How to Fit a Neural Network: Backpropagation

In general, our neural network will be trying to minimize some empirical loss function:

$$\tilde{L}(y, \hat{y}) \leftarrow \text{loss for a whole sample}$$

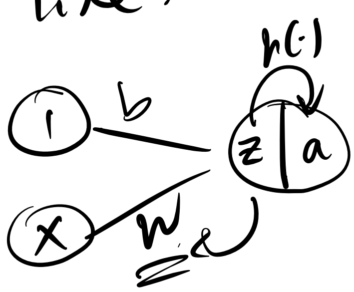
y denotes the vector $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

$\left[L(y_i, \hat{y}_i) \leftarrow \text{loss for a single obs.} \right]$

The workhorse of optimization, gradient descent, helps us minimize this loss.

Simple Example:

Say I had a neural network which looks like:



Take $\hat{y} = a$

Notation:

$\hat{y}$ = output

$z = wx + b$ = "linear predictor"

$a = h(z)$

= "activated linear predictor"

= hidden feature

x = input/predictors

$\underset{\sim}{x} = (x_i)_{i=1}^{n} = (x_1, x_2, \ldots, x_n)$

We would optimize $L(y, \hat{y})$ by the updating eqs:

$$w \leftarrow w - \eta \left(\frac{\partial L}{\partial w}\right)$$

$$b \leftarrow b - \eta \left(\frac{\partial L}{\partial b}\right)$$

For a single observation (x, y) the term $\partial L / \partial w$ can be expressed:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w} \qquad \begin{array}{l} \hat{y} = a = h(z) \\ z = Wx + b \end{array}$$

$$= \frac{\partial L}{\partial \hat{y}} \cdot h'(z) \cdot x, \qquad \&$$

So the stochastic gradient descent eq.

is: $W \leftarrow W - \eta \left(x_i \cdot h'(z_i) \cdot \frac{\partial L}{\partial \hat{y}_i} \right)$

Similarly, the bias term gradient looks like:

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial b}$$

$$= \frac{\partial L}{\partial \hat{y}} \cdot h'(z) \cdot (1)$$

If we consider the empirical loss function for the entire dataset (or a minibatch), we get:

$$\underline{\tilde{L}(y,\hat{y})} = \sum_{i=1}^{n} L(y_i,\hat{y}_i)$$

$\Rightarrow$

$$\frac{\partial \tilde{L}}{\partial W} = \sum_{i=1}^{n} \frac{\partial L(y_i,\hat{y}_i)}{\partial W}$$

$$= \sum_{i=1}^{n} \frac{\partial L}{\partial \hat{y}_i} h'(z_i)\, x_i$$

$\Rightarrow$ The updating eq. is then

$$W \leftarrow W - \eta\left(\sum_{i=1}^{n} \frac{\partial L}{\partial \hat{y}_i} h'(z_i)\, x_i\right)$$

& similarly for b:

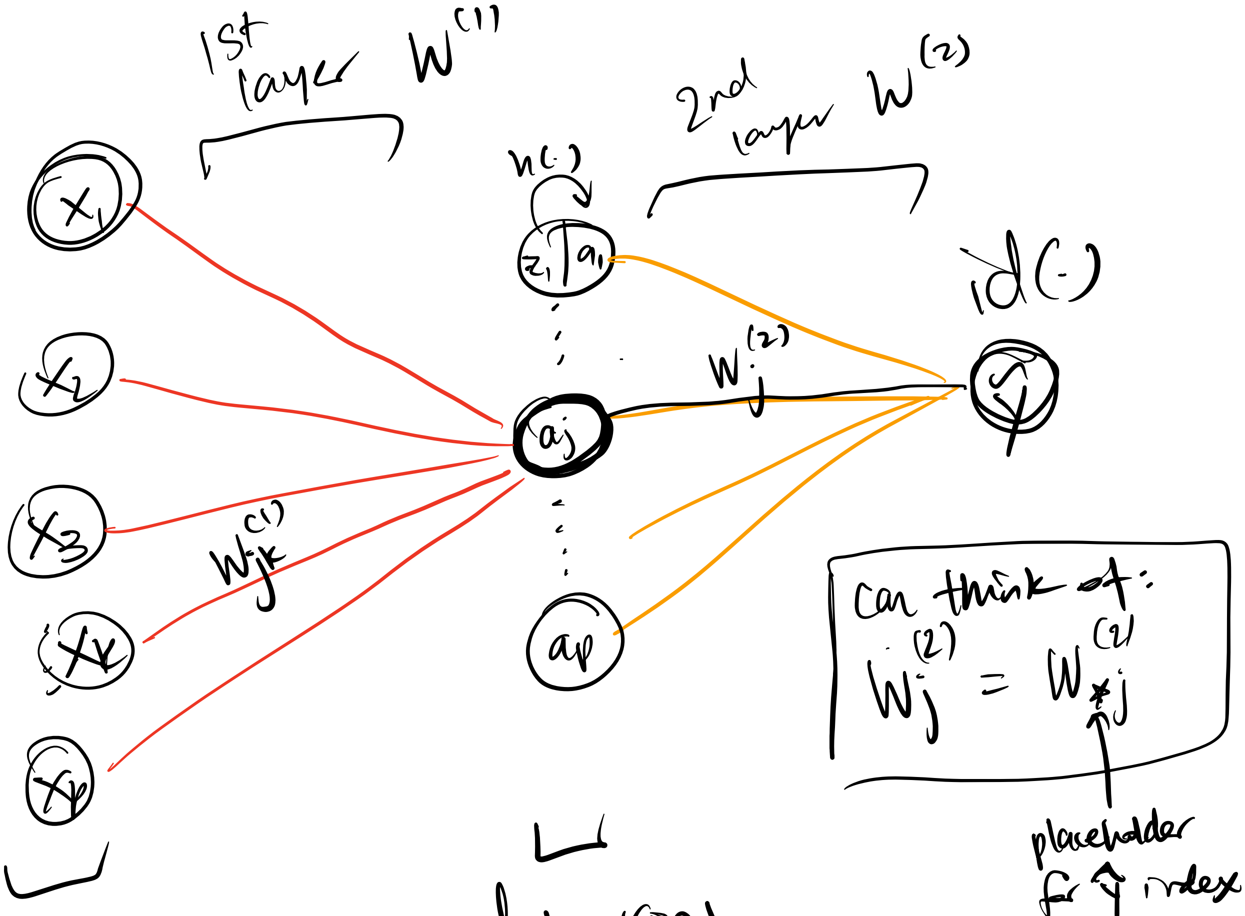
$$b \leftarrow b - \eta\left(\sum_{i=1}^{n} \frac{\partial L}{\partial \hat{y}_i} h'(z_i)\right)$$

What about a more complicated network,

with predictors $X = (X_1, X_2, \ldots, X_p)$

(for a sample $i = 1, \ldots, n$, $X_i = (X_{i1}, X_{i2}, \ldots, X_{ip})$)

We will still start w/ a single obs.



p inputs

$k = 1, \ldots, p$

d neurons

$j = 1, \ldots, d$

Let's try to update $W_{jk}^{(1)}$ w/ GD.

NN Eqs:

$$z_j^{(1)} = \sum_{k=1}^{p} w_{jk}^{(1)} x_j + b_j^{(1)} = \underbrace{W_j^{(1)}}_{\text{jth row of } W^{(1)}} x + b_j^{(1)}$$

$$a_j^{(1)} = h(z_j^{(1)})$$

$$\longrightarrow z^{(2)} = \sum_{j=1}^{d} w_j^{(2)} a_j^{(1)} + b^{(2)} = \hat{y}$$

$$\hat{y} = z^{(2)}$$

$\Rightarrow$ The gradient of L w.r.t. $w_{jk}^{(1)}$ looks like:

$$\frac{\partial L}{\partial w_{jk}^{(1)}} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_j^{(1)}} \cdot \frac{\partial a_j^{(1)}}{\partial z_j^{(1)}} \cdot \frac{\partial z_j^{(1)}}{\partial w_{jk}^{(1)}}$$

This simplifies to:

$$\frac{\partial L}{\partial w_{jk}^{(1)}} = \frac{\partial L}{\partial \hat{y}} \cdot w_j^{(2)} \cdot h'(z_j^{(1)}) \cdot x_k$$

$\therefore$ for a single obs (x_i, y_i) the SGD updating eq. looks like:

$$w_{jk}^{(1)} \leftarrow w_{jk}^{(1)} - \eta \left(\frac{\partial L}{\partial \hat{y}_i} \cdot \underbrace{w_j^{(2)}}_{\text{current value of this weight}} h'(z_{ij}^{(2)}) \, x_{ik} \right)$$

or for the entire sample:

$$w_{jk}^{(1)} \leftarrow w_{jk}^{(1)} - \eta \, w_j^{(2)} \left(\sum_{i=1}^{n} \frac{\partial L}{\partial \hat{y}_i} h'(z_{ij}^{(2)}) \underline{x_{ik}} \right)$$

If we want the updating eqs. for $b_j^{(1)}$:

$$b_j^{(1)} \leftarrow b_j^{(1)} - \eta \left(\sum_{i=1}^{n} \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial a_j^{(1)}} \cdot \frac{\partial a_j^{(1)}}{\partial z_{ij}^{(1)}} \cdot \frac{\partial z_{ij}^{(1)}}{\partial b_j^{(1)}} \right)$$

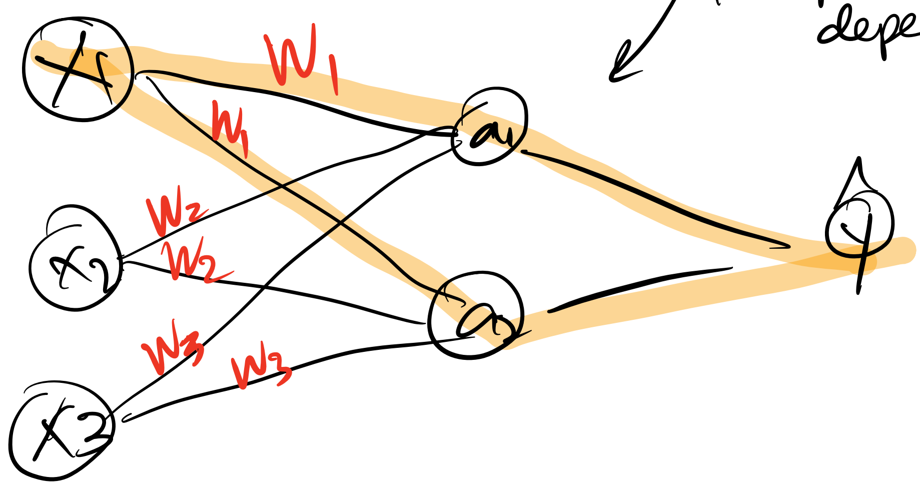
$$= b_j^{(1)} - \eta \left(\sum_{i=1}^{n} \frac{\partial L}{\partial \hat{y}_i} \cdot w_j^{(2)} h'(z_{ij}^{(1)}) \right)$$

Exercises:

- Derive the SGD & GD updating eqs. for $w_j^{(2)}$ & $b_j^{(2)}$.

- What happens if a weight applies to more than one neuron?

two paths of dependence



Hint: Consider $L(y,\hat{y})$ as

$L(y, a_1(w_1), a_2(w_1))$ & take $\frac{\partial L}{\partial w_1}$ w/

multivariate calculus.