Information Fusion 42 (2018) 146–157

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

A survey on deep learning for big data

Qingchen Zhang[a,b], Laurence T. Yang[*,a,b], Zhikui Chen[c], Peng Li[c]

[a] School of Electronic Engineering, University of Electronic Science and Technology of China China
[b] Department of Computer Science, St. Francis Xavier University,Antigonish, Canada
[c] School of Software Technology, Dalian University of Technology, Dalian,China

ARTICLE INFO

Keywords:
Deep learning
Big data
Stacked auto-encoders
Deep belief networks
Convolutional neural networks
Recurrent neural networks

ABSTRACT

Deep learning, as one of the most currently remarkable machine learning techniques, has achieved great success in many applications such as image analysis, speech recognition and text understanding. It uses supervised and unsupervised strategies to learn multi-level representations and features in hierarchical architectures for the tasks of classification and pattern recognition. Recent development in sensor networks and communication technologies has enabled the collection of big data. Although big data provides great opportunities for a broad of areas including e-commerce, industrial control and smart medical, it poses many challenging issues on data mining and information processing due to its characteristics of large volume, large variety, large velocity and large veracity. In the past few years, deep learning has played an important role in big data analytic solutions. In this paper, we review the emerging researches of deep learning models for big data feature learning. Furthermore, we point out the remaining challenges of big data deep learning and discuss the future topics.

1. Introduction

Recently, the cyber-physical-social systems, together with the sensor networks and communication technologies, have made a great progress, enabling the collection of big data [1,2]. Big data can be defined by its four characteristics, i.e., large volume, large variety, large velocity and large veracity, which is usually called 4V's model [3–5]. The most remarkable characteristic of big data is large-volume that implies an explosive in the data amount. For example, Flicker generates about 3.6 TB data and Google processes about 20,000 TB data everyday. The National Security Agency reports that approximately 1.8 PB data is gathered on the Internet everyday. One distinctive characteristic of big data is large variety that indicates the different types of data formats including text, images, videos, graphics, and so on. Most of the traditional data is in the structured format and it is easily stored in the two-dimensional tables. However, more than 75% of big data is unstructured. Typical unstructured data is multimedia data collected from the Internet and mobile devices [6]. Large velocity argues that big data is generating fast and requires to be processed in real time. The real-time analysis of big data is crucial for e-commerce to provide the online services. Another important characteristic of big data is large veracity that refers to the existence of a huge number of noisy objects, incomplete objects, inaccurate objects, imprecise objects and redundant objects [7]. The size of big data is continuing to grow at an unprecedented rate and is will reach 35 ZB by 2020. However, only having massive data is inadequate. For most of the applications such as industry and medical, the key is to find and extract valuable knowledge from big data for prediction services support. Take the physical devices that suffer mechanical malfunctions occasionally in the industrial manufacturing for an example. If we can analyze the collected parameters of devices effectively before the devices break down, we can take the immediate actions to avoid the catastrophe. While big data provides great opportunities for a broad of areas including e-commerce, industrial control and smart medical, it poses many challenging issues on data mining and information processing. Actually, it is difficult for traditional methods to analyze and process big data effectively and efficiently due to the large variety and the large veracity.

Deep learning is playing an important role in big data solutions since it can harvest valuable knowledge from complex systems [8]. Specially, deep learning has become one of the most active research points in the machine learning community since it was presented in 2006 [9–11]. Actually, deep learning can track back to the 1940s. However, traditional training strategies for multi-layer neural networks always result in a locally optimal solution or cannot guarantee the convergence. Therefore, the multi-layer neural networks have not received wide applications even though it was realized that the multi-layer neural networks could achieve the better performance for feature and representation learning. In 2006, Hinton et al. [12] proposed a two-stage strategy, pre-training and fine-tuning, for training deep learning effectively, causing the first back-through of deep learning. In addition,

* Corresponding author.
E-mail address: ltyang@stfx.ca (L.T. Yang).

http://dx.doi.org/10.1016/j.inffus.2017.10.006
Received 30 August 2017; Accepted 17 October 2017

the increase of computing power and data size also contributes to the popularity of deep learning. As the era of big data comes, a large number of samples can be collected to train the parameters of deep learning models. Meanwhile, training a large-scale deep learning model requires high-performance computing systems. Take the large-scale deep belief network with more than 100 million free parameters and millions of training samples developed by Raina et al. [13] for example. With a GPU-based framework, the training time for such the model is reduced from several weeks to about one day. Typically, deep learning models use an unsupervised pre-training and a supervised fine-tuning strategy to learn hierarchical features and representations of big data in deep architectures for the tasks of classification and recognition [14]. Deep learning has achieved state-of-the-art performance in a broad of applications such as computer vision [15,16], speech recognition [17,18] and text understanding [19,20].

In the past few years, deep learning has made a great progress in big data feature learning [21–23]. Compared to the conventional shallow machine learning techniques such as supported vector machine and Naive Bayes, deep learning models can take advantage of many samples to extract the high-level features and to learn the hierarchical representations by combining the low-level input more effectively for big data with the characteristics of large variety and large veracity. In this paper, we review the emerging research work on deep learning models for big data feature learning. We first present four types of most typical deep learning models, i.e., stacked auto-encoder, deep belief network, convolutional neural network and recurrent neural network, which are also the most widely used for big data feature learning, in Section 2. Afterwards, we provide an overview on deep learning models for big data according to the 4V's model, including large-scale deep learning models for huge amounts of data, multi-modal deep learning models and deep computation model for heterogeneous data, incremental deep learning models for real-time data and reliable deep learning models for low-quality data. Finally, we discuss the remaining challenges of deep learning on big data and point out the potential trends.

2. Typical deep learning models

Since deep learning was presented in *Science* magazine in 2006, it has become an extremely hot research topic in the machine learning community. Various deep learning models have been developed in the past few years. The most typical deep learning models include stacked auto-encoder (SAE), deep belief network (DBN), convolutional neural network (CNN) and recurrent neural network (RNN), which are also most widely used models. Most of other deep learning models can be variants of these four deep architectures. In the following parts, we review the four typical deep learning models briefly.

2.1. Stacked auto-encoder (SAE)

A stacked auto-encoder model is usually constructed by stacking several auto-encoders that are the most typical feed-forward neural networks [24–26]. A basic auto-encoder has two stages, i.e., encoding stage and decoding stage, as presented in Fig. 1.

In the encoder stage, the input x is transformed to the hidden layer h

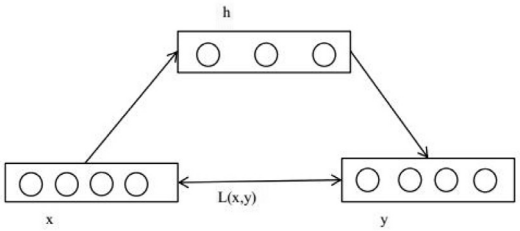

Fig. 1. Basic auto-encoder.

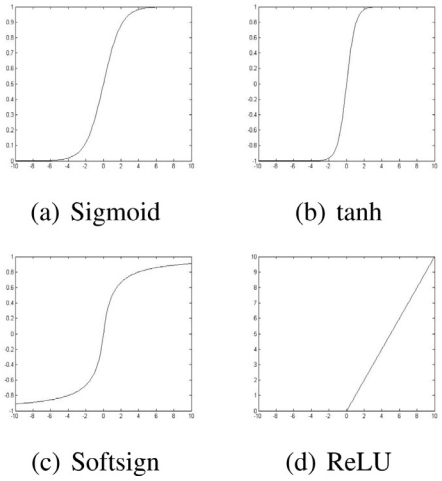

Fig. 2. Functional graph of non-linear activation functions.

via the encoding function f:

$$h = f(W^{(1)}x + b^{(1)}). \tag{1}$$

Afterwards, the hidden representation h is reconstructed back to the original input that is denoted by y in the decoding stage:

$$y = g(W^{(2)}h + b^{(2)}). \tag{2}$$

Typically, the encoding function and the decoding function are non-linear mapping functions. Four widely used non-linear activation functions are the Sigmoid function $f(x) = 1/(1 + e^{-x})$, the tanh function $f(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$, the softsign function $f(x) = x/(1 + |x|)$ and the ReLu (Rectified Linear Units) function $f(x) = max(0, x)$. The functional graph of the four non-linear activation functions is presented in Fig. 2.

$\theta = \{W^{(1)}, b^{(1)}; W^{(2)}, b^{(2)}\}$ is the parameter set of the basic auto-encoder and it is usually trained by minimizing the loss function J_θ with regard to m training samples:

$$J_\theta = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - x^{(i)})^2, \tag{3}$$

where $x^{(i)}$ denotes the ith training sample.

Obviously, the parameters of the basic auto-encoder are trained in an unsupervised strategy. The hidden layer h is viewed as the extracted feature or the hidden representation for the input data x. When the size of h is smaller than that of x, the basic auto-encoder can be viewed as an approach for data compression.

The basic auto-encoder model has some variants. For example, a regularization named wight-decay is usually integrated into the loss function to prevent the over-fitting:

$$J_\theta = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - x^{(i)})^2 + \lambda\sum_{j=1}^{2}\|W^{(j)}\|, \tag{4}$$

where λ is a hiper-parameter used to control the strength of the weight-decay.

Another representative variant is sparse auto-encoder [27,28]. To make the learned features sparse, the sparse auto-encoder adds a sparsity constraint into the hidden units, leading to the corresponding loss function as:

$$J_\theta = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - x^{(i)})^2 + \sum_{j}^{n}KL(p\|p_j), \tag{5}$$

where n denotes the number of neurons in the hidden layer and the second item denotes the KL-divergence. Specially, the KL-divergence with regard to the jth neuron is defined as: