DD 2424 - Assignment 1

Jia Fu (jiafu@kth.se)

April 7, 2021

Gradient Check

The following is the function I computed the gradients.

```
function [grad_W, grad_b] = ComputeGradients(X, Y, P, W, lambda)
    n = size(X, 2);
    G = - (Y - P);
    grad_W = G * X' / n + 2 * lambda * W;
    grad_b = G * ones(n, 1) / n;
end
```

Next thing is to do the gradient check, here I computed the relative error between numerically computed gradients g_n and analytically computed gradients g_a:

$$\frac{|g_a - g_n|}{\max(\text{eps}, |g_a| + |g_n|)}$$

Here I set `eps` = 1e-6. The numerically computed gradients were obtained by the given MATLAB function `ComputeGradsNumSlow` which is more accurate than `ComputeGradsNum`. The comparison is based on the max and mean values in the relative error matrix of `W` and `b`. I chose different numbers of training samples and dimensions, also different λ to testify whether my gradient computation function is robust in various parameter settings.

	$N = 1$, $d = 20$, $\lambda = 0$	$N = 100$, $d = 3072$, $\lambda = 0$	$N = 100$, $d = 3072$, $\lambda = 0.1$
Max	7.7147e-07	1.9133e-04	2.8663e-04
Mean	2.2541e-08	4.1434e-08	4.5330e-08

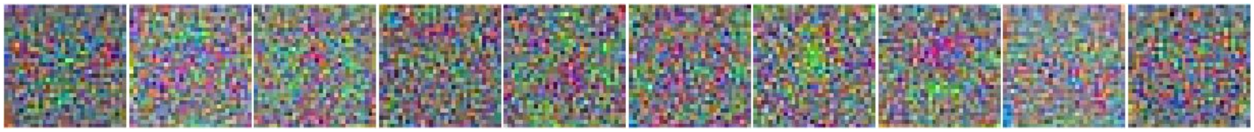
Table 1: Relative Error of `W`

	$N = 1$, $d = 20$, $\lambda = 0$	$N = 100$, $d = 3072$, $\lambda = 0$	$N = 100$, $d = 3072$, $\lambda = 0.1$
Max	1.0514e-09	1.9643e-08	1.9643e-08
Mean	3.7921e-10	5.7345e-09	6.0942e-09

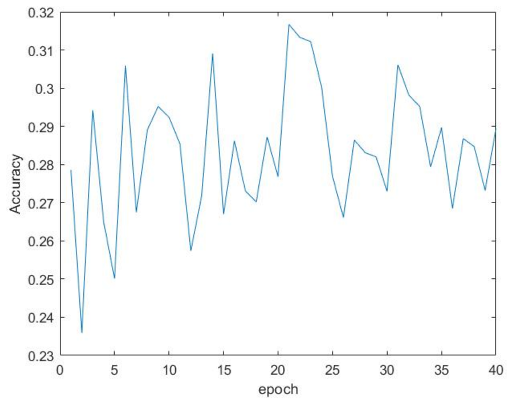
Table 2: Relative Error of `b`

Here it can be seen that the mean relative errors of `W` and `b` are all below 1e-7, indicating that the analytically computed gradients are right for the following steps.

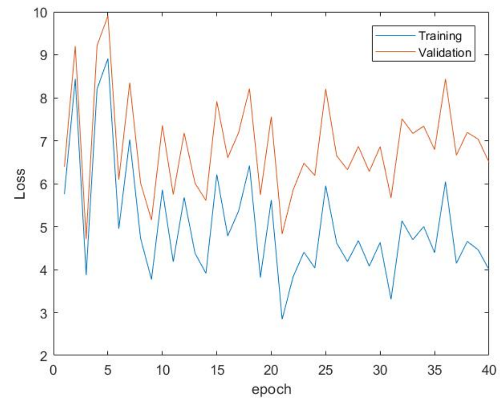
Results



(a) Images representing the learnt `W`

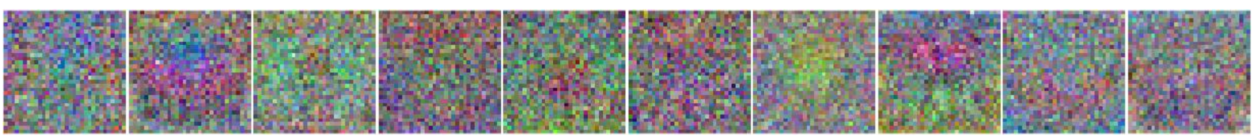


(b) Accuracy

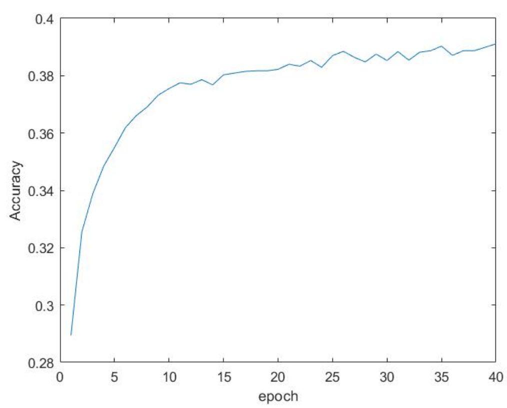


(c) Training and Validation Cost

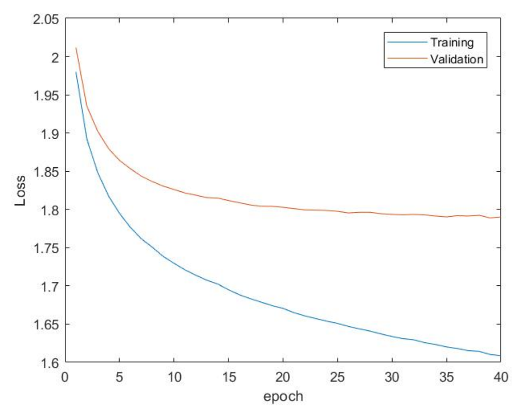
Figure 1: `lambda=0`, `n_epoch=40`, `n_batch=100`, `eta=.1`



(a) Images representing the learnt `W`

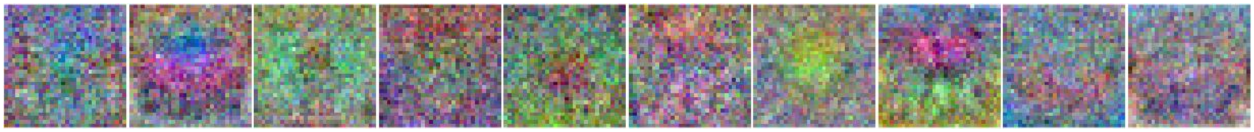


(b) Accuracy

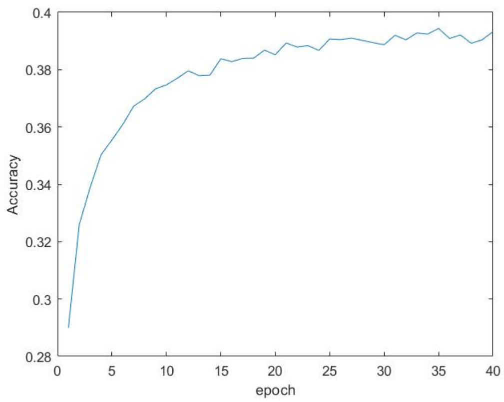


(c) Training and Validation Cost

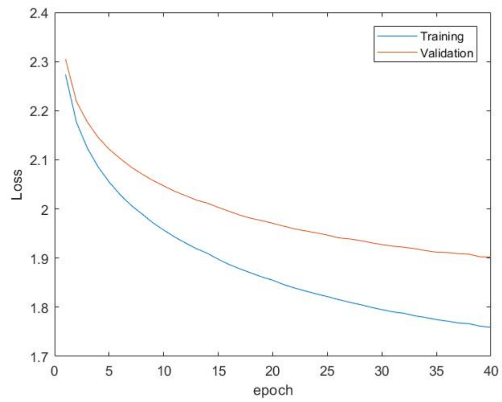
Figure 2: `lambda=0`, `n_epoch=40`, `n_batch=100`, `eta=.001`



(a) Images representing the learnt `W`



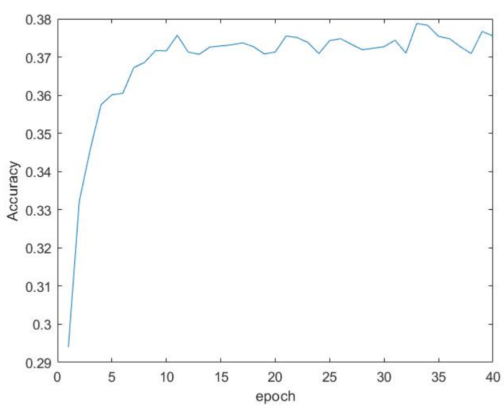
(b) Accuracy



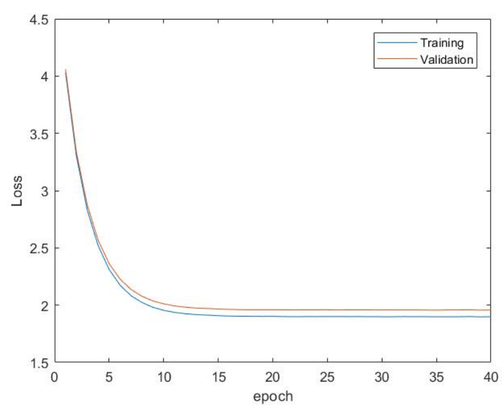
(c) Training and Validation Cost

Figure 3: `lambda=.1`, `n_epoch=40`, `n_batch=100`, `eta=.001`

(a) Images representing the learnt `W`



(b) Accuracy



(c) Training and Validation Cost

Figure 4: `lambda=1`, `n_epoch=40`, `n_batch=100`, `eta=.001`

From above four different parameter settings, `lambda=.1`, `eta=.001` gives the best outcome, the accuracy can reach around 39% after training.

Conclusions

Compare figure 1 with other figures, we can drive the conclusion that if learning rate η is set too high, the `W` matrix will update too much between steps. The accuracy fluctuates between 23% and 32%, the curves of training and validation cost also fluctuate a lot which shows the convergence problem. But this does not mean that the smaller the η, the better the outcome. When η is set too low, the model will need more time to train because of the inefficiency. Thus, an appropriate η is vital for gradient descent algorithm.
Figure 2 and 3 show the influence when the regularization term is introduced. The regularization can help model to accelerate the convergence towards the local minimum. Using regularization is to avoid overfitting and obtain a better generalization. However, λ is also cannot be set too high. Figure 4 shows the rapid convergence of model, which prevents it from improving its performance by the following training steps.