Deeplearning assingment 1

Benedith Mulongo

April 2019

1 Introduction

In this lab, we use softmax classification for classifying the CIFAR data-set. The CIFAR data-set is a dataset that contains 10 different classes and the softmax classification are used for multi-class classification.

1.1 Softmax Classification

Softmax classification is a multi-class classification. In this lab, a one-layer network is used with softmax classification train with mini-batch gradient descent.

We need the following :

- **W** [KXd] The weight of the classification
- **b** [Kx1] The bias term
- **X** [dXN] The data-set with images
- **Y** [KXN] The one-hot representation of the class for each data image
- **y** [1XN] The class of each corresponding images

1.1.1 Algorithm description

$$s = Wx + b$$

$$p = SOFTMAX(s)$$

The algorithm taken from lecture 3 presented in figure 1. The gradient of the cost function is implemented in Matlab.

- Let $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{n_b}, \mathbf{y}_{n_b})\}$ be the data in the mini-batch $\mathcal{D}^{(t)}$.
  - Gather all $\mathbf{x}_i$'s from the batch into a matrix, similarly for $\mathbf{y}_i$'s

$$\mathbf{X}_{\text{batch}} = \begin{pmatrix} \uparrow & & \uparrow \\ \mathbf{x}_1 & \cdots & \mathbf{x}_{n_b} \\ \downarrow & & \downarrow \end{pmatrix}, \quad \mathbf{Y}_{\text{batch}} = \begin{pmatrix} \uparrow & & \uparrow \\ \mathbf{y}_1 & \cdots & \mathbf{y}_{n_b} \\ \downarrow & & \downarrow \end{pmatrix}$$

  - Complete the **forward pass**

$$\mathbf{P}_{\text{batch}} = \text{SoftMax}\left(W\mathbf{X}_{\text{batch}} + \mathbf{b}\mathbf{1}_{n_b}^T\right)$$

  - Complete the **backward pass**

    1. Set

$$\mathbf{G}_{\text{batch}} = -\left(\mathbf{Y}_{\text{batch}} - \mathbf{P}_{\text{batch}}\right)$$

    2. Then

$$\frac{\partial L}{\partial W} = \frac{1}{n_b}\mathbf{G}_{\text{batch}}\mathbf{X}_{\text{batch}}^T, \quad \frac{\partial L}{\partial \mathbf{b}} = \frac{1}{n_b}\mathbf{G}_{\text{batch}}\mathbf{1}_{n_b}$$

- Add the gradient for the regularization term

$$\frac{\partial J}{\partial W} = \frac{\partial L}{\partial W} + 2\lambda W, \qquad \frac{\partial J}{\partial \mathbf{b}} = \frac{\partial L}{\partial \mathbf{b}}$$

Figure 1: Gradient descent from lecture 3

2 Test

The implementation of the gradient of the cost function is implemented in Matlab and is prone of error. Therefore different test is run in order to check the accuracy of the implementation.

Due to the computation time of running the analytical gradient and the numerical gradient, the test if run for 10 and 100 randomly selected images from the 10000-data.

We use the following formula :

$$\Delta = \frac{|g_a - g_n|}{max(\epsilon, |g_a| + |g_n|)}$$

Every entries of the difference matrix Δ are summed up, the error is the number of entries where the value is $> 1e-6$. The result is presented in the table above.

Gradient check		
Number Images	Number Entries	Error
10	10*3072	2 (0.0065 %)
100	100*3072	11 (0.0036 %)

The table above shows that the difference between the numerical and analytical gradient is almost insignificant for 10 and 100 randomly chosen images from the training data-set due to computation time.

3 Result

3.1 lambda=0, n epochs=40, n batch=100, eta=.1

Train accuracy = 0.2861 Test accuracy = 0.2457

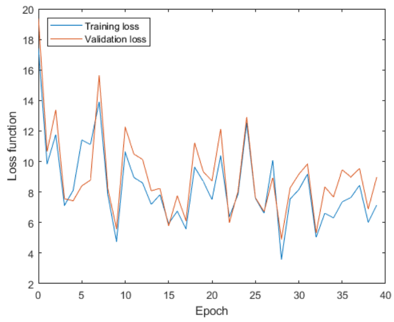

Figure 2: The loss curve

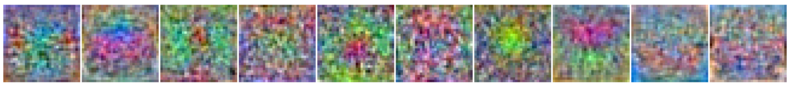

Figure 3: The learned weights

3.2 lambda=0, n epochs=40, n batch=100, eta=.01

Train accuracy = 0.4161 Test accuracy = 0.3665

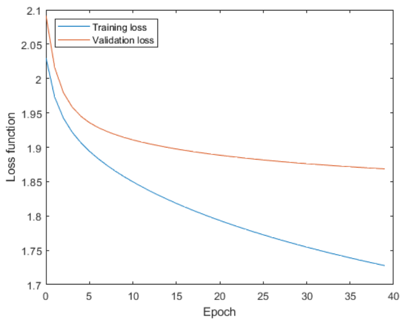

Figure 4: The loss curve

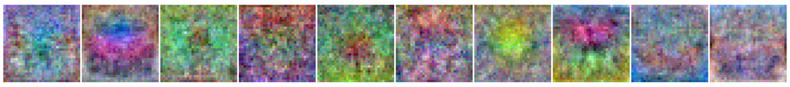

Figure 5: The learned weights

3.3 lambda=.1, n epochs=40, n batch=100, eta=.01

Train accuracy = 0.3095 Test accuracy = 0.3073

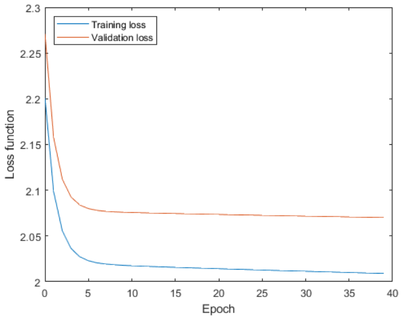

Figure 6: The loss curve

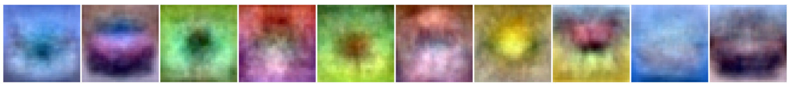

Figure 7: The learned weights

3.4 lambda=1, n epochs=40, n batch=100, eta=.01

Train accuracy = 0.1829 Test accuracy = 0.1822

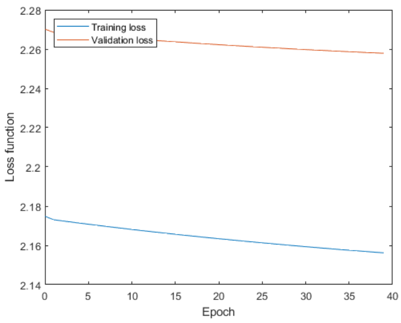

Figure 8: The loss curve

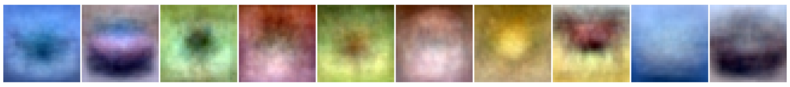

Figure 9: The learned weights

4 Conclusion

The result shows that the softmax classifier can be successfully been used for multi-class classification.

References