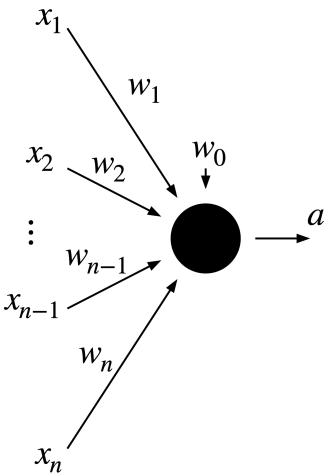
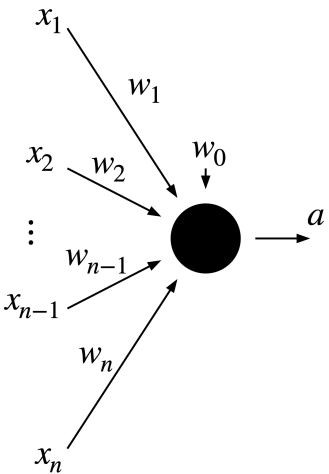
Feedforward Artificial Neural Networks (FANN)

Alejandro Veloz

Single neuron operations



Single neuron operations

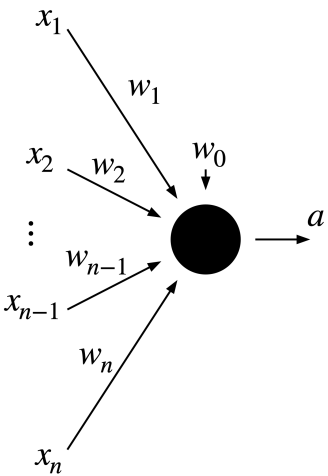


- It receives **input data point:** $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$
- and computes the **output:**

$$\underbrace{a}_{\text{activation}} = \sigma\big(\underbrace{z}_{\text{preactivation}}\big) = \sigma\left(\underbrace{\sum_{j=1}^{n} x_j w_j + w_0}_{z}\right)$$

- $z = \sum_{j=1}^{n} x_j w_j + w_0$ is called **preactivation.**
- $\sigma(z)$ is the **activation function.**
- $\mathbf{w} = [w_1, \dots, w_n]^\top$ are **weights** (parameters).
- w_0 is the **bias parameters.**

Single neuron operations

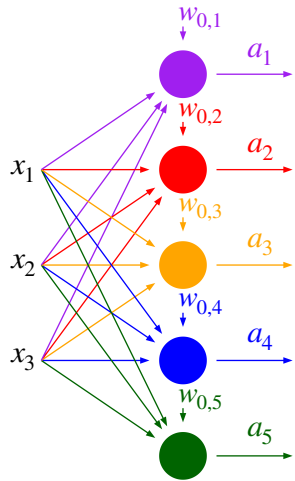


$$
\begin{aligned}
a &= \sigma\left(\underbrace{\sum_{j=1}^{n} x_j w_j + w_0}_{z}\right) \\
&= \sigma\left(\underbrace{\begin{bmatrix} w_1 & \dots & w_n \end{bmatrix}}_{\mathbf{w}^\top} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} + w_0\right) \\
&= \sigma\left(\mathbf{w}^\top \mathbf{x} + w_0\right)
\end{aligned}
$$

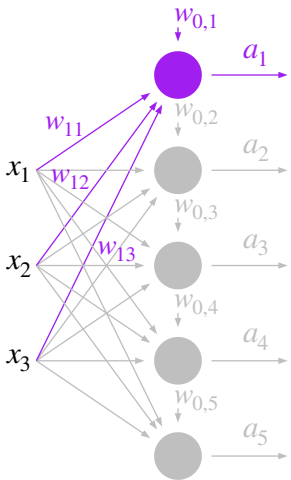
Single layer operations

Consider 5 neurons that operates “in parallel”.

It receives **input** $\mathbf{x}$ and computes the **output** $\mathbf{a}$



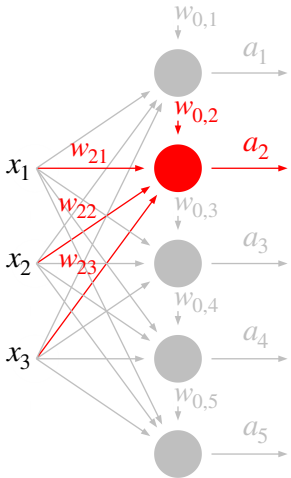
Single layer operations



$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \sigma\left(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}\right)$$

$$= \sigma\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{0,1} \\ w_{0,2} \\ w_{0,3} \\ w_{0,4} \\ w_{0,5} \end{bmatrix}\right)$$

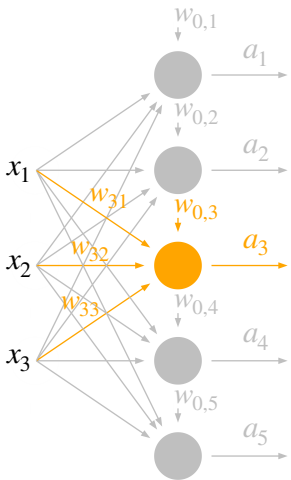
Single layer operations



$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \sigma\left(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}\right)$$

$$= \sigma\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{0,1} \\ w_{0,2} \\ w_{0,3} \\ w_{0,4} \\ w_{0,5} \end{bmatrix}\right)$$

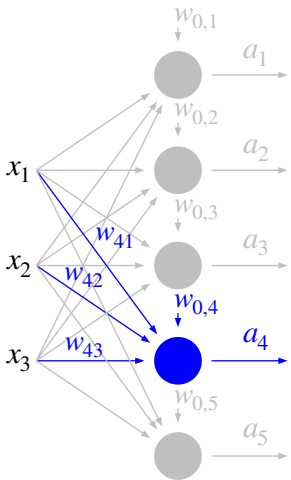
Single layer operations



$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \sigma\left(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}\right)$$

$$= \sigma\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{0,1} \\ w_{0,2} \\ w_{0,3} \\ w_{0,4} \\ w_{0,5} \end{bmatrix}\right)$$

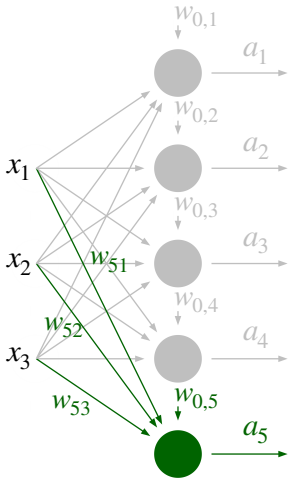
Single layer operations



$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \sigma\left(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}\right)$$

$$= \sigma\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{0,1} \\ w_{0,2} \\ w_{0,3} \\ w_{0,4} \\ w_{0,5} \end{bmatrix}\right)$$

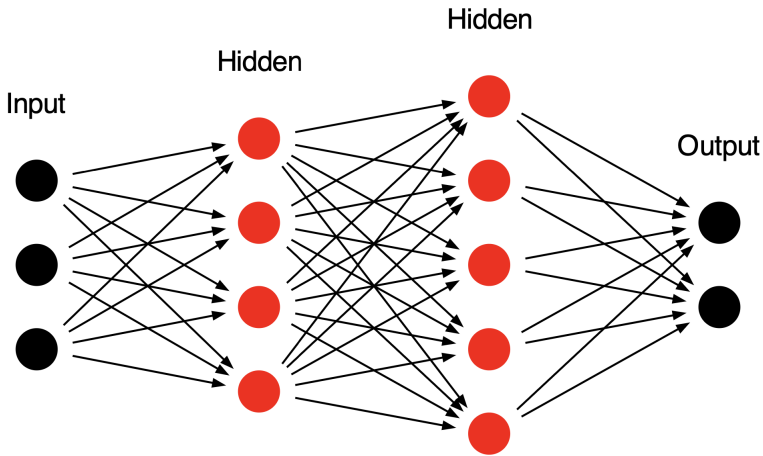
Single layer operations



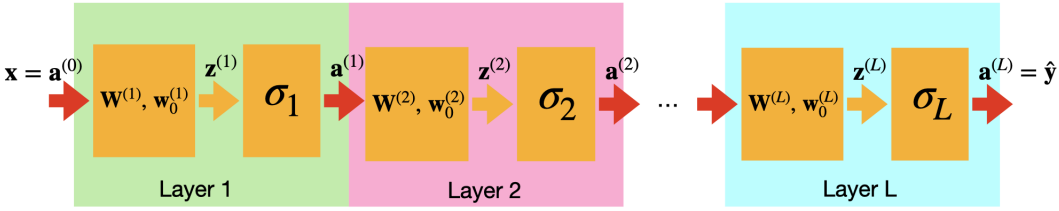
$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \sigma\left(\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}\right)$$

$$= \sigma\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} w_{0,1} \\ w_{0,2} \\ w_{0,3} \\ w_{0,4} \\ w_{0,5} \end{bmatrix}\right)$$

Network with multiple layers



Network with multiple layers



Single layer operations

Consider a network of L layers (one input layer and $L-1$ layers of neurons).

Each layer is composed by $n^{(\ell)}$ neurons, $\ell \in \{0, 1, \dots, L\}$.

The input vector for layer ℓ is:

$$\mathbf{a}^{(\ell-1)} = \begin{bmatrix} a_1^{(\ell-1)} \\ \vdots \\ a_{n^{(\ell-1)}}^{(\ell-1)} \end{bmatrix}$$

The output vector for layer ℓ is:

$$\mathbf{a}^{(\ell)} = \begin{bmatrix} a_1^{(\ell)} \\ \vdots \\ a_{n^{(\ell)}}^{(\ell)} \end{bmatrix}$$

Single layer operations

$$\underbrace{\begin{bmatrix} a_1^{(\ell)} \\ \vdots \\ a_{n^{(\ell)}}^{(\ell)} \end{bmatrix}}_{\mathbf{a}^{(\ell)}} = \sigma_\ell \left(\begin{bmatrix} z_1^{(\ell)} \\ \vdots \\ z_{n^{(\ell)}}^{(\ell)} \end{bmatrix} \right)$$

$$= \sigma_\ell \left(\begin{bmatrix} \mathbf{w}_1^{(\ell)\top} \\ \vdots \\ \mathbf{w}_{n^{(\ell)}}^{(\ell)\top} \end{bmatrix} \begin{bmatrix} a_1^{(\ell-1)} \\ \vdots \\ a_{n^{(\ell-1)}}^{(\ell-1)} \end{bmatrix} + \begin{bmatrix} w_{0,1}^{(\ell)} \\ \vdots \\ w_{0,n^{(\ell)}}^{(\ell)} \end{bmatrix} \right) = \sigma \left(\mathbf{W}^{(\ell)} \mathbf{a}^{(\ell-1)} + \mathbf{w}_0^{(\ell)} \right)$$

Feedforward network operations

Given the input vector $\mathbf{x}$ and a network of L layers:

Layer 1: $\mathbf{a}^{(1)} = \sigma_1\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{w}_0^{(1)}\right)$

Feedforward network operations

Given the input vector $\mathbf{x}$ and a network of L layers:

Layer 1: $\mathbf{a}^{(1)} = \sigma_1\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{w}_0^{(1)}\right)$

Layer 2: $\mathbf{a}^{(2)} = \sigma_2\left(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{w}_0^{(2)}\right)$

Feedforward network operations

Given the input vector $\mathbf{x}$ and a network of L layers:

Layer 1: $\mathbf{a}^{(1)} = \sigma_1\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{w}_0^{(1)}\right)$

Layer 2: $\mathbf{a}^{(2)} = \sigma_2\left(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{w}_0^{(2)}\right)$

$\vdots$

Feedforward network operations

Given the input vector $\mathbf{x}$ and a network of L layers:

Layer 1: $\mathbf{a}^{(1)} = \sigma_1\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{w}_0^{(1)}\right)$

Layer 2: $\mathbf{a}^{(2)} = \sigma_2\left(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{w}_0^{(2)}\right)$

$\vdots$

Layer ℓ: $\mathbf{a}^{(\ell)} = \sigma_\ell\left(\mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{w}_0^{(\ell)}\right)$

Feedforward network operations

Given the input vector $\mathbf{x}$ and a network of L layers:

Layer 1: $\mathbf{a}^{(1)} = \sigma_1\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{w}_0^{(1)}\right)$

Layer 2: $\mathbf{a}^{(2)} = \sigma_2\left(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{w}_0^{(2)}\right)$

$\vdots$

Layer ℓ: $\mathbf{a}^{(\ell)} = \sigma_\ell\left(\mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{w}_0^{(\ell)}\right)$

$\vdots$

Feedforward network operations

Given the input vector $\mathbf{x}$ and a network of L layers:

Layer 1: $\mathbf{a}^{(1)} = \sigma_1\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{w}_0^{(1)}\right)$

Layer 2: $\mathbf{a}^{(2)} = \sigma_2\left(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{w}_0^{(2)}\right)$

$\vdots$

Layer ℓ: $\mathbf{a}^{(\ell)} = \sigma_\ell\left(\mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{w}_0^{(\ell)}\right)$

$\vdots$

Layer L: $\hat{\mathbf{y}} = \mathbf{a}^{(L)} = \sigma_L\left(\mathbf{W}^{(L)}\mathbf{a}^{(L-1)} + \mathbf{w}_0^{(L)}\right)$

Feedforward network operations

By defining $\mathbf{a}^{(0)} = \mathbf{x}$:

Layer 1: $\mathbf{a}^{(1)} = \sigma_1\left(\mathbf{W}^{(1)}\mathbf{a}^{(0)} + \mathbf{w}_0^{(1)}\right)$

Layer 2: $\mathbf{a}^{(2)} = \sigma_2\left(\mathbf{W}^{(2)}\mathbf{a}^{(1)} + \mathbf{w}_0^{(2)}\right)$

$\vdots$

Layer ℓ: $\mathbf{a}^{(\ell)} = \sigma_\ell\left(\mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{w}_0^{(\ell)}\right)$

$\vdots$

Layer L: $\mathbf{a}^{(L)} = \sigma_L\left(\mathbf{W}^{(L)}\mathbf{a}^{(L-1)} + \mathbf{w}_0^{(L)}\right)$

Learning the network parameters: Backpropagation

Training process

Two steps:

1. **Backward Pass (Backpropagation of Error):** Compute the derivatives of the error function with respect to the weights by propagating errors backward through the network.
2. **Weight Update:** Adjust the weights using the calculated derivatives, typically by applying an optimization method such as gradient descent.

Training process

- Training dataset: $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_D, \mathbf{y}_D)\}$

Training process

- Training dataset: $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_D, \mathbf{y}_D)\}$
- Loss function $\text{loss}(\texttt{guess}, \texttt{actual})$:

$$J(\mathbf{W}, \mathbf{W}_0) = \sum_{d=1}^{D} \text{loss}\left(\text{NN}(\mathbf{x}_d; \mathbf{W}, \mathbf{W}_0), \mathbf{y}_d\right),$$

where NN is the output of our neural network for a given input.

Training process

- Training dataset: $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_D, \mathbf{y}_D)\}$
- Loss function $\text{loss}(\texttt{guess}, \texttt{actual})$:

$$J(\mathbf{W}, \mathbf{W}_0) = \sum_{d=1}^{D} \text{loss}\left(\text{NN}(\mathbf{x}_d; \mathbf{W}, \mathbf{W}_0), \mathbf{y}_d\right),$$

  where NN is the output of our neural network for a given input.

- We can do (stochastic/batch) gradient descent, adjusting the weights $\mathbf{W}, \mathbf{W}_0$ to minimize J.

Define $\hat{\mathbf{y}} = \text{NN}\left(\mathbf{x}; \mathbf{W}, \mathbf{W}_0\right)$ the network response under the data point $\mathbf{x}$.

Define $\hat{\mathbf{y}} = \text{NN}\left(\mathbf{x}; \mathbf{W}, \mathbf{W}_0\right)$ the network response under the data point $\mathbf{x}$.

Gradient descent rule:

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta \nabla_{\mathbf{W}} \text{loss}(\hat{\mathbf{y}}, \mathbf{y})$$
$$\mathbf{W}_0(t+1) = \mathbf{W}_0(t) - \eta \nabla_{\mathbf{W}_0} \text{loss}(\hat{\mathbf{y}}, \mathbf{y})$$

Recall:

$$\mathbf{a}^{(\ell)} = \sigma_\ell\left(\mathbf{z}^{(\ell)}\right) \qquad \mathbf{z}^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{w}_0^{(\ell)}$$

Recall:

$$\mathbf{a}^{(\ell)} = \sigma_\ell\left(\mathbf{z}^{(\ell)}\right) \qquad \mathbf{z}^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{a}^{(\ell-1)} + \mathbf{w}_0^{(\ell)}$$

The derivative of the loss w.r.t. $\mathbf{W}^{(L)}$ is:

$$\frac{\partial \text{loss}}{\partial \mathbf{W}^{(L)}} = \underbrace{\frac{\partial \text{loss}}{\partial \mathbf{a}^{(L)}}}_{\text{depends on loss function}} \cdot \underbrace{\frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{z}^{(L)}}}_{\sigma_L'(\mathbf{z}^{(L)})} \cdot \underbrace{\frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{W}^{(L)}}}_{\mathbf{a}^{(L-1)}}$$

$$\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(L)}} = \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{z}^{(L)}} \frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{W}^{(L)}}$$

$$\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(L)}} = \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{z}^{(L)}} \frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{W}^{(L)}}$$

$$\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(L-1)}} = \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{z}^{(L)}} \frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{a}^{(L-1)}} \frac{\partial \mathbf{a}^{(L-1)}}{\partial \mathbf{z}^{(L-1)}} \frac{\partial \mathbf{z}^{(L-1)}}{\partial \mathbf{W}^{(L-1)}}$$

$$\vdots$$

$$\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(L)}} = \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{z}^{(L)}} \frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{W}^{(L)}}$$

$$\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(L-1)}} = \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{z}^{(L)}} \frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{a}^{(L-1)}} \frac{\partial \mathbf{a}^{(L-1)}}{\partial \mathbf{z}^{(L-1)}} \frac{\partial \mathbf{z}^{(L-1)}}{\partial \mathbf{W}^{(L-1)}}$$

$$\vdots$$

$$\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(L)}} = \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \sigma'_L(\mathbf{z}^{(L)}) \mathbf{a}^{(L-1)}$$

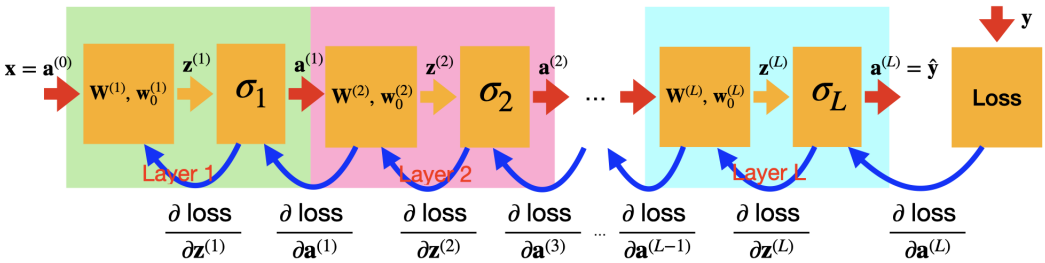
$$\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(L-1)}} = \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \sigma'_L(\mathbf{z}^{(L)}) \mathbf{W}^{(L)} \sigma'_{L-1}(\mathbf{z}^{(L-1)}) \mathbf{a}^{(L-2)}$$

$$\vdots$$

The general case

$$
\begin{aligned}
\frac{\partial \,\text{loss}}{\partial \mathbf{W}^{(\ell)}} &= \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \frac{\partial \mathbf{a}^{(L)}}{\partial \mathbf{z}^{(L)}} \frac{\partial \mathbf{z}^{(L)}}{\partial \mathbf{a}^{(L-1)}} \frac{\partial \mathbf{a}^{(L-1)}}{\partial \mathbf{z}^{(L-1)}} \frac{\partial \mathbf{z}^{(L-1)}}{\partial \mathbf{a}^{(L-2)}} \frac{\partial \mathbf{a}^{(L-2)}}{\partial \mathbf{z}^{(L-2)}} \\
&\qquad \dots \frac{\partial \mathbf{z}^{(\ell+1)}}{\partial \mathbf{a}^{(\ell)}} \frac{\partial \mathbf{a}^{(\ell)}}{\partial \mathbf{z}^{(\ell)}} \frac{\partial \mathbf{z}^{(\ell)}}{\partial \mathbf{W}^{(\ell)}} \\
&= \frac{\partial \,\text{loss}}{\partial \mathbf{a}^{(L)}} \sigma'_L \mathbf{W}^{(L)} \sigma'_{L-1} \mathbf{W}^{(L-1)} \sigma'_{L-2} \mathbf{W}^{(L-2)} \\
&\qquad \dots \mathbf{W}^{(\ell+1)} \sigma'_\ell \mathbf{a}^{(\ell-1)} \\
&= \frac{\partial \,\text{loss}}{\partial \mathbf{z}^{(\ell)}} \mathbf{a}^{(\ell-1)}
\end{aligned}
$$

Error backpropagation



Backpropagation

for $l = 1$ **to** L **do**
  $w_{ij}^{(l)} \sim \text{Gaussian}\left(0, 1/m^l\right)$; $w_{0j}^{(l)} \sim \text{Gaussian}(0,1)$
end for
for $t = 1$ **to** T **do**
  $i =$ random sample from $\{1, \ldots, n\}$, then define $\mathbf{a}^0 = \mathbf{x}^{(i)}$
  for $l = 1$ **to** L **do**
    $\mathbf{z}^{(l)} = \mathbf{W}^{l\top}\mathbf{a}^{(l-1)} + \mathbf{W}_0^{(l)}$; $\mathbf{a}^{(l)} = \sigma^{(l)}\left(\mathbf{z}^{(l)}\right)$
  end for
  $\text{loss} = \text{Loss}\left(\mathbf{a}^{(L)}, \mathbf{y}^{(\mathrm{i})}\right)$
  for $l = L$ **to** 1 **do**
    $\partial\text{loss}/\partial\mathbf{a}^{(l)} =$ if $l < L$ then $\partial\text{loss}/\partial\mathbf{z}^{(l+1)} \cdot \partial\mathbf{z}^{(l+1)}/\partial\mathbf{a}^{(l)}$ else $\partial\text{loss}/\partial\mathbf{a}^{(L)}$
    $\partial\text{loss}/\partial\mathbf{z}^{(l)} = \partial\text{loss}/\partial\mathbf{a}^{(l)} \cdot \partial\mathbf{a}^{(l)}/\partial\mathbf{z}^{(l)}$
    $\partial\text{loss}/\partial\mathbf{W}^{(l)} = \partial\text{loss}/\partial\mathbf{z}^{(l)}\partial\mathbf{z}^{(l)}/\partial\mathbf{W}^{(l)}$, $\partial\text{loss}/\partial\mathbf{W}_0^{(l)} = \partial\text{loss}/\partial\mathbf{z}^{(l)}\partial\mathbf{z}^{(l)}/\partial\mathbf{W}_0^{(l)}$
    $\mathbf{W}^{(l)} = \mathbf{W}^{(l)} - \eta(t)\partial\text{loss}/\partial\mathbf{W}^{(l)}$, $\mathbf{W}_0^{(l)} = \mathbf{W}_0^{(l)} - \eta(t)\partial\text{loss}/\partial\mathbf{W}_0^{(l)}$
  end for
end for
return $\mathbf{W}, \mathbf{W}_0$