Neurocomputing 273 (2018) 634–642

Contents lists available at ScienceDirect

Neurocomputing

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

Multi-target deep neural networks: Theoretical analysis and implementation

Zeng Zeng[a], Nanying Liang[a,*], Xulei Yang[b], Steven Hoi[c]

[a] Institute for Infocomm Research, 1 Fusionopolis Way, #21-01 Connexis (South Tower), 138632 Singapore
[b] Institute of High Performance Computing, 1 Fusionopolis Way, #16-16 Connexis, 138632 Singapore
[c] Singapore Management University, 81 Victoria Street, 188065 Singapore

ARTICLE INFO

Article history:
Received 8 June 2017
Revised 11 August 2017
Accepted 31 August 2017
Available online 8 September 2017

Communicated by Prof. Zidong Wang

Keywords:
Deep neural networks
Multi-target deep learning
Object detection
Segmentation
Learning path

ABSTRACT

In this work, we propose a novel deep neural network referred to as Multi-Target Deep Neural Network (MT-DNN). We theoretically prove that different stable target models with shared learning paths are stable and can achieve optimal solutions respectively. Based on GoogleNet, we design a single model with three different targets, one for classification, one for regression, and one for masks that is composed of 256 × 256 sub-models. Unlike bounding boxes used in ImageNet, our single model can draw the shapes of target objects, and in the meanwhile, classify the objects and calculate their sizes. We validate our single MT-DNN model via rigorous experiments and prove that the multiple targets can boost each other to achieve optimization solutions.

1. Introduction

Deep neural networks have shown great promise in many practical applications. State-of-the-art performance has been reported in several domains, ranging from image classification [1], speech recognition [2], to text processing [3], playing Atari games [4]. The latest and greatest honor of deep neural networks belongs to AlphaGo [5] that has defeated Lee Sedol, one of the best human professional Go game players in the world, a feat previously thought to be at least a decade away.

Deep neural networks are networks of neurons, which can execute different simple functions and are connected following predefined topologies [6]. Unlike the networks we are familiar with, e.g., mobile networks, computer networks, sensor networks, which have multiple entries and multiple exits, deep neural networks have only one entry, where data can be poured into the networks, and one exit, where the target functions can obtain the results. All the neurons in the networks learn synchronously (the neurons within the same layers) or asynchronously (the neurons in different layers) to achieve optimal solutions of single target models [7]. Hence, from the view of layers, deep neural networks are end-to-end links, instead of networks.

In our real world, we have many multi-label problems [8]. In the famous ImageNet data set, millions of images have got at least one label [1]. Initially, ImageNet required the competitors to identify which class the target image belongs to within 1*k* or 22*k* known categories. Little by little, ImageNet starts to provide images with additional locations of target objects that are indicated by bounding boxes, and then, the competitions become solving multi-label problems. Now, the most successful solutions are to deliberately integrate the different labels into a single target function, and then using deep neural networks to achieve optimization results of the targets [8,9]. However, no matter how many labels the target models may have, the deep neural networks are end-to-end links, solving "Single-Target" problems, instead of end-to-ends networks that can solve "Multi-Target" problems.

Inspired by communication networks [6,10,11], we propose a novel learning network, Multi-Target Deep Neural Networks (MT-DNN), based on which we can construct scalable deep neural networks. In each *Source–Destination* pair, there is at least one learning path, through which the source data can be transformed into some kinds of values that can be used in the destination as the target. Some learning paths may be shared by different Source–Destination pairs where some *shared* features can be extracted by different target models. We theoretically prove the stabilities of

* Corresponding author.
E-mail addresses: zengz@i2r.a-star.edu.sg (Z. Zeng), liangny@i2r.a-star.edu.sg, nanying@gmail.com (N. Liang), yangx@ihpc.a-star.edu.sg (X. Yang), chhoi@smu.edu.sg (S. Hoi).

http://dx.doi.org/10.1016/j.neucom.2017.08.044
0925-2312/

MT-DNN and prove that all the target learning paths can converge to their optimal solutions, respectively.

Based on the concept of MT-DNN, we design a novel model that has three branches above the main layers of GoogleNet [12]. Branch 1 is aiming at object classifications, branch 2 is used to calculate the size of the objects, and branch 3 is composed of $W \times H$ sub-models working as compound eyes of bees. The target of each sub-model is to figure out whether the corresponding pixel in the image belongs to the object or not. Unlike ImageNet that uses bounding boxes to indicate the locations and sizes of objects, we use masks with size of $W \times H$, where $mask[w, h] = 1$ if the point of $image[w, h]$ belongs to the target object and $mask[w, h] = 0$ otherwise. If we set $W = H = 256$, that means 65,536 sub-models have to be trained. We carry out rigorous experiments with respect to several influencing conditions and prove that our MT-DNN with multi-targets are stable and convergent, and can solve some problems that single-target models cannot do.

1.1. Our contributions

The specific contributions of this work are as follows: a). We propose the concept of MT-DNN; b). We theoretically demonstrate that multiple targets can converge respectively by using Stochastic Gradient Descent (SGD) [5,8,12]. Adjusting the learning rate of each target, we can roughly guarantee all the targets can converge synchronously; c). We present a study case to describe the proposed MT-DNN and design a single model with three different target models; d) We carry out series of experiments to examine the performance of the model and demonstrate that multiple targets can boost each other to achieve optimization solutions. It is the first time in the domain that a single model can identify objects, obtain their sizes, and point out their locations and shapes at the same time.

The rest of the paper is organized as follows. Section 2 discusses relevant research work. Section 3 illustrates the main definitions and theorems. In Section 4, we describe a study case for ease of understanding, and discuss the main target functions. In Section 5, we show the results of our experiments. We conclude our work and discuss some future work in Section 6.

2. Related work

The winner of ILSVRC 14 is "GoogleNet", which is a 22 layers deep network [12]. The main hallmark of this architecture is the improved utilization of the computing resources inside the network by a carefully crafted design. Besides of the top layer that is a softmax function used to calculate the logarithm loss and the beginning of back-forwards learning, there are two more same branches below that carry out the same procedures in different lower layers. In this way, the depth and width of the network are increased while the computational budget is kept constant. This crafted design can help to converge faster, but contributes little to the final accuracy. Although there are three result outputs in GoogleNet, the outputs are aiming at the same target and hence, in our definition, it is still an end-to-end deep neural network.

The problem of object classification is the recognition of the class of an object belongs to. CNNs represent the-state-of-the-art approaches to address this problem. However, to solve this problem alone cannot fully fulfill the requirement in some other real applications. For example, it could be favorable to extraction the location and size information about the object in addition to its class information at the same time, which represents a multiple targets application scenario.

In order to achieve $f_1(x_2)$, $f_2(x_2)$, ..., minimized (or maximized) at the same time, linear integration of multiple models into a single target optimization problem is applied [13,14], i.e., these functions can be added together with some trade-off parameters, λ, as: $minimized: \ f_{all} = \lambda_1 f_1(x_1) + \lambda_2 f_2(x_2) + \cdots$. One type of the method applied in deep neural networks is named as shared computation of convolutions that has been attracting increasing attention for efficient, yet accurate, visual recognition [9,15–19]. In [7], Andrew *et. al.* used linear combination to integrate sparsity and reconstruction models as the target optimization problem with $\lambda_1 = 1$ and $\lambda_2 = 0.1$. They trained their network to obtain 15.8% accuracy in recognizing 22,000 object categories from ImageNet, a leap of 70% relative improvement over the previous state-of-the-art. In [8], Ren et al. proposed Faster R-CNN model that can obtain very high object classification accuracy, while detect the positions of the objects with low errors. They define a loss function, $L_{cls}(p)$, where p is object's probability, for object classification, and a loss function, $L_{reg}(t)$, where t is Euclidean position of object, for position detection, respectively. Then, they obtain the optimization problem $L(p, t) = L_{cls}(p) + \lambda L_{reg}(t)$ with λ set to be 10, and hence, both $L_{cls}(p)$ and $L_{reg}(t)$ are roughly equally weighted. In ILSVRC and COCO 2015 competitions, the model is the foundation of the 1*st*-place winning entries in several tracks.

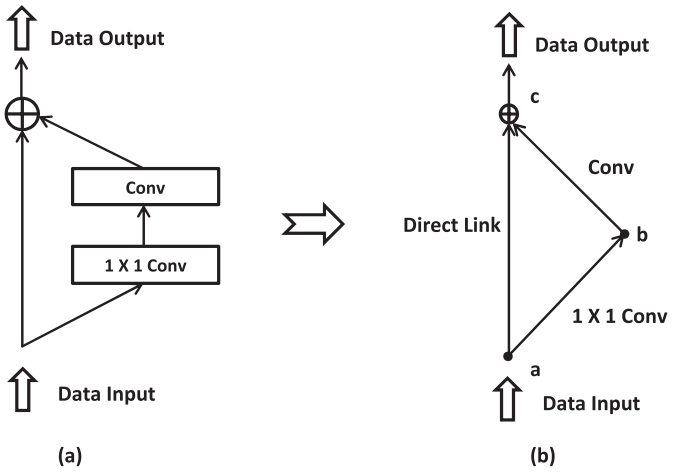

Fig. 1. Transformations of layers to learning links.

3. Concept of multi-target deep neural network

Layers of neural networks consist of neurons that are connected in pre-defined topologies and have one data input and one data output as shown in Fig. 1(a). Neural network layers can "learn" from batches of data by Stochastic Gradient Descent (SGD) functions and update their own parameters through back-forward in up-to-down fashion [5,16]. Referring to Fig. 1, we can observe that layer $1 \times 1\, Conv$, referred to as i, can be transformed to a link $(a \to b)$ or l_i and the function of the layer can be denoted as $out_i = f_{a \to b}(in_i)$ or $out_i = f_{l_i}(in_i)$, where in_i and out_i are layer i's data input and data output, respectively. Layer i can be transferred to link l_i with a mapping function f_{l_i} from $in_i \mapsto out_i$. When two links l_i and l_j need to be combined to a single link, we define two different combination functions: element-wise addition denoted as $l_i \oplus l_j \mapsto l_a$ and concatenation addition denoted as $l_i \otimes l_j \mapsto l_a$. For example, in Fig. 1(a), layer $1 \times 1\, Conv$ and layer $Conv$ can be transferred into learning link $(a \to b)$ and learning link $(b \to c)$, respectively as shown in Fig. 1(b).

AlphaGo has two deep neural networks: policy network and value network. We transfer the networks into two independent learning paths as shown in Fig. 2(a). Now, if we merge the two learning paths together, we can have many potential topologies. In Fig. 2(b), there are two main learning paths that are left one ($S \Rightarrow n \Rightarrow D_1$) and right one ($S \Rightarrow n \Rightarrow D_2$). We can observe that