Assignment 3

Guiliang Liu

November 15, 2016

1 Error Backpropagation

1.1 Calculate $\frac{\partial E_n(w)}{\partial a_1^{(4)}}$

For the output layer $h(a) = a$, so here the output function becomes:

$$E_n(w) = \frac{1}{2}(h(a_1^{(4)}) - t_n)^2 = \frac{1}{2}(a_1^{(4)} - t_n)^2$$

so $\frac{\partial E_n(w)}{\partial a_1^{(4)}}$ becomes:

$$\frac{\partial E_n(w)}{\partial a_1^{(4)}} = \frac{1}{2}\frac{\partial(a_1^{(4)} - t_n)^2}{\partial a_1^{(4)}} = (a_1^{(4)} - t_n)$$

1.2 Calculate $\frac{\partial E_n(\omega)}{\partial w_{12}^{(3)}}$

$$\frac{\partial E_n(w)}{\partial w_{12}^{(3)}} = \frac{\partial E_n(w)}{\partial a_1^{(4)}}\frac{\partial a_1^{(4)}}{\partial w_{12}^{(3)}} = (a_1^{(4)} - t_n)\frac{\partial a_1^{(4)}}{\partial w_{12}^{(3)}}$$

here, as

$$\frac{\partial a_1^{(4)}}{\partial w_{12}^{(3)}} = \frac{\partial}{\partial w_{12}^{(3)}}\sum_{k=1}^{3} w_{1k}^{(3)} z_k^{(3)} = z_2^{(3)}$$

we can conclude that:

$$\frac{\partial E_n(w)}{\partial w_{12}^{(3)}} = (a_1^{(4)} - t_n)z_2^{(3)}$$

if $h(a) = \frac{1}{1+e^{-a}}$

$$\frac{\partial E_n(w)}{\partial w_{12}^{(3)}} = (a_1^{(4)} - t_n)\frac{1}{1 + e^{-a_2^{(3)}}}$$

1.3 Write an expression for $\frac{\partial E_n(w)}{\partial a_1^{(3)}}$

$$\frac{\partial E_n(w)}{\partial a_1^{(3)}} = \sum_{k=1}^{1}\frac{\partial E_n(w)}{\partial a_k^{(4)}}\frac{\partial a_k^{(4)}}{\partial a_j^{(3)}}$$

as

$$\sum_{k=1}^{1}\frac{\partial E_n(w)}{\partial a_k^{(4)}} = \delta_1^{(4)}$$

and

$$\frac{\partial a_m^{(4)}}{\partial a_1^{(3)}} = \frac{\partial \sum_{m=1}^{3}\sum_{k=1}^{3} w_{km}^{(3)} z_m^{(3)}}{\partial a_1^{(3)}} = \frac{\partial \sum_{m=1}^{3}\sum_{k=1}^{3} w_{km}^{(3)} h(a_m^{(3)})}{\partial a_1^{(3)}}$$

$$= h'(a_1^{(3)})\sum_{k=1}^{1} w_{k1}^{(3)} = h'(a_1^{(3)})w_{11}^{(3)}$$

so

$$\frac{\partial E_n(w)}{\partial a_1^{(3)}} = h'(a_1^{(3)}) \sum_k w_{k1}^{(3)} \delta_k^{(4)} = h'(a_1^{(3)}) w_{11}^{(3)} \delta_1^{(4)}$$

if $h(a) = \frac{1}{1+e^{-a}}$

$$\frac{\partial E_n(w)}{\partial a_1^{(3)}} = \sigma(a_1^{(3)})(1 - \sigma(a_1^{(3)}) w_{11}^{(3)} \delta_1^{(4)}$$

1.4 Use this result to calculate $\frac{\partial E_n(w)}{\partial w_{11}^{(2)}}$

$$\frac{\partial E_n(w)}{\partial w_{11}^{(2)}} = \delta_1^{(3)} \frac{\partial a_1^{(3)}}{w_{11}^{(2)}}$$

similarly,

$$\frac{\partial a_1^{(3)}}{w_{11}^{(2)}} = z_1^{(2)}$$

and similarly,

$$\delta_1^{(3)} = \frac{\partial E_n(w)}{\partial a_1^{(3)}} = h'(a_1^{(3)}) w_{11}^{(3)} \delta_1^{(4)}$$

so,

$$\frac{\partial E_n(w)}{\partial w_{11}^{(2)}} = h'(a_1^{(3)}) w_{11}^{(4)} \delta_1^{(4)} z_1^{(2)}$$

if $h(a) = \frac{1}{1+e^{-a}}$

$$\frac{\partial E_n(w)}{\partial w_{11}^{(2)}} = \sigma(a_1^{(3)})(1 - \sigma(a_1^{(3)}) w_{11}^{(3)} \delta_1^{(4)} z_1^{(2)}$$

1.5 Write an expression for $\frac{\partial E_n(w)}{\partial a_1^{(2)}}$

$$\frac{\partial E_n(w)}{\partial a_1^{(2)}} = \sum_{k=1}^{3} \frac{\partial E_n(w)}{\partial a_k^{(3)}} \frac{\partial a_k^{(3)}}{\partial a_j^{(2)}}$$

as

$$\sum_{k=1}^{3} \frac{\partial E_n(w)}{\partial a_k^{(3)}} = \sum_{k=1}^{3} \delta_k^{(3)}$$

similarly, as 1.3, we could write:

$$\frac{\partial a_m^{(3)}}{\partial a_1^{(2)}} = \frac{\partial \sum_{m=1}^{3} \sum_{k=1}^{3} w_{km}^{(2)} z_m^{(2)}}{\partial a_1^{(2)}} = \frac{\partial \sum_{m=1}^{3} \sum_{k=1}^{3} w_{km}^{(2)} h(a_m^{(2)})}{\partial a_1^{(2)}}$$

$$= h'(a_1^{(2)}) \sum_{k=1}^{3} w_{k1}^{(2)}$$

so

$$\frac{\partial E_n(w)}{\partial a_1^{(2)}} = h'(a_1^{(2)}) \sum_{k=1}^{3} w_{k1}^{(2)} \delta_k^{(3)}$$

if $h(a) = \frac{1}{1+e^{-a}}$

$$\frac{\partial E_n(w)}{\partial a_1^{(2)}} = \sigma(a_1^{(2)})(1 - \sigma(a_1^{(2)}) \sum_{k=1}^{3} w_{k1}^{(2)} \delta_k^{(3)}$$

1.6 Use this result to calculate $\frac{\partial E_n(w)}{\partial w_{11}^{(1)}}$

$$\frac{\partial E_n(w)}{\partial w_{11}^{(1)}} = \delta_1^{(2)} \frac{\partial a_1^{(2)}}{w_{11}^{(1)}}$$

similarly,

$$\frac{\partial a_1^{(2)}}{w_{11}^{(1)}} = z_1^{(1)}$$

and the same as 1.4,

$$\delta_1^{(2)} = \frac{\partial E_n(w)}{\partial a_1^{(2)}} = h'(a_1^{(2)}) \sum_{k=1}^{3} w_{k1}^{(2)} \delta_k^{(3)}$$

so,

$$\frac{\partial E_n(w)}{\partial w_{11}^{(1)}} = h'(a_1^{(2)}) \sum_{k=1}^{3} w_{k1}^{(2)} \delta_k^{(3)} z_1^{(1)}$$

if $h(a) = \frac{1}{1+e^{-a}}$

$$\frac{\partial E_n(w)}{\partial w_{11}^{(1)}} = \sigma(a_1^{(2)})(1 - \sigma(a_1^{(2)}) \sum_{k=1}^{3} w_{k1}^{(2)} \delta_k^{(3)} x_1^{(1)}$$

2 Vanishing Gradients

2.1 Write an expression for $\frac{\partial E_n(w)}{\partial w_{11}^{(l)}}$ for all layers l in the network.

as it is proven in the above problem

$$\frac{\partial E_n(w)}{\partial w_{11}^{(152)}} = \frac{\partial E_n(w)}{\partial a_1^{(153)}} \frac{\partial a_1^{(153)}}{\partial w_{11}^{(152)}} = (a_1^{(153)} - t_n) \frac{\partial a_1^{(153)}}{\partial w_{11}^{(152)}} = (a_1^{(153)} - t_n) z_1^{(152)}$$

as we mentioned in the problem above, so

if $(n \geq 153)$ and $(l \geq 152)$

$$\frac{\partial E_n(w)}{\partial w_{11}^{(l)}} = (a_1^{(153)} - t_n)$$

else **(noted when n =152, l = 151, l = n-1)**

$$\frac{\partial E_n(w)}{\partial w_{11}^{(l)}} = (a_1^{(153)} - t_n) \prod_{n=l+1}^{153} h'(a_1^{(n)}) w_{11}^{(n)} z_1^{l}$$

2.2 Describe what would happen to the gradient for weights early in the network $\frac{\partial E_n(w)}{\partial w_{11}^{(l)}}$ for smaller l

when some output of the sigmoid function goes to 0 or 1, the gradient decent generated by them will tend to be 0.$(w_{11}^{(L)} h'(a_1^{(l)})$ will become close to 0). as for the layers earlier than it (whose gradient decent is generated from it), their gradient decent will tend to be 0 too.
So the gradient for weights early in the network tends to be very small, the "Vanishing Gradients" problem will happen in the "front" layers in an n-layer network, but in the "back" layers in the network, where the $h'(a_1^{(l)})$ is not zero, the gradients will be reasonable in magnitude.

2.3 Suppose we use rectified linear units (ReLU) in activation functions. When would the gradients be zero?

The function of ReLU is defined as $h(a_j) = \max(0, a_j)$ When (for some $x > L$) its input $a_j^{(l)}$ is smaller or equal to 0, it will goes to 0, so the corresponding gradients will become 0.
but the ReLU will make the $h(a_j)' = 1$ when its input is bigger than 0. It could help to solve the "Vanishing Gradients" problem. But can not completely solve it.

2.4 Suppose we modify the graph to have bipartite connections at each layer, still using ReLU activation functions. When would the gradients be zero?

When we modify the graph to have bipartite connections, $\frac{\partial E_n(w)}{\partial w_{11}^{(l)}}$ will become:

$$\frac{\partial E_n(w)}{\partial w_{11}^{(l)}} = \sigma(a_1^{(l+1)})(1 - \sigma(a_1^{(l+1)})\sum_{k=1}^{2} w_{k1}^{(l+1)}\delta_k^{(l+2)}z_1^{(l)}$$

For the case of bipartite graph, for the layer earlier than the $l+1$ layer, we need to consider both two nodes. So their gradient will become zero when $h'(a_1^{(l+1)})$ and $h'(a_2^{(l+1)})$ goes zero(sum of them, which is $\frac{\partial E_n(w)}{\partial w_{11}^{(l)}}$ goes to 0).

3 Fine-Tuning a Pre-Trained Network

3.1 Question 1:

question: The current code does not train the layers in Inception V3. After training the new layers for a while (until good values have been obtained), turn on training for the Inception V3 layers to see if better performance can be achieved.
solution: for the normal version of code, where the the layers in Inception V3 has not been trained, it performance is present as follow:

Epoch	train loss	train accuracy	validate loss	validate accuracy
Epoch 1	1.1317	0.4062	1.2732	0.3125
Epoch 2	0.9963	0.5000	1.2741	0.2917
Epoch 3	1.1019	0.4688	1.2862	0.3021
Epoch 4	0.9927	0.5625	1.2741	0.3438
Epoch 5	0.9829	0.5000	1.3897	0.2708

After we turn on training for the Inception V3 layers, the performance become:

Epoch	train loss	train accuracy	validate loss	validate accuracy
Epoch 1	1.1679	0.4375	1.3524	0.3125
Epoch 2	1.2176	0.3125	1.1783	0.3646
Epoch 3	1.0265	0.5312	1.0440	0.4375
Epoch 4	1.1536	0.5312	1.1133	0.4062
Epoch 5	0.7945	0.5938	0.8825	0.5938

Without the training on the Inception V3 layers, we could observe in the last epoch, the validation accuracy become nearly 0.27(event lower than the random guess). Comparably, provide we add the training on that layer, the validation accuracy become nearly 0.6, the performance become better.

3.2 Question 2:

question: Write a Python function to be used at the end of training that generates HTML output showing each test image and its classification scores. You could produce an HTML table output for example.
solution: as the validation accuracy of the original network is very low(around 30 percent, event lower than the random guess), so I modify the parameters of the network and set the input training example to 200 and add the training on the base model, so the training accuracy become nearly 80 percent and the validation accuracy become nearly 75 percent. describe as follow:

Epoch	train loss	train accuracy	validate loss	validate accuracy
Epoch 1	0.9930	0.5089	0.7100	0.7396
Epoch 2	0.5747	0.8036	0.4352	0.8750
Epoch 3	0.5107	0.7991	0.4607	0.8333
Epoch 4	0.5092	0.8170	0.4027	0.8333
Epoch 5	0.4633	0.8304	0.2946	0.8958

form the table above we could observe the increasing of accuracy of both validation and training as the epoch increase. And the validation rate becomes nearly 90 percent in the epoch 5. However, the training time become longer and we need more than 15 min to train(origin network only requires 5 min to train in the same laptop). the HTML table is represent int **the next page**(the HTML file is in the code.zip).