S&DS 355 / 555

Introductory Machine Learning

Some Notes on Backpropagation

If $\mathcal{L}$ is a scalar function and $A = BC$, then

$$\frac{\partial \mathcal{L}}{\partial B} = \frac{\partial \mathcal{L}}{\partial A} C^T \tag{1}$$

$$\frac{\partial \mathcal{L}}{\partial C} = B^T \frac{\partial \mathcal{L}}{\partial A}. \tag{2}$$

This can be shown directly using the "usual" chain rule. You should check that the dimensions match up!

The function $\mathcal{L}$ is our loss function. The use of this is called "backpropagation" because we start with the derivatives in the last layer of the network, and recursively send these "back" to compute the derivatives in the earlier part of the network.

So, if $f = Wx + b$ then

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}}{\partial f} x^T. \tag{3}$$

If the loss $\mathcal{L}$ is squared error $\mathcal{L} = \frac{1}{2}(y - f)^2$ then

$$\frac{\partial \mathcal{L}}{\partial f} = (f - y). \tag{4}$$

Now, if

$$f = W_2 h + b_2 \tag{5}$$

$$h = W_1 x + b_1 \tag{6}$$

then

$$\frac{\partial \mathcal{L}}{\partial W_2} = \frac{\partial \mathcal{L}}{\partial f} h^T = (f - y) h^T \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial h} = W_2^T \frac{\partial \mathcal{L}}{\partial f} = W_2^T (f - y) \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial b_2} = \frac{\partial \mathcal{L}}{\partial f} = (f - y). \tag{9}$$

Then, we have that

$$\frac{\partial \mathcal{L}}{\partial W_1} = \frac{\partial \mathcal{L}}{\partial h} x^T = W_2^T \frac{\partial \mathcal{L}}{\partial f} x^T = W_2^T (f - y) x^T \tag{10}$$

$$\frac{\partial \mathcal{L}}{\partial b_1} = \frac{\partial \mathcal{L}}{\partial h} = W_2^T (f - y). \tag{11}$$

This is just another linear model, so it will train to be equivalent to least squares regression. But if we add a ReLU nonlinearity, we get the two-layer neural network

$$h = \text{ReLU}(W_1 x + b_1) \tag{12}$$

$$f = W_2 h + b_2. \tag{13}$$

Let the loss for an example (x, y) be $\mathcal{L} = \frac{1}{2}(y - f(x))^2$. From the above calculations we then have for the second layer

$$\frac{\partial \mathcal{L}}{\partial W_2} = \frac{\partial \mathcal{L}}{\partial f} h^T \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial h} = W_2^T \frac{\partial \mathcal{L}}{\partial f} \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial b_2} = \frac{\partial \mathcal{L}}{\partial f}. \tag{16}$$

For the hidden layer we have

$$\frac{\partial \mathcal{L}}{\partial W_1} = \mathbb{1}(h > 0) \frac{\partial \mathcal{L}}{\partial h} x^T \tag{17}$$

$$= \mathbb{1}(h > 0) W_2^T \frac{\partial \mathcal{L}}{\partial f} x^T \tag{18}$$

$$\frac{\partial \mathcal{L}}{\partial b_2} = \mathbb{1}(h > 0) \frac{\partial \mathcal{L}}{\partial h} \tag{19}$$

$$= \mathbb{1}(h > 0) W_2^T \frac{\partial \mathcal{L}}{\partial f}. \tag{20}$$

For a two-layer network *for classification* we have

$$h = \text{ReLU}(W_1 x + b_1) \tag{21}$$
$$f = W_2 h_1 + b_2 \tag{22}$$
$$p = \text{Softmax}(f) \tag{23}$$

where again W_j and b_j are matrices or vectors of the appropriate dimensions. Let the loss for an example (x, y) be the log-loss $\mathcal{L} = -\log p(y \,|\, x)$. Then

$$\frac{\partial \mathcal{L}}{\partial f} = \begin{pmatrix} p_1 - \mathbb{1}(y = 1) \\ p_2 - \mathbb{1}(y = 2) \\ p_3 - \mathbb{1}(y = 3) \end{pmatrix} \in \mathbb{R}^3 \tag{24}$$

Backpropagating to the second layer we have

$$\frac{\partial \mathcal{L}}{\partial W_2} = \frac{\partial \mathcal{L}}{\partial f} h_1^T \tag{25}$$
$$\frac{\partial \mathcal{L}}{\partial h} = W_2^T \frac{\partial \mathcal{L}}{\partial f} \tag{26}$$
$$\frac{\partial \mathcal{L}}{\partial b_2} = \frac{\partial \mathcal{L}}{\partial f} \tag{27}$$

Then backpropagating to the first layer

$$\frac{\partial \mathcal{L}}{\partial W_1} = \mathbb{1}(h > 0) \frac{\partial \mathcal{L}}{\partial h} x^T \tag{28}$$
$$= \mathbb{1}(h > 0) W_2^T \frac{\partial \mathcal{L}}{\partial f} x^T \tag{29}$$
$$\frac{\partial \mathcal{L}}{\partial b_1} = \mathbb{1}(h > 0) \frac{\partial \mathcal{L}}{\partial h} \tag{30}$$
$$= \mathbb{1}(h > 0) W_2^T \frac{\partial \mathcal{L}}{\partial f}. \tag{31}$$

We'll leave it to you to extend the calculations (and implementation!) to a three-layer network, if you are so-inclined.