

1 Neural Network

Assume we have the following data point \mathbf{x} :

$$\mathbf{x} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

with corresponding binary label:

$$y = [1]$$

Which is part of a larger dataset \mathbf{X} with binary labels \mathbf{y} .

We want to create a neural network to solve this classification problem. We want to use squared error as the loss function: $\ell = \frac{1}{2}(y - \hat{y})^2$. (Please take a moment and think about why do we use this loss function.) We will use stochastic gradient descent to train our network by minimizing the loss. Suppose we want only one hidden layer with two neurons, sigmoid activation functions and we want to include all bias terms.

1. Draw what our neural network will look like.

2. What will be the shape of our weight matrices (Hint: we view \mathbf{x} as a column vector and compute the product as $\mathbf{W}^T \mathbf{x}$.)

(a) $\mathbf{W}^1 =$

(b) $\mathbf{W}^2 =$

Assume our weights are as follows:

$$\mathbf{W}^1 = \begin{bmatrix} -1 & 2 \\ -2 & 1 \\ 1 & 4 \end{bmatrix} \quad \mathbf{W}^2 = \begin{bmatrix} 1 \\ 2 \\ -4 \end{bmatrix}$$

3. What are the values being passed into our hidden neurons? (Hint: the bias term is considered the first feature.)

4. What are the outputs of our hidden neurons?

Recall we need to add a +1 to the start of this vector to represent the Bias term in this layer!

5. What is the value being passed into our output layer?

6. What is our final output \hat{y} ?

7. What does our model classify \mathbf{x}_1 as?

8. What is the loss ℓ on this example?

Now we do backpropagation to update the weights. I hope you stored your values because you will be reusing them!

First you need to calculate the derivative of ℓ with respect to all of the weights in \mathbf{W}^1 and \mathbf{W}^2 , then you use gradient descent to update them. Be sure to do it in this order because if you update the weights in

\mathbf{W}^2 before calculating the derivative with respect to \mathbf{W}^1 then your calculations will be incorrect.

In the interest of time let's just calculate the derivatives: $\frac{d\ell}{d\mathbf{W}_{11}^2}$ and $\frac{d\ell}{d\mathbf{W}_{21}^1}$.

Let's start with $\frac{d\ell}{d\mathbf{W}_{11}^2}$. It helps to examine the path that this weight effects \mathbf{W}_{11}^2 :

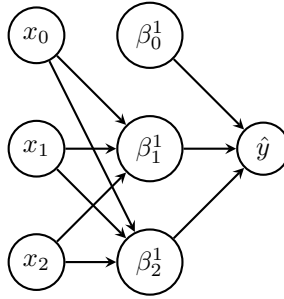


Figure 2: Neural Network

\mathbf{W}_{11}^2 follows a simple propagation path from β_1^1 to \hat{y} .

So our derivative chain will be:

$$\frac{d\ell}{d\mathbf{W}_{11}^2} = \frac{d\ell}{d\hat{y}} \frac{d\hat{y}}{d\alpha^2} \frac{d\alpha^2}{d\mathbf{W}_{11}^2}$$

1. What is $\frac{d\ell}{d\hat{y}}$?

2. What is $\frac{d\hat{y}}{d\alpha^2}$?

3. What is $\frac{d\alpha^2}{d\mathbf{W}_{11}^2}$?

4. Finally, what is $\frac{d\ell}{d\mathbf{W}_{11}^2}$?

Now we will calculate the derivative with respect to \mathbf{W}_{21}^1 : $\frac{d\ell}{d\mathbf{W}_{21}^1}$

$$\frac{d\ell}{d\mathbf{W}_{21}^1} = \frac{d\ell}{d\hat{\mathbf{y}}} \frac{d\hat{\mathbf{y}}}{d\alpha^2} \frac{d\alpha^2}{d\beta_1^1} \frac{d\beta_1^1}{d\alpha_1^1} \frac{d\alpha_1^1}{d\mathbf{W}_{21}^1}$$

We already had the first two derivatives calculated from the previous question.

5. What is $\frac{d\alpha^2}{d\beta_1^1}$?

6. What is $\frac{d\beta_1^1}{d\alpha_1^1}$?

7. What is $\frac{d\alpha_1^1}{d\mathbf{W}_{21}^1}$?

8. Finally what is $\frac{d\ell}{d\mathbf{W}_{21}^1}$?

9. What is our updated \mathbf{W}_{11}^2 and \mathbf{W}_{21}^1 if we use learning rate $\lambda = 2$?