Big Data and Deep Learning

Jyothi Pranavi Devineni
Indiana University Bloomington
Bloomington, Indiana
jyodevin@umail.iu.edu

ABSTRACT

Big Data is providing new opportunities for various industries in different sectors to enhance their performance by performing analysis on the huge amounts of data available. However, this is not as easy as it is said. Storing, transforming and performing analysis on such large amounts of data requires good storing and computational power. Many solutions have been proposed to handle big data and use it to the benefit of the company like Map Reduce, Spark and so on. Deep learning is one of the popular branches of machine learning which plays a key role when it comes to big data analytics.

KEYWORDS

Deep Learning, Big Data, Deep Belief Network, Convolutional Neural Network

1 INTRODUCTION

Big data is the latest hot topic in the technical world and so is deep learning. Any data which consumes more than 1 Terra Bytes of memory is considered as big data. Social Media websites like Facebook generate more than 500 Terra Bytes No conventional data base cannot store or manage more than 1 Terra Bytes of data. Hence, new technologies like Hadoop, Spark and so on have emerged to store and process large amounts of data. Hadoop used HDFS for storing the data and Map Reduce for processing the data. Scripting languages like Pig, Hive and Spark can also be used to process the data but, Map Reduce is a better option for processing unstructured data.

Deep Learning is another hot topic which is being discussed almost everywhere. Deep learning is one of the branches of machine learning, which uses machine learning techniques to solve the problems of data analysis and prediction. It does not follow any pre-defined algorithms, rather learns from the data. The learning can be supervised or unsupervised. Deep learning is used along with the systems with high computational power to address the big data problems. Many companies like Facebook, Apple, Google, Samsung are using the deep learning techniques to manage the huge amounts of data that is being generated daily by their search engines and websites. Not only this, deep learnig is also used in speech recognition, image processing, weather forecasting and so on. Hence, the voice assistants like Siri, Google home, Alexa make use of deep learning as well. As the data keeps getting huge, deep learning comes into play to process the data.

2 DEEP LEARNING

Deep learning learns multiple levels of the deep architectures such as Deep Belief Networks(DBN), Convolutional Neural Networks(CNN) and so on. In this paper, a brief overview of DBN and CNN is given.

2.1 Deep Belief Networks

Any conventional neural network can only learn from the labelled data. But, most of the big data available is unlabelled. To take advantage of this massive amounts of unlabelled data, deep belief networks are used. They can not only learn from the labelled data, but also from the unlabelled data. They use both supervised and unsupervised learning techniques. It uses unsupervised techniques for pre- training and then to tune the data, it uses supervised techniques. Figure1 shows the architecture of DBN.

To achieve this, DBNs use Restricted Boltzmann Machines(RBMs). RBM consists of input layer, hidden layer and an output layer. Nodes in each layer are connected to all the nodes in the adjacent layer(input to hidden) and nodes in same layer are not connected to each other. Hence, we can say that nodes in same layer are independent of each other. The nodes in hidden layer are connected to the nodes in output layer according to the output to be generated. The network is pre-trained layer by layer using unlabelled data and the generative weights of each RBM are found using Gibbs sampling[7]. The output of an RBM is fed as input to the RBM in the next layer. This process is repeated until all the RBMs in a network are pre-trained. The weights represent the input data. Then, the output layer is constructed according to the required outputs. Then, fine tuning is performed using labels and by back propagation. RBMs can be trained on unlabelled and large amounts of data. The sampling probabilities of hidden and visible layers of an RBM with bernoulli distribution are as follows:

$$p\left(h_j = 1 \mid \mathbf{v}; W\right) = \sigma\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right) \tag{1}$$

$$p\left(v_i = 1 \mid \mathbf{h}; W\right) = \sigma\left(\sum_{j=1}^{J} w_{ij} h_j + b_i\right) \tag{2}$$

The weights are updated using the following equation. The $(t+1)^{th}$ weight is updated as:

$$\Delta w_{ij}(t+1) = c\Delta w_{ij}(t) + \alpha\left(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{model}\right) \tag{3}$$

2.2 Convolutional Neural Networks(CNN)

CNN is another multi-layer neural network which is used for deep learning. CNN consists of multiple layers of convolution, activation and pooling. Figure2 depicts the architecture of a CNN

Convolution is a mathematical operation on two functions. It is defined by the following equation:

$$s(t) = x(t) * w(t) = \sum_{a=-\infty}^{\infty} x(a) w(t-a) \tag{4}$$

Convolution is used to extract the features from the input given to CNN, hence, it is called feature map stage. The output of convolution is called a feature map. The size of the feature map depends on three parameters:

(1) Depth
(2) Stride
(3) Zero-Padding

Depth is the number of filters used for convolution. Stride is the number of pixels used to slide a filter across the input during convolution and appending zeros for the sake of convolution is called zero-padding.

After the convolution operation, activation function is applied to the output, to introduce non-linearity. Any non-linear activation function can be used for this purpose.

After applying the activation function, the output is passed through pooling stage where different pooling operations such as average, sum, minimum, maximum, etc are applied. A window is defined and pooling is done in that window. The window slides over the input of the pooling stage by stride amount as discussed earlier.

After all these layers, a final layer is added using MLP for classifying the data or performing the regression, as per the required output. The steps followed in a CNN are, the filters and weights are randomly initialized first and then the data is passed through all the stages of CNN to get some output, then compare the actual and desired output and perform back propagation to update the weights accordingly.

3 DEEP LEARNING FOR BIG DATA

Deep Learning is very useful for prediction, especially when it comes to unlabelled data. It has proved to be efficient in many applications. But, when it comes to big data, deep learning algorithms are not very efficient, as training the nodes requires iterative computing of the weights which is very difficult when the data is huge. Hence, parallel algorithms must be used for training deep architectures, when dealing with big data.

In 2012, Deng et al.[6] proposed the concept of Deep Stacking Network(DSN) for parallel processing in deep architectures. Also there are other methods to address the big data problems, like improving the computation power, parallely computing the weights of hidden and visible layers, by distributing them across different machines and then integrating. It is nothing but a multi-node cluster in Hadoop. Each system is considered as a slave node computing the weights of a part of the hidden and visible neurons.

In addition to these techniques, systems with high computing power are used. One such example is GPU. Figure3 shows the architecture of a GPU:

The above GPU consists of four multi processors(MP), each MP consists of multiple streaming processors(SM) and each SM consists of multiple stream processors(SP). The stream processors share a common control logic and memory. The GPU also has a global memory. The architecture shown in the figure is a Single Instruction Multiple Thread architecture(SIMT). Such architecture is used when multiple computations are to be performed with less access time to the memory. The global memory in GPU is also a high-latency memory with high bandwidth. Here, the host represents the CPU. This architecture supports two levels of parallelism, namely memory level(MP) and thread level(SP). It facilitates multi-threading by running many hundreds of thousands of threads at a time.

3.1 Deep Belief Networks for Big Data

In deep learning architectures, millions of free parameters are considered to reduce the risk of over-fitting, in contrast to conventional architectures. For example Hinton and Salakhutdinov[8] have used 3.8 million parameters for images and Ranzato and Szummer[1] used three million parameters. But, the model proposed by Raina et al.[2] is far better than these models. Raina's model uses hundred million parameters for parallelizing the learning models which learn from unlabelled data, like DBNs.

Using GPU for parallelizing the DBN is not enough. Because, a considerable amount of time is wasted in transfering the data between the host and the global memories. Hence, to overcome this, a part of the training samples and the parameters are stored in the global memory itself while training. Also updating the parameters is done in GPU. In addition to memory and thread processing, data processing is also facilitated.

In DBNs, the weights are generated using Gibbs sampling using the same equations as for a non-parallel DBN, by generating sampling matrices P(x/h) and P(h/x) where the $(i, j)^{th}$ element is P(x_j/h_i) in P(x/h) and p(h_j/x_i) in P(h/x). Then, GPU is used to implement these two matrices. The weights are also updated in parallel using GPU.

3.2 CNN for Big Data

CNNs use GPUs for parallel processing to deal with big data. Both forward and backward propagation are used in training a CNN. Hence, both the propagations should be parallelized. To parallelize the forward propagation, each feature map in a CNN is assigned with some memory blocks, based on the size of the feature map and every thread in a block corresponds to a single neuron in a feature map. The CNN computations for each neuron in a map, such as convolution, applying activation function and pooling are performed in SP and the outputs are stored in the global memory.

In CNN, the weights are updated by back propagating the error. Back propagation can be parallelized by pushing or pulling the error signals. Although using GPUs facilitates parallel processing of data, it is only possible to process limited number of feature maps at any given time. For this purpose, Scherer et al[3] proposed an efficient method to use a circular buffer, which holds a small part of each feature map, loaded from global memory. Then, the threads parallely perform the convolution and the results are written back to the global memory. Krizhevsky et al.[4] proposed another yet faster method for processing big data using CNNs, by using two GPUs. Also, the speed of operation of CNN can be improved by using a ReLU or Rectified Linear Units activation function instead of any other activation functions.

4 DEEP LEARNING FOR HIGH VOLUMES, VARIETY AND VELOCITY OF DATA

The major concerns when dealing with big data are the volume of the data, variety of the data and velocity of the data. When dealing with large volumes of data, it is very difficult to train a

deep learning algorithm using a single storage and CPU. Hence, distributed processing is preferred, which makes use of the multi-node cluster environment as in Hadoop. In such environment, the data and processing is distributed among different systems or nodes in the cluster for parallel processing and the outputs are again integrated at the master node.

Also, there are three types of data to be handled, structured, semi-structured and unstructured. Whatever the form the data might come in, it has to be stored and processed. There is not much difficulty in storing and processing structured data, but it is the semi-structured and un-structured data one faces a problem with. Also there is high velocity of data generated in many online site like the social media and so on, which needs to be accounted for in a timely manner. Deep learning can handle data of different varieties and with high velocity by using domain adaptation as discussed by Xue-Wen Chen and Xiaotong Lin.[5]

5 CONCLUSION

This paper discusses two of the available deep learning architectures and how they are used to to address the big data problems. Deep learning has proved to be useful whenever one encounters big data. Deep learning architectures can be used along with systems which have high computation power and by performing parallel processing.

ACKNOWLEDGMENTS

The authors would like to thank Professor Gregor Von Laszewski and all the associate instructors of the course I-523 for guiding us through.

REFERENCES

[1] 2008. *Semi-supervised learning of compact document representations with deep networks.*
[2] 2009. *Large-scale deep unsupervised learning using graphics processors.*
[3] 2010. *Evaluation of pooling operations in convolutional architectures for object recognition.*
[4] 2012. *ImageNet classification with deep convolutional neural networks.*
[5] Xue-Wen Chen and Xiaotong Lin. 2014. Big Data Deep Learning: Challenges and Perspectives. *IEEE* (2014).
[6] L. Deng and J. Platt D. Yu. 2012. Scalable stacking and learning for building deep architectures. *IEEE* (2012).
[7] G. Hinton and Y. Teh S. Osindero. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18 (2006), 7.
[8] G. Hinton and R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313 (2006), 5786.

[Figure 1 about here.]

[Figure 2 about here.]

[Figure 3 about here.]

6 BIBTEX ISSUES

Warning–no key, editor or organization in Ranzato2008

Warning–to sort, need editor, organization, or key in Ranzato2008

Warning–no key, editor or organization in 2009

Warning–to sort, need editor, organization, or key in 2009

Warning–no key, editor or organization in 2010

Warning–to sort, need editor, organization, or key in 2010

Warning–no key, editor or organization in 2012

Warning–to sort, need editor, organization, or key in 2012

Warning–no key, editor or organization in 2009

Warning–no key, editor or organization in 2009

Warning–no key, editor or organization in 2010

Warning–no key, editor or organization in 2010

Warning–no key, editor or organization in 2012

Warning–no key, editor or organization in 2012

Warning–no key, editor or organization in Ranzato2008

Warning–no key, editor or organization in Ranzato2008

Warning–no key, editor or organization in Ranzato2008

Warning–no key, editor or organization in 2009

Warning–no key, editor or organization in 2010

Warning–no key, editor or organization in 2012

Warning–no number and no volume in Chen2014

Warning–page numbers missing in both pages and numpages fields in Chen2014

Warning–no number and no volume in Deng2012

Warning–page numbers missing in both pages and numpages fields in Deng2012

(There were 24 warnings)

7 ISSUES

DONE:

Example of done item: Once you fix an item, change TODO to DONE

7.1 Formatting

Incorrect number of keywords or HID and i523 not included in the keywords

7.2 Writing Errors

Do not use the phrase *In this paper/report we show* instead use *We show.* It is not important if this is a paper or a report and does not need to be mentioned

7.3 Citation Issues and Plagiarism

It is your responsibility to make sure no plagiarism occurs. The instructions and resources were given in the class

Most of the sentences and paragraphs do not have any reference

Need to quote directly cited material. Are you sure you have quoted all of them?

The citation mark should be in the end of the sentence, before the period mark. example: ... a library called Message Passing Interface(MPI) [7].

Put a space between the citation mark and the previous word

7.4 Character Errors

Erroneous use of quotation marks, i.e. use “quotes” , instead of ” ”

To emphasize a word, use *emphasize* and not “quote”

When using the characters & # % _ put a backslash before them so that they show up correctly

Pasting and copying from the Web often results in non-ASCII characters to be used in your text, please remove them and replace accordingly. This is the case for quotes, dashes and all the other special characters.

If you see a ffigure and not a figure in text you copied from a text that has the fi combined as a single character

7.5 Details about the Figures and Tables

If you are using an image from another reference, the reference should be cited. Otherwise it is correct

Remove any figure that is not referred to explicitly in the text (As shown in Figure ..) When used in the text, they should be referred to the image correctly using the ref key

Figures should be reasonably sized and often you just need to add columnwidth

e.g.

```
/includegraphics[width=\columnwidth]{images/myimage.pdf}
```

re

List of Figures

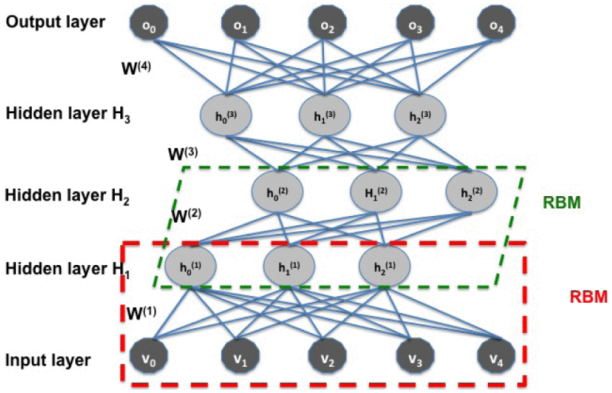

Figure 1: DBN

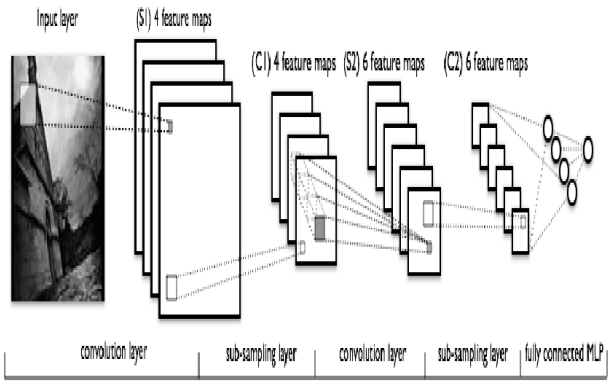

Figure 2: CNN

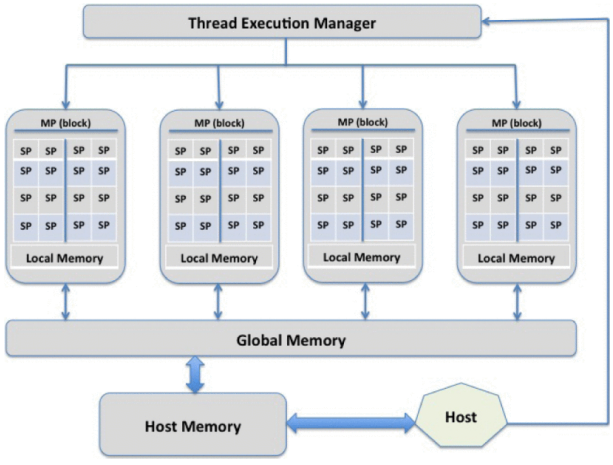

Figure 3: GPU