Neural Networks Optimized with Matrix Calculation

Abstract

I implemented a neural network optimized with matrix calculation. Though neural networks can be implemented with many 'for' loops, which may be easy to read, the traininig time may become longer because we need at leas 3 'for' loops. When we implement such a network with 30 epochs, 50000 mnist training data, and mini_batch size 10, the network has to implement 150000 loops in tota.

This is quite expensive, so I implemented a neural network which uses two for loops and some matrix calculation. I used matrix calculation on update over mini-batch example. The update shares the same weights and biases in the network, so we can conduct the update simultaneously over the examples in the mini-batch.

$$\begin{aligned}\mathbf{w} &\leftarrow \mathbf{w} - \eta \nabla_w C_{all} \\ &= \mathbf{w} - \eta(\nabla_w C_{0all} + \frac{\lambda}{n}\mathbf{w}) \\ &= (1 - \frac{\eta\lambda}{n})\mathbf{w} - \eta\nabla_w C_{0all}\end{aligned} \tag{1}$$