Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 122 (2017) 622–628

Procedia Computer Science

www.elsevier.com/locate/procedia

Information Technology and Quantitative Management (ITQM 2017)

Hybrid Learning Network: A Novel Architecture for Fast Learning

Ying Liu[a,b], Chao Xiang[a]

[a]School of Computer and Engineering, University of Chinese Academy of Sciences, Beijing, 100190 China
[b]Key Lab of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, 100190 China

Abstract

There're many effective architectures of the artificial neural network(ANN). For which the training is a hard work. The cost for training an ANN increases exponentially when the ANN gets deeper or wider. We therefore propose a novel architecture, the Hybrid Learning Network(HLN), to achieve a fast learning with good stablity. The HLN can learn from both labeled data and unlabeled data at the same time in a hybrid learning manner. It uses a Self Organizing Map unified by the specially designed nonlinear function as the sparsity mask for a hidden layer to improve the training speed. We experiment our architecture on a synthetic dataset to test its regression capability against the traditional architecture, the result is promising.

Peer-review under responsibility of the scientific committee of the 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.

Keywords: hybrid learning; neural network; sparsity mask; self organizing map; fast learning

1. Introduction

Since the first mathematical model of the artificial neural network(ANN) was proposed in 1943[1], lots of different architectures have been proposed to develop the model, such as the Convolutional Neural Network(CNN) for image recognition [2], the Recurrent Convolutional Neural Network(R-CNN) for object detection in videos [3], and the Long Short Term Memory(LSTM) for speech recognition [4], etc. These specially designed neural networks are trained by a few ecient methods such as the stochastic gradient descent(SGD)[5]. Such architectures suit well for their specific applications but may have plain or worse performances on others, and their best performances rely heavily on the hyperparameter configuration. Therefore a relatively universal architecture that enables equal or similar performances among varied applications is in real demand.

Although a number of different neural network architectures have been proposed in the past years, literally all of them can be classifed into 3 categories, supervised [6], unsupervised [7], and semi-supervised learning [8]. In supervised learning, one can only train a model on labeled samples. However in real applications, labeling a large dataset is infeasible as a result unlabeled samples are the major. To make use of unlabeled data, a few unsupervised training algorithms emerged. These unsupervised learning methods attempt to produce an optimized parameter initialization [9]. Thus, such unsupervised techniques can only be applied before the supervised training phase.

*Corresponding author. Tel.: +86-183-0114-2368;
E-mail address: xiangchao215@mails.ucas.ac.cn.

1877-0509
Peer-review under responsibility of the scientific committee of the 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.
10.1016/j.procs.2017.11.415

Once the supervised begins, unsupversied learning will quit. Semi-supervised learning allows the model to learn from both labeled and unlabeled samples at the same time, where a regularizer is embedded in the supervised optimizing object, and generally a balance constraint is required to avoid the trival solution [10]. Such an enhanced learner brings more hyperparameters(such as the balance constraint), making it even more difficult to search for the best settings for current architectures. Additionally, this integration manner makes it impossible to separate the two learners as the semi-supervised regularizer is built on the supervised learner, and therefore an early supervised training alone with profound labeled data is necessary before the semi-supervised learning starts.

Therefore in the paper, we introduce a new architecture that combines the supervised and unsupervised learning requiring no extra techniques in training. Our architecture, the Hybrid Learning Network(HLN), consists of stacked hybrid learning layers, each of which embeds a Self Organizing Map(SOM). In the similar manner of backpropagation, HLNs demonstrate higher learning capability and stablity.

The main contributions of our work are:

- HLN is proposed as a hybrid learning architecture to learn both unlabeled and labeled data simultaneously. HLN overcomes the problem of traditional semi-supervised learning methods which requires an early standalone training for the supervised learner.
- A SOM-embedding layer structure is designed to learn a cluster mapping function from unlabeled data to speed up the supervised learning from labeled data.
- A nonlinear function $h(x)$, is proposed to measure the state of a SOM in each iteration of the training, and then determine a sparsity mask for every hidden layer.

Our proposed HLN is implemented in Python and all our code and results of experiments are available at `https://github.com/hiroki-kyoto/hybrid-learning-net`.

The rest of the paper is organized as follows. In section 2 we introduce existing semi-supervised algorithms for neural network models and SOM. In section 3, we present our novel architecture HLN and how to embed SOMs into deep architectures of neural networks. In section 4 we explain the training theory for HLNs. Section 5 presents the experimental results, and the last section concludes our work.

2. Related work and background Knowledge

2.1. Semi-supervised learning for neural network

A key assumption in semi-supervised algorithms, is the structure assumption: two samples with similar distribution on the same mapping structure tend to have high probability belonging to the same class. Based on this assumption, one can use large unlabeled data to uncover such structures. There're already a few algorithms dedicated to it, such as cluster kernels [11], Low Density Separation(LDS) [12], label propagation [13], etc. In such algorithms, designing a regularizer to enable the model to learn the representation or structure of raw data becomes the key point.

Let's firstly focus on the general algorithm description of semi-supervised learning. Given a set of unlabeled samples, $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}(\mathbf{x} \in \mathbb{R}^d)$, and the similarity labels between any $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{W} = \{W_{ij}|i, j = 1, \cdots, N\}$, our goal is to find the best embedding function,$f(\mathbf{x})$, for each sample $\mathbf{x}_i$, in order to minimize:

$$\Delta_f = \sum_{i=1}^{N}\sum_{j=1}^{N} L\left(f(\mathbf{x}_i), f(\mathbf{x}_j), W_{ij}\right) \tag{1}$$

where,

- $L(\cdot)$ is the loss function of 3 variables: $\left\langle f(\mathbf{x}_i), f(\mathbf{x}_j), W_{ij}\right\rangle$. For example, if Euclidean distance is used,

$$L\left(f(\mathbf{x}_i), f(\mathbf{x}_j), W_{ij}\right) = \left(\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\| - W_{ij}\right)^2$$

- $f(\mathbf{x}) \in \mathbb{R}^n$ is the embedding function, trying to produce an output vector for $\mathbf{x}_i$, similar to that for $\mathbf{x}_j$ with $W_{ij} = 0$, and disimilar with $W_{ij} = 1$.