Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 72 (2015) 137 – 144

Procedia Computer Science

The Third Information Systems International Conference

Simulated Annealing Algorithm for Deep Learning

L.M. Rasdi Rere, Mohamad Ivan Fanany, Aniati Murni Arymurthy

[a]Faculty of Computer Science, Universitas Indonesia, Depok 16424, Indonesia

Abstract

Deep learning (DL) is a new area of research in machine learning, in which the objective is moving us closer to the goal of artificial intelligent. This method can learn many levels of abstraction and representation to create a common sense of data such as text, sound and image. Although DL is useful for a variety of tasks, it's hard to train. Some methods in training deep learning to make it optimal have been proposed, including Stochastic Gradient Descent, Conjugate Gradient, Hessian-free optimization, and Krylov Subspace Descent. In this paper, we proposed Simulated Annealing (SA) to improve the performance of Convolution Neural Network (CNN), as an alternative approach for optimal DL using modern optimization technique, i.e. metaheuristic algorithm. MNIST dataset is used to ensure the accuracy and efficiency of the proposed method. Moreover, we also compare our proposed method with the original of CNN. Although there is an increase in computation time, the experiment results show that the proposed method can improve the performance of original CNN.

Peer-review under responsibility of organizing committee of Information Systems International Conference (ISICO2015)

Keywords: Simulated Annealing algorithm; Deep Learning, Optimization

1. Introduction

Deep learning refers to a new class of machine learning methods, where the process of information from many layers in hierarchical architectures can be used to classify pattern and learning of feature. This technique is in the intersections between the areas of research in the graphical model, optimization, signal processing, pattern recognition and neural network. Three principal reasons for the current popularity of deep learning are it drastically increases the capabilities of chip processing, significantly lowers the cost of computing hardware and recent advances in research of machine learning [1].

In general, techniques of deep learning can be classified into deep discriminative models and generative models [2]. Examples of discriminative models are deep neural networks (DNNs), recurrent neural network (RNNs), and convolutional neural networks (CNNs). On the other hand, generative models, for instance, are restricted Boltzmann machine (RBMs), deep belief networks (DBNs),

Peer-review under responsibility of organizing committee of Information Systems International Conference (ISICO2015)
doi:10.1016/j.procs.2015.12.114

regularized autoencoders, and deep Boltzmann machines (DBMs). Among those techniques, CNNs is the focus of this paper. CNNs is a deep supervised-learning while this model is usually efficient to train and test, flexible to construct, and suitable for end-to-end learning of complex system [2].

Although DL has good reputation for solving a variation of learning task, it is not easy to train [3]. The Recent proposal of optimization techniques for training DL used layered wise pre-training [4]. Some examples of the successful methods for training of this technique are Stochastic Gradient Descent, Conjugate gradient, Hessian-free Optimization and Krylov Subspace Descent.

Stochastic Gradient Descent (SGD) is simple to implement and also fast in the process for problems that have many training examples. However, this method needs a lot of manual tunings to optimize its parameters and essential sequential, so that to distribute them using computer clusters or to parallelize them using GPUs is difficult. In contrast, Conjugate Gradient (CG) can distribute the computation across machines and parallelize for computing the gradient on GPU as well as more stable to train and easier to check for convergence. But this method usually is slow, and it can be fast due to the availability of large amounts of RAMs, multicore CPUs, GPU and computer cluster with fast network hardware [5].

Hessian-free optimization (HFO), proposed by J. Marten [6], uses the basis of 2^{nd} order optimization approach, the truncated-Newton. HFO has been applied to train deep auto-encoders neural network and able to overcome the under-fitting problem as well as more efficient than pre-training + fine-tuning approach. Krylov Subspace Descent (KSD), proposed by Vinyals [7], is another second order optimization method. Compare to HFO, optimization using KSD has some advantages. First, KSD has greater simplicity and robustness, in which it does not need heuristic to initialize and update the smoothing value. Second, KSD can be applied even if H is not positive semi-definite, and third, KSD seems to work better than HFO in both optimization speed and classification performance. However, the weakness of optimization using SGD is more memory required.

In fact, a vast majority of modern optimization techniques are usually heuristic and metaheuristic [8]. These optimization methods are very powerful in solving hard optimization problems, and they have been applied in almost all the main areas of science and engineering as well as industrial application. However, the research of metaheuristic algorithms to optimize DL is rarely conducted. Because of that in this worked, we used metaheuristic algorithm as an alternative approach for optimal performance of DL.

Metaheuristic work for three main purposes: solving the problem faster, solving large problems and obtaining robust algorithms. Moreover, they are flexible, simple to design and also not difficult to implement [9]. The combination of rules and randomness to duplicate natural phenomena usually is employed in metaheuristic algorithms. The phenomena may include biological evolutionary process like Genetic Algorithms (GA), Genetic Programming (GP), Evolution Strategy (ES), and Differential Evolution (DE). Animal behaviors, as an ethology phenomenon, for instance, are Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Bacterial Foraging Optimization Algorithms (BFOA) and Bee Colony Optimization (BCO). Examples of physics phenomena are Simulated Annealing (SA), Threshold Accepting method (TA) and Micro canonic Annealing (MA). TA and MA are variants of SA [10]. Another form of metaheuristic, for instance, is Harmony Search method, is inspired by musical phenomena [11].

Metaheuristic algorithms can also be classified into trajectory-based (single-solution) and population-based [10]. Some of single solution based metaheuristic are SA, TA, MA, Tabu Search (TS), Noising Method (NM), and Guided Local Search (GLS). In case of Population-based metaheuristic, it can be separated into Evolutionary Computation and Swarm Intelligent. The general term of Evolutionary Computation is inspired by the Darwinian principles of nature's capability to evolve living beings well adapted to their environment. Examples of these algorithms are GA, ES, GP, and DE. Furthermore, Swarm Intelligent takes inspiration from the collective behavior of a group of social insect colonies and of other animal societies. Including these methods are ACO, PSO, BFOA, and BCO. Among these algorithms, SA is used in our study.