Neural Networks

Seminar - Week 3

Igor Janos

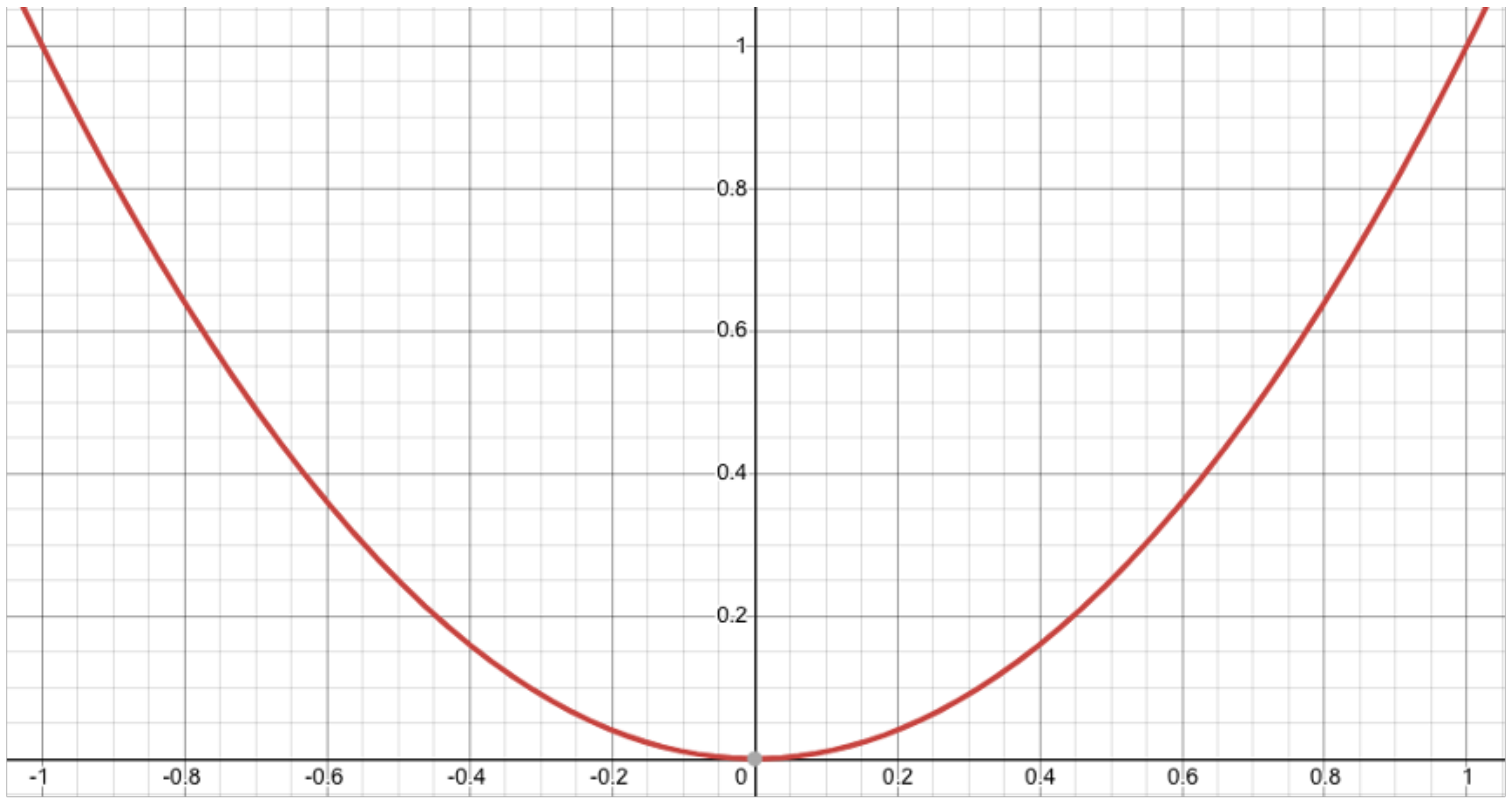
Seminar - Week 3

Assignment 2 Goals

- Implement loss functions - **Mean Square Error, Binary Cross-Entropy**
- Implement backward pass and derivatives for
    - Linear layer
    - Activation functions
    - Loss functions
- Implement backward pass for Multi-layer Perceptron

Loss Functions

Square Error - L2 Loss



$$\mathcal{L}(\hat{y}, y) = (y - \hat{y})^2$$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}} = -2(y - \hat{y})$$

Example - Logistic Regression

Functions and Their Derivatives

- Logistic loss

$$\mathcal{L}(a, y) = -(y \log a + (1 - y) \log(1 - a))$$

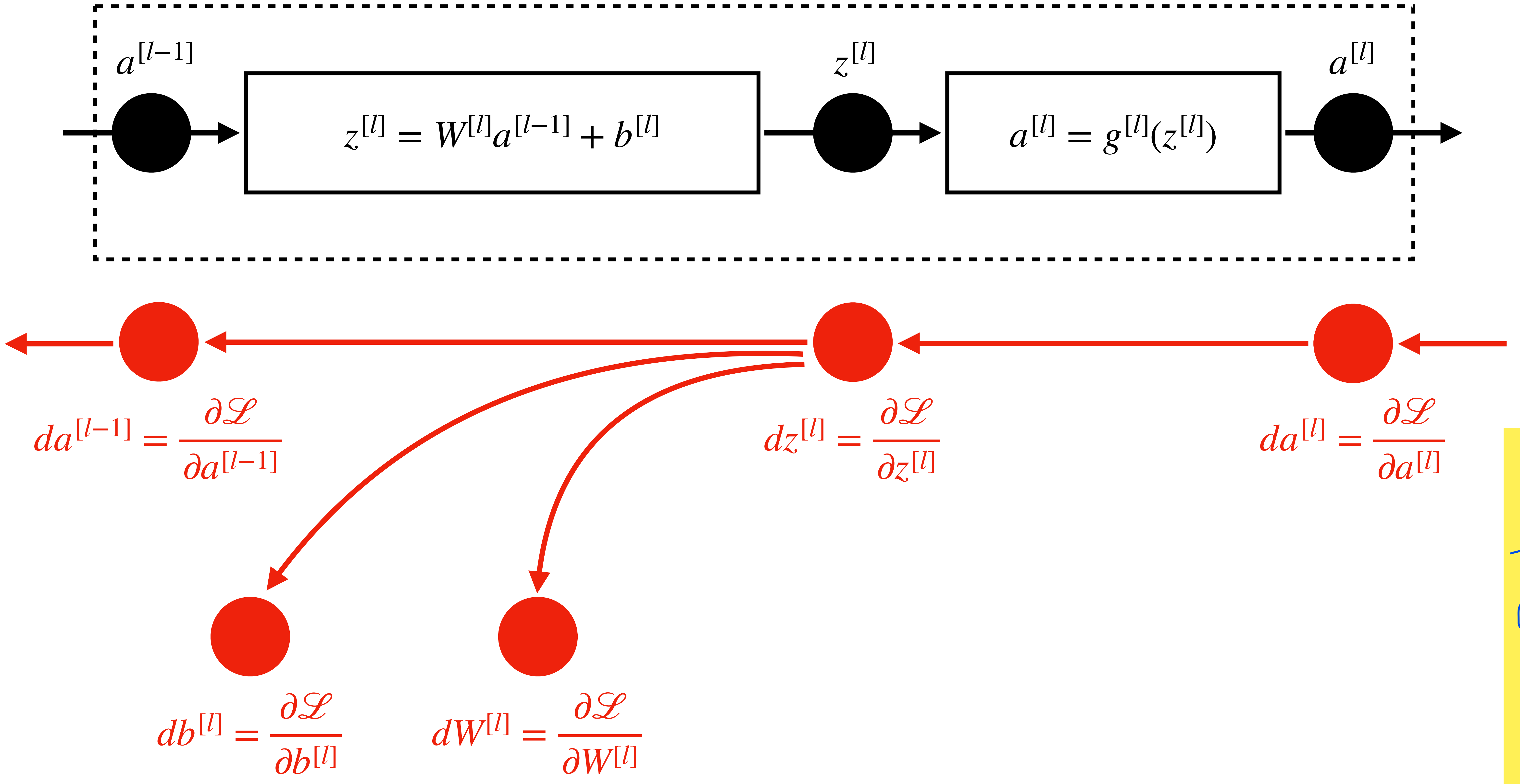
$$\frac{\partial \mathcal{L}(a, y)}{\partial a} = \frac{-y}{a} + \frac{(1 - y)}{(1 - a)}$$

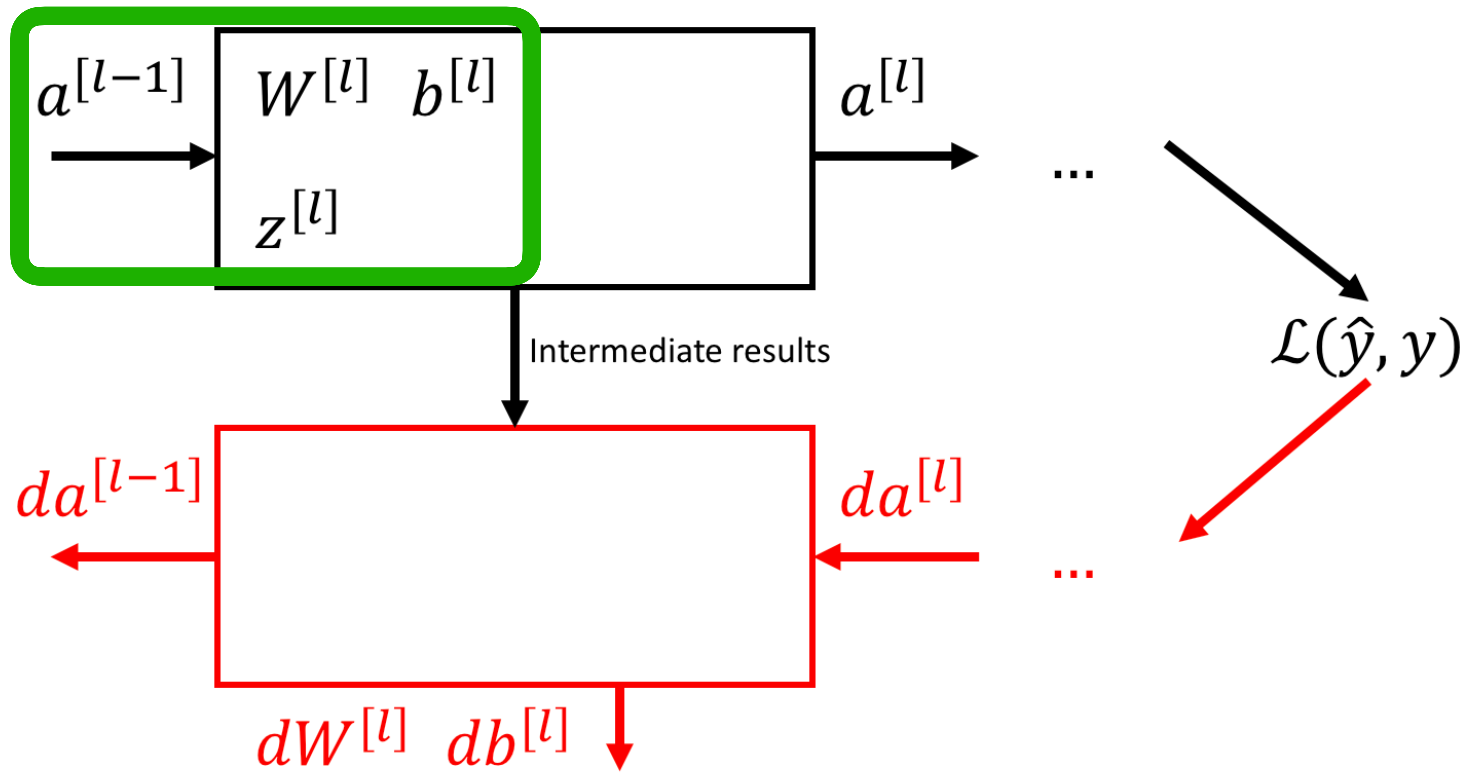
$$\frac{d}{dx} \ln(x) = \frac{1}{x}$$

$$\frac{d}{dx} \ln\left[f(x)\right] = \frac{1}{f(x)} f'(x)$$

Back-propagation with Neural Network

Backward Pass





$$dz^{[l]} = g^{[l]\prime}(z^{[l]}) * da^{[l]}$$

$$da^{[l-1]} = W^{[l]^T} dz^{[l]}$$

$$db^{[l]} = dz^{[l]}$$

$$dW^{[l]} = dz^{[l]} a^{[l-1]^T}$$

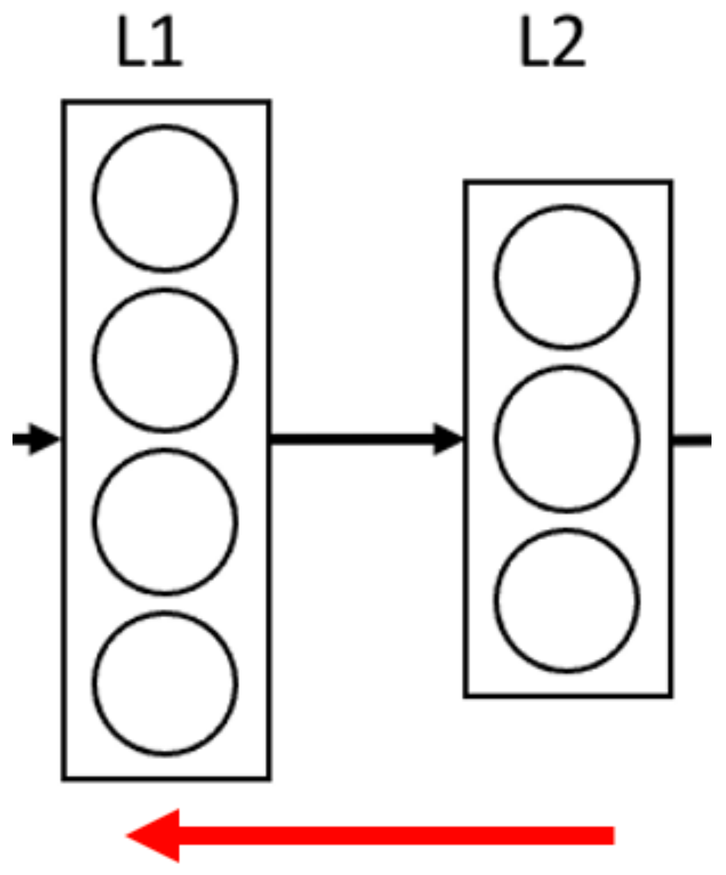
Back-propagation with Neural Network

Shapes - Vectorized

$$dZ^{[2]} = g^{[2]'}(Z^{[2]}) * dA^{[2]}$$

$$dA^{[1]} = W^{[2]^T} dZ^{[2]}$$

$$db^{[2]} = mean(dZ^{[2]})$$



$$\begin{bmatrix} db^{[2]} \\ (3 \times 1) \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \overbrace{\cdots \qquad\qquad \cdots}^{m} \\ \cdots \quad dZ^{[2]} \quad \cdots \\ (3 \times m) \end{bmatrix} \left. \begin{bmatrix} 1 \\ 1 \\ \vdots \\ (m \times 1) \end{bmatrix} \right\} m$$

$$dZ^{[l]} = g^{[l]'}(Z^{[l]}) * dA^{[l]}$$

$$dA^{[l-1]} = W^{[l]^T} dZ^{[l]}$$

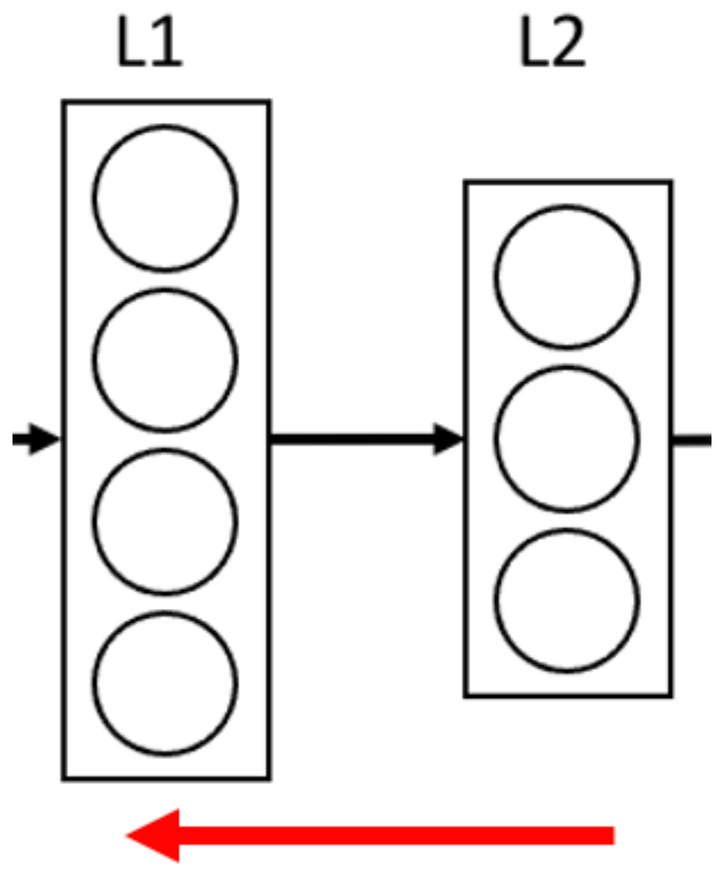
$$db^{[l]} = mean(dZ^{[l]})$$

$$dW^{[l]} = mean(dZ^{[l]} A^{[l-1]^T})$$

Back-propagation with Neural Network

Shapes - Vectorized

$$dZ^{[2]} = g^{[2]'}(Z^{[2]}) * dA^{[2]}$$
$$dA^{[1]} = W^{[2]^T} dZ^{[2]}$$
$$db^{[2]} = mean(dZ^{[2]})$$
$$dW^{[2]} = mean(dZ^{[2]} A^{[1]^T})\ m$$

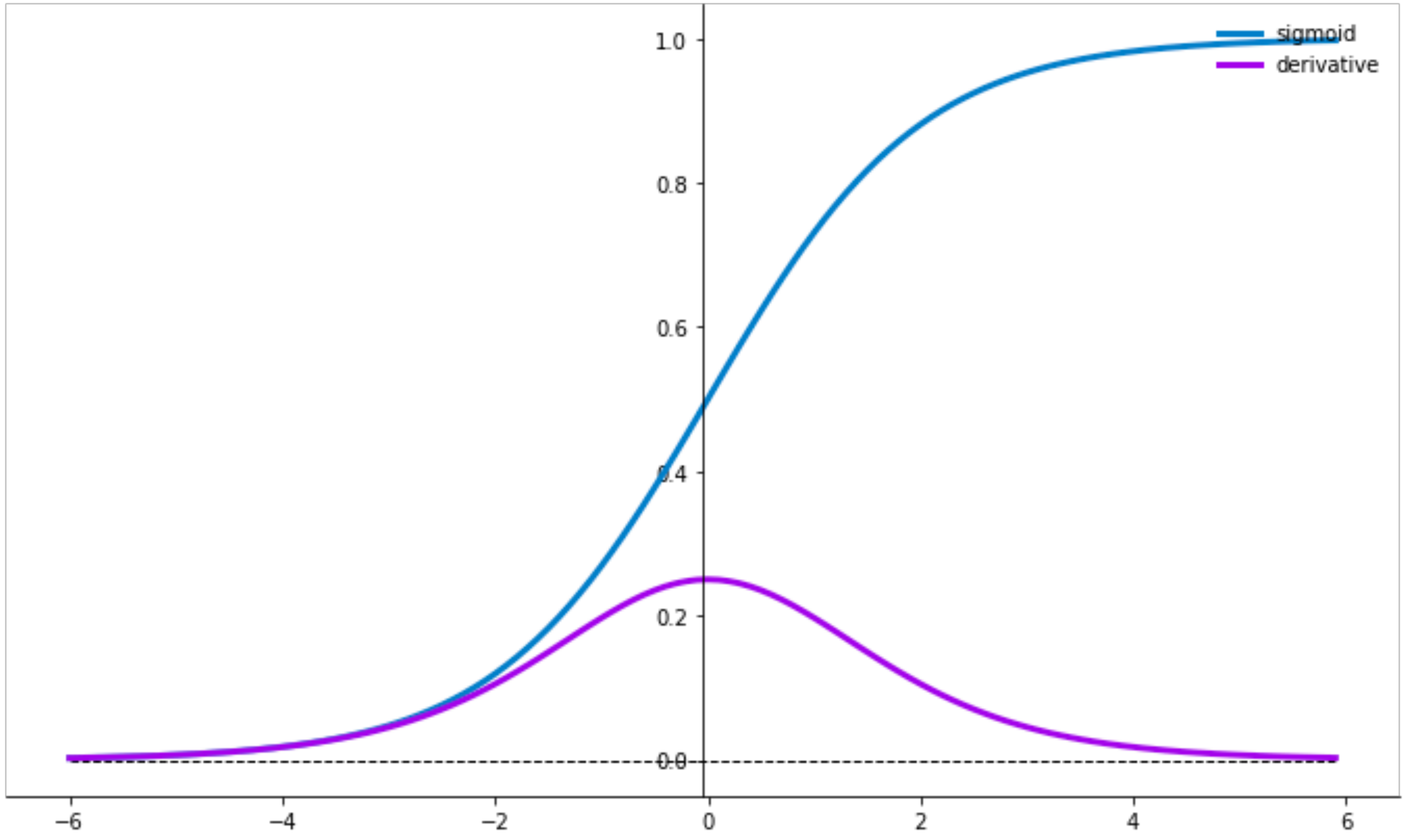


$$\begin{bmatrix} dW^{[2]} \\ (3 \times 4) \end{bmatrix} = \frac{1}{m} \overbrace{\begin{bmatrix} \cdots & & \cdots \\ \cdots & dZ^{[2]} & \cdots \\ & (3 \times m) & \end{bmatrix}}^{m} \left.\begin{bmatrix} - & A^{[1]^T} & - \\ \cdots & (m \times 4) & \cdots \\ \cdots & & \cdots \\ \cdots & & \cdots \end{bmatrix}\right\} m$$

$$dZ^{[l]} = g^{[l]'}(Z^{[l]}) * dA^{[l]}$$
$$dA^{[l-1]} = W^{[l]^T} dZ^{[l]}$$
$$db^{[l]} = mean(dZ^{[l]})$$
$$dW^{[l]} = mean(dZ^{[l]} A^{[l-1]^T})$$

Activation Functions

Sigmoid

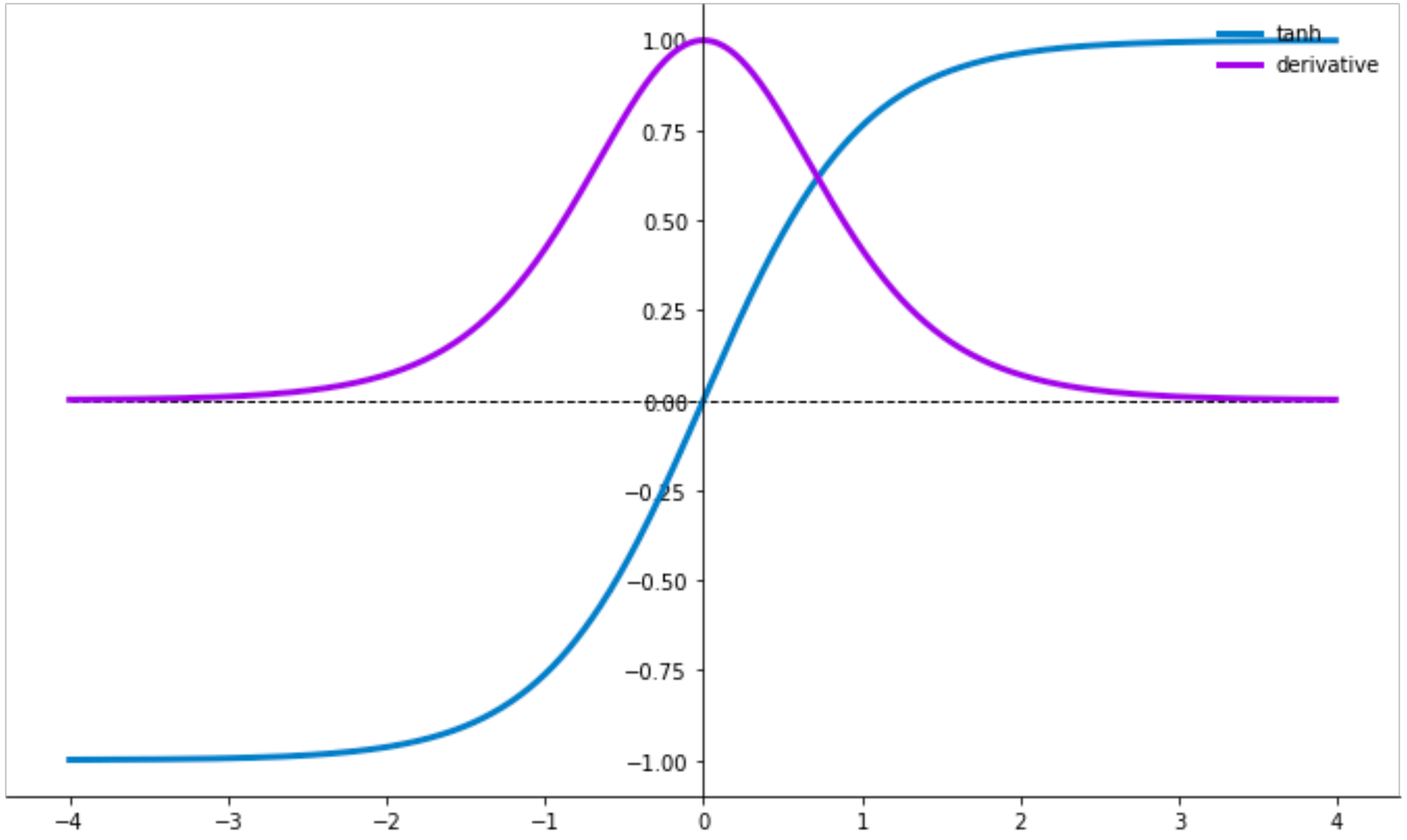


$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

Activation Functions

TanH

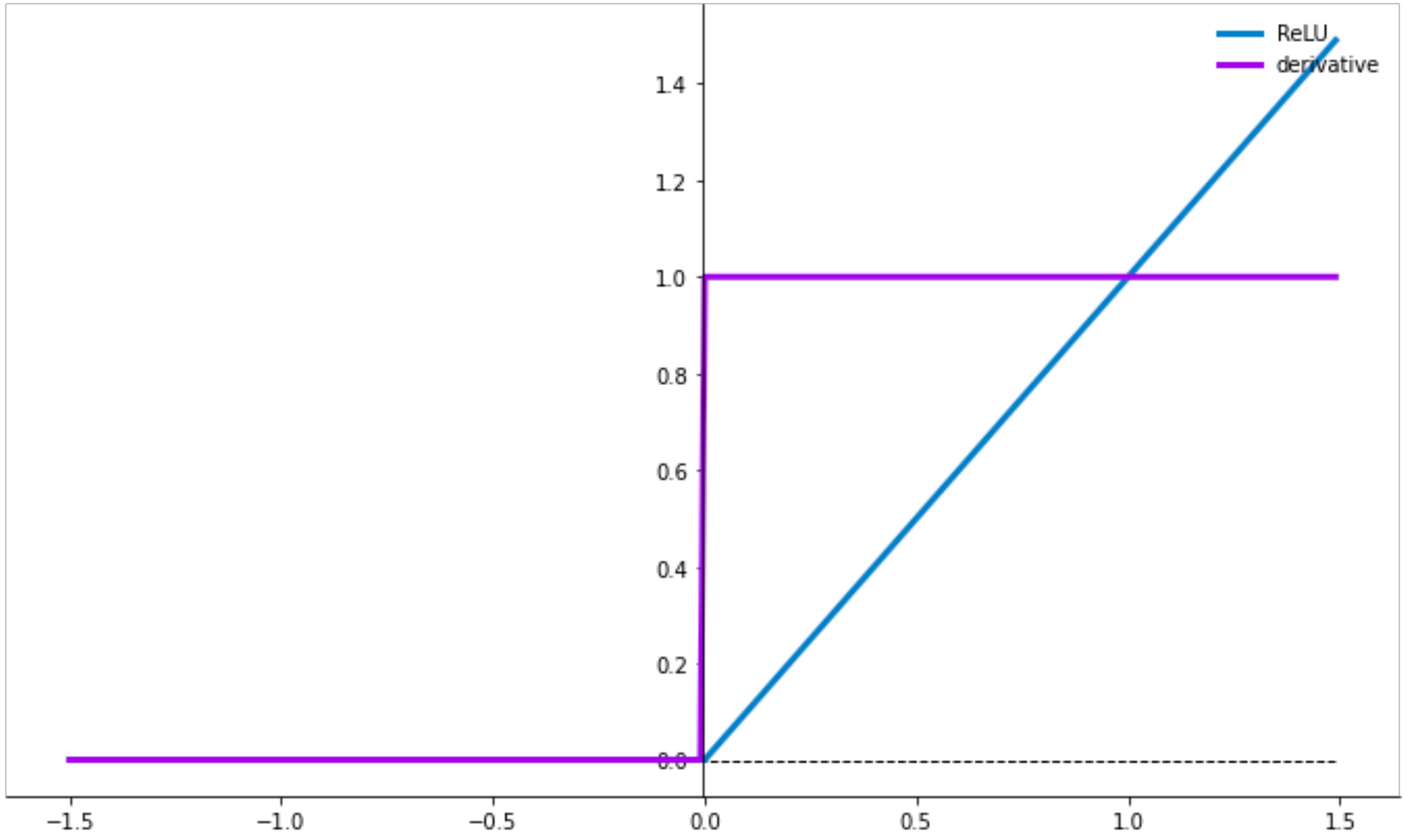


$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$\tanh'(z) = 1 - \tanh^2(z)$$

Activation Functions

ReLU

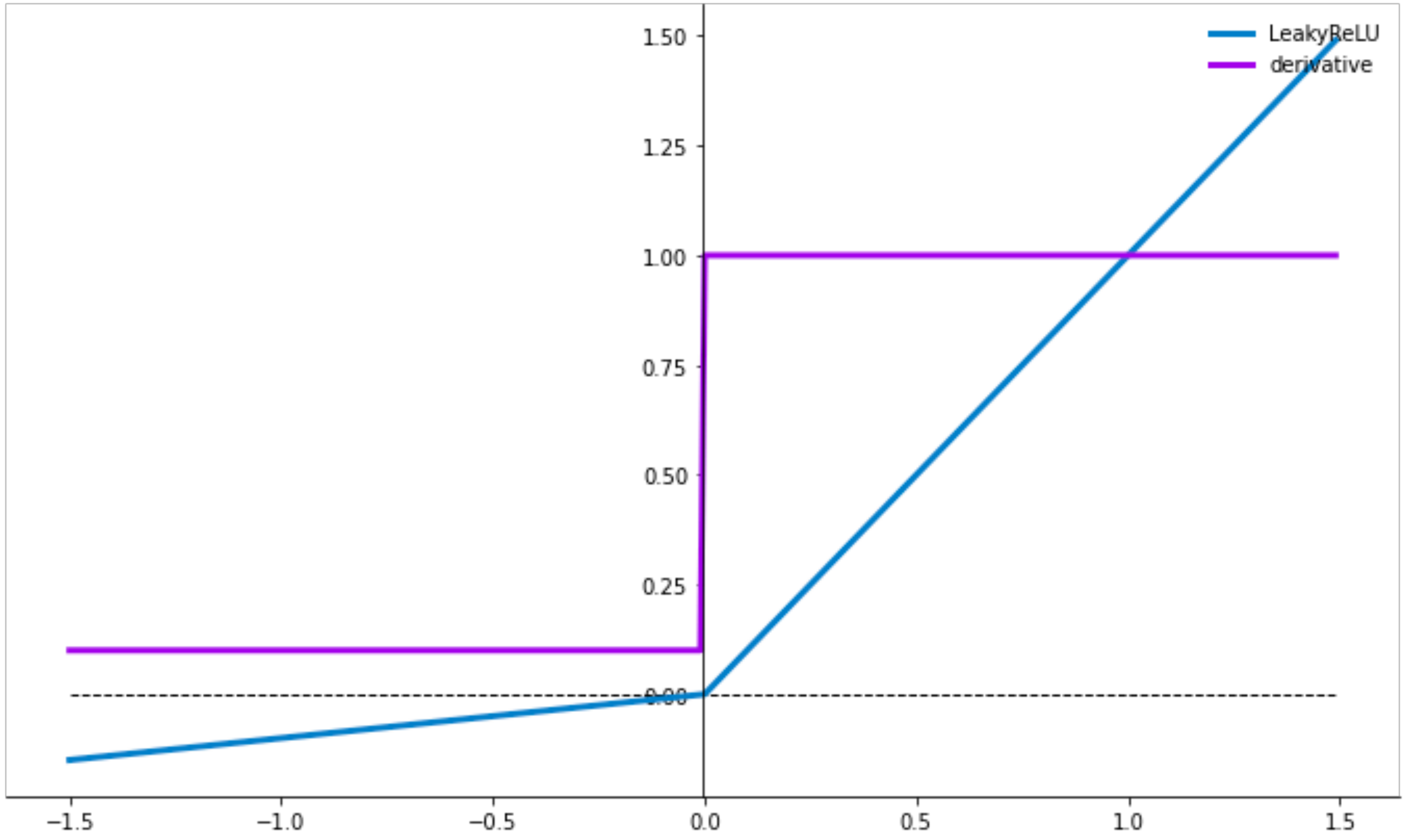


$$ReLU(z) = \max(0, z)$$

$$ReLU'(z) = \begin{cases} 1 & if\ z > 0 \\ 0 & else \end{cases}$$

Activation Functions

LeakyReLU



$$\alpha = 0.1$$

$$lReLU(z) = \max(\alpha z, z)$$

$$lReLU'(z) = \begin{cases} 1 & if\, z > 0 \\ \alpha & else \end{cases}$$