Gradients MLP

Steve Gutfreund

October 2018

1 Our model

Let's consider our samples (x_i, y_i) for i $\in \{1, ..., m\}$

h = tanh(Wx + b)
$\hat{y}$ = softmax(Uh + $\hat{b}$)

$$[\ \hat{y}[i] = softmax(Uh + \hat{b})[i] = \frac{e^{U_i h + b_i}}{\sum\limits_k e^{U_k h + b}}]$$

L = -$\sum\limits_k y_{[k]} log \hat{y}_{[k]}$

2 Gradients

We have to calculate:
$\frac{dL}{dU}, \frac{dL}{db}, \frac{dL}{dW}, \frac{dL}{d\hat{b}}$

We use SGD so let's focus on sample i
Let's denote: $z_i = Uh_i + \hat{b}$

$\frac{dL}{dU} = \frac{dL}{dz_i} \cdot \frac{dz_i}{dU}$
$\frac{dL}{d\hat{b}} = \frac{dL}{dz_i} \cdot \frac{dz_i}{d\hat{b}}$
$\frac{dL}{dW} = \frac{dL}{dz_i} \cdot \frac{dz_i}{dh} \cdot \frac{dh}{dW}$
$\frac{dL}{db} = \frac{dL}{dz_i} \cdot \frac{dz_i}{dh} \cdot \frac{dh}{db}$

[1] $\frac{dL}{d\hat{y}} = -\sum\limits_k \frac{y_{[k]}}{\hat{y}_{[k]}}$, $\frac{d\hat{y}}{dz_i}_{[k]} = y_{[k]}(1\{k = y_i\} - y_i)$

$\Rightarrow \frac{dL}{dz_i} = \hat{y}_i - y_i$

$\frac{dz_i}{dU} = h_i$, $\frac{dz_i}{d\hat{b}} = 1$, $\frac{dz_i}{dh} = U$

[1] It's nicely explained at this site:
http://lyy1994.github.io/machine-learning/2016/05/11/softmax-cross-entropy-derivative.html

$\frac{dh}{dW} = (1 - tanh^2(Wx_i + b)) \cdot x_i$
$\frac{dh}{db} = (1 - tanh^2(Wx_i + b))$

3 Conclusion

$\frac{dL}{dU} = (\hat{y}_i - y_i) \cdot h_i$
$\frac{dL}{d\hat{b}} = (\hat{y}_i - y_i)$
$\frac{dL}{dW} = (\hat{y}_i - y_i) \cdot U \cdot (1 - tanh^2(Wx_i + b)) \cdot x_i$
$\frac{dL}{db} = (\hat{y}_i - y_i) \cdot U \cdot (1 - tanh^2(Wx_i + b))$

4 Generalization

$x^{(2)} = g^{(1)}(W^{(1)}x^{(1)} + b^{(1)})$

$x^{(3)} = g^{(2)}(W^{(2)}x^{(2)} + b^{(2)})$

$x^{(4)} = g^{(3)}(W^{(3)}x^{(3)} + b^{(3)})$
$\cdots$
$\hat{y} = g^{(n)}(W^{(n)}x^{(n)} + b^{(n)})$

$\frac{dL}{db^{(i)}} = (\hat{y}_i - y_i) \cdot \prod_{j=n-1}^{i} [W^{(j+1)} \cdot g^{(j)'}(W^{(j)}x^{(j)} + b^{(j)})]$

$\frac{dL}{dW^{(i)}} = (\hat{y}_i - y_i) \cdot \prod_{j=n-1}^{i} [W^{(j+1)} \cdot g^{(j)'}(W^{(j)}x^{(j)} + b^{(j)})] \cdot x^{(i)}$

OR (for programming purposes)

<u>i=n</u>

$\frac{dL}{db^{(n)}} = (\hat{y}_i - y_i)$

$\frac{dL}{dW^{(n)}} = (\hat{y}_i - y_i) \cdot x^{(n)}$

<u>$\forall i < n$</u>

$\frac{dL}{db^{(i)}} = (\hat{y}_i - y_i) \cdot W^{(n)} \cdot \prod_{j=n-1}^{i+1} [g^{(j)'}(W^{(j)}x^{(j)} + b^{(j)}) \cdot W^{(j)}]$

$\frac{dL}{dW^{(i)}} = (\hat{y}_i - y_i) \cdot W^{(n)} \cdot \prod_{j=n-1}^{i+1} [g^{(j)'}(W^{(j)}x^{(j)} + b^{(j)}) \cdot W^{(j)}] \cdot x^{(i)}$

The red part is an expression which is being calculated by the layers before and pushed up the back-propagation process, s.t. every layer i (**after** calculating its gradients) must multiply the red part by $g^{(i)'}(W^{(i)}x^{(i)} + b^{(i)})$ and pass it on to the next layer.