MLP Framework Mathamatics Support

April 6, 2019

Helper Functions

Relu Function

$$f(x) = max(0, x)$$
$$f'(x) = \begin{cases} 1, & \text{if } x > 0. \\ 0, & \text{otherwise.} \end{cases}$$

Softmax Function

$$f(x_j) = \frac{e^{-x_j}}{\sum_i e^{-x_i}}$$
$$f'(x_j) = f(x_j)(1 - f(x_j))$$

Linear Forward Activation Module

The linear forward module (vectorized over all the examples) computes the following equations:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l-1]}$$

where $A^{[0]} = X$. After linear forward part, we should apply activation function to it. Utilizing relu in hidden layers and softmax in output layer.

Cross Entropy Error Function

We need to know the derivative of loss function to back propagate. Our cost function is multi-class cross entropy function. Its defintion follows,

$$E = -\sum_{i=\#class} c_i log(A_i^L)$$

Where c_i is the lable for class i, and A_i^L is the prediction of probability belonging to class i.

Notice that we would apply softmax to calculated neural network scores(Z^L) and predict out probabilities first. Cross entropy is applied to softmax applied probabilities and one hot encoded

classes calculated second. That's why, we need to calculate the derivative of total error with respect to the each score.

We apply chain rule to calculate the derivative. Calculating it step by step, for a specific score with index i,

$$\frac{\partial E}{\partial Z_i^L} = \sum_j (\frac{\partial E}{\partial A_j^L})(\frac{\partial A_j^L}{\partial Z_i^L}) = (\frac{\partial E}{\partial A_i^L})(\frac{\partial A_i^L}{\partial Z_i^L})$$

Considering about $\frac{\partial E}{\partial A_i^L}$ first,

$$\begin{aligned}\frac{\partial E}{\partial A_i^L} &= \frac{\partial (E = -\sum_{i=\#class} c_i log(A_i^L))}{\partial A_i^L} \\ &= \frac{\partial (-c_i log(A_i^L))}{\partial A_i^L} \\ &= -\frac{c_i}{A_i^L}\end{aligned}$$

We have known that the partial derivative of $\frac{\partial A_i^L}{\partial Z_i^L}$ is that,

$$\frac{\partial A_i^L}{\partial Z_i^L} = A_i^L(1 - A_i^L)$$

Thus, we now can calculate out $\frac{\partial E}{\partial Z_i^L}$,

$$\begin{aligned}\frac{\partial E}{\partial Z_i^L} &= (\frac{\partial E}{\partial A_i^L})(\frac{\partial A_i^L}{\partial Z_i^L}) \\ &= -\frac{c_i A_i^L (1 - A_i^L)}{A_i^L} \\ &= -c_i(1 - A_i^L) \\ \frac{\partial E}{\partial Z^L} &= -c \odot (1 - A^L)\end{aligned}$$

Backpropagation Algorithm

Backpropagation is about understanding how changing the weights and biases in a network changes the cost function. Ultimately, this means computing the partial derivatives $\frac{\partial C}{\partial W_{jk}^l}$ and $\frac{\partial C}{\partial b_j^l}$, where W_{jk}^l denotes the weight connecting between k_{th} neuron in the $l-1_{th}$ layer and j_{th} in the l_{th} layer. Having a revisiting of we have said in Linear activation forward, for every layer $l = 1, ..., L$,

$$A^l = \sigma(W^l A^{l-1} + b^l)$$

where σ is the activation function, it can be $softmax$ or $relu$.
Since we aleady have output error,

$$\frac{\partial E}{\partial Z_i^L} = -c_i(1 - A_i^L)$$
$$\frac{\partial E}{\partial Z^L} = -c \odot (1 - A^L)$$

where $\odot$ is element wise multiply.
Now considering about the error $\frac{\partial E}{\partial Z^l}$ in terms of the error in the next layer, $\frac{\partial E}{\partial Z^{l+1}}$. We can do this using the chain rule,

$$\frac{\partial E}{\partial Z_j^l} = \sum_k \frac{\partial E}{\partial Z_k^{l+1}} \frac{\partial Z_k^{l+1}}{\partial Z_j^l}$$

To evaluate the first term on the last line, note that,

$$Z_k^{l+1} = \sum_j W_{kj}^{l+1} A_j^l + b_k^{l+1} = \sum_j W_{kj}^{l+1} \sigma(Z_j^l) + b_k^{l+1}$$

Differentiating, we obtain,

$$\frac{\partial Z_k^{l+1}}{\partial Z_j^l} = W_{kj}^{l+1} \sigma'(Z_j^l)$$

Substituting back,

$$\frac{\partial E}{\partial Z_j^l} = \sum_k W_{kj}^{l+1} \frac{\partial E}{\partial Z_k^{l+1}} \sigma'(Z_j^l)$$
$$\frac{\partial E}{\partial Z^l} = ((W^{l+1})^T) \frac{\partial E}{\partial Z_k^{l+1}}) \odot \sigma'(Z^l)$$

Next part we will calculate $\frac{\partial C}{\partial W_{jk}^l}$ and $\frac{\partial C}{\partial b_j^l}$,

$$\frac{\partial E}{\partial b_j^l} = \sum_k \frac{\partial E}{\partial Z_k^l} \frac{\partial Z_k^l}{\partial b_j^l} = \frac{\partial E}{\partial Z_j^l}$$
$$\frac{\partial E}{\partial b^l} = \frac{\partial E}{\partial Z^l}$$

$$\frac{\partial E}{\partial W_{jk}^l} = \sum_i \frac{\partial E}{\partial Z_i^l} \frac{\partial Z_i^l}{\partial W_{jk}^l} = \frac{\partial E}{\partial Z_j^l} A_k^{l-1}$$
$$\frac{\partial E}{\partial W^l} = \frac{\partial E}{\partial Z^l} (A^{l-1})^T$$

Summary: the equations of backpropagation

(i) $\frac{\partial E}{\partial Z^L} = -c \odot (1 - A^L)$

(ii) $\frac{\partial E}{\partial Z^l} = ((W^{l+1})^T) \frac{\partial E}{\partial Z_k^{l+1}}) \odot \sigma'(Z^l)$

(iii) $\frac{\partial E}{\partial b^l} = \frac{\partial E}{\partial Z^l}$

(iv) $\frac{\partial E}{\partial W^l} = \frac{\partial E}{\partial Z^l} (A^{l-1})^T$