

Gradient of Loss w.r.t Output Layer Weights

October 2020

Let $\mathbf{y}, \hat{\mathbf{y}} \in \mathbb{R}^{m \times 1}$, be true and predicted values for m examples.
Further

$$\hat{\mathbf{y}} = \mathbf{a} * W_{out} + \mathbf{b}^T \quad (1)$$

where $W_{out} \in \mathbb{R}^{u \times 1}$, $\mathbf{b} \in \mathbb{R}^{1 \times 1}$ are weights and biases of the output layer.
 $\mathbf{a} \in \mathbb{R}^{m \times u}$ be the activation at the last hidden layer.

MSE loss can be written as

$$\mathbf{L} = \frac{1}{m}(\mathbf{y} - \hat{\mathbf{y}})^2 \quad (2)$$

$$\mathbf{L} = \frac{1}{m}(\mathbf{y} - (\mathbf{a}W_{out} + \mathbf{b}^T))^2 \quad (3)$$

Note that $\mathbf{L} \in \mathbb{R}^{m \times 1}$. To write gradient of \mathbf{L} with respect to W_{out} , let us consider a simple case in which we compute gradient of i -th entry of \mathbf{L} (i.e. i -th example) with respect to W_{out} .

Now, for i -th example, let y be true value and \hat{y} be predicted value. Note that both y and \hat{y} are scalars. The loss is (ignore $\frac{1}{m}$ for the moment)

$$l = (y - \hat{y})^2 \quad (4)$$

$$l = (y - (\mathbf{a}W_{out} + \mathbf{b}^T))^2 \quad (5)$$

And gradient can be written as

$$\frac{\partial l}{\partial W_{out}} = 2(y - \mathbf{a}W_{out} - \mathbf{b}^T) \frac{\partial (y - \mathbf{a}W_{out} - \mathbf{b}^T)}{\partial W_{out}} \quad (6)$$

$$= 2(y - \mathbf{a}W_{out} - \mathbf{b}^T) * (-\mathbf{a}) \quad (7)$$

$$= 2(\hat{y} - y) * \mathbf{a} \quad (8)$$

To compute the gradient of all examples at the same time, suppose $m = 5$ and $u = 4$. Then we can write the vector W_{out} as

$$W_{out} = [w_1, w_2, w_3, w_4]^T \quad (9)$$

And we can write the expression $\frac{\partial l}{\partial W_{out}}$ as

$$\frac{\partial l}{\partial W_{out}} = \begin{bmatrix} \frac{\partial l}{\partial w_1} \\ \frac{\partial l}{\partial w_2} \\ \frac{\partial l}{\partial w_3} \\ \frac{\partial l}{\partial w_4} \end{bmatrix} = 2(\hat{y} - y) \begin{bmatrix} a_{i,1} \\ a_{i,2} \\ a_{i,3} \\ a_{i,4} \end{bmatrix} = 2(\hat{y} - y)\mathbf{a} \quad (10)$$

Hence the gradient for all m examples can be obtained by stacking $\frac{\partial l}{\partial W_{out}}$ for all examples:

$$\frac{\partial \mathbf{L}}{\partial W_{out}} = \begin{bmatrix} \frac{\partial \mathbf{L}_1}{\partial W_{out}} \\ \frac{\partial \mathbf{L}_2}{\partial W_{out}} \\ \frac{\partial \mathbf{L}_3}{\partial W_{out}} \\ \frac{\partial \mathbf{L}_4}{\partial W_{out}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{L}_1}{\partial w_1} & \frac{\partial \mathbf{L}_1}{\partial w_2} & \frac{\partial \mathbf{L}_1}{\partial w_3} & \frac{\partial \mathbf{L}_1}{\partial w_4} \\ \frac{\partial \mathbf{L}_2}{\partial w_1} & \frac{\partial \mathbf{L}_2}{\partial w_2} & \frac{\partial \mathbf{L}_2}{\partial w_3} & \frac{\partial \mathbf{L}_2}{\partial w_4} \\ \frac{\partial \mathbf{L}_3}{\partial w_1} & \frac{\partial \mathbf{L}_3}{\partial w_2} & \frac{\partial \mathbf{L}_3}{\partial w_3} & \frac{\partial \mathbf{L}_3}{\partial w_4} \\ \frac{\partial \mathbf{L}_4}{\partial w_1} & \frac{\partial \mathbf{L}_4}{\partial w_2} & \frac{\partial \mathbf{L}_4}{\partial w_3} & \frac{\partial \mathbf{L}_4}{\partial w_4} \\ \frac{\partial \mathbf{L}_5}{\partial w_1} & \frac{\partial \mathbf{L}_5}{\partial w_2} & \frac{\partial \mathbf{L}_5}{\partial w_3} & \frac{\partial \mathbf{L}_5}{\partial w_4} \end{bmatrix}_{5 \times 4} = 2(\hat{\mathbf{y}} - \mathbf{y}) * \mathbf{a} \quad (11)$$

Note that we have compactly written the expression as $2(\hat{\mathbf{y}} - \mathbf{y}) * \mathbf{a}$. Also, the multiplication $(\hat{\mathbf{y}} - \mathbf{y}) * \mathbf{a}$ is supported by numpy-broadcast.

Finally, take average of gradient vectors of all examples to get the gradient that you can use to update the weights of the output layer as follows:

$$W_{out} = W_{out} - \frac{1}{m} * \eta * \text{Sum}\left(\frac{\partial \mathbf{L}}{\partial W_{out}}, \text{axis}=0\right) \quad (12)$$

where axis=0 denotes sum along rows.