Recitation 7: Backpropagation Examples
LOrtiz (Orig. by K Koile)

An efficient method of implementing gradient descent for neural networks

\[ w_{i \rightarrow j} = w_{i \rightarrow j} - r \delta_j y_i \quad \text{Descent Rule} \]

\[ \delta_j = \frac{dE_j}{dz_j} = \frac{d s(z_j)}{dz_j} \]

Backprop rule (\( s(z_j) \) is the sigmoid function)

\[ \delta_j = \frac{ds(z_j)}{dz_j} \sum_k \delta_k w_{j \rightarrow k} \]

1. Initialize weights to small random values
2. Choose a random sample input feature vector
3. Compute total input (\( z_j \)) and output (\( y_j \)) for each unit (forward prop)
4. Compute \( \delta_n \) for output layer. Assuming sigmoid neural net and standard error function:
   \[ \delta_n = \frac{d s(z_n)}{dz_n} (y_n - y_n^*) = y_n (1 - y_n)(y_n - y_n^*) \]
5. Compute \( \delta_j \) for preceding layer by backprop rule (repeat for all layers)
6. Compute weight change by descent rule (repeat for all weights)

From the online tutor:

![Backprop Example](image)

Forward prop: Compute \( z_j \) and \( y_j \), given \( x_i \), \( w_i \)

\[ \delta_3 = y_3 (1 - y_3)(y_3 - y_3^*) \]
\[ \delta_2 = y_2 (1 - y_2) \delta_3 w_{23} \]
\[ \delta_1 = y_1 (1 - y_1) \delta_3 w_{13} \]

\[ w_{03} = w_{03} - r \delta_3 (-1) \]
\[ w_{02} = w_{02} - r \delta_2 (-1) \]
\[ w_{01} = w_{01} - r \delta_1 (-1) \]

Compare to the direct derivation earlier. Note that all computations are local!
\[
\frac{\partial E}{\partial w_1} = s(z_1)(1-s(z_1))\left[(y_2 - y_2^*)(s(z_2)(1-s(z_2)))w_2 + (y_3 - y_3^*)(s(z_3)(1-s(z_3)))w_3\right]x_1
\]

\[
\frac{\partial E}{\partial w_1} = y_1(1-y_1)\left[(y_2 - y_2^*)y_2(1-y_2)w_2 + (y_3 - y_3^*)y_3(1-y_3)w_3\right]x_1
\]

To compute \(w_j\):

1. Substitute above expression into: \(w_i = w_i - r \frac{\partial E}{\partial w_i}\)

or

2. Compute \(\delta_1\) and substitute into: \(w_i = w_i - r\delta_1x_1\)

From the expression for \(\frac{\partial E}{\partial w_1}\), we can derive the expression for \(\delta_1\):

\[
\delta_1 = y_1(1-y_1)[(y_2 - y_2^*)y_2(1-y_2)w_2 + (y_3 - y_3^*)y_3(1-y_3)w_3]
\]

\[
\delta_1 = y_1(1-y_1)(\delta_2 w_2 + \delta_3 w_3)
\]

(Note: This is a restatement of the backprop rule.)