Deep Learning

MNIST Digits Classification with CNN

Author: Yantian Luo

Institute: Electronic Engineering

ID: 2018310742

Update: April 21, 2019

Contents

Chapter 1 Introduction

MNIST digits dataset is a widely used database for image classification in machine learning field. It contains 60,000 training samples and 10,000 testing samples. Each sample is a 784×1 column vector, which is transformed from an original 28×28 pixels grayscale image.

In this homework, we will continue working on MNIST digits classification problem by utilizing Pytorch framework to implement neural networks. Pytorch provides good abstraction for different modules, and its auto-differentiation feature can save us from the backward details.

Chapter 2 Algorithm Design

In the last two homework, we have implement Linear layer, activation layer, convolution layer and loss layer, in this homework, we only implement an important technique discovered recently and utilize Pytorch framework to implement others.

2.1 Training phase

For every input x_i across a mini-batch, the output can be computed as :

$$BN(x_i) = \gamma \cdot \frac{x_i - \mu_B}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{2.1}$$

where γ and β are learnable parameters, μ_B is the mean of the mini-batch and σ^2 is the variance of the mini-batch.

2.2 Evaluation phase

When we test our samples one by one, the normalization may not work because we don’t have mini- batch concept this time. Therefore, during training we should keep running estimates of its computed mean and variance, which are then used for normalization during evaluation. The running estimates are kept with a default momentum of 0.9, which can be mathematically expressed as

$$\begin{aligned} \hat{\mu}_{new} &= momentum \times \hat{\mu} + (1 - momentum) \times \mu_t \\ \hat{\sigma}^2_{new} &= momentum \times \hat{\sigma}^2 + (1 - momentum) \times \sigma_t^2 \end{aligned} \tag{2.2}$$

where where $\hat{\mu}, \hat{\sigma}^2$ are the estimated statistic and μ_t, σ_t^2 is the new observed value.

2.3 Results

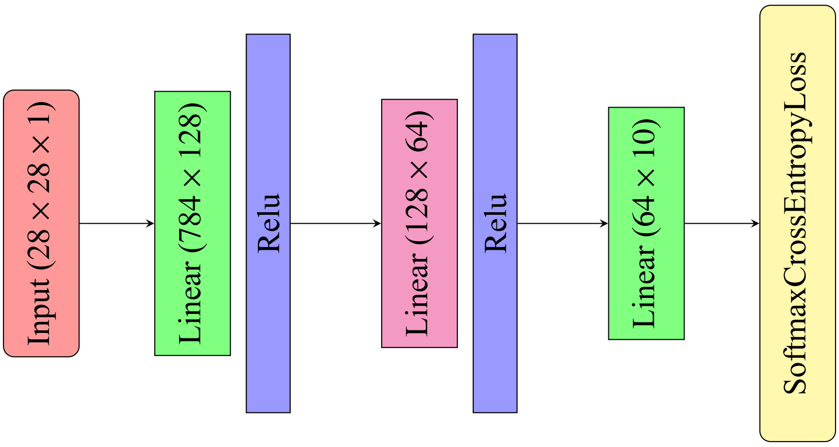

Figure 2.1: MLP Network Structure