How to Fit a Neural Network: Backpropagation

In general, our neural network will be trying to minimize some **empirical loss function**. Let $\tilde{L}(\boldsymbol{y}, f(\boldsymbol{x}))$ denote the loss computed over an entire sample and $L(y_i, f(x_i))$ the loss for a single observation, so that

$$\tilde{L}(\boldsymbol{y}, f(\boldsymbol{x})) = \sum_{i=1}^{n} L(y_i, f(x_i)).$$

Stochastic gradient descent (SGD) helps us minimize this loss.
Consider a simple feed-forward neural network which looks like the following, where $\hat{y} = a$:

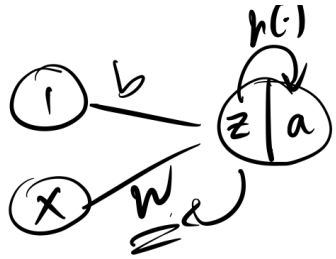

Figure 1: Simple neural network architecture

Notation

- **Output:** $\hat{y}$
- **Linear predictor:**
$$z = wx + b,$$
where w is the weight associated with x, the input/ predictor, and b is the bias/intercept term.
- **Activated linear predictor:**
$$\begin{aligned} a &= h(z) \\ &= \text{“activated linear predictor”} \\ &= \text{hidden feature} \end{aligned}$$
where h is the activation function.
- $\boldsymbol{x} = (x_1, ..., x_n)$

SGD for a Single Observation

For a single observation (x, y), the gradient of the loss with respect to the weights is computed by the chain rule:

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

Since

$$\hat{y} = a = h(z), \quad \text{and} \quad z = wx + b,$$

we have

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

and so the stochastic gradient descent equation is

$$w \leftarrow w - \eta \cdot (\frac{\partial L}{\partial \hat{y}_i} \cdot h'(z_i) \cdot x_i).$$

Similarly, because

$$\frac{\partial z}{\partial b} = 1,$$

the gradient with respect to the bias is

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}_i} \cdot h'(z_i)$$

and the bias update is

$$b \leftarrow b - \eta \ \frac{\partial L}{\partial \hat{y}_i} \cdot h'(z_i)$$

SGD for the Entire Dataset (or Mini-Batch)

When using the entire dataset (or a mini-batch) of n observations, we average the gradients:

$$w \leftarrow w - \eta \ \sum_{i=1}^{n} [\frac{\partial L}{\partial \hat{y}_i} \cdot h'(z_i) \cdot x_i]$$

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

A More Complicated Network

Now consider a network with multiple predictors $x_1, x_2, \ldots, x_p$ for a given observation, i.e.,

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T,$$

a $p \times 1$ column vector of predictors.

(Or for a sample, $i = 1, ..., n$: $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$.)

In a more complex (multi-layer) network the gradient of the loss with respect to a given weight involves contributions from multiple layers of dependence.

We will start in the context of a regression problem with a neural network with a single hidden layer with d neurons, producing an output $\hat{y}$:

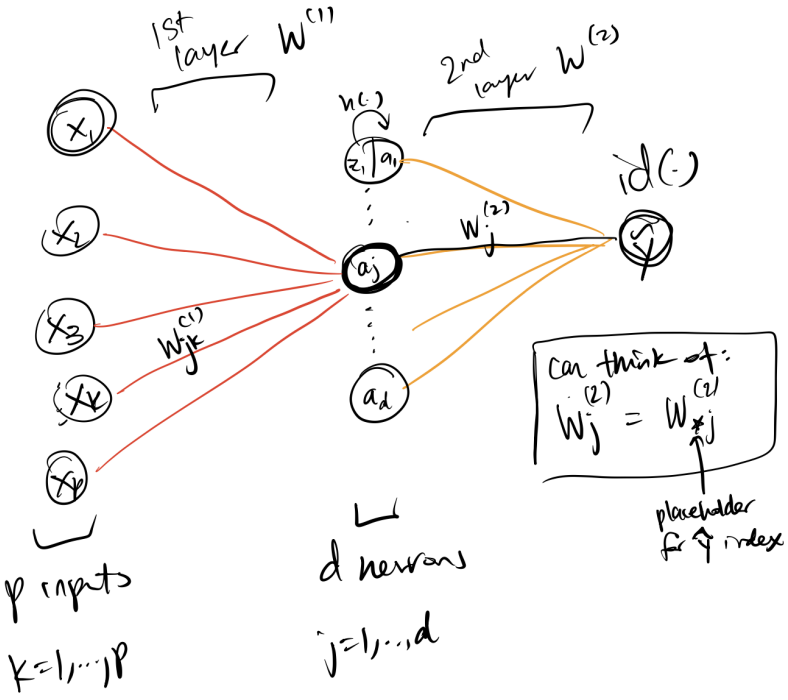

Figure 2: Simple neural network architecture

Here we have for the first layer:

$$\boldsymbol{z}^{[1]} = W^{[1]}\boldsymbol{x} + b^{[1]},$$

where $W^{[1]}$ is the first layer's weight matrix, $\boldsymbol{x}$ is the vector of input predictors, and b is the bias. Together these combine to get $\boldsymbol{z}^{[1]}$ the d-dimensional vector of unactivated linear predictors, which will pass through $h(\cdot)$ to get activated.

Let's try to update $w_{jk}^{[1]}$.

The NN equations are:

$$z_j^{[1]} = \sum_{k=1}^{p} w_{jk}^{[1]} x_k \; + \; b_j^{[1]}$$

$$a_j^{[1]} = h\big(z_j^{[1]}\big)$$

$$z^{[2]} = \sum_{j=1}^{d} w_j^{[2]} a_j^{[1]} \; + \; b^{[2]} \; = \; \hat{y}$$

Hence,

$$\hat{y} = z^{[2]}.$$

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 observation, the SGD updating equation looks like:

$$w_{jk}^{[1]} = w_{jk}^{[1]} - \eta \cdot [\frac{\partial L}{\partial \hat{y}_i} \; \cdot \; w_j^{[2]} \cdot h'(z_{ij}^{[1]}) \; \cdot \; x_{ik}]$$

or for the whole sample:

$$w_{jk}^{[1]} = w_{jk}^{[1]} - \eta \cdot \sum_{i=1}^{n} [\frac{\partial L}{\partial \hat{y}_i} \; \cdot \; w_j^{[2]} \cdot h'(z_{ij}^{[1]}) \; \cdot \; x_{ik}]$$

Similarly, the bias update for a

$$\begin{aligned} b_j &\leftarrow b_j - \eta \frac{\partial L}{\partial \hat{y}_i} \; \cdot \; \frac{\partial \hat{y}_i}{\partial a_j^{[1]}} \cdot \frac{\partial a_j^{[1]}}{\partial z_{ij}^{[1]}} \; \cdot \; \frac{\partial z_{ij}^{[1]}}{\partial b_j^{[1]}} \\ &= b_j - \eta \frac{\partial L}{\partial \hat{y}_i} w_j^{[2]} h'(z_{ij}^{[1]}) \end{aligned}$$

And the bias update for a whole sample is:

$$\begin{aligned} b_j &\leftarrow b_j - \eta \sum_{i=1}^{n} [\frac{\partial L}{\partial \hat{y}_i} \; \cdot \; \frac{\partial \hat{y}_i}{\partial a_j^{[1]}} \cdot \frac{\partial a_j^{[1]}}{\partial z_{ij}^{[1]}} \; \cdot \; \frac{\partial z_{ij}^{[1]}}{\partial b_j^{[1]}}] \\ &= b_j - \eta \sum_{i=1}^{n} [\frac{\partial L}{\partial \hat{y}_i} w_j^{[2]} h'(z_{ij}^{[1]})] \end{aligned}$$

Exercise

1. Continuing with the previous example, derive the SGD (or gradient descent) updating equations for $w_j^{[2]}$ and $b^{[2]}$.