Maseeh College of Engineering and Computer Science
PORTLAND STATE UNIVERSITY

Revised backpropagation algorithm of artificial neural network

Mark Musil, Jian Meng, Yuhao Zhou, Peilong Ning

Background & Introduction

Backpropagation algorithms are a family of a methods that can efficiently train the neural network based on gradient descent. In other words, backpropagation algorithm can adjust the weight value of each neuron by calculating the gradient of the loss function. The objective of gradient descent is to reach the minimum value of the loss function (error).

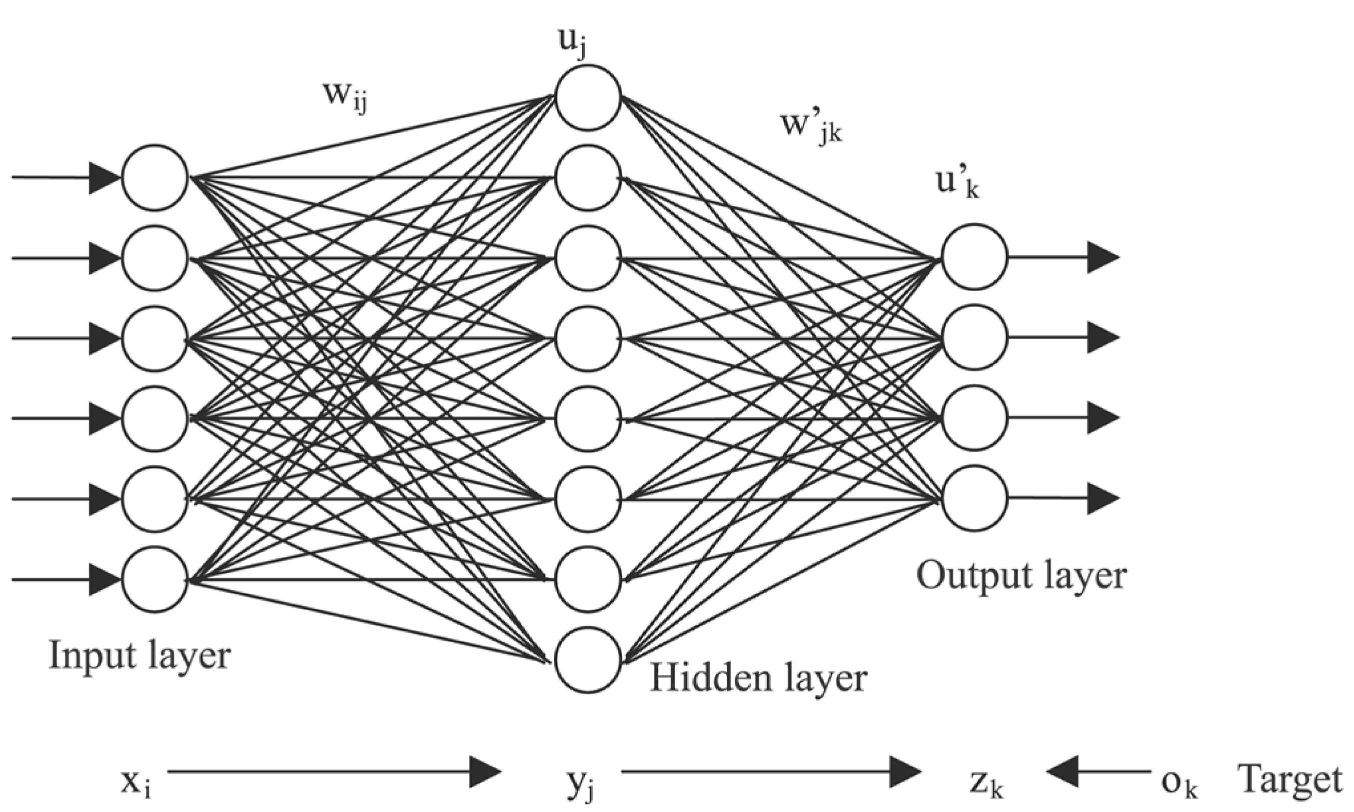

Figure 1: A simple neural network.

Problem statement

1. The error of the input features will propagate through the neural network and then get amplified.
2. The gradient descent algorithm is generally slow because it requires small learning rate for stable learning process (descent process)
3. The disadvantage of the traditional SGD is the direction of the descent is not the shortest direction of the optimization.

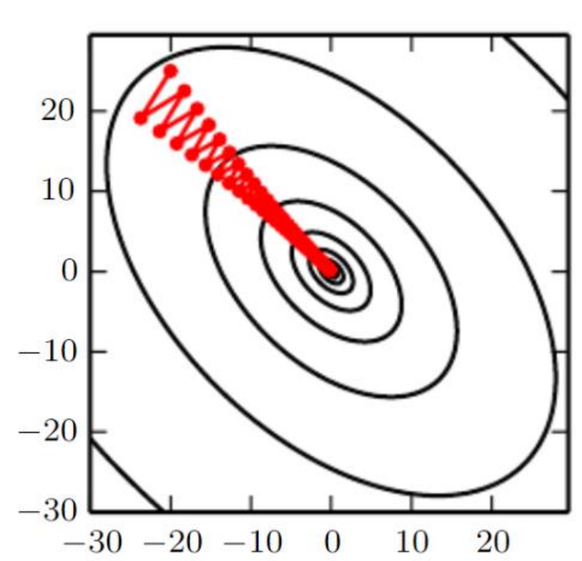

Figure 2: Example of SGD making redundant steps along the gradient.

Algorithm & Solution

By adding an adaptive learning rate term, we can focus learning adjustments in layers with the most impact of the overall layer. Due to the structure of the neural network, the first layers of the network (near the input) get more error. The experiments shown used layer coefficients $[K^1 = 2, K^2 = 1, K^3 = 0.5, K^4 = 0.3]$.

Algorithm #1: Adaptive learning rate adjustment – Error by layer method
Pick the learning rate coefficient for each layer.

$$w^i_{jk} -= \eta\left(a^{i-1}\delta^I_i\right)K^I$$
$$b^I_i -= \eta\delta^I_i K^I$$

Algorithm #2: Neuron-wise method
Pick the principle directions in δ – The error sensitive term. By applying SVD on δ:

$$\delta = U\Sigma V^T$$

Then pick the first n vectors in the decomposition matrices, denoted as U_n, Σ_n, V_n^T. Then use $\delta_n = U_n\Sigma_n V_n^T$ to calculate the new gradients and update weights.

Experiment & Verification

Method Comparison:
The new update method was tested based on two different perspectives:
1. The speed of convergence
2. The accuracy of the method in both training set and test set.

The objective of all the test and comparison is to find the difference between the revised method and the traditional stochastic gradient descent algorithm.

Data sets:
1. Iris dataset .
2. MNIST dataset.
3. Covertype dataset
4. Olivetti Face dataset

Experimental Results – Table

The following table summarize the results of each method based on two dataset:

Method	Final training loss	Final test loss
SGD – iris	0.23	0.24
SGD – MNIST	0.22	0.23
Error by layer – MNIST	0.24	0.24
Error by layer – Iris	0.23	0.23
Error to neuron – MNIST	0.13	0.13
Error by neuron – Iris	0.24	0.24

Experimental Results – Plots

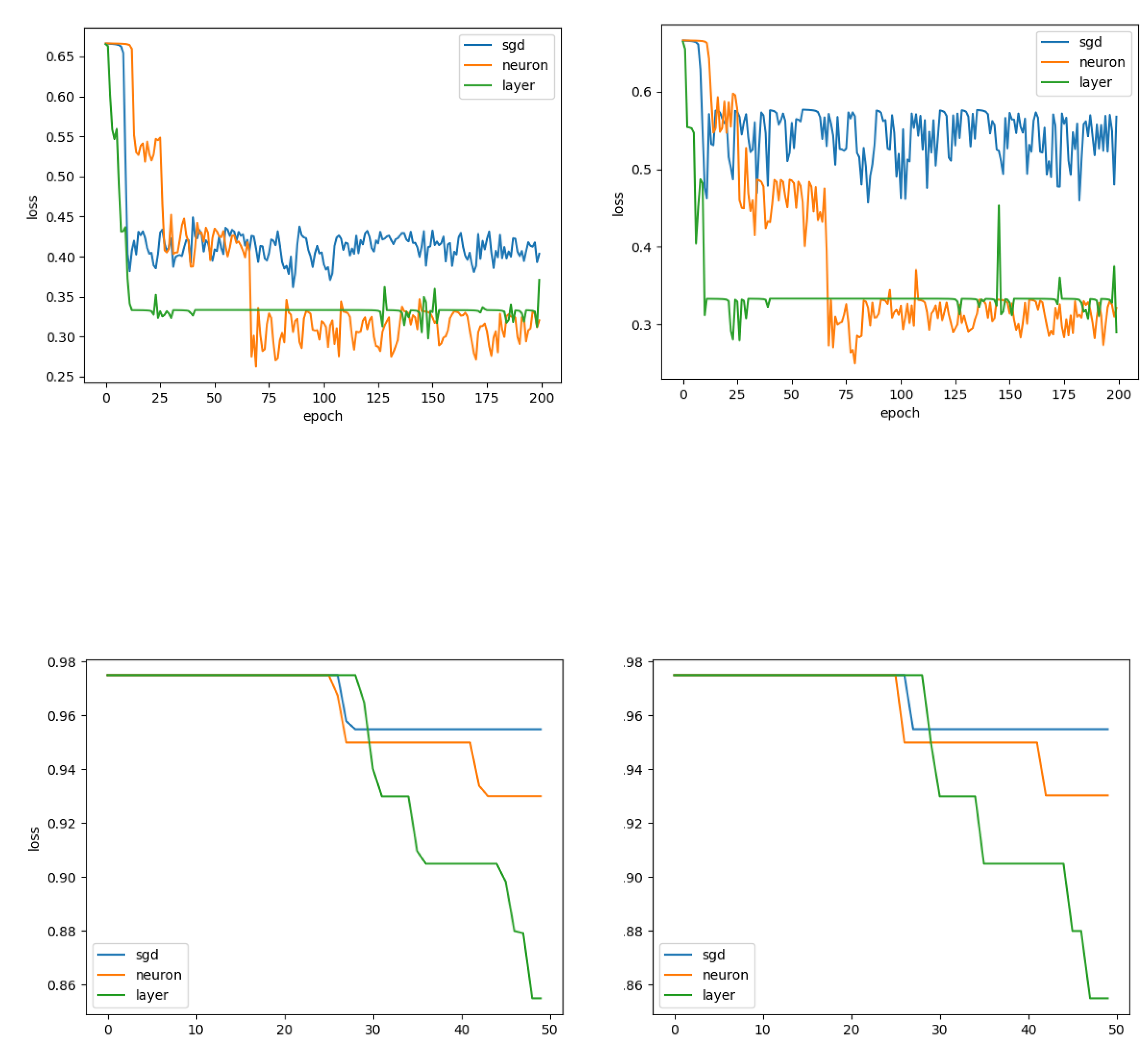

Figure 4: Test on MNIST dataset. Hidden Dims: [200 80 40]; Learning Rate 0.1

Department of Electrical and Computer Engineering

Portland State UNIVERSITY
Maseeh College of Engineering and Computer Science