Deep Learning
Assignment 1 Report

Tsinghua University

Albert Millan – 2019280366

Contents

1 Introduction **1**

2 Model: Logistic Regression **1**

3 Hyperparameter Selection **2**

4 Results **2**

1 Introduction

This report explains the setup and methodology involved to deploy a logistic regression model on the MNIST dataset.

2 Model: Logistic Regression

The model deployed is known as logistic regression. Given an input image, it computes a probability distribution function outlining the likelihood for each of the possible outcomes. In this experiment, only two outcomes can be observed, either the image denotes number three or number six.

Correct probabilities are inferred by iteratively altering the weight values in order to minimize a cost/error function. Weight values can be adjusted using the gradient of the loss on each iteration. The error function employed is the cross-entropy loss, outlined in equation 1 for a single example, and equation 2 for the total loss.

$$L_i = -\log(P(Y = y_i|X = x_i) = -\log\left(\frac{e^{w_{y\,i} * x_i}}{\sum_j e^{w_j * x_i}}\right) \tag{1}$$

$$L = \frac{1}{N} * \sum L_i + \lambda ||W||_2 \tag{2}$$

To estimate the extend to which to alter the weights, the gradient of the losss w.r.t. the weights is computed on each iteration and added to the weights. Gradient derivation is also outlined. Iteratively repeating this process eventually reduces the loss while fine-tuning the weights.

$$\nabla_{w_{y_i}} L_i = x_i * \frac{e^{w_{y\,i} * x_i}}{\sum_j e^{w_j * x_i}} - x_i$$
$$\nabla_{w_{k \neq y_i}} L_i = x_i * \frac{e^{w_k * x_i}}{\sum_j e^{w_j * x_i}}$$

3 Hyperparameter Selection

Hyperparameter selection should be conducted through an exhaustive search of the potential combinations of parameters over the validation set. Notwithstanding, outstanding results were achieved on the first correct implementation of the model, indicating that the initial estimation was sufficient. The following are the hyperparameters:

Type	**Value**
Batch size	200
Batches	60
Learning rate	0.001
Weight decay	0.0001
Max epoch	100

Table 1: Hyperparameters employed to train the model.

4 Results

The deployed model achieves an accuracy of 98 100% and 98.88% on the training and test datasets respectively. It is able to distinguish the threes from the sixes. A summary of the loss and the accuracy is shown across the iterations in figure 1.

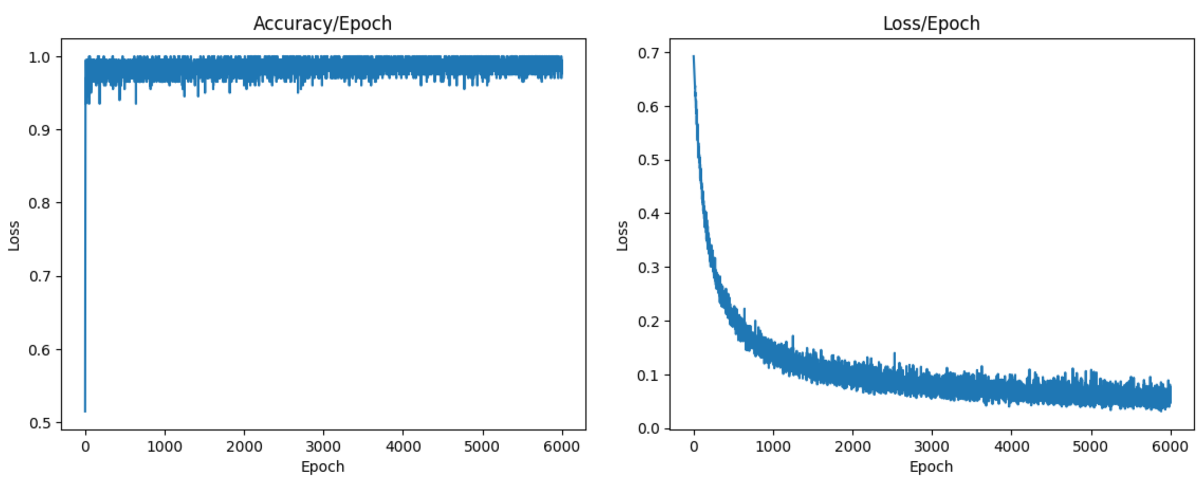

Figure 1: Accuracy and loss over training iteration/epoch.

The models achieves high accuracy with a substantially low volume of training iterations. Precisely, above 95% accuracy is obtained within the first five full iterations (includes batch iterations). The loss, takes longer to converge, slowing the rate of decline significantly after 3000 epochs. To conclude, figure 2 outlines the resulting weight templates for each of the possible outcomes upon completing the training procedure.

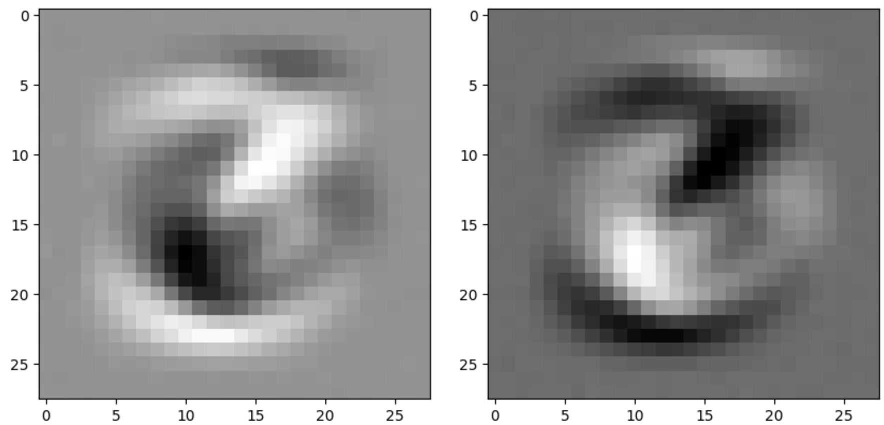

Figure 2: Weight templates for the number three and six respectively.