Journal of Computational Mathematics
Vol.39, No.1, 2021, 147–158.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1912-m2018-0279

CONVERGENCE OF BACKPROPAGATION WITH MOMENTUM FOR NETWORK ARCHITECTURES WITH SKIP CONNECTIONS*

Chirag Agarwal
Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA
Email: chiragagarwall12@gmail.com
Joe Klobusicky[1)]
Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
Email: klobuj@rpi.edu
Dan Schonfeld
Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA
Email: dans@uic.edu

Abstract

We study a class of deep neural networks with architectures that form a directed acyclic graph (DAG). For backpropagation defined by gradient descent with adaptive momentum, we show weights converge for a large class of nonlinear activation functions. The proof generalizes the results of Wu et al. (2008) who showed convergence for a feed-forward network with one hidden layer. For an example of the effectiveness of DAG architectures, we describe an example of compression through an AutoEncoder, and compare against sequential feed-forward networks under several metrics.

Mathematics subject classification: 68M07, 68T01.
Key words: Backpropagation with momentum, Autoencoders, Directed acyclic graphs.

1. Introduction

Neural networks have recently enjoyed an acceleration in popularity, with new research adding to several decades of foundational work. From multilayer perceptron (MLP) networks to the more prominent recurrent neural networks (RNNs) and convolutional neural networks (CNNs), neural networks have become a dominant force in the fields of computer vision, speech recognition, and machine translation [11]. Increase in computational speed and data collection have legitimized the training of increasingly complex deep networks. The flow of information from input to output is typically performed in a strictly sequential feed-forward fashion, in which for a network consisting of L layers, nodes in the ith layer receive input from the $(i-1)$st layer, compute an output for each neuron through an activation function, and in turn use this output as an input for the $(i+1)$st layer. A natural extension to this network structure is the addition of “skip connections” between layers. Specifically, we are interested in the class of architectures in which the network of connections form a directed acyclic graph (DAG). The defining property of a DAG is that it can always be decomposed into a *topological ordering* of L layers, in which

* Received December 18, 2018 / Revised version received August 4, 2019 / Accepted December 9, 2019 / Published online August 30, 2020 /

[1)] Corresponding author

nodes in layer i may be connected to layer j, where $j > i$. A skip connections is a connection between nodes in layers i and j, with $j > i+1$. There has been an increasing interest in studying networks with skip connections which skip a small number of layers, with examples including Deep Residual Networks (ResNet) [5], Highway Networks [13], and FractalNets [8]. ResNets, for instance, use "shortcut connections" in which a copy of previous layers is mapped through an identity mapping to future layers. Kothari and Agyepong [7] introduced "lateral connections" in the form of a chain, with each unit in a hidden layer connected to the next. The full generality of neural networks for DAG architectures was considered in [6], which demonstrated superior performance of neural networks, entitled DenseNets, under a wide variety of skip-connections.

As an example of the efficacy of DAG architectures considered in [6], we consider AutoEncoders, a class of neural networks which provide a means of data compression. For an AutoEncoder, input data, such as a pixelated image, is also the desired output for a neural network. During an encoding phase, input is compressed through several hidden layers before arriving at a middle hidden layer, called the code, having dimension smaller than the input. The next phase is decoding, in which input from the code is fed through several more hidden layers until arriving at the output, which is of the same dimension as the input. The goal of compression is to minimize the difference between input data and output. In [1], Agarwal et al. introduced CrossEncoders and demonstrated its superior performance against AutoEncoders with no skip-connections. In Section 3, we extend the previous results to include the MNIST and Olivetti faces public datasets. We validate our results against several commonly used compression based performance metrics.

Our main theoretical result is the convergence of backpropagation with DAG architectures using gradient descent with momentum. It is well known that feed-forward architectures converge under backpropagation, which is essentially gradient descent applied to an error function (see [3], for instance). Updates for weights in backpropagation may be generalized to include a momentum term, which can help with increasing the convergence rate [12]. Momentum can help with escaping local minima, but concerns of overshooting require careful arguments for establishing convergence. Formal arguments for convergence have so far been restricted to simple classes of neural networks. Bhaya [2] and Torii [15] studied the convergence with backpropagation using momentum under a linear activation function. Zhang et al. [17] generalized convergence for a class of common nonlinear activation functions, including sigmoids, for the case of a zero hidden layer networks. Wu et al. [16] further generalized to one layer by demonstrating that error is monotonically decreasing under backpropagation iterations for sufficiently small momentum terms. The addition of a hidden layer required [16] to make the additional assumption of bounded weights during the iteration procedure.

It is not evident whether applying the methods of [16] would generalize to networks with several hidden layers and skip connections, or if they would require stronger assumptions on boundedness of weights or the class of activation functions. We show in Section 4 that convergence indeed does hold, with similar assumptions to the proof of convergence of one hidden layer. In Theorem 4.1, we give the key inequality for proving Theorem 2.1, a recursive form for increments of error and output values of hidden layers after each iteration. This estimate allows us to show that for sufficiently small momentum parameters (including the case of zero momentum), error decreases with each iteration. Our approach to convergence is somewhat more explicit than the traditional proof of gradient descent, which minimizes a loss function without considering network architecture.