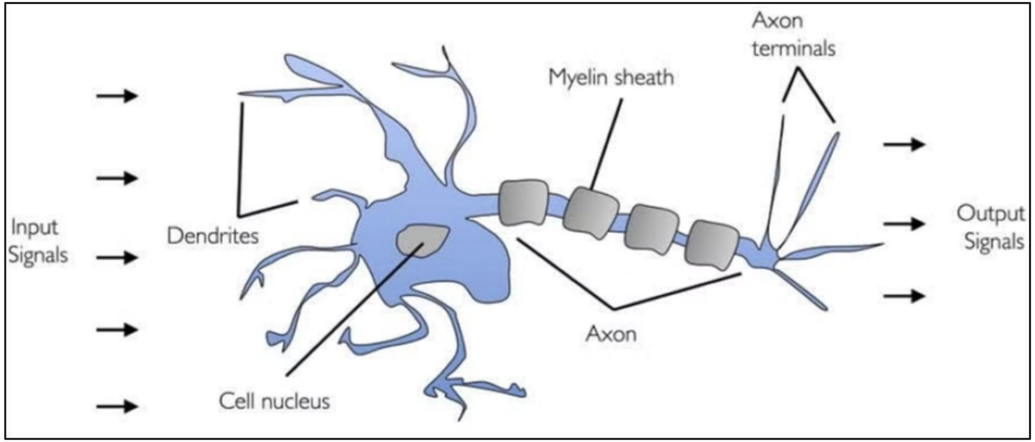
CIS 678 Machine Learning

Introduction to Neural Networks

Perceptron: the first NN

Motivation src: Biological neuron

Perceptron was introduced by **Frank Rosenblatt** in 1957.



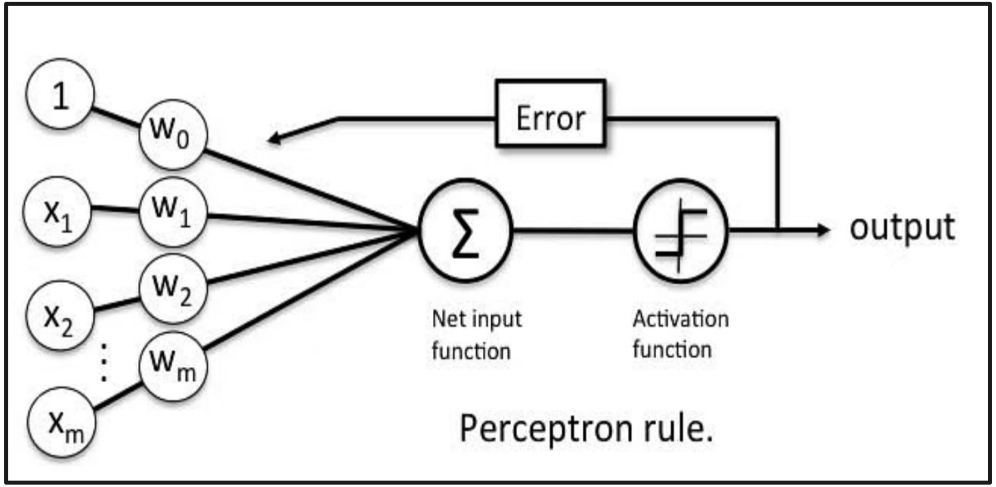
perceptron

Perceptron: the first NN

Motivation src: Biological neuron

Perceptron was introduced by **Frank Rosenblatt** in 1957.

A binary classifier



[perceptron]

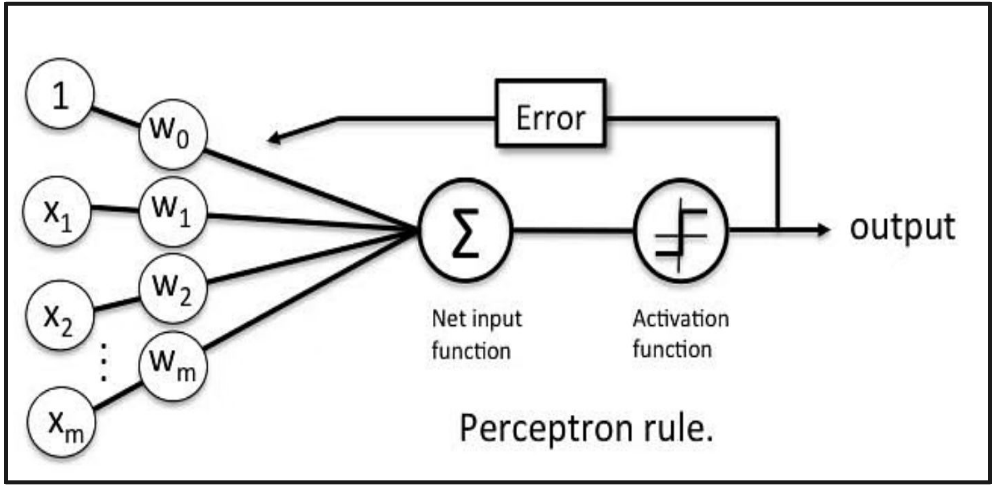
Perceptron: the first NN

Motivation src: Biological neuron

Perceptron was introduced by **Frank Rosenblatt** in 1957.

A binary classifier

[Professor's perceptron paved the way for AI – 60 years too soon]



perceptron

Feed-forward neural networks

x_1

x_2

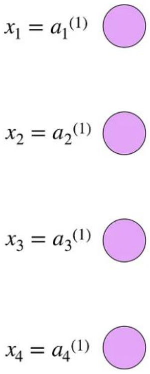
x_3

x_4

Input
(X)

mainly adapted from

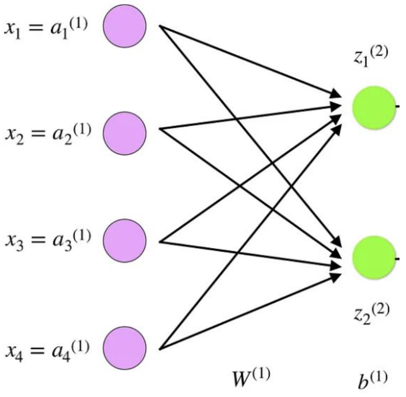
Feed-forward neural networks



A = X

[mainly adapted from]

Feed-forward neural networks



Input layer

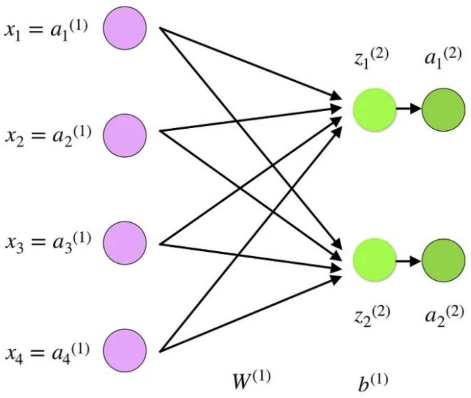
Hidden_1 layer

A = X

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

mainly adapted from

Feed-forward neural networks



Input layer

Hidden_1 layer

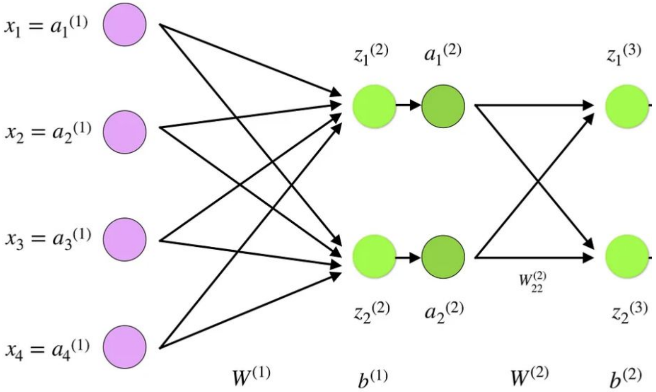
A = X

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

$$a^{(2)} = f(z^{(2)})$$

mainly adapted from

Feed-forward neural networks



Input layer

A = X

Hidden_1 layer

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

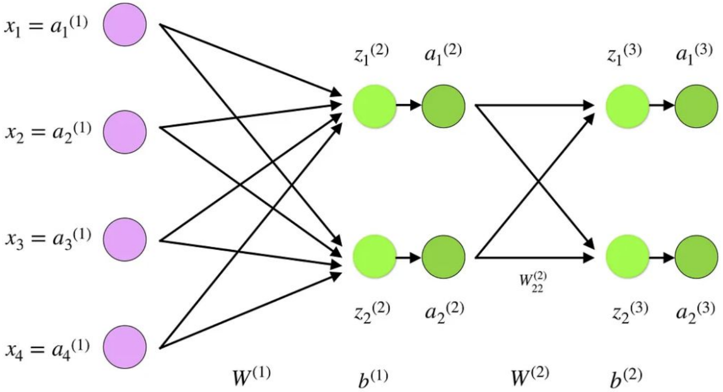
$$a^{(2)} = f(z^{(2)})$$

Hidden_2 layer

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)}$$

[mainly adapted from]

Feed-forward neural networks



Input layer

A = X

Hidden_1 layer

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

$$a^{(2)} = f(z^{(2)})$$

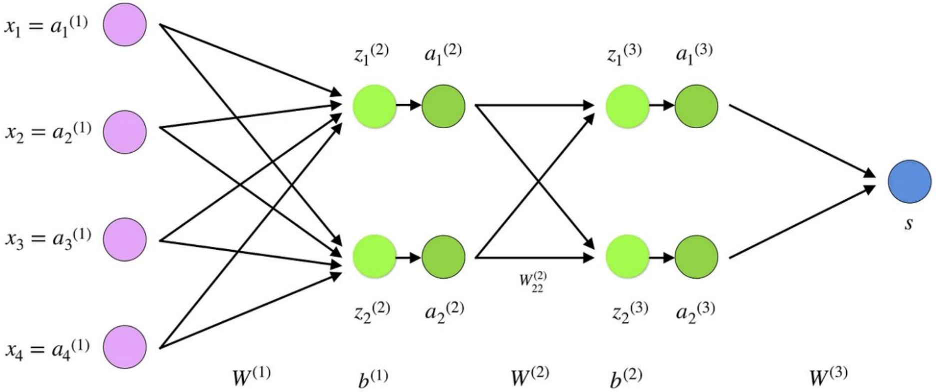
Hidden_2 layer

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)}$$

$$a^{(3)} = f(z^{(3)})$$

mainly adapted from

Feed-forward neural networks



Input layer

A = X

Hidden_1 layer

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

$$a^{(2)} = f(z^{(2)})$$

Hidden_2 layer

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)}$$

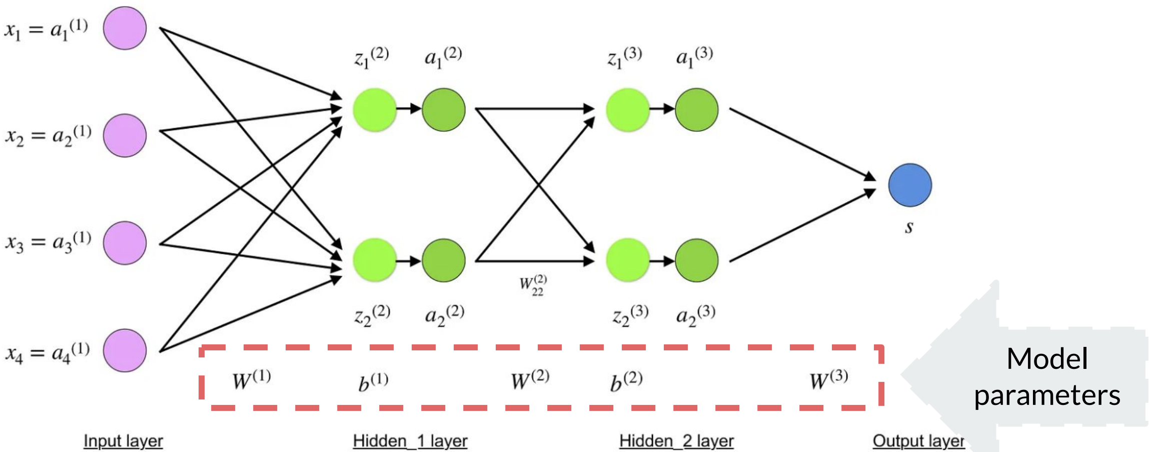
$$a^{(3)} = f(z^{(3)})$$

Output layer

$$s = W^{(3)}a^{(3)}$$

mainly adapted from

Feed-forward neural networks



A = X

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

$$a^{(2)} = f(z^{(2)})$$

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)}$$

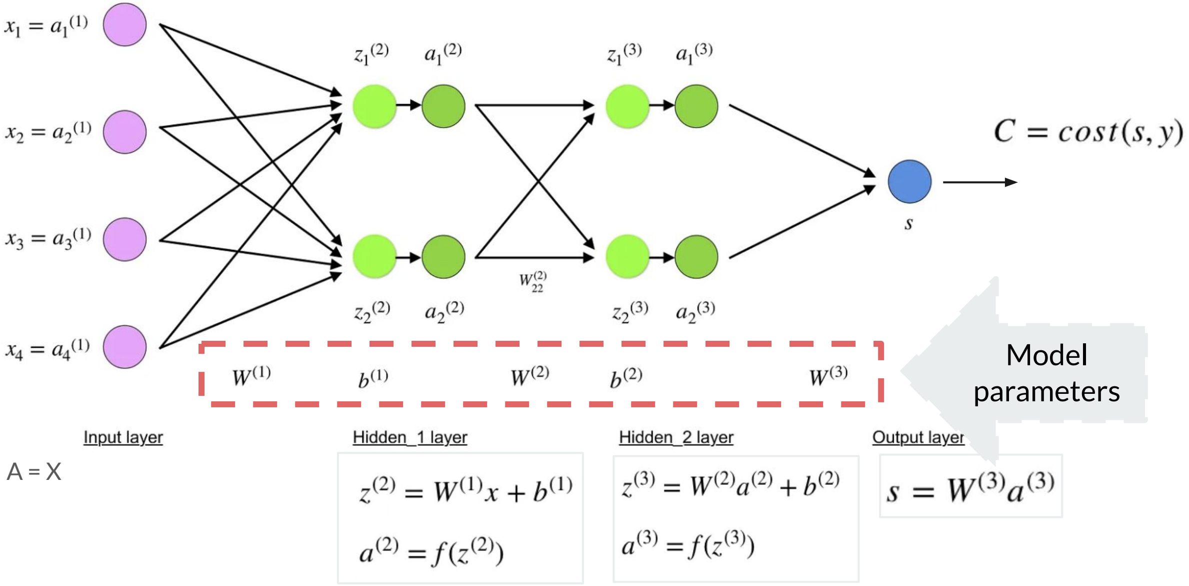
$$a^{(3)} = f(z^{(3)})$$

$$s = W^{(3)}a^{(3)}$$

mainly adapted from

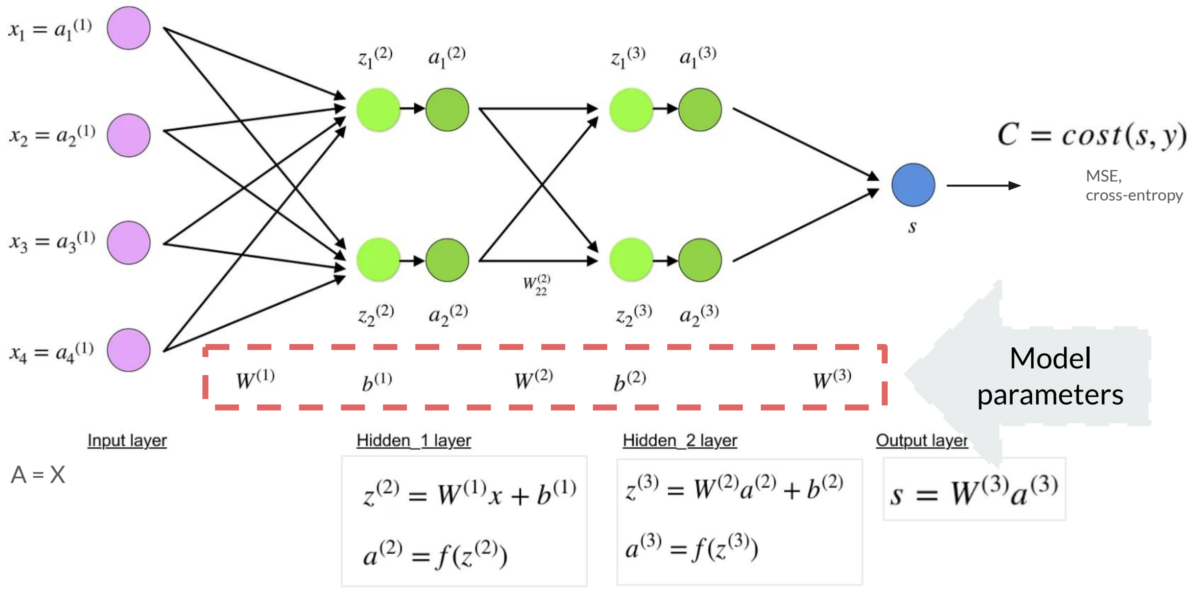
NN Training

Training



mainly adapted from

Training



[mainly adapted from]

Gradients

- **x** is your parameter vector
- Partial derivatives
- Only the last (hidden) layer parameters can have direct derivatives
- Rest (including the input layer) requires to apply a chain rule

$$\frac{\partial C}{\partial x} = [\frac{\partial C}{\partial x_1}, \frac{\partial C}{\partial x_2}, \ldots, \frac{\partial C}{\partial x_m}]$$

mainly adapted from

Gradients

- **x** is your parameter vector
- Partial derivatives
- Only the last (hidden) layer parameters can have direct derivatives
- Rest (including the input layer) requires to apply a chain rule

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} \frac{\partial z_j^l}{\partial w_{jk}^l} \qquad \textit{chain rule}$$

$$z_j^l = \sum_{k=1}^{m} w_{jk}^l a_k^{l-1} + b_j^l \qquad \textit{by definition}$$

$$m \ - \ \textit{number of neurons in } l-1 \textit{ layer}$$

$$\frac{\partial z_j^l}{\partial w_{jk}^l} = a_k^{l-1} \qquad \textit{by differentiation (calculating derivative)}$$

$$\frac{\partial C}{\partial w_{jk}^l} = \frac{\partial C}{\partial z_j^l} a_k^{l-1} \qquad \textit{final value}$$

[mainly adapted from]

Gradients

- **x** is your parameter vector
- Partial derivatives
- Only the last (hidden) layer parameters can have direct derivatives
- Rest (including the input layer) requires to apply a chain rule

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l}\frac{\partial z_j^l}{\partial b_j^l} \qquad \textit{chain rule}$$

$$\frac{\partial z_j^l}{\partial b_j^l} = 1 \qquad \textit{by differentiation (calculating derivative)}$$

$$\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial z_j^l}1 \qquad \textit{final value}$$

mainly adapted from

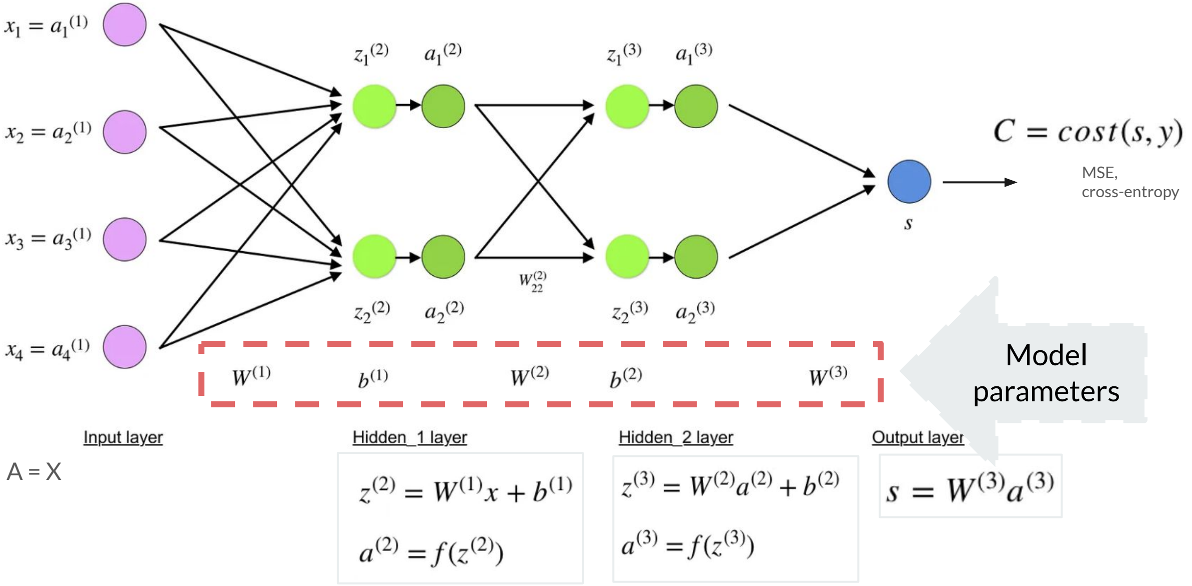
Gradient descent

- Can you recall our gradient descent Linear Regression model training?

$$
\begin{aligned}
&\textit{while (termination condition not met)}\\
&\qquad w := w - \epsilon \frac{\partial C}{\partial w}\\
&\qquad b := b - \epsilon \frac{\partial C}{\partial b}\\
&\textit{end}
\end{aligned}
$$

mainly adapted from

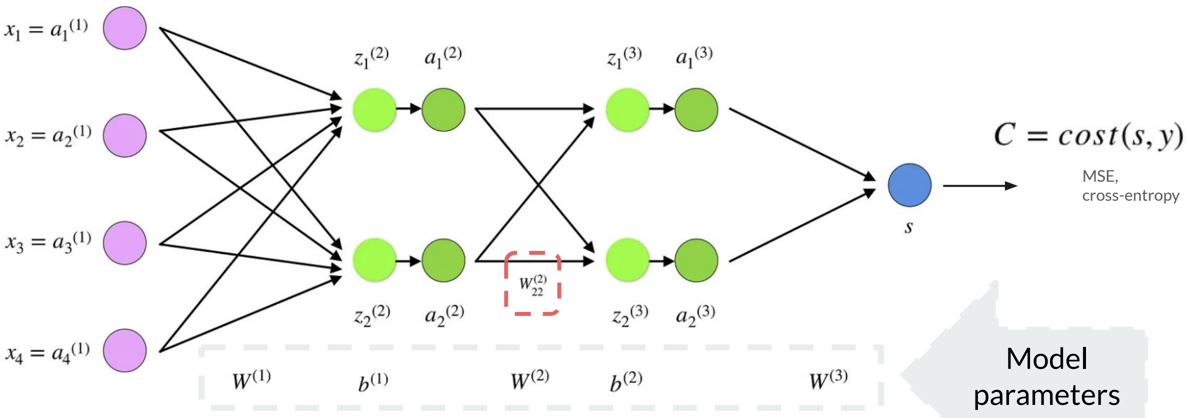
Training



mainly adapted from

Training

One Random Parameter, $W^{(2)}_{22}$



Input layer

A = X

Hidden_1 layer

$$z^{(2)} = W^{(1)}x + b^{(1)}$$

$$a^{(2)} = f(z^{(2)})$$

Hidden_2 layer

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)}$$

$$a^{(3)} = f(z^{(3)})$$

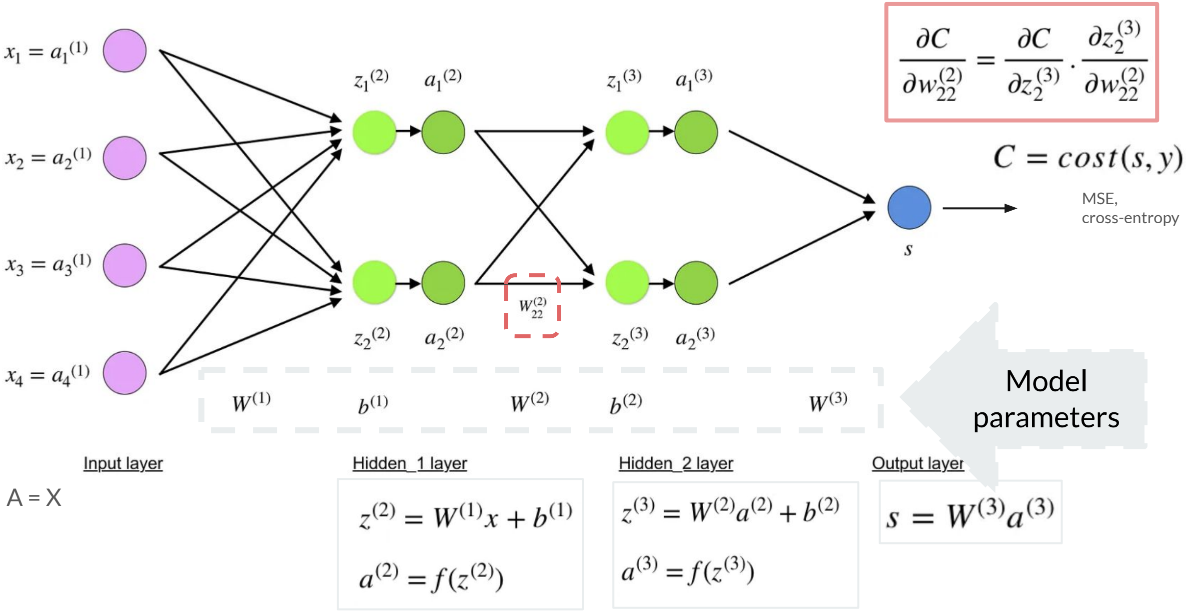
Output layer

$$s = W^{(3)}a^{(3)}$$

[mainly adapted from]()

Training

One Random Parameter, **$W^{(2)}_{22}$**

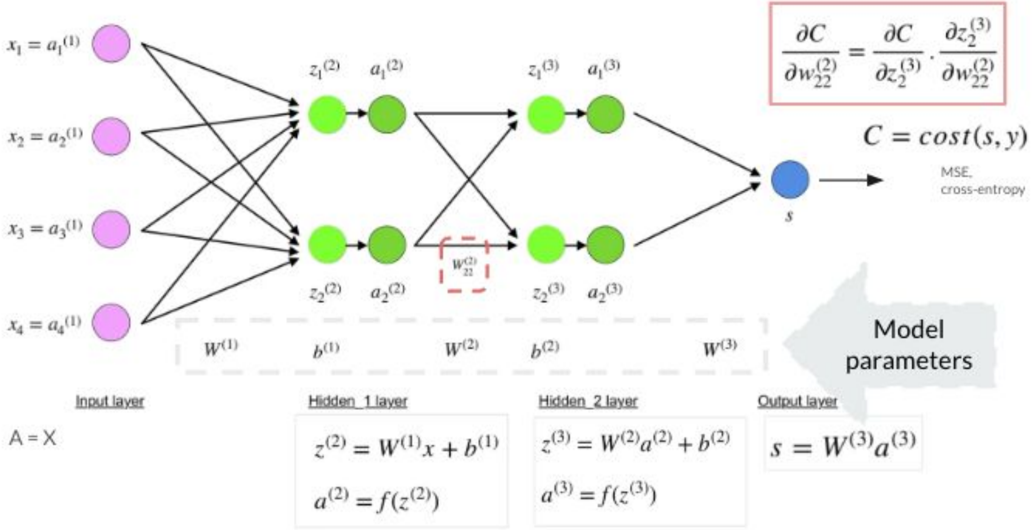


mainly adapted from

Error Backpropagation

One Random Parameter, $W^{(2)}_{22}$

$$\frac{\partial C}{\partial w_{22}^{(2)}} = \frac{\partial C}{\partial z_2^{(3)}} \cdot \frac{\partial z_2^{(3)}}{\partial w_{22}^{(2)}}$$



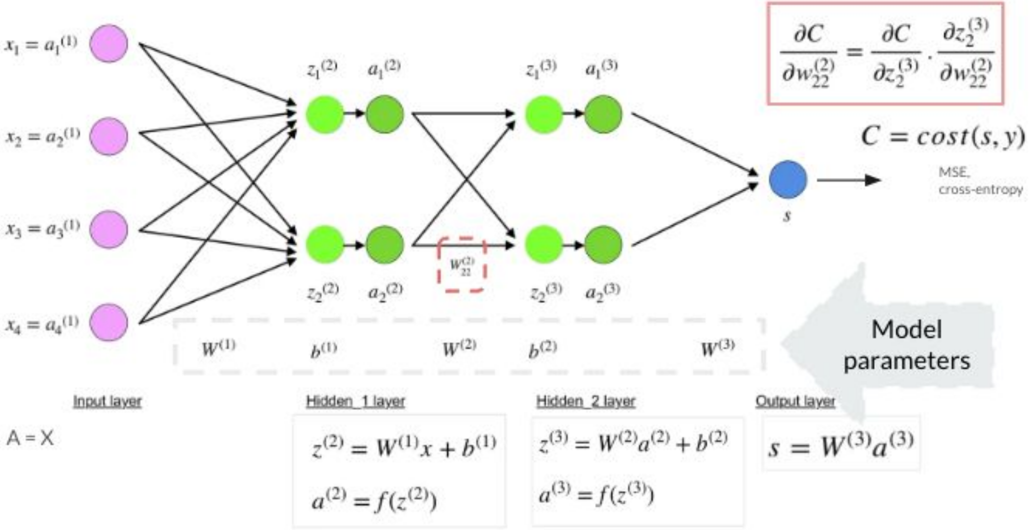
mainly adapted from

Error Backpropagation

One Random Parameter, $W^{(2)}_{22}$

$$\frac{\partial C}{\partial w_{22}^{(2)}} = \frac{\partial C}{\partial z_2^{(3)}} \cdot \frac{\partial z_2^{(3)}}{\partial w_{22}^{(2)}}$$

$$= \frac{\partial C}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \cdot a_2^{(2)}$$



mainly adapted from

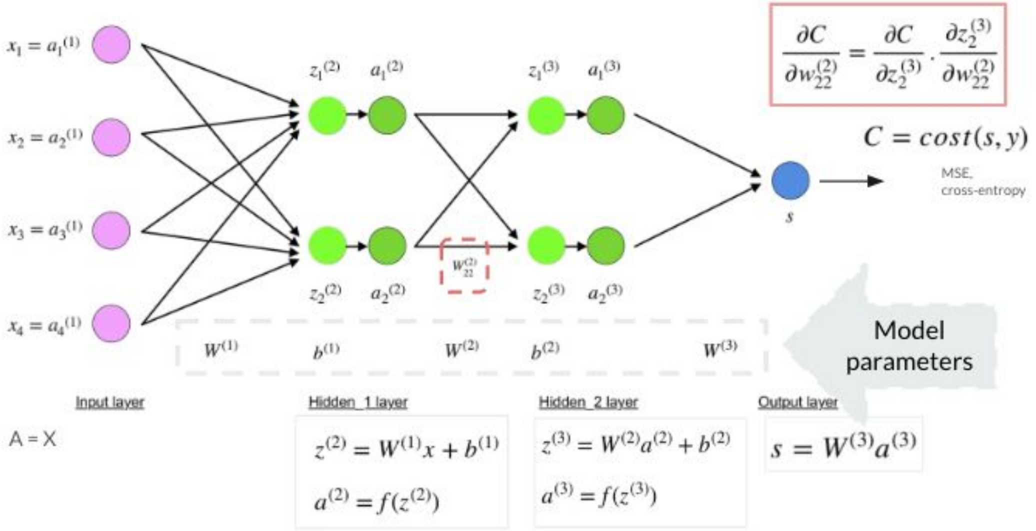
Error Backpropagation

One Random Parameter, $W^{(2)}_{22}$

$$\frac{\partial C}{\partial w_{22}^{(2)}} = \frac{\partial C}{\partial z_2^{(3)}} \cdot \frac{\partial z_2^{(3)}}{\partial w_{22}^{(2)}}$$

$$= \frac{\partial C}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \cdot a_2^{(2)}$$

$$= \frac{\partial C}{\partial a_2^{(3)}} \cdot f'(z_2^{(3)}) \cdot a_2^{(2)}$$



mainly adapted from

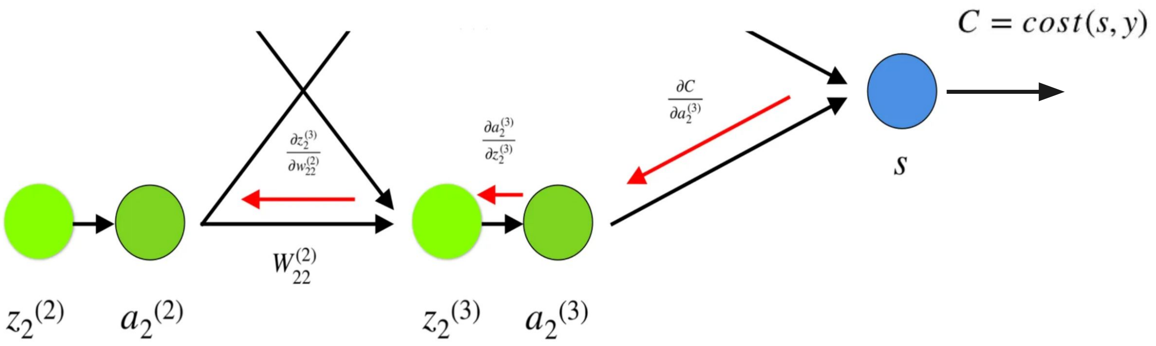
Error Backpropagation

One Random Parameter, $W^{(2)}_{22}$

$$\frac{\partial C}{\partial w_{22}^{(2)}} = \frac{\partial C}{\partial z_2^{(3)}} \cdot \frac{\partial z_2^{(3)}}{\partial w_{22}^{(2)}}$$

$$= \frac{\partial C}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \cdot a_2^{(2)}$$

$$= \frac{\partial C}{\partial a_2^{(3)}} \cdot f'(z_2^{(3)}) \cdot a_2^{(2)}$$



mainly adapted from

QA