Answers to The Questions

September 29, 2016

1 The Structure of our network

Most time, we use 1-hidden-layer neural network with 300 units. Mathmatically, $y_{train} = W_2 * h(W_1 * a_0)$, where a_0 is the vectorized images and function h is the active function. Here, we choose the ReLu function. i.e.

$$h(x) = \begin{cases} x, & x >= 0 \\ 0, & x < 0 \end{cases} \tag{1}$$

For notations, we set $z_l = W_l * a_{l-1}$ and $a_l = h(z_l)$, where l means the l^{th} layer (in our network, l can only be 1 referring the hidden layer or 2 referring the output layer). We use L to represent the last layer (in our network, $L = 2$).
However, in section 2.2 "Comparison of different network structure", we modified the number of layers and units in each layer. [300] means 1 hidden layer with 300 units. [300,150] means 2 hidden layers and 300 units in the first layer and 150 units in the second layer. [150,300] means 2 hidden layers and 150 units in the first layer and 300 units in the second layer.

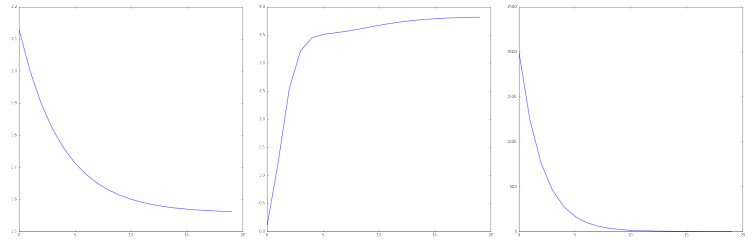
Hyper parameters:
β: the weight before $z_l = W_l * a_{l-1}$ constraint, in .
γ: the weight before $a_l = h(z_l)$ constraint.
λ: the Lagrange multiplier term.
ϵ: a constant that initialize z and a.
In our program, we initialized them with $z_l = \epsilon * randn(sizeof(z_l))$ and $a_l = \epsilon * randn(sizeof(a_l))$. W_l is directly calculated with $W_l = z_l a_{l-1}^{\dagger}$

2 Explaination about ϵ

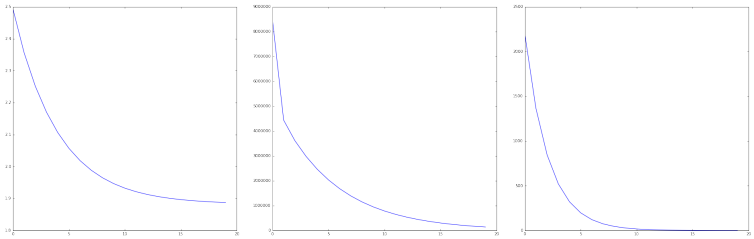
In the paper, they mentioned that a and z should be initialized with normal distribution. We got this initializing method from the open course "CS231n Convolutional Neural Networks for Visual Recognition"of Stanford University. Interestingly, as showed in the next section, when ϵ is 1, all the energies will go down but the accuracy is lower. While ϵ is 0.0001, one of the energies will go up but the accuracy is higher. We found this result yesterday and haven't got any thinking about this.

3 Energy

As our network has 1 hidden-layer, we plotted the energy of $||z_1 - W_1 * a_0||^2$, $||z_2 - W_2 * a_1||^2$ and loss energy which is softmax here. The results are as follows.



(a) loss energy (b) $||z_1 - W_1 * a_0||^2$ (c) $||z_2 - W_2 * a_1||^2$



(d) loss energy (e) $||z_1 - W_1 * a_0||^2$ (f) $||z_2 - W_2 * a_1||^2$

Figure 1: (a), (b), (c) when $\epsilon = 0.0001$ and (d), (e), (f) when $\epsilon = 1$.

4 Update λ

We used the same way indicated in the paper to update λ, i.e. $\lambda \leftarrow \lambda + \beta * (z_L - W_L * a_{L-1})$.As the authors explained in their answers to the review, they thought only the z_L is related to the loss function. We followed their idea, but it's still unclear for us.
Besides, we are also confused about "quadratic decoupling", which was mentioned in the email.

5 Bottleneck

Until now, we haven't spent any time on the efficiency of our program yet, for we want to make sure our implementation is correct first. However, at present we believe the updating for z_L to be the most time consuming part. Therefore, we will try to implement Newton's method for softmax.
The next consuming part should be the matrix inversion in W update. Because of the size of dataset, calculating the inverse or pseudo inverse is difficult even

though we used the least square.