A Practical Analysis of the Convergence of Back Propagation

Andrei Purcarus Sean Stappas

ECSE 526
McGill University

November 30, 2017

Table of Contents

Table of Contents

Objectives

- Implement a fully connected neural network.
- Analyze the effects on performance and convergence of several improvements to back propagation, including those proposed by LeCun's 1998 paper *Efficient BackProp* [1].
- Perform practical tests on the MNIST data set.

MNIST

Figure 1: Sample digits from the MNIST dataset [2].

Table of Contents

Input Normalization

Average of the input training data should be close to 0.

$$x \leftarrow x - \bar{x} \tag{1}$$

Variance of the input training data should be the same for each feature (≈ 1).

$$x_i \leftarrow \frac{x_i}{\sigma_{x_i}} \tag{2}$$

Sigmoid

Logistic sigmoid:

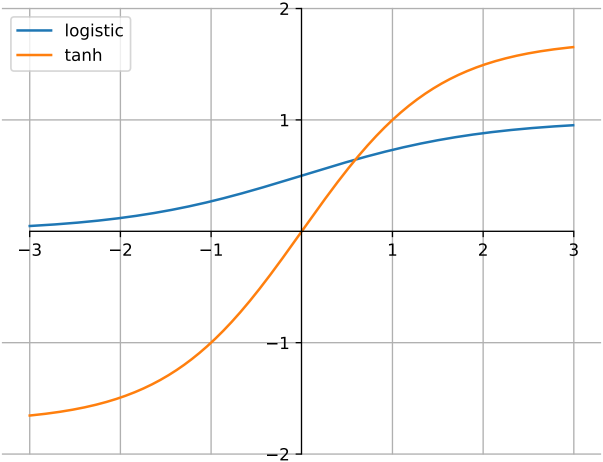
$$f_{logistic}(x) = \frac{1}{1+e^{-x}} \tag{3}$$

Tanh sigmoid:

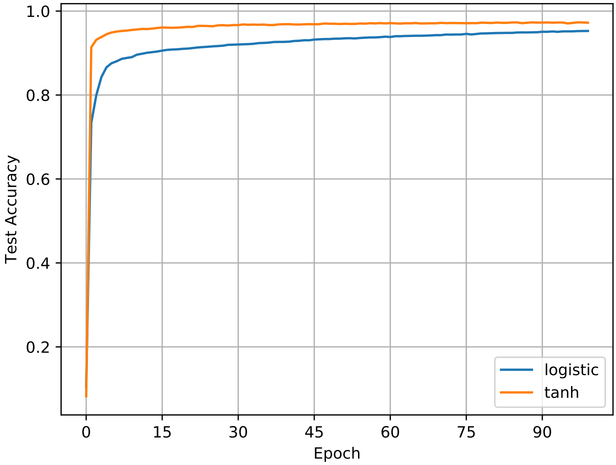
$$f_{tanh}(x) = 1.7159 \tanh\left(\frac{2}{3}x\right) \tag{4}$$

- Symmetric about the origin.
- $f(\pm 1) = \pm 1$.
- Second derivative maximum at $x = 1$.
- Output variance close to 1 if input is normalized.

Sigmoid



Sigmoid


1.0
0.8
0.6
0.4
0.2
Test Accuracy
0
15
30
45
60
75
90
Epoch
logistic
tanh

Weight Initialization

Initial weights should be randomly drawn from a distribution with zero mean and standard deviation σ_w (m: fan-in).

$$\sigma_w = m^{-1/2} \tag{5}$$

This ensures that the initial weights range over the sigmoid's linear region (assuming input normalization and tanh sigmoid).

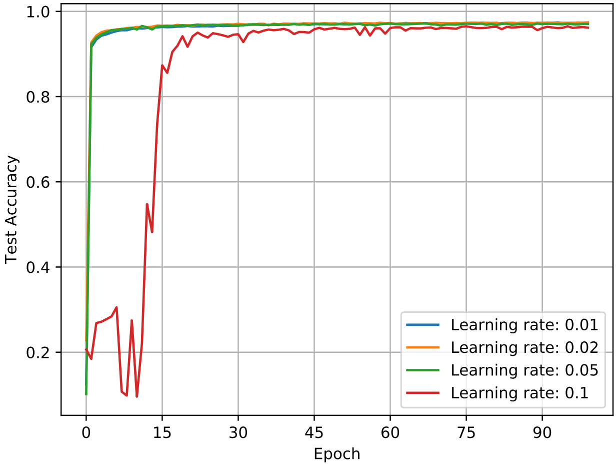
- Gradients large enough for learning to proceed.
- Linear learning occurs before more difficult nonlinear learning.
- Distribution of outputs of each node has $\sigma \approx 1$.

Learning Rate

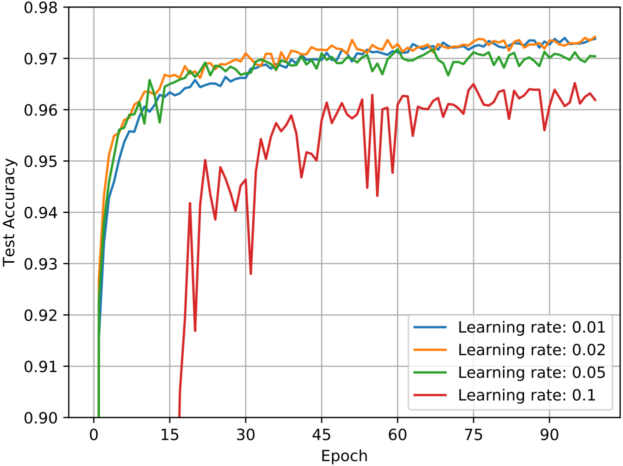
$$w_i \leftarrow w_i - \eta_i \frac{\partial E}{\partial w_i} \tag{6}$$

- η_i: learning rate for weight w_i.
- Large learning rate: oscillations.
- Small learning rate: slow learning.

Learning Rate



Learning Rate



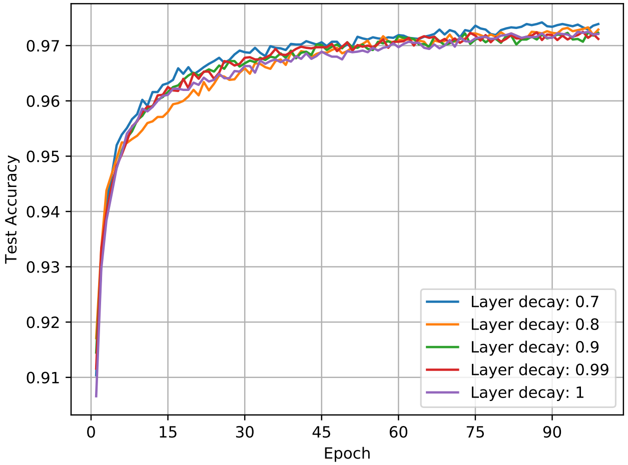
Layer Decay

Learning rate for weights in lower layers should be larger than for those in higher layers.

$$\eta_i = \delta \eta_{i-1} \tag{7}$$

- η_i: learning rate in layer i (layer i is higher than layer $i-1$).
- δ: layer decay $(0 < \delta \leq 1)$.

Layer Decay

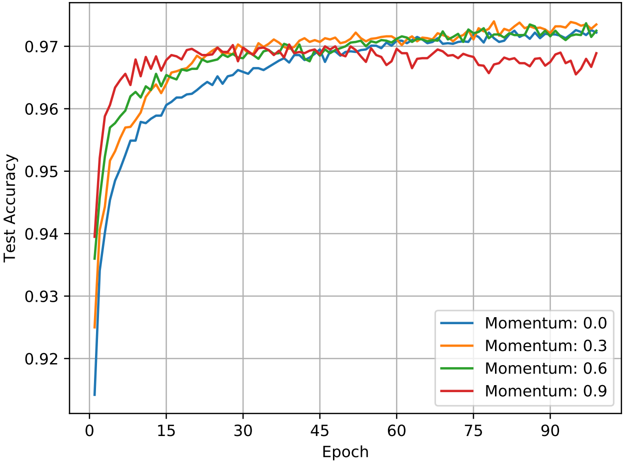

0.97
0.96
0.95
0.94
0.93
0.92
0.91
Test Accuracy
0
15
30
45
60
75
90
Epoch
Layer decay: 0.7
Layer decay: 0.8
Layer decay: 0.9
Layer decay: 0.99
Layer decay: 1

Momentum

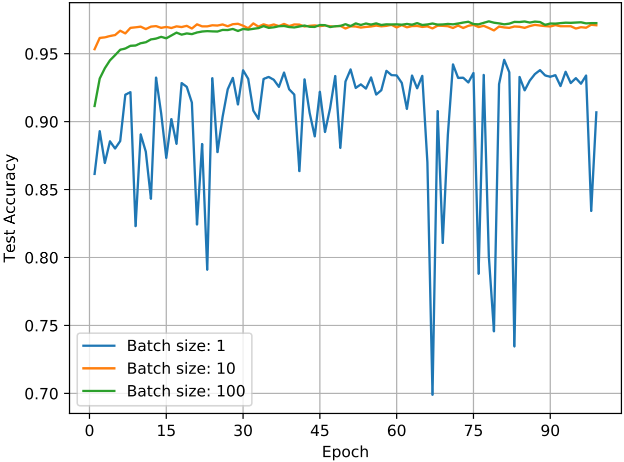
$$\Delta w(t+1) \leftarrow -\eta \nabla E + \mu \Delta w(t) \tag{8}$$

- μ: momentum.
- Avoids getting stuck in local minima.
- Large momentum: overshoot the minimum.
- Small momentum: slow learning.

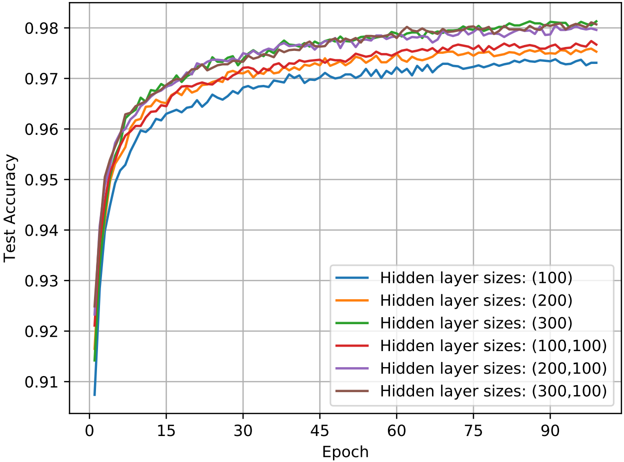
Momentum


0.97
0.96
0.95
0.94
0.93
0.92
Test Accuracy
0
15
30
45
60
75
90
Epoch
Momentum: 0.0
Momentum: 0.3
Momentum: 0.6
Momentum: 0.9

Batch Size


Test Accuracy
0.95
0.90
0.85
0.80
0.75
0.70
0
15
30
45
60
75
90
Epoch
Batch size: 1
Batch size: 10
Batch size: 100

Layer Sizes


0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
Test Accuracy
0
15
30
45
60
75
90
Epoch
Hidden layer sizes: (100)
Hidden layer sizes: (200)
Hidden layer sizes: (300)
Hidden layer sizes: (100,100)
Hidden layer sizes: (200,100)
Hidden layer sizes: (300,100)

Comparison to Other Approaches

- Deep convolutional network using TensorFlow [3].
- Learns localized features of the input.
- Learned features are translation invariant.
- Better for image recognition problems (like MNIST).

Comparison to Other Approaches

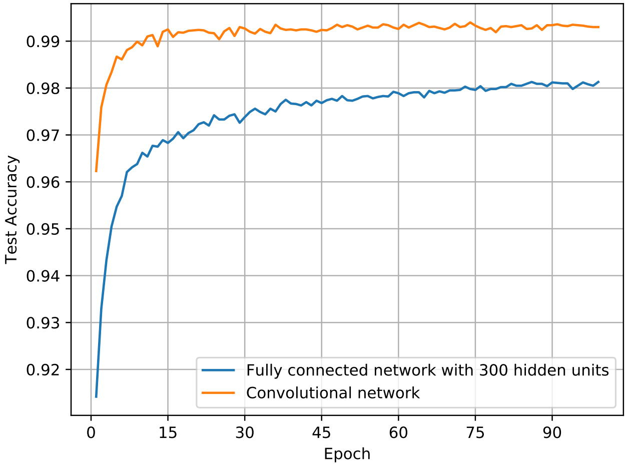


Table of Contents

Demo

- Inspired by the OCHRE applet [4].

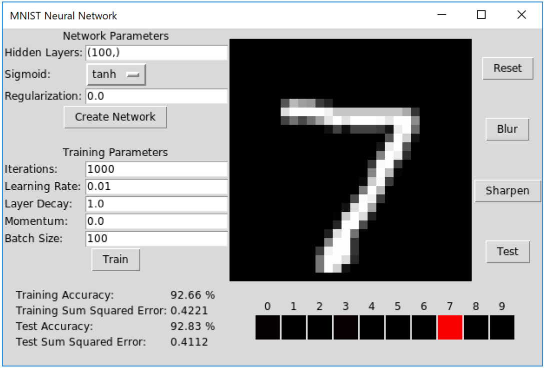


Table of Contents

- Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 9–50, DOI: 10.1007/3-540-49430-8_2. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-49430-8_2
- J. Steeves. (2015, Sept) MNIST Hand Written Digits Classification Benchmark. [Online]. Available: https://knowm.org/mnist-hand-written-digits-classification-benchmark/
- Deep MNIST for Experts. [Online]. Available: https://www.tensorflow.org/get_started/mnist/pros
- OCHRE - Optical Character Recognition. [Online]. Available: http://www.sund.de/netze/applets/BPN/bpn2/ochre.html