Journal of The Gujarat Research Society

ISSN: 0374-8588
Volume 22 Issue 1, January 2020

Deep Learning Research: An Overview

Vijay Maheshwari, Mr. Kuldeep Chauhan
Shobhit Institute of Engineering and Technology (Deemed to be University), Meerut
Email Id- vijay@shobhituniversity.ac.in, kuldeep.chauhan@shobhituniversity.ac.in

ABSTRACT: ***Deep learning technology has been a significant study area in the field of machine learning with the advent of big data, and it has been extensively used in image processing, natural language processing, voice recognition, and online advertising, among other applications. This article covers many elements of deep learning techniques, such as typical deep learning models and optimization approaches, widely used open-source frameworks, current issues, and future research prospects. First, we'll go over some of the most common deep learning models and optimization techniques; second, we'll go over some of the most common deep learning models and optimization methods; and third, we'll go over some of the most common deep learning models and optimization methods. Finally, we go through several popular deep learning frameworks and platforms. Finally, we discuss deep learning's most recent acceleration technologies as well as its future research.***

KEYWORDS: ***Applications, Computer Vision, Deep Learning, Developments, Trends***

INTRODUCTION

The rise of different applications in the Internet area has led to the exponential expansion of data size, thanks to the fast development of technologies such as cloud computing, Big data, and Internet of things. Global total data is projected to be 22 times 2011 by 2020, according to a study published by the International Data Corporation (IDC) in 2012. Big data has a lot of worth and a lot of promise; it will change and improve human civilization, but it will also cause significant issues with overload[1]. The ability to rapidly and effectively extract useful information from a variety of complicated data sets has become a major problem. Deep learning has produced significant advances in image processing, natural language comprehension, and voice recognition in recent years. Given the fast advancement of deep learning technology, deep learning can map diverse data to the same hidden space by conducting automated feature learning from multi-source heterogeneous data and obtaining a unified data representation. This article examines deep learning approaches from a variety of perspectives, including popular deep learning models and optimization methods, frequently used frameworks, current issues, and future research objectives[2].

Deep learning's first use was image recognition. For image recognition, deep convolutional neural networks are used to learn the end-to-end mapping connection between low-resolution and high-resolution pictures. The greatest results were obtained at the time using a neural network to identify handwritten digital. Using a deep convolutional neural network, the R-CNN object detection technique is faster. Image recognition using deep convolutional neural networks. Images are classified using an auto encoder and trained support vector machines[3][4]. In the 2016 Image Net Competition, deep learning accuracy surpassed 97 percent. CNNH (Convolutional Neural Network Hashing) is a supervised depth hashing method for image recognition. Deep learning technology has been used in voice recognition in recent years. Accuracy of Chinese voice recognition based on deep learning had surpassed 97 percent[5]. Microsoft's research on deep neural network-based voice recognition has totally altered the basic technological foundation of speech recognition. The deep neural network model has resulted in significant improvements in speech recognition accuracy. Deep neural network models are being utilized in voice recognition algorithms used by well-known Internet businesses. In, they used a convolutional neural network (CNN) to extract voice features.

Speech recognition using a multilayer perceptron-based speech synthesis model[6]. The LSTM technique for extracting speech characteristics, which significantly increases feature efficiency. The error rate on the TIMIT core test set was reduced to 20.7 percent using the Gaussian mixture model (GMM) in the conventional model with DBN, which was a substantial improvement.

Recently, Google developed a speech recognition system based on the feed forward sequential memory network, which employs a high number of convolutional layers to directly model the whole phrase speech signal and better convey long-term speech relevance. Baidu used deep convolutional neural networks in voice recognition research, and the recognition error rate was significantly reduced by combining visual geometry group networks with deep convolutional neural networks. Deep learning is also used for natural language processing[7]. K. A recurrent neural network (RNN)-based vector constant length representation paradigm for machine translation. In natural language processing, artificial neural networks have gotten a lot of attention. The bilingual assessment understudy rating method was used to evaluate similar models in statistical machine translation jobs. For standard natural language processing problems like semantic role labeling, embedding and multi-layered one-dimensional convolutional architectures are used.

The performance of the neural network model may be enhanced by adding additional recursive layers, according to the researchers. To map words into a vector representation space using embedding techniques, and then represent the language model using nonlinear neural networks. a Machine Translation RNN search model. Auto encoder is a backpropagation-based unsupervised learning method that sets the target values to be identical to the inputs[7]. The idea of auto encoder (AE) was first introduced in 1986, and it was first utilized for high-dimensional complicated data processing. By rebuilding the input data to create the output data, an auto encoder may extract the hidden feature. The fundamental construction of an auto encoder is a three-layer neural network with input layer x, hidden layer h, and output layer y, where the output layer and input layer have the same size. The input and output layers of auto encoders contain the same neurons, while the intermediate layer has more than the input layer. The output layer reconstructs the input data by training the network and ensuring that the input and output data are as comparable as feasible. The similarity is represented by the training error. Some researchers suggested a sparse Auto encoder, in which the average value of the output signal is penalized by a l2 penalty or encouraged by a tiny mean value that is approximated by a Gaussian distribution. A coarse-to-fine auto encoder was trained to locate important spots on the face in the paper[8]. De noising auto encoder, which introduced artificial noise to the training samples to enable the network to reconstruct the original clean input from the noisy signals, was used to improve the generalization of auto encoder. To produce a compact picture high-level description and image retrieval, the deep auto encoder is employed. The deep auto encoder has been successfully used to picture feature representation by a number of researchers. To create a deep neural network, auto encoders may reconstruct the input data by training the network, adjusting the parameters, and cascading. The Boltzmann Machine is a kind of feedback neural network that is made up of random neural networks. The Boltzmann Machine is made up of visible unit's visible variables, i.e. data samples and hidden units (hidden variables), with each visible unit connected to all hidden units[9].

The visible variables and hidden variables are binary variables with states of 0 or 1, with 0 indicating that the neuron is suppressed and 1 indicating that the neuron is active. Boltzmann machine with restrictions proposed (RBM). The visible layer receives the training data, while

the hidden layer identifies the characteristics of the input data. The neurons are disconnected within the same layer, but completely linked between the two layers. Restricted Boltzmann machine training is quicker than Auto encoder training. Based on the stochastic gradient descent method, a more efficient optimization algorithm has been developed. RBM's conventional training technique requires a large number of sample steps, resulting in a low training efficiency. Hinton's suggested contrastive divergence addressed the issue[10].

DISCUSSION

A deep deconvolution network that learns the hierarchical structural features from the bottom layer to the top layer directly from the global picture by concatenating several convolutional sparse Auto encoder and maximum pooling layers. Many expansion models based on limited Boltzmann machines have been proposed by certain researchers. Discriminative learning should be integrated into RBM's generative learning algorithm so that it can be used more effectively for discriminative tasks like categorization. The suggested deep Boltzmann machine is immediately cascaded into a multi-layer structure. For learning the latent characteristics of picture pixel blocks, a deep sparse Auto encoder model is used. A restricted Boltzmann machine may be cascaded to create a deep neural network, which can then be optimized using the layer-by-layer training technique. The model may learn possible feature representations straight from the original 2D picture using a deep belief network with convolution operations. There are various hierarchical generation models than the RBM-based deep structure. To create a deep belief network, a multi-layered directed sigmoid belief network was cascaded using RBM. The Restricted Boltzmann machine, which has a Gaussian kernel, accepts continuous variables as input signals.

By changing the structure of the RBM or probability distribution, the restricted Boltzmann machine may be expanded to tackle increasingly complicated problems. In these models, a more complicated energy function is typically specified, which reduces the efficiency of learning and inference. Max pooling has the disadvantage of being susceptible to overfitting the training set, making it difficult to generalize to test data. To address this issue, a stochastic pooling method was developed, which substitutes traditional deterministic pooling procedures with a stochastic procedure based on a multinomial distribution for activation inside each pooling area. It's similar to traditional maximum pooling, but with multiple copies of the input picture, each with tiny local deformations. This stochastic character aids in avoiding the issue of overfitting. CNN-based techniques often need a fixed-size input picture. This limitation may lower the recognition accuracy for pictures of any size. To get around this restriction, final pooling layer was replaced with a spatial pyramid pooling layer in the standard CNN design. The spatial pyramid pooling technique can extract fixed-length representations from any pictures (or areas), resulting in a versatile approach for dealing with various scales, sizes, and aspect ratios, and it may be used in any CNN structure to improve its performance. Handling deformation is a major issue in computer vision, particularly when it comes to object recognition. Max and average pooling are helpful for dealing with deformation, but they can't learn the deformation constraint or the geometric model of object components.

A novel deformation constrained pooling layer, termed pooling layer, was created to enhance the deep model by learning the deformation of visual patterns in order to cope with deformation more efficiently. At any level of information abstraction, it may take the place of the conventional max-pooling layer. Fully-connected layers mimic the behavior of a conventional neural network and include about 90% of the parameters found in a CNN. It allows us to feed forward the neural network into a pre-defined length vector. We could either feed the vector

forward into a set of number categories for picture classification, or we could use it as a feature vector for further processing. The transferred learning method, which maintained the parameters learnt by ImageNet but replaced the final fully-connected layer with two new fully-connected layers to adapt to the new visual identification tasks, is an example of changing the structure of the fully-connected layer. The disadvantage of these layers is that they have a lot of parameters, which means that training them requires a lot of computing power. As a result, removing these layers or reducing connections using a specific technique is a promising and widely used strategy. Switching from fully linked to sparsely connected topologies, for example, may create a deep and broad network while keeping the computational budget unchanged. Deep learning has the benefit of being able to construct deep architectures to learn more abstract knowledge when compared to shallow learning. The high number of factors added, however, may lead to another issue: overfitting. Several regularization techniques, like the stochastic pooling method described above, have recently developed in defense against overfitting. In this part, we'll go over a few more regularization methods that may have an impact on training results. When a CNN is used to recognize visual objects, data augmentation is often used to produce more data without incurring additional labeling expenses. The well-known used two different types of data augmentation: the first kind involves creating picture translations and horizontal reflections, while the second involves changing the intensities of the RGB channels in training images. Alex Net was used as the basic model, and additional transformations were applied to enhance translation and color invariance by expanding picture cropping with more pixels and adding additional color manipulations. Some of the more recent research used this data augmentation technique extensively.

It initially randomly selected a collection of picture patches and declared each of them as a surrogate class, then extended these classes by adding transformations corresponding to translation, scale, color, and contrast. Finally, a CNN was taught to distinguish between these surrogate classes. The network's learned features performed well on a range of categorization tests. Aside from the traditional methods of scaling, rotating, and cropping, color casting, visualization, and lens distortion techniques were also used, resulting in more training examples with broader coverage. Fine-tuning is an important step in refining models so that they can adapt to different tasks and datasets. In general, fine-tuning necessitates the creation of class labels for the fresh training dataset, which are then used to calculate the loss functions. Except for the final output layer, which is dependent on the number of class labels in the new dataset and will therefore be randomly initialized, all layers of the new model will be initialized based on the pre-trained model, such as Alex Net. However, obtaining the class labels for any new dataset may be challenging in certain cases.

To solve this issue, a similarity learning objective function was suggested to be utilized as the loss functions in the absence of class labels, allowing back-propagation to operate normally and the model to be improved layer by layer. There are also many study findings explaining how to effectively transfer the pre-trained model. A new method is described for determining how generic or specialized a layer is, namely how effectively characteristics at that layer move from one job to another. After fine-tuning to a fresh dataset, they discovered that initializing a network using transferred characteristics from nearly any number of layers may improve generalization performance. There are additional popular regularization techniques, such as weight decay, weight tying, and many more, in addition to the ones mentioned above. Weight decay works by penalizing the parameters by adding an additional term to the cost function, preventing them from perfectly replicating the training data and therefore aiding generalization to new instances. By decreasing the number of parameters in a Convolutional Neural Network,

weight tying enables models to develop effective representations of the input data. Another deep learning method, the Deep Boltzmann Machine (DBM), uses layers to organize the units. The DBM has connections throughout its structure, unlike DBNs, which have an undirected graphical model at the top and a directed generative model at the bottom.

The DBM is a subset of the Boltzmann family, much as the RBM. The DBM differs in that it has several levels of concealed units, with units in odd-numbered layers conditionally independent of units in even-numbered layers and vice versa. Because of the interactions between the hidden units, computing the posterior distribution over the hidden units is no longer tractable given the visible units. Instead of directly maximizing the likelihood, a DBM uses a stochastic maximum likelihood (SML) based algorithm to maximize the lower bound on the likelihood, i.e., performing only one or a few updates using a Markov chain Monte Carlo (MCMC) method between each parameter update when training the network. When pre-training the DBM network, a greedy layer-wise training strategy is applied to the layers, much like the DBN, to prevent ending up in bad local minima that leave many hidden units essentially dead.

This combined learning has resulted in promising increases in the deep feature learner's probability and classification performance. However, a significant drawback of DBMs is that approximate inference requires much more time than DBNs, making simultaneous optimization of DBM parameters unfeasible for big datasets. To improve the efficiency of DBMs, some researchers developed an approximation inference method that uses a separate "recognition" model to establish the values of latent variables in all layers, significantly speeding up the inference process. There is also a slew of additional methods aimed at improving the efficacy of DBMs. Improvements may occur during the pre-training period or during the training session. The centering technique, for example, improved the stability of a DBM while also making it more discriminative and generative. The DBM was jointly trained using the multi-prediction training strategy, which surpasses the prior techniques in image classification presented in. To simulate the energy landscape, the model employs deep feed forward neural networks, which can train all layers at the same time. It was shown that combined training of several layers provides qualitative and quantitative gains over greedy layer-wise training by assessing the performance on real pictures.

A sparse auto encoder is a program that extracts sparse features from raw data. The representation's sparsity may be obtained either by punishing hidden unit biases or by penalizing the output of hidden unit activations directly. The projected gradient method, which renormalizes each column of the weight matrix immediately after each update of the conventional Gradient descent process, is a widely used technique for updating the weights. For the sparsity penalty to have any impact, it must be normalized. Iterative projections often result in delayed convergence. A significantly more efficient approach than gradient-based methods is the Lagrange dual method. The article also suggested a feature-sign search method to learn the sparse representation given a dictionary. The combination of these two algorithms resulted in substantially improved performance over the prior ones. However, it is unable of handling extremely large training sets or dynamically changing training data. As a result, during each iteration, it automatically retrieves the whole training set. To solve this problem, an online method for learning dictionaries was suggested, in which one element or a tiny subset of the training set is processed at a time. The dictionary is then updated via block-coordinate de-scent with warm restarts, which eliminates the need for learning rate adjustment. We will discuss several well-known algorithms related to sparse coding, in particular those that are

employed in computer vision problems, in this subsection, as we have quickly indicated how to create the sparse representation given the objective function.

CONCLUSION

This article provides a thorough overview of deep learning and proposes a classification system for analyzing the current literature on the subject. Convolutional Neural Networks, Restricted Boltzmann Machines, Auto encoder, and Sparse Coding are the four categories in which deep learning algorithms are classified based on the fundamental model from which they are formed. The four courses' state-of-the-art methods are discussed and evaluated in depth. The article focuses on advances in CNN-based methods for applications in the computer vision sector, since they are the most widely used and best suited for images. Most notably, several recent papers have claimed breakthroughs demonstrating that certain CNN-based algorithms have already outperformed human raters in terms of accuracy. Despite the encouraging outcomes thus far, there is still a lot of potential for improvement. For instance, the theoretical basis does not yet explain under what circumstances they would perform well or outperform other methods, or how to select the best structure for a given job. This article discusses these issues and highlights recent developments in the design and training of deep neural networks, as well as various avenues that may be pursued in the future.

REFERENCES

[1] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, “Joint learning of words and meaning representations for open-text semantic parsing,” 2012.

[2] N. Ref, “XXX-bad document,” *Comput. Educ.*, 2011.

[3] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*. 2018, doi: 10.1016/j.compag.2018.02.016.

[4] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *J. Big Data*, 2015, doi: 10.1186/s40537-014-0007-7.

[5] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: Review, opportunities and challenges,” *Brief. Bioinform.*, 2017, doi: 10.1093/bib/bbx044.

[6] D. Ravi *et al.*, “Deep Learning for Health Informatics,” *IEEE J. Biomed. Heal. Informatics*, 2017, doi: 10.1109/JBHI.2016.2636665.

[7] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*. 2015, doi: 10.1038/nature14539.

[8] D. Shen, G. Wu, and H. Il Suk, “Deep Learning in Medical Image Analysis,” *Annu. Rev. Biomed. Eng.*, 2017, doi: 10.1146/annurev-bioeng-071516-044442.

[9] C. Cao *et al.*, “Deep Learning and Its Applications in Biomedicine,” *Genomics, Proteomics and Bioinformatics*. 2018, doi: 10.1016/j.gpb.2017.07.003.

[10] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, “Deep learning for smart manufacturing: Methods and applications,” *J. Manuf. Syst.*, 2018, doi: 10.1016/j.jmsy.2018.01.003.