

There are various variations of auto-encoders. The simple version discussed above does the task of regenerating the same input at the output. The reason for using such a model is to extract latent features at the hidden layer and then reconstruct the same input. So the hidden layer is used to extract the features from the input data. [4][5]

Another version of auto-encoders is the De-Noising Auto Encoder used to extract the data output from a noised input. The input provided is used to predict at the output layer in presence of noise at the hidden layer. DAE learn more efficiently and are more susceptible to noise. The DAEs are used to train using simple back propagation algorithms and variations of it. [4][5]

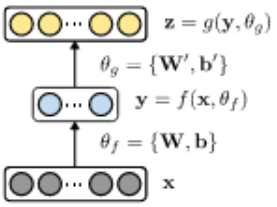

Figure 6: Depicts the structure of an auto-encoder with the function of encoding and decoding.[4]

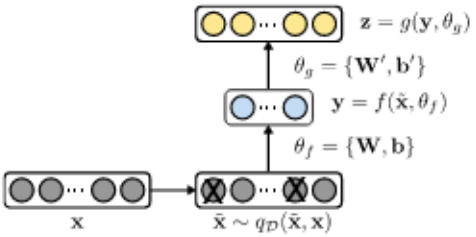

Figure 7: Depicts the structure of a denoising auto-encoder where the Q_D is the unit which induces noise in the input to enable the network to correctly identify the features in case of noise too. [4]

$$\tilde{\boldsymbol{x}} \sim q_{\mathcal{D}}(\tilde{\boldsymbol{x}}|\boldsymbol{x})$$
$$\boldsymbol{y} - f(\tilde{\boldsymbol{x}}, \theta_f) - \sigma(\boldsymbol{W}\tilde{\boldsymbol{x}} + \boldsymbol{b})$$
$$\boldsymbol{z} - g(\boldsymbol{y}, \theta_g) - \boldsymbol{W}'\boldsymbol{y} + \boldsymbol{b}'$$

Figure 8: Depicts the functions of the denoising auto-encoder.[4]

Another variation of auto-encoders is stacked auto-encoders, where the AEs are stacked on top of each other to identify more detailed features for the input provided. Also, the network is trained using noiseless and noisy data to get the correct mapping of data.

The stacked DAEs are explained using a research done by Kang, Lee, Eun, Park and Choi. The experiment involves using non-frontal images to be converted to frontal images for effective face recognition techniques. The paper is titled- 'Stacked Denoising Autoencoders for Face Pose Normalisation'.

The paper describes the experiment is 3 easy steps:
1) Train the network for non-frontal to frontal image normalization.
2) Use the images to train the network in the right direction.
3) Provide test data which would convert image to frontal image and then feed it to face recognition system.

The reason for using SDAEs is because it provides acceptable result in spite of the type of noise provided in the network. Therefore, the input can be corrupted with more complex noisy data as compared to simple linear noises. The non-frontal images are considered as the noise for the frontal face dataset. So, the work of the network is to convert non-frontal to frontal face image. The non-frontal images were provided in the input layer along with some Gaussian noise.
The dataset used had 50 subjects with 15 different poses of their faces, where 1 pose was of the frontal face. The dataset had only 750 images which were transformed to 26550 images by adding some other types of noises like rotating the images, flipping the image horizontally, etc.[4]

Figure 8: Depicts the sample of the dataset provided for the SDAEs in the experiment.[4]

The dataset was partitioned into 0.8 as training set and 0.2 as test data. Also, to analyse the performance of the algorithm of face recognition, they used 0.8 of the subjects for the training dataset and rest of the subjects for test data.

The architecture of the network was 1024-2000-1000-2000-1024, which shows that it had 3 SDAEs stacked on each other to give 3 hidden layers.

After running the training phase, the weights captured the rotation of the faces in the images and made them as the features for identifying the frontal or non-frontal faces.[4]

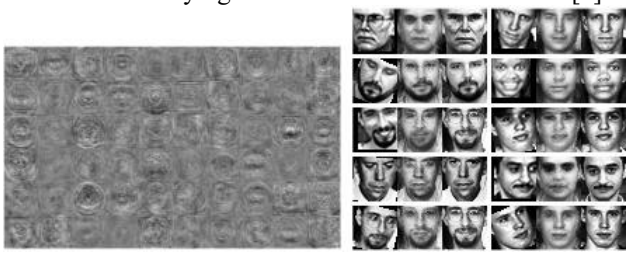

Figure 9: Depicts two images, (Left) features captured by the SDAEs weights, (Right) sample of the dataset.[4]

The results obtained by converting the image to frontal from non-frontal images were blurry but had shown a dramatic improvement as compared to the SVMs used in the experiment for the purpose of comparison. The accuracy was about 84.3% as compared to 10.8% with SVMs. [4]

The above results prove that SDAEs are better at face recognition and transformation algorithm than SVMs or any other algorithm which involves face recognition.

4. Conclusion

The deep learning algorithms are the new era approach towards approaching a problem via computationally. The CNNs are best suited for image processing and generating text out of images. These are by far the most efficient algorithm in this scenario and has resulted in highest efficiency till date.
The Autoencoders are used for face recognition approaches and deals with it in the most effective way. They give a high accuracy rate as compared to conventional SVMs for face recognition.

References

[1] Review Paper on 'Deep Learning' by Yann LeCun, Yoshua Bengio and Geoffrey Hinton. (347 citations)
[2] Geoffrey Hinton, Ilya Sutskever, Alex Krizhevsky, 'ImageNet Classification with Deep Convolutional Networks.' (4543 citations)
[3] https://stats.stackexchange.com/questions/114385/what-is-the-difference-between-convolutional-neural-networks-restricted-boltzma
[4] Yoonseop Kang, Kang-Tae Lee, Jihyun Eun, Sung Eun Park and Seungjin Choi, 'Stacked Denoising Autoencoders for Face Pose Normalization.'
[5] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio and Pierre-Antoine Manzagol, 'Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.' (801 citations)