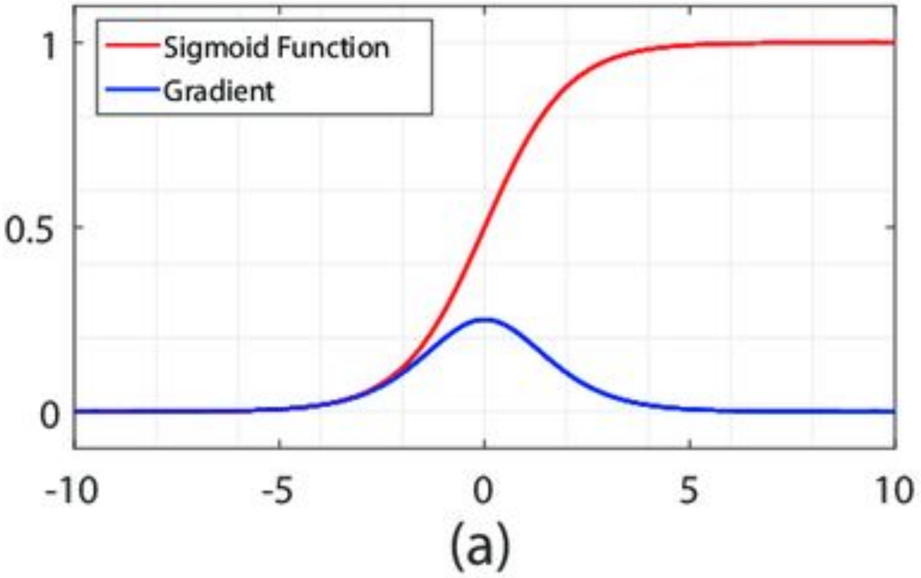
Neural Net From Scratch

https://shala2020.github.io/

Sigmoid Forward



$$Sigmoid(x) = \frac{1}{1 + e^{-x}}$$

Image source: Ranking to Learn and Learning to Rank: On the Role of Ranking in Pattern Recognition Applications - Scientific Figure on ResearchGate.

Sigmoid Derivative

$$\frac{d}{dx}S(x) = \frac{d}{dx}\frac{1}{1+e^{-x}}$$

$$\frac{d}{dx}S(x) = \frac{(1+e^{-x})(0) - (1)(-e^{-x})}{(1+e^{-x})^2}$$

$$\frac{d}{dx}S(x) = \frac{e^{-x}}{(1+e^{-x})^2}$$

(Are we done yet?)

Source

Sigmoid Derivative Continued (1)

$$\frac{d}{dx}S(x) = \frac{1 - 1 + e^{-x}}{(1 + e^{-x})^2}$$

$$\frac{d}{dx}S(x) = \frac{1 + e^{-x}}{(1 + e^{-x})^2} - \frac{1}{(1 + e^{-x})^2}$$

$$\frac{d}{dx}S(x) = \frac{1}{(1 + e^{-x})} - \frac{1}{(1 + e^{-x})^2}$$

Source

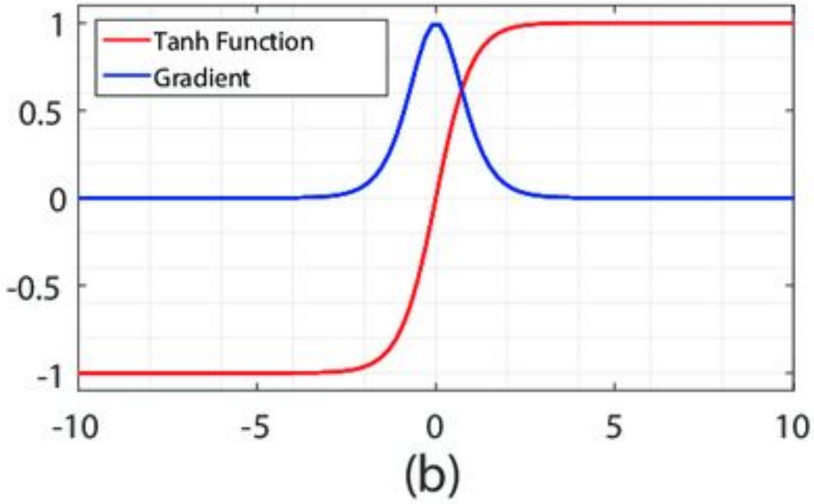
Sigmoid Derivative Continued (2)

$$\frac{d}{dx}S(x) = \frac{1}{(1+e^{-x})}(1 - \frac{1}{1+e^{-x}})$$

$$\boxed{\frac{d}{dx}S(x) = S(x)(1 - S(x))}$$

Source

Tanh Forward



$$g(z) = tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

Image source: Ranking to Learn and Learning to Rank: On the Role of Ranking in Pattern Recognition Applications - Scientific Figure on ResearchGate.

Tanh Derivative

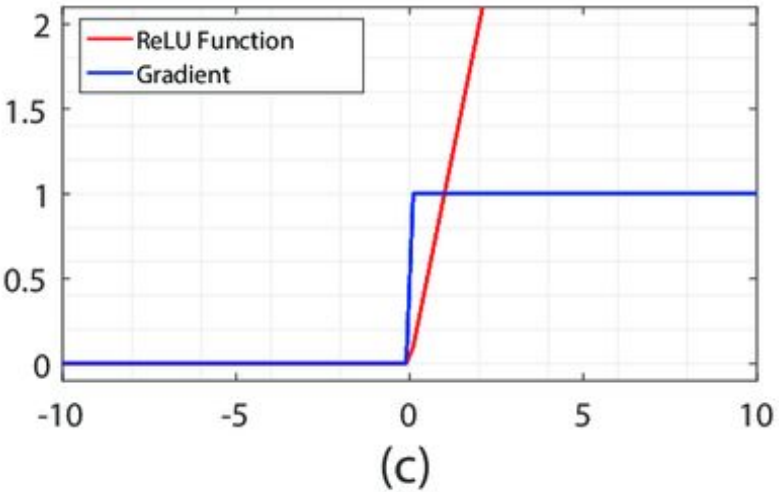
$$\frac{d}{dz}g(z) = \frac{(e^z+e^{-z})(e^z+e^{-z})-(e^z-e^{-z})(e^z-e^{-z})}{(e^z+e^{-z})^2}$$

$$\frac{(e^z+e^{-z})^2-(e^z-e^{-z})^2}{(e^z+e^{-z})^2}$$

$$1-\left(\frac{e^z-e^{-z}}{e^z+e^{-z}}\right)^2 = \boxed{1-tanh(z)^2}$$

Source

ReLU Forward



$$R(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}$$

Image source: Ranking to Learn and Learning to Rank: On the Role of Ranking in Pattern Recognition Applications - Scientific Figure on ResearchGate.

ReLU Derivative

$$R'(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases}$$

Note: ReLU's derivative is undefined at 0, however we will implement the above derivative for this homework.

Source

Softmax Cross Entropy Forward

$$\vec{y} = [0, 1, 0, 0] \qquad \vec{p} = [0.1, 0.3, 0.5, 0.1]$$

$$-\sum_{i=1}^{M} y_i log(p_i)$$

(M Classes)

$$-\sum_{i=1}^{M} y_i log(\frac{e^{x_i}}{\sum_{k=1}^{M} e^{x_k}})$$

Softmax Cross Entropy Forward Continued (1)

$$-y_c log(\frac{e^{x_c}}{\sum_{k=1}^{M} e^{x_k}})$$

('c' is the true class)

$$-(y_c log(e^{x_c}) - y_c log(\sum_{k=1}^{M} e^{x_k}))$$

(using the property of log)

$$-y_c x_c + y_c log(\sum_{k=1}^{M} e^{x_k})$$

(log and exp are inverse functions of each other)

Softmax Cross Entropy Forward Continued (2)

$$-y_c x_c + y_c(a + log(\sum_{k=1}^{M} e^{x_k - a}))$$

(LogSumExp trick)

$$-x_c + a + log(\sum_{k=1}^{M} e^{x_k - a})$$

(since $y_c = 1$)

Softmax Cross Entropy Derivative

$$
\begin{aligned}
\frac{\partial L(\hat{y}, y)}{\partial x_j} &= \frac{\partial}{\partial x_j}\left(-\sum_{i=1}^{K} y_i \log \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}}\right) \\
&= \frac{\partial}{\partial x_j}\left(-\sum_{i=1}^{K} y_i (\log e^{x_i} - \log \sum_{k=1}^{K} e^{x_k})\right) \\
&= \frac{\partial}{\partial x_j}\left(\sum_{i=1}^{K} -y_i \log e^{x_i} + \sum_{i=1}^{K} y_i \log \sum_{k=1}^{K} e^{x_k}\right) \\
&= \frac{\partial}{\partial x_j}\left(\sum_{i=1}^{K} -y_i x_i + \sum_{i=1}^{K} y_i \log \sum_{k=1}^{K} e^{x_k}\right)
\end{aligned}
$$

[Source](#)

Softmax Cross Entropy Derivative Continued (1)

$$
\begin{aligned}
\frac{\partial L(\hat{y}, y)}{\partial x_j} &= \frac{\partial}{\partial x_j}(-y_j x_j) + \frac{\partial}{\partial x_j} \log \sum_{k=1}^{K} e^{x_k} \\
&= -y_j + \frac{1}{\sum_{k=1}^{K} e^{x_k}} \cdot \left(\frac{\partial}{\partial x_j} \sum_{k=1}^{K} e^{x_k} \right) \\
&= -y_j + \frac{1}{\sum_{k=1}^{K} e^{x_k}} \cdot (e^{x_j}) \\
&= -y_j + \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}
\end{aligned}
$$

Sourc
e

Softmax Cross Entropy Derivative Continued (2)

$$
\begin{aligned}
\frac{\partial L(\hat{y}, y)}{\partial x_j} &= -y_j + \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \\
&= -y_j + \sigma(x_j) \\
&= \sigma(x_j) - y_j \\
&= \hat{y}_j - y_j
\end{aligned}
$$

Sourc
e

Linear Layer Derivative

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix} \qquad W = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{pmatrix} \tag{1}$$

$$Y = XW \tag{2}$$

$$= \begin{pmatrix} x_{1,1}w_{1,1} + x_{1,2}w_{2,1} & x_{1,1}w_{1,2} + x_{1,2}w_{2,2} & x_{1,1}w_{1,3} + x_{1,2}w_{2,3} \\ x_{2,1}w_{1,1} + x_{2,2}w_{2,1} & x_{2,1}w_{1,2} + x_{2,2}w_{2,2} & x_{2,1}w_{1,3} + x_{2,2}w_{2,3} \end{pmatrix} \tag{3}$$

$$\frac{\partial L}{\partial Y} = \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \tag{4}$$

[Source](#)

Linear Layer Derivative Continued (1)

By the chain rule, we know that:

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial X} \qquad\qquad \frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial W} \tag{5}$$

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix} \implies \frac{\partial L}{\partial X} = \begin{pmatrix} \frac{\partial L}{\partial x_{1,1}} & \frac{\partial L}{\partial x_{1,2}} \\ \frac{\partial L}{\partial x_{2,1}} & \frac{\partial L}{\partial x_{2,2}} \end{pmatrix} \tag{6}$$

[Sourc
e]

Linear Layer Derivative Continued (2)

$$\frac{\partial L}{\partial x_{1,1}} = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\partial L}{\partial y_{i,j}} \frac{\partial y_{i,j}}{\partial x_{1,1}} = \frac{\partial L}{\partial Y} \cdot \frac{\partial Y}{\partial x_{1,1}} \tag{7}$$

$$\frac{\partial Y}{\partial x_{1,1}} = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ 0 & 0 & 0 \end{pmatrix} \tag{8}$$

Sourc
e

Linear Layer Derivative Continued (3)

Now combining Equations 6, 7, and 8 gives:

$$\frac{\partial L}{\partial x_{1,1}} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial x_{1,1}} \tag{9}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \cdot \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ 0 & 0 & 0 \end{pmatrix} \tag{10}$$

$$= \frac{\partial L}{\partial y_{1,1}} w_{1,1} + \frac{\partial L}{\partial y_{1,2}} w_{1,2} + \frac{\partial L}{\partial y_{1,3}} w_{1,3} \tag{11}$$

Source

Linear Layer Derivative Continued (4)

$$\frac{\partial L}{\partial x_{1,2}} = \frac{\partial L}{\partial Y}\frac{\partial Y}{\partial x_{1,2}} \tag{12}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \cdot \begin{pmatrix} w_{2,1} & w_{2,2} & w_{2,3} \\ 0 & 0 & 0 \end{pmatrix} \tag{13}$$

$$= \frac{\partial L}{\partial y_{1,1}} w_{2,1} + \frac{\partial L}{\partial y_{1,2}} w_{2,2} + \frac{\partial L}{\partial y_{1,3}} w_{2,3} \tag{14}$$

Sourc
e

Linear Layer Derivative Continued (5)

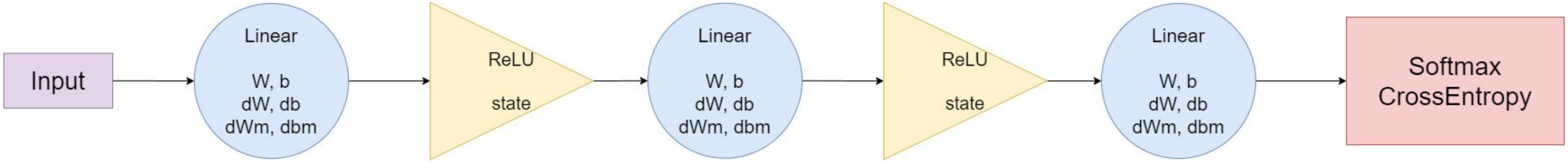
$$\frac{\partial L}{\partial X} = \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} w_{1,1} + \frac{\partial L}{\partial y_{1,2}} w_{1,2} + \frac{\partial L}{\partial y_{1,3}} w_{1,3} & \frac{\partial L}{\partial y_{1,1}} w_{2,1} + \frac{\partial L}{\partial y_{1,2}} w_{2,2} + \frac{\partial L}{\partial y_{1,3}} w_{2,3} \\ \frac{\partial L}{\partial y_{2,1}} w_{1,1} + \frac{\partial L}{\partial y_{2,2}} w_{1,2} + \frac{\partial L}{\partial y_{2,3}} w_{1,3} & \frac{\partial L}{\partial y_{2,1}} w_{2,1} + \frac{\partial L}{\partial y_{2,2}} w_{2,2} + \frac{\partial L}{\partial y_{2,3}} w_{2,3} \end{pmatrix} \tag{22}$$

$$= \begin{pmatrix} \frac{\partial L}{\partial y_{1,1}} & \frac{\partial L}{\partial y_{1,2}} & \frac{\partial L}{\partial y_{1,3}} \\ \frac{\partial L}{\partial y_{2,1}} & \frac{\partial L}{\partial y_{2,2}} & \frac{\partial L}{\partial y_{2,3}} \end{pmatrix} \begin{pmatrix} w_{1,1} & w_{2,1} \\ w_{1,2} & w_{2,2} \\ w_{1,3} & w_{2,3} \end{pmatrix} \tag{23}$$

$$= \boxed{\frac{\partial L}{\partial Y} W^T} \tag{24}$$

[Source](#)

MLP Forward



Input
Linear
W, b
dW, db
dWm, dbm
ReLU
state
Linear
W, b
dW, db
dWm, dbm
ReLU
state
Linear
W, b
dW, db
dWm, dbm
Softmax
CrossEntropy

MLP Backward