Stats 102B - Week 3, Lecture 3

Miles Chen, PhD

Department of Statistics

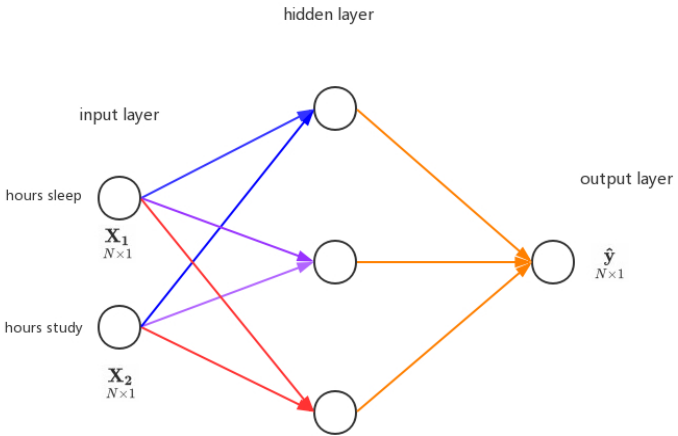
Week 3 Friday

UCLA

Section 1

Quick Review Quiz

What are the dimensions of $\mathbf{X}$, $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and $\mathbf{y}$?



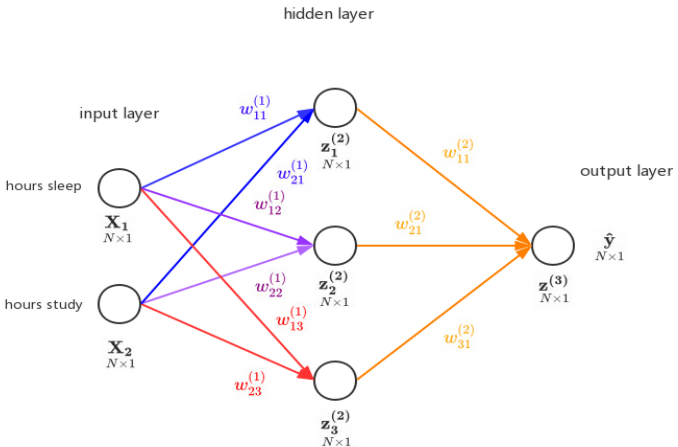
Quiz answers:

- $\mathbf{X}$ is $N \times 2$
- $\mathbf{W}^{(1)}$ is 2×3
- $\mathbf{W}^{(2)}$ is 3×1
- $\mathbf{y}$ is $N \times 1$

Section 2

Neural Networks: Review

Example Neural Network



Review: Weight Matrices

$$\mathbf{W}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix}$$

$$\mathbf{W}^{(2)} = \begin{bmatrix} w_{11}^{(2)} \\ w_{21}^{(2)} \\ w_{31}^{(2)} \end{bmatrix}$$

Review: Sigmoid activation function

Our activation function is the sigmoid (logisitc) activation function. This will 'squish' values in the domain $(-\infty, +\infty)$ to the range $(0, 1)$.

$$f(t) = \frac{1}{1 + e^{-t}}$$

Review: Forward Propagation (no bias terms)

$$\underset{N\times 2}{\mathbf{X}}\ \underset{2\times 3}{\mathbf{W}^{(1)}} = \underset{N\times 3}{\mathbf{z}^{(2)}}$$

$$f(\underset{N\times 3}{\mathbf{z}^{(2)}}) = \underset{N\times 3}{\mathbf{a}^{(2)}}$$

$$\underset{N\times 3}{\mathbf{a}^{(2)}}\underset{3\times 1}{\mathbf{W}^{(2)}} = \underset{N\times 1}{\mathbf{z}^{(3)}}$$

$$f(\underset{N\times 1}{\mathbf{z}^{(3)}}) = \underset{N\times 1}{\hat{\mathbf{y}}}$$

$$\hat{\mathbf{y}} = f(f(\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)})$$

Review: The Loss Function

To see how good the weights are, we need a Loss function to measure its performance.

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2$$

Keep in mind that

$$\hat{\mathbf{y}} = f(f(\mathbf{X}\mathbf{W}^{(1)})\mathbf{W}^{(2)})$$

To make our derivative calculations a little bit easier to work with, we can multiply the Loss function with a positive constant (multiply $\mathcal{L}$ by $N/2$).

$$\mathcal{J} = \frac{1}{2}\sum_{n}(y_n - \hat{y}_n)^2$$

The values of $\mathbf{W}$ that minimize $\mathcal{J}$ will be the same values of $\mathbf{W}$ that minimize $\mathcal{L}$.

Section 3

Back Propagation

Training the network

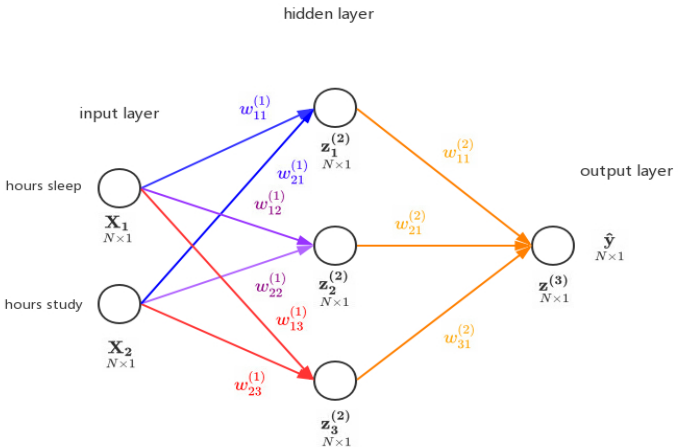
Training the network means finding a set of weights to use in our weight matrices $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ so that the network makes good predictions.

We train by finding weights that minimize the loss function.

To minimize the loss function, we will use gradient descent. This does not guarantee that we find the global minimum, but should help us find some weights that perform well. Gradient descent can often get stuck in flat-ish regions, so we may need to try different starting locations.

Gradient descent requires that we find the gradient.

Our Example Neural Network



Back Propagation is the process of finding the gradient

We have two gradients to calculate:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \begin{bmatrix} \frac{\partial \mathcal{J}}{\partial w_{11}^{(1)}} & \frac{\partial \mathcal{J}}{\partial w_{12}^{(1)}} & \frac{\partial \mathcal{J}}{\partial w_{13}^{(1)}} \\ \frac{\partial \mathcal{J}}{\partial w_{21}^{(1)}} & \frac{\partial \mathcal{J}}{\partial w_{22}^{(1)}} & \frac{\partial \mathcal{J}}{\partial w_{23}^{(1)}} \end{bmatrix}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \begin{bmatrix} \frac{\partial \mathcal{J}}{\partial w_{11}^{(2)}} \\ \frac{\partial \mathcal{J}}{\partial w_{21}^{(2)}} \\ \frac{\partial \mathcal{J}}{\partial w_{31}^{(2)}} \end{bmatrix}$$

If we had bias terms, we would also need to calculate $\frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(1)}}$ and $\frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(2)}}$. (future homework exercise)

Back Propagation

$$
\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} &= \frac{\partial}{\partial \mathbf{W}^{(2)}} \frac{1}{2} \sum (y_n - \hat{y}_n)^2 \\
&= \sum \frac{\partial}{\partial \mathbf{W}^{(2)}} \frac{1}{2} (y_n - \hat{y}_n)^2 \\
&= \sum (y_n - \hat{y}_n)(-1) \frac{\partial \hat{y}_n}{\partial \mathbf{W}^{(2)}} \\
&= \sum -(y_n - \hat{y}_n) \frac{\partial \hat{y}_n}{\partial z_n^{(3)}} \frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(2)}}
\end{aligned}
$$

Back Propagation

Remember: $\hat{y}$ is the result of passing the value $z_n^{(3)}$ through the activation function. So the derivative of $\hat{y}$ with respect to $z_n^{(3)}$ is just the derivative of the activation function.

$$\hat{y}_n = f(z_n^{(3)})$$
$$\frac{\partial \hat{y}_n}{\partial z_n^{(3)}} = f'(z_n^{(3)})$$

The activation function is the sigmoid (logistic) function:

$$f(t) = \frac{1}{1 + e^{-t}}$$

Back Propagation - derivative of the sigmoid

Recall quotient rule:

$$\text{If } f = \frac{u}{v}; \text{ then } f' = \frac{u'v - uv'}{v^2}$$

$$f(t) = \frac{1}{1 + e^{-t}}; \text{let: } u = 1, v = 1 + e^{-t}$$

$$f'(t) = \frac{u'v - uv'}{v^2} = \frac{0 \cdot (1 + e^{-t}) - 1 \cdot -e^{-t}}{(1 + e^{-t})^2}$$

$$f'(t) = \frac{e^{-t}}{(1 + e^{-t})^2}$$

$$\text{so we have: } \frac{\partial \hat{y}_n}{\partial z_n^{(3)}} = f'(z_n^{(3)}) = \frac{e^{-z_n^{(3)}}}{(1 + e^{-z_n^{(3)}})^2}$$

Back Propagation

To keep our work 'tidy', I'm going to replace $\frac{\partial \hat{y}_n}{\partial z_n^{(3)}}$ with $f'(z_n^{(3)})$, knowing that $f'(z_n^{(3)}) = \frac{e^{-z_n^{(3)}}}{(1+e^{-z_n^{(3)}})^2}$.

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} &= \sum -(y_n - \hat{y}_n)\frac{\partial \hat{y}_n}{\partial z_n^{(3)}}\frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(2)}} \\ &= \sum -(y_n - \hat{y}_n)f'(z_n^{(3)})\frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(2)}}\end{aligned}$$

Back Propagation

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \sum -(y_n - \hat{y}_n) f'(z_n^{(3)}) \frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(2)}}$$

Recall that the matrix $\mathbf{z}^{(3)}$ has these dimensions:

$$\underset{N \times 1}{\mathbf{z}^{(3)}} = \underset{N \times 3}{\mathbf{a}^{(2)}} \underset{3 \times 1}{\mathbf{W}^{(2)}}$$

Let's focus on one value $z_n^{(3)}$:

$$\underset{1 \times 1}{z_n^{(3)}} = \underset{1 \times 3}{\mathbf{a}_n^{(2)}} \underset{3 \times 1}{\mathbf{W}^{(2)}}$$

Back Propagation

Let's focus on one value $z_n^{(3)}$:

$$\underset{1\times 1}{z_n^{(3)}} = \underset{1\times 3}{\mathbf{a}_n^{(2)}}\underset{3\times 1}{\mathbf{W}^{(2)}}$$

The derivative will be:

$$\frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(2)}} = \frac{\partial}{\partial \mathbf{W}^{(2)}}\mathbf{a}_n^{(2)}\mathbf{W}^{(2)}$$

$$\underset{3\times 1}{\frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(2)}}} = \underset{3\times 1}{\mathbf{a}_n^{(2)T}}$$

Back Propagation

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \sum -(y_n - \hat{y}_n) f'(z_n^{(3)}) \frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(2)}}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \sum \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \underbrace{\mathbf{a}_n^{(2)T}}_{3\times 1}$$

We try to get rid of the summation by expanding the individual values into matrices and use matrix multiplication.

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \underbrace{-(\mathbf{y} - \hat{\mathbf{y}})}_{N\times 1} \underbrace{f'(\mathbf{z}^{(3)})}_{N\times 1} \underbrace{\mathbf{a}^{(2)T}}_{3\times N}$$

However, the dimensions don't quite line up after expanding them to matrices.

Let's revisit the individual values. If we look at the multiplication, we can rearrange the scalar terms being multiplied.

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \sum \underbrace{-(y_n - \hat{y}_n)}_{1\times 1(\text{scalar})} \underbrace{f'(z_n^{(3)})}_{1\times 1(\text{scalar})} \underbrace{\mathbf{a}_n^{(2)T}}_{3\times 1}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \sum \underbrace{\mathbf{a}_n^{(2)T}}_{3\times 1} \underbrace{-(y_n - \hat{y}_n)}_{1\times 1(\text{scalar})} \underbrace{f'(z_n^{(3)})}_{1\times 1(\text{scalar})}$$

After rearranging terms, I can expand from the individual values to the full matrix, and the dimensions line up appropriately. Because summation is inherent to matrix multiplication, we can get rid of the summation terms. We will also use element-wise multiplication where we were multiplying scalar values together.

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \underbrace{\mathbf{a}^{(2)T}}_{3\times N} \underbrace{-(\mathbf{y} - \hat{\mathbf{y}})}_{N\times 1} \odot \underbrace{f'(\mathbf{z}^{(3)})}_{N\times 1}$$

The $\odot$ represents element wise multiplication, so $-(\mathbf{y} - \hat{\mathbf{y}}) \odot f'(\mathbf{z}^{(3)})$ has dimensions N x 1.

The video calls this:

$$\delta^{(3)} = \underbrace{-(\mathbf{y} - \hat{\mathbf{y}}) \odot f'(\mathbf{z}^{(3)})}_{N\times 1}$$

Back Propagation - Gradient with respect to second weight matrix

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \underbrace{\mathbf{a}^{(2)T}}_{3\times N} \underbrace{-(\mathbf{y} - \hat{\mathbf{y}}) \odot f'(\mathbf{z}^{(3)})}_{N\times 1}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \underbrace{\mathbf{a}^{(2)T}\delta^{(3)}}_{3\times 1}$$

Where:

$$\delta^{(3)} = -(\mathbf{y} - \hat{\mathbf{y}}) \odot f'(\mathbf{z}^{(3)})$$

Symbol $\odot$ represents elementwise multiplication between $(\mathbf{y} - \hat{\mathbf{y}})$ and $f'(\mathbf{z}^{(3)})$ which both have dimension $n \times 1$.

Back Propagation - Gradient with respect to first weight matrix

Now we need to get the gradient with respect to $\mathbf{W}^{(1)}$. The beginning is the same as finding the derivative with respect to $\mathbf{W}^{(2)}$

$$
\begin{aligned}
\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} &= \frac{\partial}{\partial \mathbf{W}^{(1)}} \frac{1}{2} \sum (y_n - \hat{y}_n)^2 \\
&= \sum \frac{\partial}{\partial \mathbf{W}^{(1)}} \frac{1}{2} (y_n - \hat{y}_n)^2 \\
&= \sum (y_n - \hat{y}_n)(-1) \frac{\partial \hat{y}_n}{\partial \mathbf{W}^{(1)}} \\
&= \sum -(y_n - \hat{y}_n) \frac{\partial \hat{y}_n}{\partial z_n^{(3)}} \frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(1)}} \\
&= \sum -(y_n - \hat{y}_n) f'(z_n^{(3)}) \frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(1)}}
\end{aligned}
$$

We split the partial derivative of $z_n^{(3)}$ with respect to weight matrix $\mathbf{W}^{(1)}$ into the partial with respect to $\mathbf{a}_n^{(2)}$ and the partial of $\mathbf{a}_n^{(2)}$ with respect to weight matrix $\mathbf{W}^{(1)}$.

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} &= \sum -(y_n - \hat{y}_n) f'(z_n^{(3)}) \frac{\partial z_n^{(3)}}{\partial \mathbf{W}^{(1)}} \\ &= \sum \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \frac{\partial z_n^{(3)}}{\partial \mathbf{a}_n^{(2)}} \frac{\partial \mathbf{a}_n^{(2)}}{\partial \mathbf{W}^{(1)}}\end{aligned}$$

To find the derivative of $z_n^{(3)}$ (a scalar) with respect to $\mathbf{a}_n^{(2)}$ (a row matrix with dim 1 x 3),

$$\underset{1\times 1}{z_n^{(3)}} = \underset{1\times 3}{\mathbf{a}_n^{(2)}} \underset{3\times 1}{\mathbf{W}^{(2)}}$$

$$\frac{\partial z_n^{(3)}}{\partial \mathbf{a}_n^{(2)}} = \underset{1\times 3}{(\mathbf{W}^{(2)})^T}$$

We plug in the results from the previous page, and we further split the partial derivatives using chain rule:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \sum \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} \frac{\partial \mathbf{a}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$$

$$= \sum \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} \frac{\partial \mathbf{a}_n^{(2)}}{\partial \mathbf{z}_n^{(2)}} \frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$$

The vector $\mathbf{a}_n^{(2)}$ is the result of applying the sigmoid activation function $f()$ element-wise to the vector $\mathbf{z}_n^{(2)}$. The derivative of $\mathbf{a}_n^{(2)}$ with respect to $\mathbf{z}_n^{(2)}$ is the derivative of the function $f()$.

$$\underset{1\times 3}{\mathbf{a}_n^{(2)}} = f(\underset{1\times 3}{\mathbf{z}_n^{(2)}})$$

$$\frac{\partial \mathbf{a}_n^{(2)}}{\partial \mathbf{z}_n^{(2)}} = f'(\underset{1\times 3}{\mathbf{z}_n^{(2)}})$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \sum \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} f'(\underset{1\times 3}{\mathbf{z}_n^{(2)}}) \frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$$

The last bit is $\frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$. We got $\mathbf{z}_n^{(2)}$ by multiplying row n in the input matrix $\mathbf{X}$ by $\mathbf{W}^{(1)}$.

$$\underset{1\times 3}{\mathbf{z}_n^{(2)}} = \underset{1\times 2}{\mathbf{X}_n} \underset{2\times 3}{\mathbf{W}^{(1)}}$$

Finding the derivative of the 1 x 3 vector $\mathbf{z}_n^{(2)}$ with respect to the 2 x 3 matrix $\mathbf{W}^{(1)}$ feels a bit strange. However, if we pay attention to the matrix multiplication between $\mathbf{X}$ and $\mathbf{W}$, we can take the partial derivative without too much pain.

I'll write out the complete relationship for one observation: $\mathbf{x}_n \times \mathbf{W}^{(1)} = \mathbf{z}_n^{(2)}$.

$$\begin{bmatrix} x_{n1} & x_{n2} \end{bmatrix} \times \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \end{bmatrix} =$$

$$\begin{bmatrix} x_{n1}w_{11}^{(1)} + x_{n2}w_{21}^{(1)} & x_{n1}w_{12}^{(1)} + x_{n2}w_{22}^{(1)} & x_{n1}w_{13}^{(1)} + x_{n2}w_{13}^{(1)} \end{bmatrix}$$

Let's use the idea of "perturbing" each element in $\mathbf{W}^{(1)}$, akin to calculating a numeric gradient. We can see that if I add ϵ to any of the elements in the top row of $\mathbf{W}^{(1)}$, the elements in $\mathbf{z}_n^{(2)}$ change by $x_{n1} \cdot \epsilon$. Likewise, if I perturb any of the elements in the lower row of $\mathbf{W}^{(1)}$, the elements in $\mathbf{z}_n^{(2)}$ change by $x_{n2} \cdot \epsilon$.

For this operation, the partial derivative $\frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}}$ will end up being a 2 x 1 matrix: the transpose of $\mathbf{X}_n$

$$\frac{\partial \mathbf{z}_n^{(2)}}{\partial \mathbf{W}^{(1)}} = \underset{2\times 1}{(\mathbf{X}_n)^T}$$

When we plug that back in we get:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \sum \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} \underbrace{f'(\mathbf{z}_n^{(2)})}_{1\times 3} \underbrace{(\mathbf{X}_n)^T}_{2\times 1}$$

Again, we have some dimension mismatch, so we need to rearrange terms:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \sum \underbrace{(\mathbf{X}_n)^T}_{2\times 1} \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} \odot \underbrace{f'(\mathbf{z}_n^{(2)})}_{1\times 3}$$

So for one observation, $\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}}$ has dimension 2 x 3.

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \sum \underbrace{(\mathbf{X}_n)^T}_{2\times 1} \underbrace{-(y_n - \hat{y}_n) f'(z_n^{(3)})}_{1\times 1 \text{(scalar)}} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} \odot \underbrace{f'(\mathbf{z}_n^{(2)})}_{1\times 3}$$

Let's try to get rid of the summation sign:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \underbrace{(\mathbf{X})^T}_{2\times N} \left(\underbrace{-(\mathbf{y} - \hat{\mathbf{y}})}_{N\times 1} \odot \underbrace{f'(\mathbf{z}^{(3)})}_{N\times 1} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} \right) \odot \underbrace{f'(\mathbf{z}^{(2)})}_{N\times 3}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \underbrace{(\mathbf{X})^T}_{2\times N} \left(\underbrace{\delta^{(3)}}_{N\times 1} \underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3} \right) \odot \underbrace{f'(\mathbf{z}^{(2)})}_{N\times 3}$$

Where:

$$\delta^{(3)} = -(\mathbf{y} - \hat{\mathbf{y}}) \odot f'(\mathbf{z}^{(3)})$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \underbrace{(\mathbf{X})^T}_{2\times N}\left(\underbrace{\delta^{(3)}}_{N\times 1}\underbrace{(\mathbf{W}^{(2)})^T}_{1\times 3}\right) \odot \underbrace{f'(\mathbf{z}^{(2)})}_{N\times 3}$$

$$= \underbrace{(\mathbf{X})^T}_{2\times N}\left(\underbrace{\delta^{(3)}(\mathbf{W}^{(2)})^T}_{N\times 3}\right) \odot \underbrace{f'(\mathbf{z}^{(2)})}_{N\times 3}$$

$$= \underbrace{(\mathbf{X})^T}_{2\times N}\underbrace{(\delta^{(3)}(\mathbf{W}^{(2)})^T) \odot f'(\mathbf{z}^{(2)})}_{N\times 3}$$

$$\underbrace{\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}}}_{2\times 3} = \underbrace{(\mathbf{X})^T}_{2\times N}\underbrace{\delta^{(2)}}_{N\times 3}$$

Where

$$\delta^{(2)} = \delta^{(3)}(\mathbf{W}^{(2)})^T \odot f'(\mathbf{z}^{(2)})$$

Section 4

Coding it up

The activation Function

$$f(t) = \frac{1}{1 + e^{-t}}$$

```r
# We define a function called sigmoid. This will apply the sigmoid function
#  to a vector or matrix element-wise.
sigmoid <- function(Z){
    1/(1 + exp(-Z))
}
```

The input data X and Y

```
# training data
x <- matrix(c(
    3, 5,
    5, 1,
    10, 2), ncol = 2, byrow=TRUE)
y <- matrix(c(75,82,93))

# scaling the data
colmax <- apply(x,2,max)
# divide the x values by the column max, so it is on a scale from 0 to 1
X <- t(t(x)/colmax)
# y has a range of 100, so this puts it on a scale from 0 to 1 also
Y <- y/100
```

Variables to specify properties of the neural network

We use variables to specify properties of the neural network to maintain flexibility

```
input_layer_size <- 2
output_layer_size <- 1
hidden_layer_size <- 3
```

Forward Feed (the math)

$$\underset{N\times 2}{\mathbf{X}}\ \underset{2\times 3}{\mathbf{W}^{(1)}} = \underset{N\times 3}{\mathbf{z}^{(2)}}$$

$$f(\underset{N\times 3}{\mathbf{z}^{(2)}}) = \underset{N\times 3}{\mathbf{a}^{(2)}}$$

$$\underset{N\times 3}{\mathbf{a}^{(2)}}\underset{3\times 1}{\mathbf{W}^{(2)}} = \underset{N\times 1}{\mathbf{z}^{(3)}}$$

$$f(\underset{N\times 1}{\mathbf{z}^{(3)}}) = \underset{N\times 1}{\hat{\mathbf{y}}}$$

Forward Feed

```
# define weight matrices W_1 (2 x 3) and W_2 (3x1)
W_1 <- matrix(1, nrow = input_layer_size, ncol = hidden_layer_size)
W_2 <- matrix(1, nrow=hidden_layer_size, ncol=output_layer_size)

# Matrix Z
Z_2 <- X %*% W_1

# apply the sigmoid function to the matrix Z_2
A_2 <- sigmoid(Z_2)

# multiply the A matrix by the weight matrix
Z_3 <- A_2 %*% W_2

# applying the sigmoid function to the Z_3 matrix produces our Y-hats
Y_hat <- sigmoid(Z_3)
```

The Cost function

```
# define our cost function to quantify the error between Y and Y-hat
cost <- function(y,y_hat){
    0.5*sum((y - y_hat)^2)
}
cost(Y,Y_hat)
```

```
## [1] 0.01532851
```

Derivative of the activation function

```r
# we define a function to calculate the derivative of the sigmoid function
sigmoidprime <- function(z){
    exp(-z)/((1+exp(-z))^2)
}
```

The gradients

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \underbrace{(\mathbf{a}^{(2)})^T}_{3 \times N} \underbrace{-(\mathbf{y} - \hat{\mathbf{y}}) \odot f'(\mathbf{z}^{(3)})}_{N \times 1} = \underbrace{(\mathbf{a}^{(2)})^T \delta^{(3)}}_{3 \times 1}$$

We let:

$$\delta^{(3)} = -(\mathbf{y} - \hat{\mathbf{y}}) \odot f'(\mathbf{z}^{(3)})$$

```
delta_3 <- (-(Y - Y_hat) * sigmoidprime(Z_3))
djdw2 <- t(A_2) %*% delta_3
```

The gradients

$$\underbrace{\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}}}_{2\times 3} = \underbrace{(\mathbf{X})^T}_{2\times N} \underbrace{(\delta^{(3)}(\mathbf{W}^{(2)})^T) \odot f'(\mathbf{z}^{(2)})}_{N\times 3}$$

$$= \underbrace{(\mathbf{X})^T}_{2\times N} \underbrace{\delta^{(2)}}_{N\times 3}$$

$$\delta^{(2)} = \delta^{(3)}(\mathbf{W}^{(2)})^T \odot f'(\mathbf{z}^{(2)})$$

```
delta_2 <- delta_3 %*% t(W_2) * sigmoidprime(Z_2)
djdw1 <- t(X) %*% delta_2
```

Gradient Descent

```
#initialize our weight matrices again with random weights
set.seed(1)
W_1 <- matrix(runif(6), nrow = input_layer_size, ncol = hidden_layer_size)
W_2 <- matrix(runif(3), nrow = hidden_layer_size, ncol = output_layer_size)

# for cost tracking
cost_hist <- rep(NA, 10000)

scalar <- 5
```

Gradient Descent

```
for(i in 1:10000){
    # this takes the current weights and calculates y-hat
    Z_2 <- X %*% W_1
    A_2 <- sigmoid(Z_2)
    Z_3 <- A_2 %*% W_2
    Y_hat <- sigmoid(Z_3)
    cost_hist[i] <- cost(Y, Y_hat)
    # this part calculates the gradient at the current y-hat
    delta_3 <- (-(Y - Y_hat)*sigmoidprime(Z_3))
    djdw2 <- t(A_2) %*% delta_3
    delta_2 <- delta_3 %*% t(W_2) * sigmoidprime(Z_2)
    djdw1 <- t(X) %*% delta_2
    # this updates the weights based on the gradient
    W_1 <- W_1 - scalar * djdw1
    W_2 <- W_2 - scalar * djdw2
    # repeat
}
```

Results

```
W_1 # resulting coefficients are basically uninterpretable
```

```
##             [,1]      [,2]       [,3]
## [1,]  2.1169800 1.6157951 -1.4547416
## [2,] -0.7233191 0.4304933 -0.6199876
```

```
W_2
```

```
##            [,1]
## [1,]  2.431735
## [2,]  1.227946
## [3,] -3.645386
```

Results

```
Y_hat  # these match well (probably overfit)
```

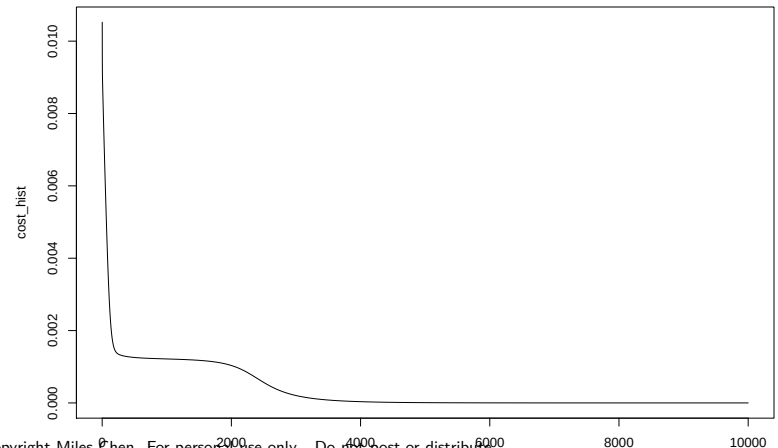
```
##           [,1]
## [1,] 0.7499998
## [2,] 0.8200548
## [3,] 0.9298944
```

```
Y
```

```
##      [,1]
## [1,] 0.75
## [2,] 0.82
## [3,] 0.93
```

A look at the cost function

```
plot(cost_hist, type = "l")
```



A look at the cost function - log scale

```
plot(log(cost_hist), type = "l")
```