MNIST SGD Neural Network

By Caleb Howard s5155126

May 18, 2020

Contents

Chapter 1

Manual Calculation

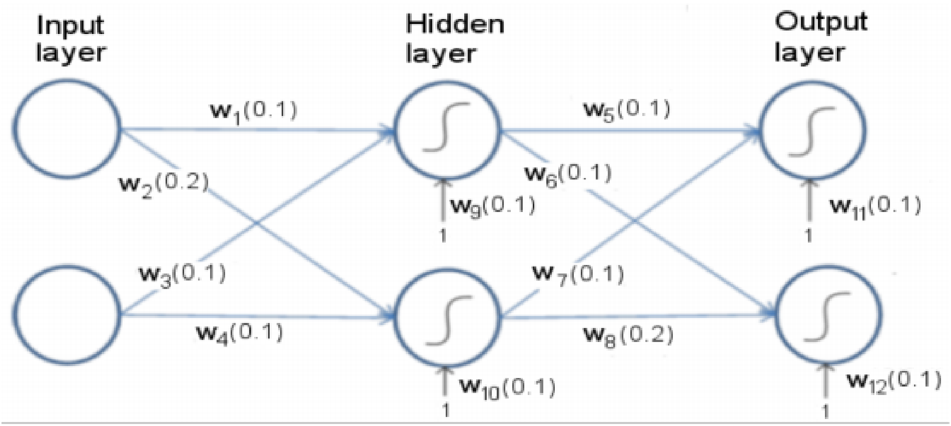

Figure 1.1: A small neural network.

Let the input layer neurons be represented by i1 and i2, the hidden layer by h1 and h2, and the output layer as o1 and o2.

1.1 Sample 1

Input: $X_1 = (0.1, 0.1)$ Output $Y_1 = (1, 0)$

1.1.1 Forward Pass

Calculations for h_1:

$$\begin{aligned} net_{h1} &= w_1 \cdot i_1 + w_3 \cdot i_2 + w_9 \\ &= (0.1 \cdot 0.1) + (0.1 \cdot 0.1) + 0.1 \\ &= 0.12 \end{aligned}$$

Squash the input with the logistic function (sigmoid) to get the output of h_1:

$$\begin{aligned} out_{h1} &= \frac{1}{1 + e^{-net_{h1}}} \\ &= 0.5299640518 \end{aligned}$$

Calculations for h_2:

$$\begin{aligned} net_{h2} &= w_2 \cdot i_1 + w_4 \cdot i_2 + w_{10} \\ &= (0.2 \cdot 0.1) + (0.1 \cdot 0.1) + 0.1 \\ &= 0.13 \end{aligned}$$

Squash the input with the logistic function (sigmoid) to get the output activation of h_2:

$$\begin{aligned} out_{h2} &= \frac{1}{1 + e^{-net_{h2}}} \\ &= \frac{1}{1 + e^{-0.13}} \\ &= 0.5324543064 \end{aligned}$$

Calculations for o_1:

$$\begin{aligned} net_{o1} &= w_5 \cdot h_1 + w_7 \cdot h_2 + w_{11} \\ &= (0.1 \cdot 0.5299640518) + (0.1 \cdot 0.5324543064) + 0.1 \\ &= 0.20624183582 \end{aligned}$$

Squash the input with the logistic function (sigmoid) to get the output activation of o_1:

$$\begin{aligned} out_{o1} &= \frac{1}{1 + e^{-net_{o1}}} \\ &= \frac{1}{1 + e^{-0.20624183582}} \\ &= 0.5513784697 \end{aligned}$$

Calculations for o_2:

$$\begin{aligned} net_{o2} &= w_6 \cdot h_1 + w_8 \cdot h_2 + w_{12} \\ &= (0.1 \cdot 0.5299640518) + (0.2 \cdot 0.5324543064) + 0.1 \\ &= 0.2594872665 \end{aligned}$$

Squash the input with the logistic function (sigmoid) to get the output activation of o_2:

$$\begin{aligned} out_{o2} &= \frac{1}{1 + e^{-net_{o2}}} \\ &= \frac{1}{1 + e^{-0.2594872665}} \\ &= 0.5645102464 \end{aligned}$$

1.1.2 Total Error Calculation

To calculate the total error for each output neuron we can use the squared error function and sum them to get the total error. $E_{total} = \sum \frac{1}{2}(target - output)^2$

Calculations for E_{o1}:

$$\begin{aligned} E_{o1} &= \frac{1}{2}(target_{o1} - output_{o1})^2 \\ &= \frac{1}{2}(1 - 0.5513784697)^2 \\ &= 0.1006306387 \end{aligned}$$

Calculations for E_{o2}:

$$\begin{aligned} E_{o2} &= \frac{1}{2}(target_{o2} - output_{o2})^2 \\ &= \frac{1}{2}(0 - 0.5645102464)^2 \\ &= 0.1593359091 \end{aligned}$$

Therefore:

$$\begin{aligned} E_{total} &= E_{o1} + E_{o2} \\ &= 0.1006306387 + 0.1593359091 \\ &= 0.2599665478 \end{aligned}$$

1.1.3 Backward Pass

The following section will calculate how much change each weight in the output layer affects the total error.

$$\frac{\partial E_{total}}{\partial w_j} = \frac{\partial E_{total}}{\partial out_{oi}} \cdot \frac{\partial out_{oi}}{\partial net_{oi}} \cdot \frac{\partial net_{oi}}{\partial w_j}$$

Output Layer

Calculations for $\frac{\partial E_{total}}{\partial w_5}$ and $\frac{\partial E_{total}}{\partial w_7}$:

How much does the total error change with respect to the output?

$$\begin{aligned} E_{total} &= \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2 \\ \frac{\partial E_{total}}{\partial out_{o1}} &= -(target_{o1} - out_{o1}) = -(1 - 0.5513784697) \\ &= -0.4486215303 \end{aligned}$$

How much does the output of o_1 change with respect to its total net input?

$$\begin{aligned} out_{o1} &= \frac{1}{1 + e^{-net_{o1}}} \\ \frac{\partial out_{o1}}{\partial net_{o1}} &= out_{o1}(1 - out_{o1}) \\ &= 0.5513784697(1 - 0.5513784697) \\ &= 0.24736025285 \end{aligned}$$

For $\frac{\partial E_{total}}{\partial w_5}$:

How much does the net input of o_1 change with respect to w_5?

$$\begin{aligned} \frac{\partial net_{o1}}{\partial w_5} &= out_{h1} \\ &= 0.5299640518 \end{aligned}$$

Putting it all together we get:

$$\begin{aligned} \frac{\partial E_{total}}{\partial w_5} &= -0.4486831301 \cdot 0.24736025285 \cdot 0.5299640518 \\ &= -0.05881878767 \end{aligned}$$

For $\frac{\partial E_{total}}{\partial w_7}$: We get the same calculations from w_5 expect we use $\frac{\partial net_{o1}}{\partial w_7} = out_{h2} = 0.5324543064$
Therefore:

$$\begin{aligned} \frac{\partial E_{total}}{\partial w_7} &= -0.4486831301 \cdot 0.24736025285 \cdot 0.5324543064 \\ &= -0.059095172 \end{aligned}$$

For bias value $\frac{\partial E_{total}}{\partial w_{11}}$: we get the same calculations from w_5 except $\frac{\partial net_{o1}}{\partial w_{11}} = 1$
Therefore:

$$\begin{aligned} \frac{\partial E_{total}}{\partial w_{11}} &= -0.4486831301 \cdot 0.24736025285 \cdot 1 \\ &= -0.11098637251 \end{aligned}$$

Calculations for $\frac{\partial E_{total}}{\partial w_6}$ and $\frac{\partial E_{total}}{\partial w_8}$:

How much does the total error change with respect to the output?

$$\begin{aligned}
E_{total} &= \frac{1}{2}(target_{o2} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2 \\
\frac{\partial E_{total}}{\partial out_{o2}} &= -(target_{o2} - out_{o2}) = -(0 - 0.5645102464) \\
&= 0.5645102464
\end{aligned}$$

How much does the output of o_2 change with respect to its total net input?

$$\begin{aligned}
out_{o2} &= \frac{1}{1 + e^{-net_{o2}}} \\
\frac{\partial out_{o2}}{\partial net_{o2}} &= out_{o2}(1 - out_{o2}) \\
&= 0.5645102464(1 - 0.5645102464) \\
&= 0.24583884281
\end{aligned}$$

For $\frac{\partial E_{total}}{\partial w_6}$:

How much does the net input of o_2 change with respect to w_6?

$$\begin{aligned}
\frac{\partial net_{o2}}{\partial w_6} &= out_{h1} \\
&= 0.5299640518
\end{aligned}$$

Putting it all together we get:

$$\begin{aligned}
\frac{\partial E_{total}}{\partial w_6} &= 0.5645102464 \cdot 0.24583884281 \cdot 0.5299640518 \\
&= 0.0735476404
\end{aligned}$$

For $\frac{\partial E_{total}}{\partial w_8}$: We get the same calculations from w_6 expect we use $\frac{\partial net_{o2}}{\partial w_8} = out_{h2} = 0.5324543064$

Therefore:

$$\begin{aligned}
\frac{\partial E_{total}}{\partial w_8} &= 0.5645102464 \cdot 0.24583884281 \cdot 0.5324543064 \\
&= 0.07389323431
\end{aligned}$$

For bias value $\frac{\partial E_{total}}{\partial w_{12}}$: we get the same calculations from w_6 except $\frac{\partial net_{o2}}{\partial w_{12}} = 1$

Therefore:

$$\begin{aligned}
\frac{\partial E_{total}}{\partial w_{12}} &= 0.5645102464 \cdot 0.24583884281 \cdot 1 \\
&= 0.13877854573
\end{aligned}$$

Hidden Layer

The following section will calculate how much change each weight in the hidden layer affects the total error.

$$\frac{\partial E_{total}}{\partial w_j} = \frac{\partial E_{total}}{\partial out_{hi}} \cdot \frac{\partial out_{hi}}{\partial net_{hi}} \cdot \frac{\partial net_{hi}}{\partial w_j}$$

We need to account for the fact that the output of each hidden layer neuron contributes to the output layer of the multiple output neurons.

$$\frac{\partial E_{total}}{\partial out_{hi}} = \frac{\partial E_{o1}}{\partial out_{hi}} + \frac{\partial E_{o2}}{\partial out_{hi}}$$

For h_1:

Starting with:

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial out_{h1}}$$

Using values calculated earlier:

$$\begin{aligned}\frac{\partial E_{o1}}{\partial net_{o1}} &= \frac{\partial E_{o1}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}}\\ &= -0.4486215303 \cdot 0.24736025285\\ &= -0.11097113517\end{aligned}$$

AND

$$\begin{aligned}\frac{\partial net_{o1}}{\partial out_{o1}} &= w_5\\ &= 0.1\end{aligned}$$

Putting them together we get:

$$\begin{aligned}\frac{\partial E_{o1}}{\partial out_{h1}} &= -0.11097113517 \cdot 0.1\\ &= -0.011097113517\end{aligned}$$

Next we will calculate:

$$\frac{\partial E_{o2}}{\partial out_{h1}} = \frac{\partial E_{o2}}{\partial net_{o2}} \cdot \frac{\partial net_{o2}}{\partial out_{h1}}$$

Using values calculated earlier:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial net_{o2}} &= \frac{\partial E_{o2}}{\partial out_{o2}} \cdot \frac{\partial out_{o2}}{\partial net_{o2}}\\ &= 0.5645102464 \cdot 0.24583884281\\ &= 0.13877854573\end{aligned}$$

AND

$$\begin{aligned}\frac{\partial net_{o2}}{\partial out_{h1}} &= w_6\\ &= 0.1\end{aligned}$$

Putting them together we get:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial out_{h1}} &= 0.13877854573 \cdot 0.1\\ &= 0.013877854573\end{aligned}$$

Therefore:

$$\begin{aligned}\frac{\partial E_{total}}{\partial out_{h1}} &= -0.011097113517 + 0.013877854573\\ &= 0.00277893348\end{aligned}$$

For h_2:

Starting with:

$$\frac{\partial E_{o1}}{\partial out_{h2}} = \frac{\partial E_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial out_{h2}}$$

We know that $\frac{\partial net_{o1}}{\partial out_{h2}} = w_7$ and $w_7 = w_5$,
therefore $\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h2}}$

Now we will calculate:

$$\frac{\partial E_{o2}}{\partial out_{h2}} = \frac{\partial E_{o2}}{\partial net_{o2}} \cdot \frac{\partial net_{o2}}{\partial out_{h2}}$$

Using values calculated earlier:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial net_{o2}} &= \frac{\partial E_{o2}}{\partial out_{o2}} \cdot \frac{\partial out_{o2}}{\partial net_{o2}} \\ &= 0.5645102464 \cdot 0.24583884281 \\ &= 0.13877854573\end{aligned}$$

AND

$$\begin{aligned}\frac{\partial net_{o2}}{\partial out_{h2}} &= w_8 \\ &= 0.2\end{aligned}$$

Putting them together we get:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial out_{h2}} &= 0.13877854573 \cdot 0.2 \\ &= 0.02775570915\end{aligned}$$

Therefore:

$$\begin{aligned}\frac{\partial E_{total}}{\partial out_{h2}} &= -0.011097113517 + 0.02775570915 \\ &= 0.01665859563\end{aligned}$$

Now that we have $\frac{\partial E_{total}}{\partial out_{hi}}$ we need $\frac{\partial out_{hi}}{\partial net_{hi}}$.

$$\begin{aligned}\frac{\partial out_{h1}}{\partial net_{h1}} &= out_{h1}(1 - out_{h1}) \\ &= 0.5299640518(1 - 0.5299640518) \\ &= 0.2491021556\end{aligned}$$

$$\begin{aligned}\frac{\partial out_{h2}}{\partial net_{h2}} &= out_{h2}(1 - out_{h2}) \\ &= 0.5324543064(1 - 0.5324543064) \\ &= 0.248946718\end{aligned}$$

Now that we have $\frac{\partial E_{total}}{\partial out_{hi}}$ and $\frac{\partial out_{hi}}{\partial net_{hi}}$ we need to calculate $\frac{\partial net_{hi}}{\partial w}$ for each weight and plug that value into our formula that we have developed.

Calculations with w_1:

$$\begin{aligned}\frac{\partial net_{h1}}{\partial w1} &= i_1 \\ &= 0.1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_1} &= \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_1} \\ &= 0.00277893348 \cdot 0.2491021556 \cdot 0.1 \\ &= 0.00006922383\end{aligned}$$

Calculations with w_3:

$$\begin{aligned}\frac{\partial net_{h1}}{\partial w3} &= i_2 \\ &= 0.1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_3} &= \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_3} \\ &= 0.00006922383\end{aligned}$$

Calculations with w_2:

$$\begin{aligned}\frac{\partial net_{h2}}{\partial w2} &= i_1 \\ &= 0.1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_2} &= \frac{\partial E_{total}}{\partial out_{h2}} \cdot \frac{\partial out_{h2}}{\partial net_{h2}} \cdot \frac{\partial net_{h2}}{\partial w_2} \\ &= 0.01665859563 \cdot 0.248946718 \cdot 0.1 \\ &= 0.00041471027\end{aligned}$$

Calculations with w_4:

$$\begin{aligned}\frac{\partial net_{h2}}{\partial w4} &= i_2 \\ &= 0.1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_4} &= \frac{\partial E_{total}}{\partial out_{h2}} \cdot \frac{\partial out_{h2}}{\partial net_{h2}} \cdot \frac{\partial net_{h2}}{\partial w_4} \\ &= 0.00041471027\end{aligned}$$

Calculations with bias w_9:

$$\begin{aligned}\frac{\partial net_{h1}}{\partial w9} &= b_1 \\ &= 1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_9} &= \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_9} \\ &= 0.0006922383\end{aligned}$$

Calculations with bias w_{10}:

$$\begin{aligned}\frac{\partial net_{h2}}{\partial w10} &= b_2 \\ &= 1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_{10}} &= \frac{\partial E_{total}}{\partial out_{h2}} \cdot \frac{\partial out_{h2}}{\partial net_{h2}} \cdot \frac{\partial net_{h2}}{\partial w_{10}} \\ &= 0.00414710271\end{aligned}$$

1.2 Sample 2

Input: $X_1 = (0.1, 0.2)$ Output $Y_1 = (0, 1)$

1.2.1 Forward Pass

Calculations for h_1:

$$\begin{aligned} net_{h1} &= w_1 \cdot i_1 + w_3 \cdot i_2 + w_9 \\ &= (0.1 \cdot 0.1) + (0.1 \cdot 0.2) + 0.1 \\ &= 0.13 \end{aligned}$$

Squash the input with the logistic function (sigmoid) to get the output of h_1:

$$\begin{aligned} out_{h1} &= \frac{1}{1 + e^{-net_{h1}}} \\ &= 0.5324543063873187 \end{aligned}$$

Calculations for h_2:

$$\begin{aligned} net_{h2} &= w_2 \cdot i_1 + w_4 \cdot i_2 + w_{10} \\ &= (0.1 \cdot 0.2) + (0.1 \cdot 0.2) + 0.1 \\ &= 0.14 \end{aligned}$$

Squash the input with the logistic function (sigmoid) to get the output activation of h_2:

$$\begin{aligned} out_{h2} &= \frac{1}{1 + e^{-net_{h2}}} \\ &= \frac{1}{1 + e^{-0.14}} \\ &= 0.5349429451582145 \end{aligned}$$

Calculations for o_1:

$$\begin{aligned} net_{o1} &= w_5 \cdot h_1 + w_7 \cdot h_2 + w_{11} \\ &= (0.1 \cdot 0.5324543063873187) + (0.1 \cdot 0.5349429451582145) + 0.1 \\ &= 0.20673972515455333 \end{aligned}$$

Squash the input with the logistic function (sigmoid) to get the output activation of o_1:

$$\begin{aligned} out_{o1} &= \frac{1}{1 + e^{-net_{o1}}} \\ &= \frac{1}{1 + e^{-0.20673972515455333}} \\ &= 0.5515016245695407 \end{aligned}$$

Calculations for o_2:

$$
\begin{aligned}
net_{o2} &= w_6 \cdot h_1 + w_8 \cdot h_2 + w_{12} \\
&= (0.1 \cdot 0.5324543063873187) + (0.2 \cdot 0.5349429451582145) + 0.1 \\
&= 0.2602340196703748
\end{aligned}
$$

Squash the input with the logistic function (sigmoid) to get the output activation of o_2:

$$
\begin{aligned}
out_{o2} &= \frac{1}{1 + e^{-net_{o2}}} \\
&= \frac{1}{1 + e^{-0.2602340196703748}} \\
&= 0.5646938181510764
\end{aligned}
$$

1.2.2 Total Error Calculation

To calculate the total error for each output neuron we can use the squared error function and sum them to get the total error. $E_{total} = \sum \frac{1}{2}(target - output)^2$

Calculations for E_{o1}:

$$\begin{aligned} E_{o1} &= \frac{1}{2}(target_{o1} - output_{o1})^2 \\ &= \frac{1}{2}(0 - 0.5515016245695407)^2 \\ &= 0.15207702095142128 \end{aligned}$$

Calculations for E_{o2}:

$$\begin{aligned} E_{o2} &= \frac{1}{2}(target_{o2} - output_{o2})^2 \\ &= \frac{1}{2}(1 - 0.5646938181510764)^2 \\ &= 0.09474573597794406 \end{aligned}$$

Therefore:

$$\begin{aligned} E_{total} &= E_{o1} + E_{o2} \\ &= 0.15207702095142128 + 0.09474573597794406 \\ &= 0.24682275692936534 \end{aligned}$$

1.2.3 Backward Pass

The following section will calculate how much change each weight in the output layer affects the total error.

$$\frac{\partial E_{total}}{\partial w_j} = \frac{\partial E_{total}}{\partial out_{oi}} \cdot \frac{\partial out_{oi}}{\partial net_{oi}} \cdot \frac{\partial net_{oi}}{\partial w_j}$$

Output Layer

Calculations for $\frac{\partial E_{total}}{\partial w_5}$ and $\frac{\partial E_{total}}{\partial w_7}$:

How much does the total error change with respect to the output?

$$\begin{aligned} E_{total} &= \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2 \\ \frac{\partial E_{total}}{\partial out_{o1}} &= -(target_{o1} - out_{o1}) = -(0 - 0.5515016245695407) \\ &= 0.5515016245695407 \end{aligned}$$

How much does the output of o_1 change with respect to its total net input?

$$\begin{aligned} out_{o1} &= \frac{1}{1 + e^{-net_{o1}}} \\ \frac{\partial out_{o1}}{\partial net_{o1}} &= out_{o1}(1 - out_{o1}) \\ &= 0.5515016245695407(1 - 0.5515016245695407) \\ &= 0.2473475826666981 \end{aligned}$$

For $\frac{\partial E_{total}}{\partial w_5}$:

How much does the net input of o_1 change with respect to w_5?

$$\begin{aligned} \frac{\partial net_{o1}}{\partial w_5} &= out_{h1} \\ &= 0.5324543063873187 \end{aligned}$$

Putting it all together we get:

$$\begin{aligned} \frac{\partial E_{total}}{\partial w_5} &= 0.5515016245695407 \cdot 0.2473475826666981 \cdot 0.5324543063873187 \\ &= 0.07263347294720225 \end{aligned}$$

For $\frac{\partial E_{total}}{\partial w_7}$: We get the same calculations from w_5 expect we use $\frac{\partial net_{o1}}{\partial w_7} = out_{h2} = 0.5349429451582145$
Therefore:

$$\begin{aligned} \frac{\partial E_{total}}{\partial w_7} &= 0.5515016245695407 \cdot 0.2473475826666981 \cdot 0.5349429451582145 \\ &= 0.0729729546166579 \end{aligned}$$

For bias value $\frac{\partial E_{total}}{\partial w_{11}}$: we get the same calculations from w_5 except $\frac{\partial net_{o1}}{\partial w_{11}} = 1$
Therefore:

$$\begin{aligned} \frac{\partial E_{total}}{\partial w_{11}} &= 0.5515016245695407 \cdot 0.2473475826666981 \cdot 1 \\ &= 0.13641259367 \end{aligned}$$

Calculations for $\frac{\partial E_{total}}{\partial w_6}$ and $\frac{\partial E_{total}}{\partial w_8}$:

How much does the total error change with respect to the output?

$$E_{total} = \frac{1}{2}(target_{o2} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2$$
$$\frac{\partial E_{total}}{\partial out_{o2}} = -(target_{o2} - out_{o2}) = -(1 - 0.5646938181510764)$$
$$= -0.43530618184892356$$

How much does the output of o_2 change with respect to its total net input?

$$out_{o2} = \frac{1}{1 + e^{-net_{o2}}}$$
$$\frac{\partial out_{o2}}{\partial net_{o2}} = out_{o2}(1 - out_{o2})$$
$$= 0.5646938181510764(1 - 0.5646938181510764)$$
$$= 0.24581470989303544$$

For $\frac{\partial E_{total}}{\partial w_6}$:

How much does the net input of o_2 change with respect to w_6?

$$\frac{\partial net_{o2}}{\partial w_6} = out_{h1}$$
$$= 0.5324543063873187$$

Putting it all together we get:

$$\frac{\partial E_{total}}{\partial w_6} = -0.43530618184892356 \cdot 0.24581470989303544 \cdot 0.5324543063873187$$
$$= -0.05697509351449143$$

For $\frac{\partial E_{total}}{\partial w_8}$: We get the same calculations from w_6 expect we use $\frac{\partial net_{o2}}{\partial w_8} = out_{h2} = 0.5349429451582145$

Therefore:

$$\frac{\partial E_{total}}{\partial w_8} = -0.43530618184892356 \cdot 0.24581470989303544 \cdot 0.5349429451582145$$
$$= -0.05724138946701667$$

For bias value $\frac{\partial E_{total}}{\partial w_{12}}$: we get the same calculations from w_6 except $\frac{\partial net_{o2}}{\partial w_{12}} = 1$

Therefore:

$$\frac{\partial E_{total}}{\partial w_{12}} = -0.43530618184892356 \cdot 0.24581470989303544 \cdot 1$$
$$= -0.10700466281$$

Hidden Layer

The following section will calculate how much change each weight in the hidden layer affects the total error.

$$\frac{\partial E_{total}}{\partial w_j} = \frac{\partial E_{total}}{\partial out_{hi}} \cdot \frac{\partial out_{hi}}{\partial net_{hi}} \cdot \frac{\partial net_{hi}}{\partial w_j}$$

We need to account for the fact that the output of each hidden layer neuron contributes to the output layer of the multiple output neurons.

$$\frac{\partial E_{total}}{\partial out_{hi}} = \frac{\partial E_{o1}}{\partial out_{hi}} + \frac{\partial E_{o2}}{\partial out_{hi}}$$

For h_1:

Starting with:

$$\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial out_{h1}}$$

Using values calculated earlier:

$$\begin{aligned}\frac{\partial E_{o1}}{\partial net_{o1}} &= \frac{\partial E_{o1}}{\partial out_{o1}} \cdot \frac{\partial out_{o1}}{\partial net_{o1}}\\ &= 0.5515016245695407 \cdot 0.2473475826666981\\ &= 0.13641259367403274\end{aligned}$$

AND

$$\begin{aligned}\frac{\partial net_{o1}}{\partial out_{o1}} &= w_5\\ &= 0.1\end{aligned}$$

Putting them together we get:

$$\begin{aligned}\frac{\partial E_{o1}}{\partial out_{h1}} &= 0.13642431178 \cdot 0.1\\ &= 0.013641259367403274\end{aligned}$$

Next we will calculate:

$$\frac{\partial E_{o2}}{\partial out_{h1}} = \frac{\partial E_{o2}}{\partial net_{o2}} \cdot \frac{\partial net_{o2}}{\partial out_{h1}}$$

Using values calculated earlier:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial net_{o2}} &= \frac{\partial E_{o2}}{\partial out_{o2}} \cdot \frac{\partial out_{o2}}{\partial net_{o2}}\\ &= -0.43530618184892356 \cdot 0.24581470989303544\\ &= -0.10700466281\end{aligned}$$

AND

$$\begin{aligned}\frac{\partial net_{o2}}{\partial out_{h1}} &= w_6\\ &= 0.1\end{aligned}$$

Putting them together we get:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial out_{h1}} &= -0.10696771903 \cdot 0.1\\ &= -0.010700466281\end{aligned}$$

Therefore:

$$\begin{aligned}\frac{\partial E_{total}}{\partial out_{h1}} &= 0.013641259367403276 + -0.010700466281\\ &= 0.00294079309\end{aligned}$$

For h_2:

Starting with:

$$\frac{\partial E_{o1}}{\partial out_{h2}} = \frac{\partial E_{o1}}{\partial net_{o1}} \cdot \frac{\partial net_{o1}}{\partial out_{h2}}$$

We know that $\frac{\partial net_{o1}}{\partial out_{h2}} = w_7$ and $w_7 = w_5$,
therefore $\frac{\partial E_{o1}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h2}}$

Now we will calculate:

$$\frac{\partial E_{o2}}{\partial out_{h2}} = \frac{\partial E_{o2}}{\partial net_{o2}} \cdot \frac{\partial net_{o2}}{\partial out_{h2}}$$

Using values calculated earlier:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial net_{o2}} &= \frac{\partial E_{o2}}{\partial out_{o2}} \cdot \frac{\partial out_{o2}}{\partial net_{o2}} \\ &= -0.43530618184892356 \cdot 0.24581470989303544 \\ &= -0.10700466281\end{aligned}$$

AND

$$\begin{aligned}\frac{\partial net_{o2}}{\partial out_{h2}} &= w_8 \\ &= 0.2\end{aligned}$$

Putting them together we get:

$$\begin{aligned}\frac{\partial E_{o2}}{\partial out_{h2}} &= -0.10700466281 \cdot 0.2 \\ &= -0.02140093256\end{aligned}$$

Therefore:

$$\begin{aligned}\frac{\partial E_{total}}{\partial out_{h2}} &= 0.013641259367403274 + -0.02140093256 \\ &= -0.00775967319\end{aligned}$$

Now that we have $\frac{\partial E_{total}}{\partial out_{hi}}$ we need $\frac{\partial out_{hi}}{\partial net_{hi}}$.

$$\begin{aligned}\frac{\partial out_{h1}}{\partial net_{h1}} &= out_{h1}(1 - out_{h1}) \\ &= 0.5324543063873187(1 - 0.5324543063873187) \\ &= 0.248946718\end{aligned}$$

$$\begin{aligned}\frac{\partial out_{h2}}{\partial net_{h2}} &= out_{h2}(1 - out_{h2}) \\ &= 0.5349429451582145(1 - 0.5349429451582145) \\ &= 0.24877899058\end{aligned}$$

Now that we have $\frac{\partial E_{total}}{\partial out_{hi}}$ and $\frac{\partial out_{hi}}{\partial net_{hi}}$ we need to calculate $\frac{\partial net_{hi}}{\partial w}$ for each weight and plug that value into our formula that we have developed.

Calculations with w_1:

$$\begin{aligned}\frac{\partial net_{h1}}{\partial w1} &= i_1 \\ &= 0.1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_1} &= \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_1} \\ &= 0.00294079309 \cdot 0.248946718 \cdot 0.1 \\ &= 0.00007321008\end{aligned}$$

Calculations with w_3:

$$\begin{aligned}\frac{\partial net_{h1}}{\partial w3} &= i_2 \\ &= 0.2\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_3} &= \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_3} \\ &= 0.00294079309 \cdot 0.248946718 \cdot 0.2 \\ &= 0.00014642016\end{aligned}$$

Calculations with w_2:

$$\begin{aligned}\frac{\partial net_{h2}}{\partial w2} &= i_1 \\ &= 0.1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_2} &= \frac{\partial E_{total}}{\partial out_{h2}} \cdot \frac{\partial out_{h2}}{\partial net_{h2}} \cdot \frac{\partial net_{h2}}{\partial w_2} \\ &= -0.00775967319 \cdot 0.24877899058 \cdot 0.1 \\ &= -0.00019304437\end{aligned}$$

Calculations with w_4:

$$\begin{aligned}\frac{\partial net_{h1}}{\partial w4} &= i_2 \\ &= 0.2\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_4} &= \frac{\partial E_{total}}{\partial out_{h2}} \cdot \frac{\partial out_{h2}}{\partial net_{h2}} \cdot \frac{\partial net_{h2}}{\partial w_4} \\ &= -0.00775967319 \cdot 0.24877899058 \cdot 0.2 \\ &= -0.00038608873\end{aligned}$$

Calculations with bias w_9:

$$\begin{aligned}\frac{\partial net_{h1}}{\partial w9} &= b_1 \\ &= 1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_9} &= \frac{\partial E_{total}}{\partial out_{h1}} \cdot \frac{\partial out_{h1}}{\partial net_{h1}} \cdot \frac{\partial net_{h1}}{\partial w_9} \\ &= 0.00294079309 \cdot 0.248946718 \cdot 1 \\ &= 0.00073210079\end{aligned}$$

Calculations with bias w_{10}:

$$\begin{aligned}\frac{\partial net_{h2}}{\partial w10} &= b_2 \\ &= 1\end{aligned}$$

Plugging it into our formula:

$$\begin{aligned}\frac{\partial E_{total}}{\partial w_9} &= \frac{\partial E_{total}}{\partial out_{h2}} \cdot \frac{\partial out_{h2}}{\partial net_{h2}} \cdot \frac{\partial net_{h2}}{\partial w_9} \\ &= -0.00775967319 \cdot 0.24877899058 \cdot 1 \\ &= -0.00193044366\end{aligned}$$

1.3 Updating Weights

For each each weight we will take the average of it's partial derivative from each sample, multiply it by a learning rate of 0.1, and minus it from the current weight to receive the new weight.

Updating w_5:

$$\frac{\sum_{t=1}^{2} \frac{\partial E^t_{total}}{\partial w^t_5}}{2} = \frac{-0.05881878767 + 0.07263347294720225}{2} = 0.00690734264$$

$$w_5^+ = w_5 - (0.1 \cdot 0.00690734264) = 0.09930886197$$

Updating w_7:

$$\frac{\sum_{t=1}^{2} \frac{\partial E^t_{total}}{\partial w^t_7}}{2} = \frac{-0.059095172 + 0.0729729546166579}{2} = 0.00693889131$$

$$w_7^+ = w_7 - (0.1 \cdot 0.00693889131) = 0.09930611087$$

Updating w_6:

$$\frac{\sum_{t=1}^{2} \frac{\partial E^t_{total}}{\partial w^t_6}}{2} = \frac{0.0735476404 - 0.05697509351449143}{2} = 0.00828627344$$

$$w_6^+ = w_6 - (0.1 \cdot 0.00828627344) = 0.09917137266$$

Updating w_8:

$$\frac{\sum_{t=1}^{2} \frac{\partial E^t_{total}}{\partial w^t_8}}{2} = \frac{0.07389323431 - 0.05724138946701667}{2} = 0.00832592242$$

$$w_8^+ = w_8 - (0.1 \cdot 0.00832592242) = 0.19916740776$$

Updating w_1:

$$\frac{\sum_{t=1}^{2} \frac{\partial E^t_{total}}{\partial w^t_1}}{2} = \frac{0.00006922383 + 0.00007321008}{2} = 0.00007121696$$

$$w_1^+ = w_1 - (0.1 \cdot 0.00007121696) = 0.0999928783$$

Updating w_3:

$$\frac{\sum_{t=1}^{2} \frac{\partial E^t_{total}}{\partial w^t_3}}{2} = \frac{0.00006922383 + 0.00014642016}{2} = 0.000107822$$

$$w_3^+ = w_3 - (0.1 \cdot 0.000107822) = 0.0999892178$$

Updating w_2:

$$\frac{\sum_{t=1}^{2}\frac{\partial E_{total}^t}{\partial w_2^t}}{2} = \frac{0.00041471027 - 0.00019304437}{2}$$
$$= 0.00011083295$$

$$w_2^+ = w_2 - (0.1 \cdot 0.00011083295) = 0.19998891671$$

Updating w_4:

$$\frac{\sum_{t=1}^{2}\frac{\partial E_{total}^t}{\partial w_4^t}}{2} = \frac{0.00041471027 - 0.00038608873}{2}$$
$$= 0.00001431077$$

$$w_4^+ = w_4 - (0.1 \cdot 0.00001431077) = 0.09999856892$$

Biases

Updating w_9:

$$\frac{\sum_{t=1}^{2}\frac{\partial E_{total}^t}{\partial w_9^t}}{2} = \frac{0.0006922383 + 0.00073210079}{2}$$
$$= 0.00001431077$$

$$w_9^+ = w_9 - (0.1 \cdot 0.00071216954) = 0.09992878305$$

Updating w_{10}:

$$\frac{\sum_{t=1}^{2}\frac{\partial E_{total}^t}{\partial w_{10}^t}}{2} = \frac{0.00414710271 - 0.00193044366}{2}$$
$$= 0.00303877319$$

$$w_{10}^+ = w_{10} - (0.1 \cdot 0.00110832953) = 0.09988916705$$

Updating w_{11}:

$$\frac{\sum_{t=1}^{2}\frac{\partial E_{total}^t}{\partial w_{11}^t}}{2} = \frac{-0.11098637251 + 0.13641259367}{2}$$
$$= -0.12369948309$$

$$w_{11}^+ = w_{11} - (0.1 \cdot 0.01271311058) = 0.09872868894$$

Updating w_{12}:

$$\frac{\sum_{t=1}^{2}\frac{\partial E_{total}^t}{\partial w_{12}^t}}{2} = \frac{0.138778545738 - 0.10700466281}{2}$$
$$= 0.00001431077$$

$$w_{12}^+ = w_{12} - (0.1 \cdot 0.01588694146) = 0.09841130585$$

Chapter 2

Neural Network Implementation

2.1 Overview

The implemented neural network is written in python. It uses stochastic gradient descent to update the weights and biases of the network. The network has 3 layers, although more can be added easily. Three cost types are calculate at the end of each epoch and printed to the console: accuracy, quadratic cost, and cross entropy cost. The program has a console menu in which the user can select specific parameters to train the network with. Other functionality such as graphing results, printing predictions to file and more are also included in the program.

This chapter will contain an analysis of the neural network on the provided MNIST data, training with different hyper-parameters and comparing results. A section on training the network with the parameters from chapter one, with a confirmation of results will also be included in this chapter.

Examples of running the program

For training with MNIST data:

```
main.py 784 30 10 TrainDigitX.csv.gz TrainDigitY.csv.gz TestDigitX.csv.gz PredictDigitY.csv.gz
```

or

```
main.py 784 30 10 TrainDigitX.csv.gz TrainDigitY.csv.gz TestDigitX2.csv.gz PredictDigitY2.csv.gz
```

For training with the simple network from chapter one:

```
main.py 2 2 2
```

2.2 MNIST Results Analysis with Different Hyper-parameters

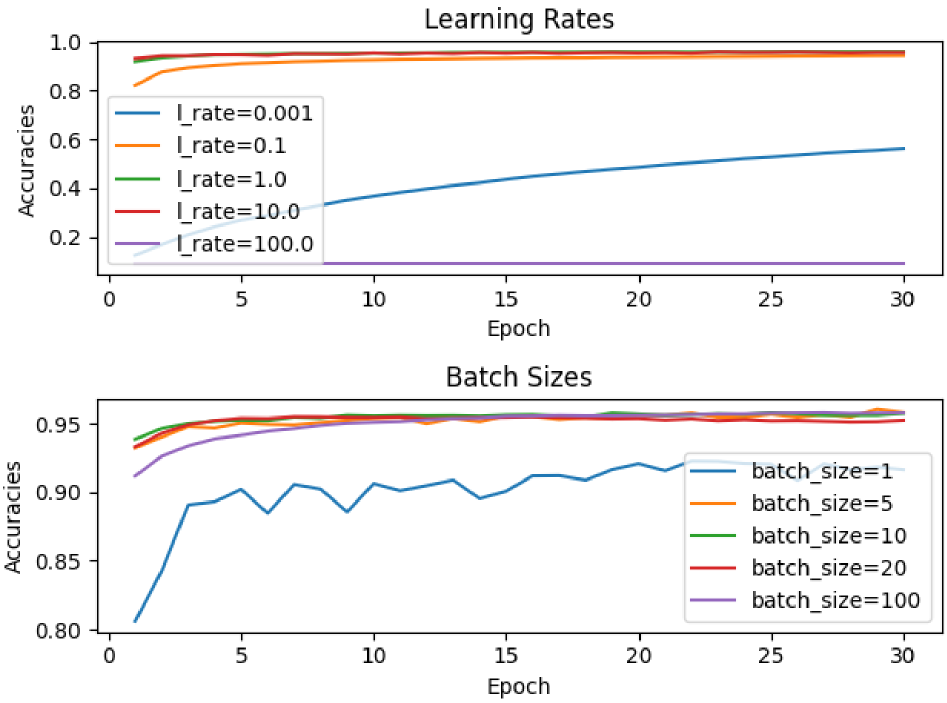

Figure 2.1: Accuracy Vs. Epoch.

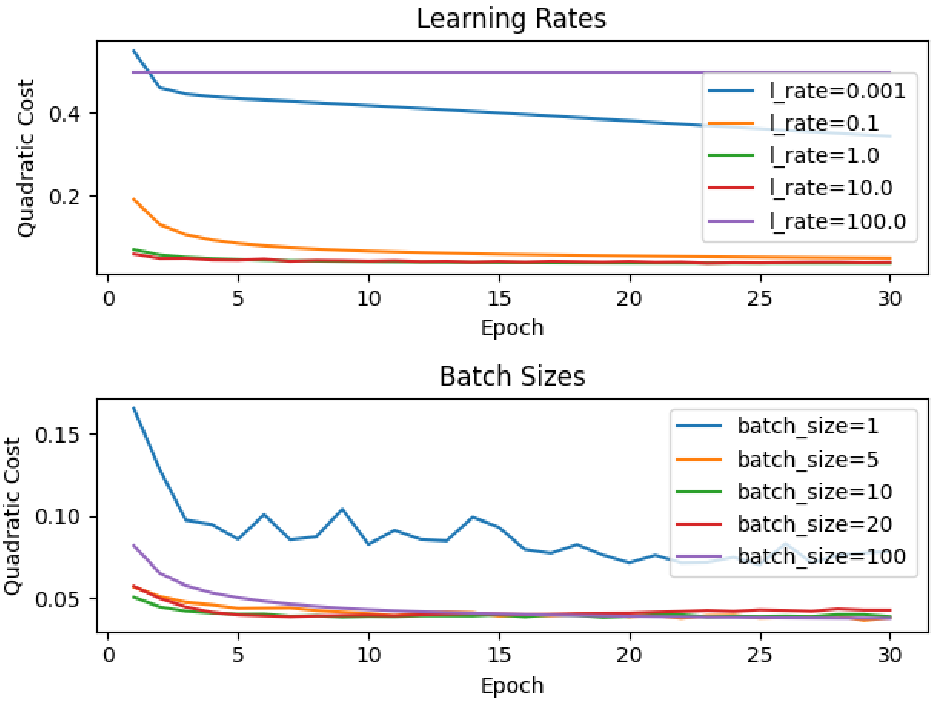

Figure 2.2: Quadratic cost Vs. Epoch.

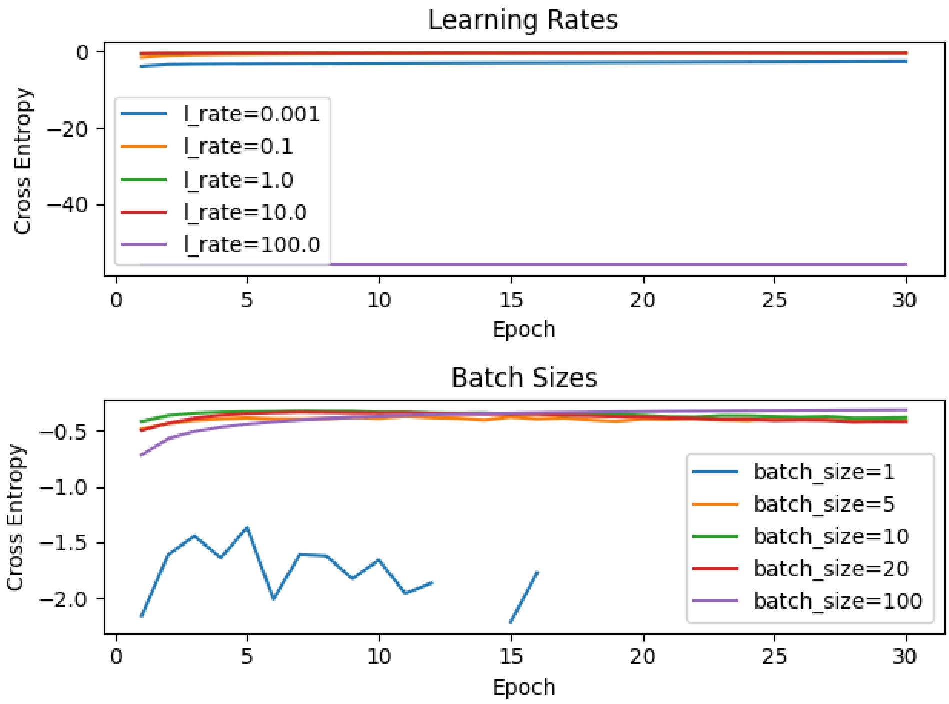

Figure 2.3: Cross Entropy cost Vs. Epoch.

Learning Rates

The results highlight the ideal parameters for this network. For learning rates, between 0.1 and 10.0 produced the optimal results. It can be inferred from the above figures that this range of rates (0.1-10.0) quickly found the high accuracy of approximately 95%, at which a local minimum was likely found. We see from the figures that the rate at which the learning rate 0.001 decreases is much less in comparison to the other range. Therefore, given more epochs, it is likely that the learning rate of 0.001 would avoid this local minimum and pass the other rates. The learning rate of 100.0 quickly found a local minimum as expected and therefore is the worst of the learning rates for this network.

Batch Sizes

The batch sizes reveal interesting results. It can be inferred that the ideal batch size for this data is between 5 and 100. For the batch size of 1, the figures highlight a higher volatility in the increase of accuracy/decrease of cost. With a larger batch size than 1, the weights are updated based on an average, therefore, for the range of 5-100 we see less volatility and faster accuracy increases. However, a batch size too large will increase approximation and therefore decrease accuracy, this can be seen for the batch size of 100, where more epochs are taken to reach the accuracy/cost of the other batch sizes.

Summary

After training the networking with various different parameters, I have concluded that the best hyper-parameter configuration for my implementation for the provided MNIST dataset is as follows:

- Batch size: 10
- Learning rate: 0.1
- Epochs: 25

However, if time were not a constraint, then the ideal hyper-parameters would have a lower learning rate and a higher number of epochs. This would help reduce the risk of reaching a local minimum around the 95% accuracy mark.

2.3 Confirmation of Manual Calculation From Chapter 1

To confirm our manual calculations from part 1 of this document I ran the neural network with the following parameters.

train_x = [0.1, 0.1], [0.1, 0.2]
train_y [1.0, 0.0], [0.0, 1.0]
Hidden layer bias = 0.1, 0.1
Output layer bias = 0.1, 0.1
Hidden layer weights = [0.1, 0.1], [0.2, 0.1]
Output layer weights = [0.1, 0.1], [0.1, 0.2]
batch_size = 2
epochs = 1
learning rate = 0.1

Results

	Neural Network	Manual Calculations
w1	0.09999287608226642	0.0999928783
w2	0.19998891676306169	0.19998891671
w3	0.09998921557833007	0.0999892178
w4	0.0999985689813837	0.09999856892
w5	0.09930886197388886	0.09930886197
w6	0.09917137885954593	0.09917137266
w7	0.09930570520953383	0.09930611087
w8	0.19916741399056967	0.19916740776
w9	0.09992876082266425	0.0999287608
w10	0.09988916763061688	0.0998891676
w11	0.09872792707492782	0.09872793
w12	0.09841131755925499	0.09841131

It can be seen from the above results table that the manually calculated weights match the weights outputted from the python neural network.
Note: All values do match, however, the manual calculations are rounded due to calculator constraints.