Aprendizado Profundo 1

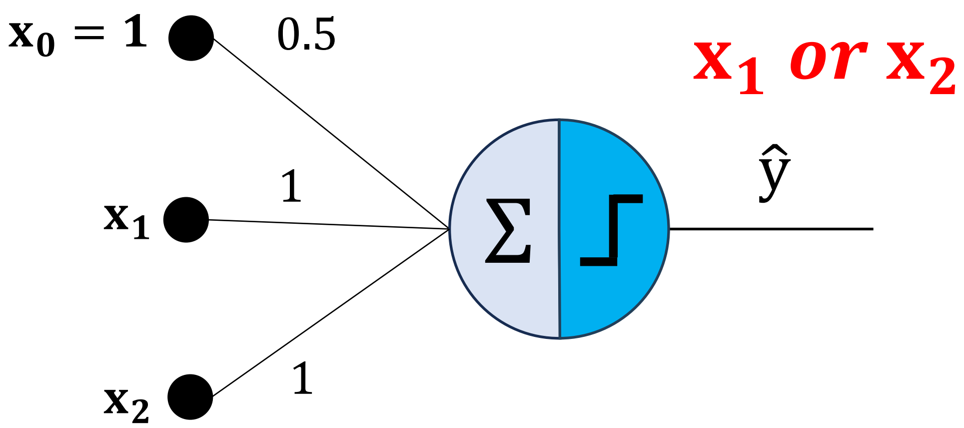
Perceptron Multicamadas e a Retropropagação de Erros

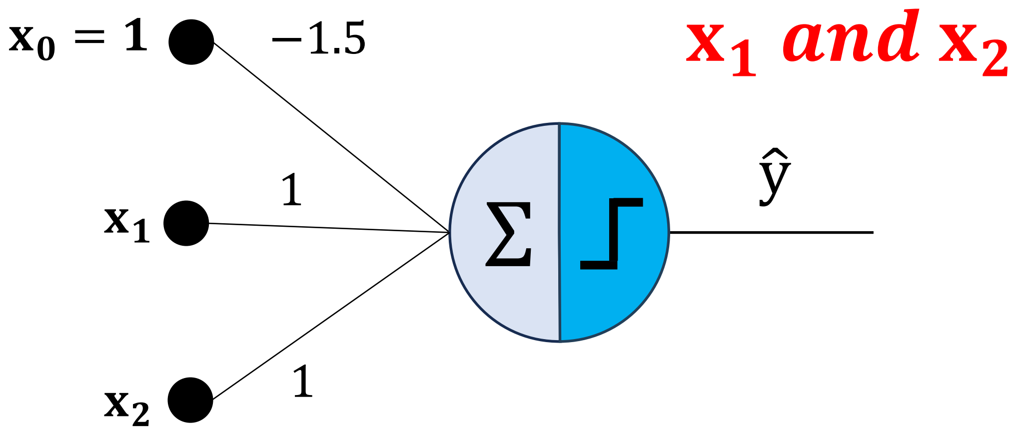
Professor: Lucas Silveira Kupssinskü

Agenda

- Breve revisão sobre o *perceptron*
- Treinando um MLP simples
  - Forward Pass
  - Backward Pass
  - Atualização de Pesos
  - Compação com código PyTorch
- Generalizando o treinamento
- Exemplo Vetorizado

Breve revisão



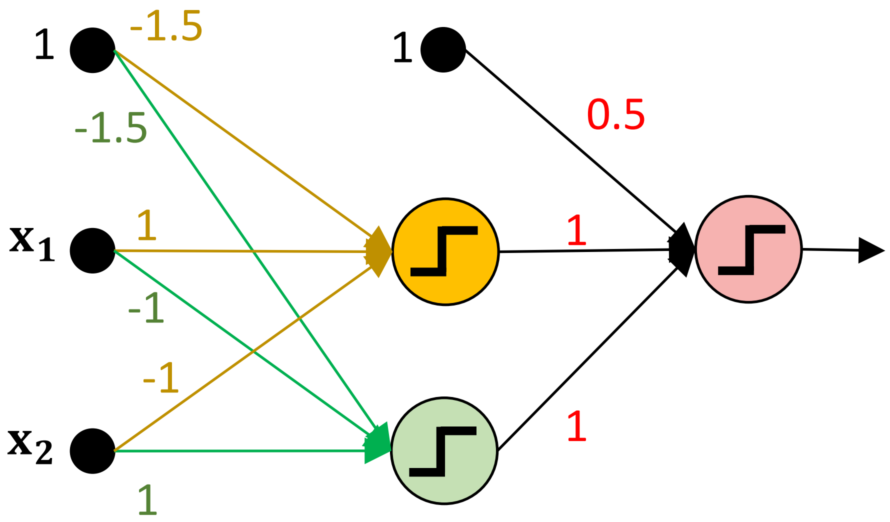


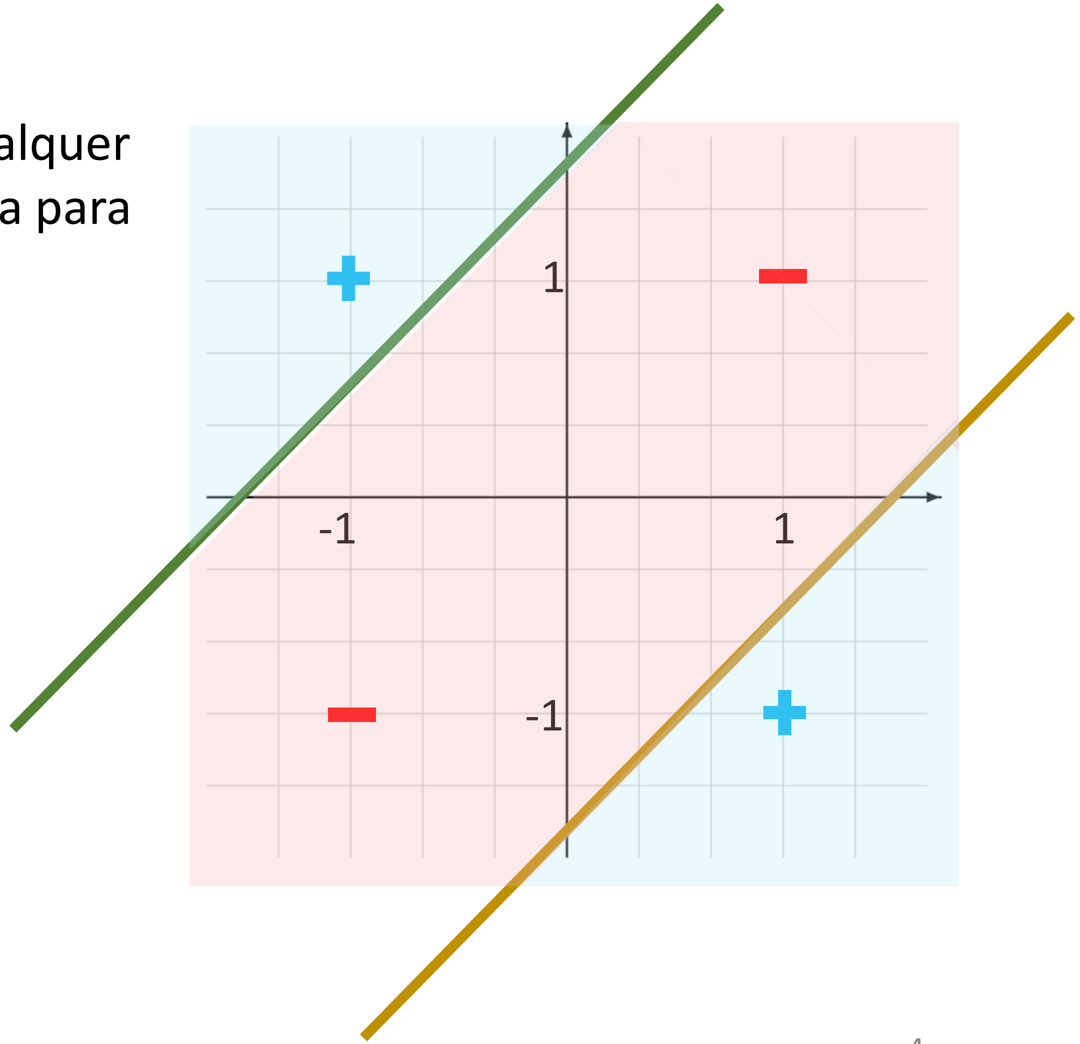
x_1	x_2	y
-1	-1	**-1**
-1	1	**1**
1	-1	**1**
1	1	**1**

x_1	x_2	y
-1	-1	**-1**
-1	1	**-1**
1	-1	**-1**
1	1	**1**

Perceptron Multi Camadas

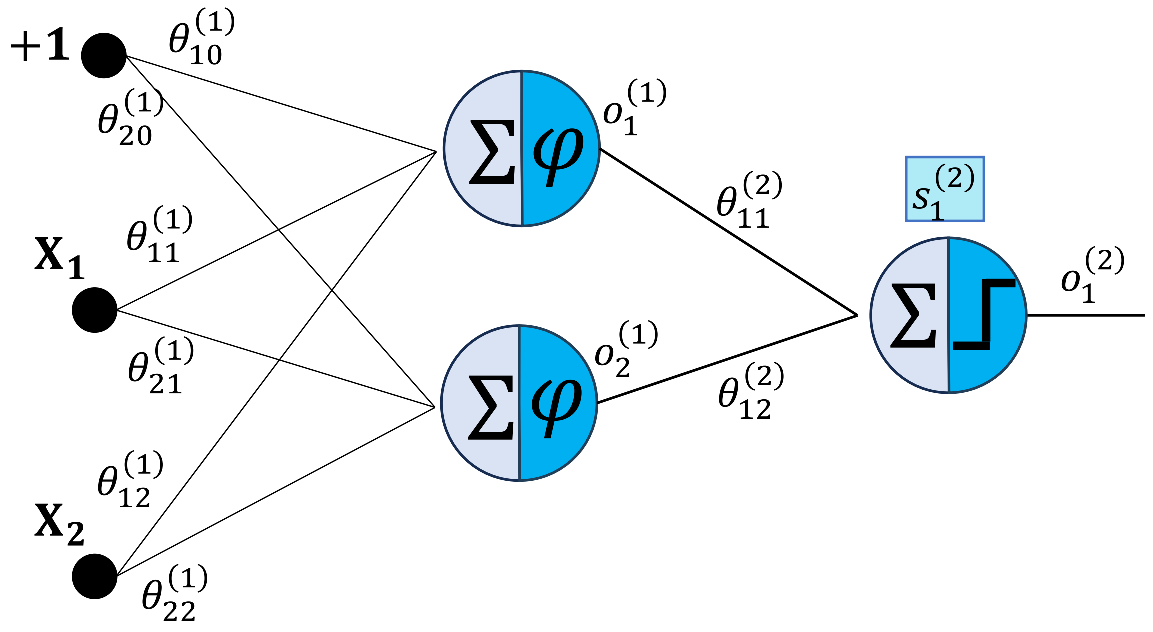
- Uma rede de perceptrons pode aproximar uma função qualquer
- Porém o *Perceptron Learning Algorithm* (PLA) não funciona para uma rede de perceptrons





Exercício

- Prove que o Perceptron Multicamadas abaixo gera uma fronteira de decisão linear
  - Considere $\varphi(x) = 2x$



“Inverno”

- Após 1968 com a publicação de Minski e Papert o interesse em redes neurais diminuiu
  - Pouca verba de pesquisa
  - Qual motivo de trabalhar em algo que não consegue nem resolver o problema XOR?
  - Sem ideias de como treinar perceptrons de múltiplas camadas
- Até que as coisas mudaram
  - RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **nature**, v. 323, n. 6088, p. 533-536, 1986.
  - Observação: Hinton não alega ter inventado o *backprop*, ele só foi um dos principais responsáveis pela sua popularização e talvez o primeiro a observar a hierarquia de *features* com grande riqueza semântica

Treinando um MLP

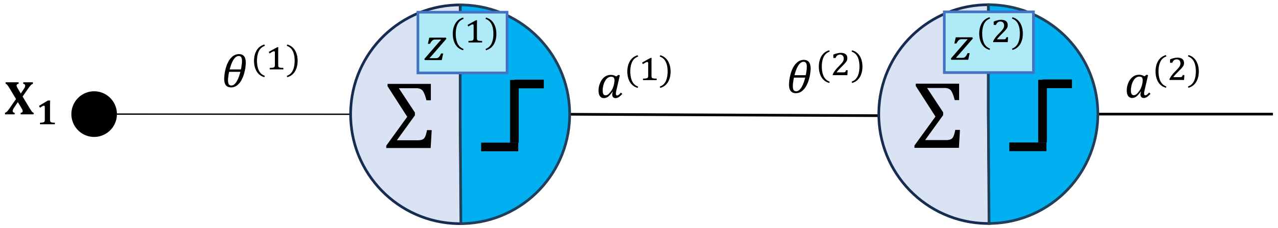
Vamos calcular manualmente uma iteração de treino em uma rede neural MLP

- Ideia central
  - Usar ~~*Half*~~ *Mean Squared Error* como função de custo
    - $J(\theta) = \frac{1}{2N} \Sigma_i \left(y^{(i)} - \hat{y}^{(i)} \right)^2$
  - Aplicar a descida de gradiente
    - $\theta_{t+1} = \theta_t - \eta \nabla J$

As premissas abaixo tornam o problema mais simples, mas não menos genérico

- 2 camadas
- 1 neurônio em cada camada
- Desconsideramos o *bias*
- Vamos tentar aprender a função identidade $y = X_1$

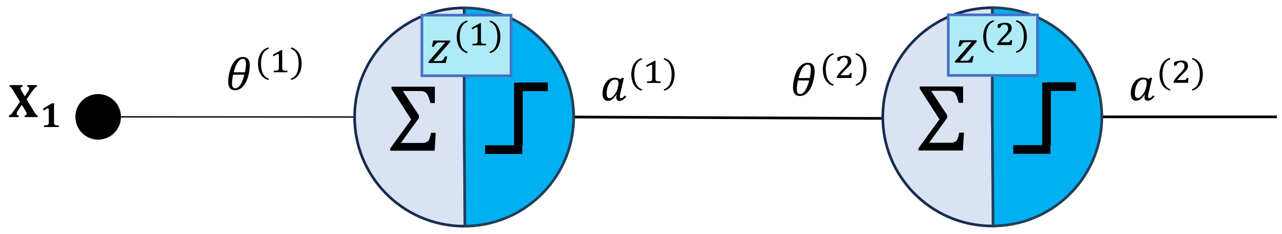
Treinando um MLP

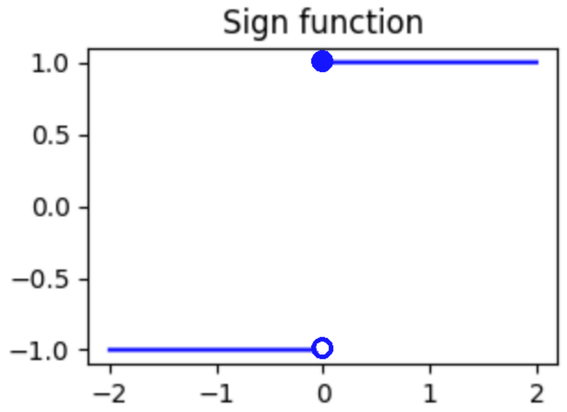


Ideia: Aplicar GD

- 2 camadas
- 1 neurônio em cada camada
- Desconsideramos o *bias*
- Vamos tentar aprender a função identidade $y = X_1$

Treinando um MLP



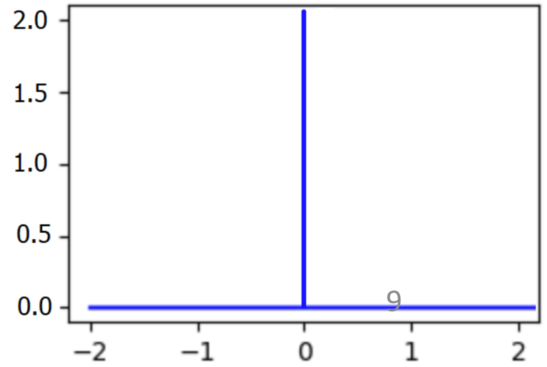


Ideia: Aplicar GD

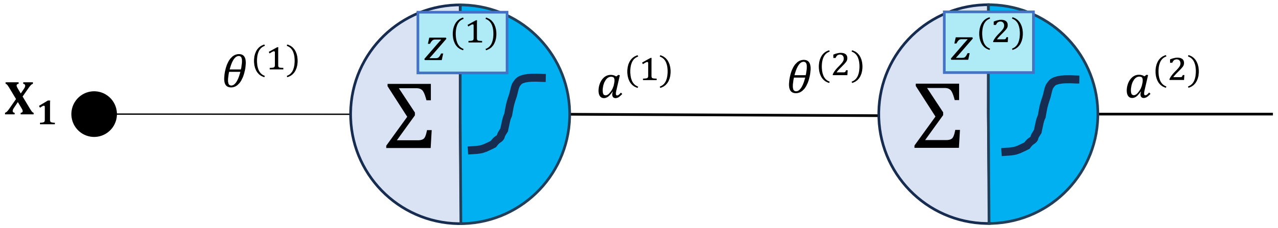
A função sinal tem dois **problemas**

1. Derivada não definida em $x = 0$
2. Valor da derivada é 0 em todos os valores de x nos quais a função é derivável

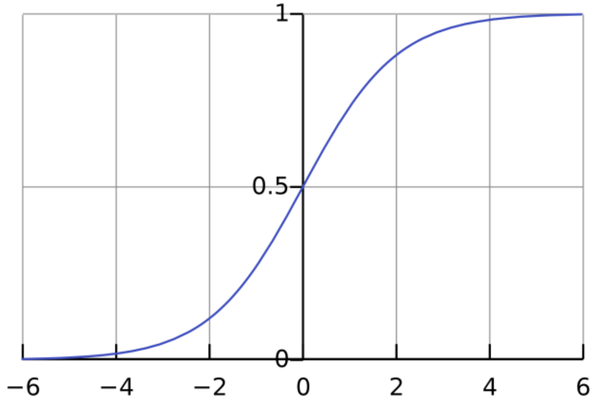
$\partial sign/\partial x$



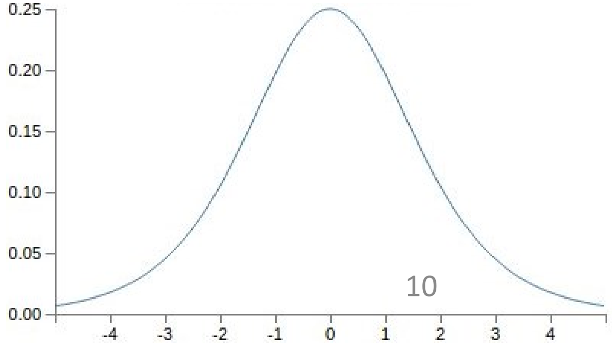
Treinando um MLP



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



$\partial\sigma/\partial x$



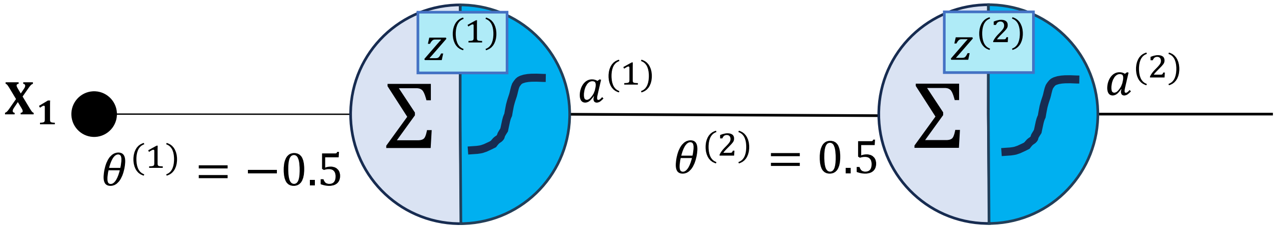
Ideia: Aplicar GD

A função sinal tem dois **problemas**

1. Derivada não definida em $x = 0$
2. Valor da derivada é 0 em todos os valores de x nos quais a função é derivável

Solução: Trocar de função de ativação para sigmoid

Treinando um MLP



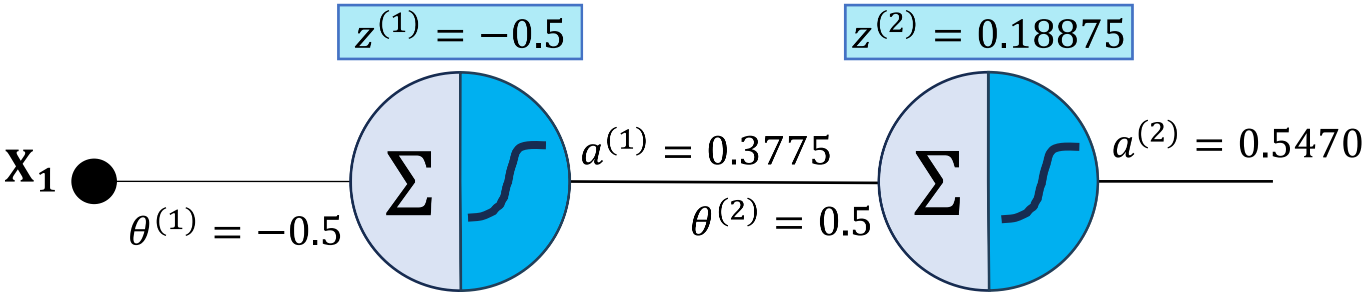
Ideia: Aplicar GD

Inicializando os pesos $\theta^{(1)} = -0.5$ e $\theta^{(2)} = 0.5$ conseguimos computar loss para o par $X_1 = 1, y = 1$

Vamos adotar loss como $J(\theta) = \frac{1}{2}(y - \hat{y})^2$

Treinando um MLP

Este é o *Forward Pass*

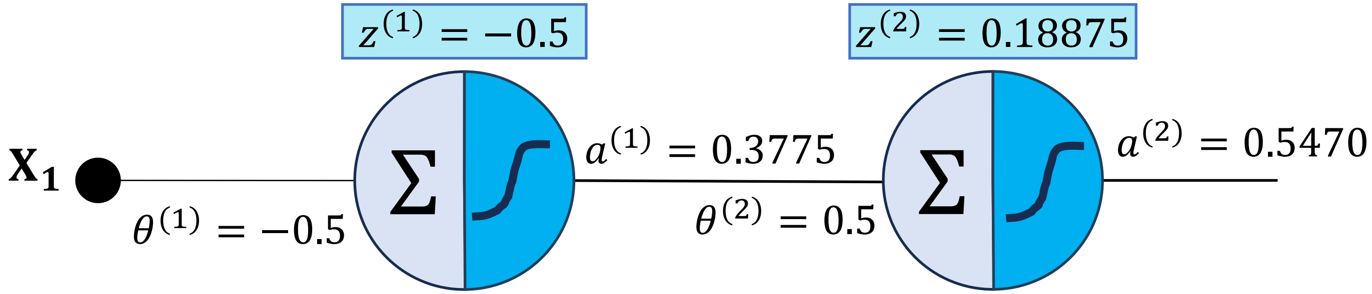


Ideia: Aplicar GD

$$\hat{y} = \sigma\left(\theta^{(2)}\sigma\left(X^{(1)}\theta^{(1)}\right)\right) = 0.5470$$

Vamos adotar loss como $J(\theta) = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(1 - 0.5470)^2 = 0.1026045$

Comparando com PyTorch



Ideia: Aplicar GD

$$\hat{y} = \sigma\left(\theta^{(2)}\sigma\left(X^{(1)}\theta^{(1)}\right)\right) = 0.5470$$

Vamos adotar loss como $J(\theta) = \frac{1}{2}(y - \hat{y})^2 = 0.1026045$

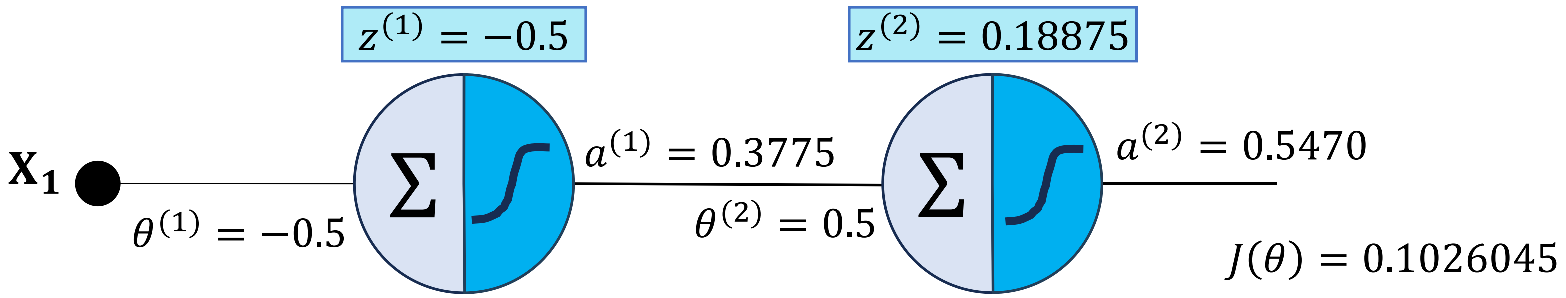
```python
import torch
```

```python
class RedeSimples(torch.nn.Module):
   def __init__(self) -> None:
      super().__init__()
      self.theta_1 = torch.nn.parameter.Parameter(
         torch.tensor(-0.5), requires_grad=True)
      self.theta_2 = torch.nn.parameter.Parameter(
         torch.tensor(0.5), requires_grad=True)

   def forward(self, x):
      x = torch.sigmoid(x * self.theta_1)
      x = torch.sigmoid(x * self.theta_2)
      return x
```

```python
x = torch.tensor(1.0)
y = torch.tensor(1.0)
model = RedeSimples()
y_hat = model(x)
loss = ((y - y_hat)**2)/2
print(f'y_hat: {y_hat}')
print(f'loss: {loss}')
```

```
y_hat: 0.547052919846545
loss: 0.10258052498102188
```

Treinando um MLP



Vamos relembrar a descida de gradiente:

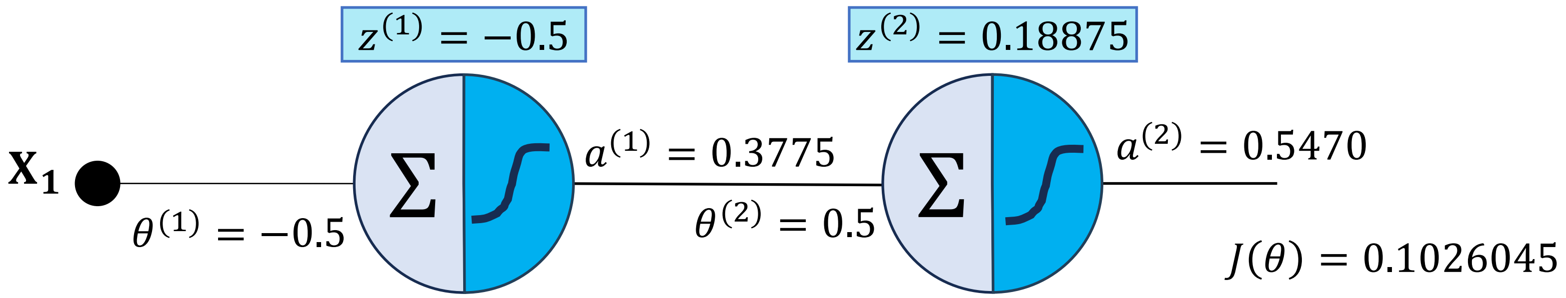
$$\theta_t = \theta_{t-1} - \eta \nabla J$$

Onde:

η: taxa de aprendizado

∇J: gradiente da função de custo em relação aos parâmetros θ

Treinando um MLP

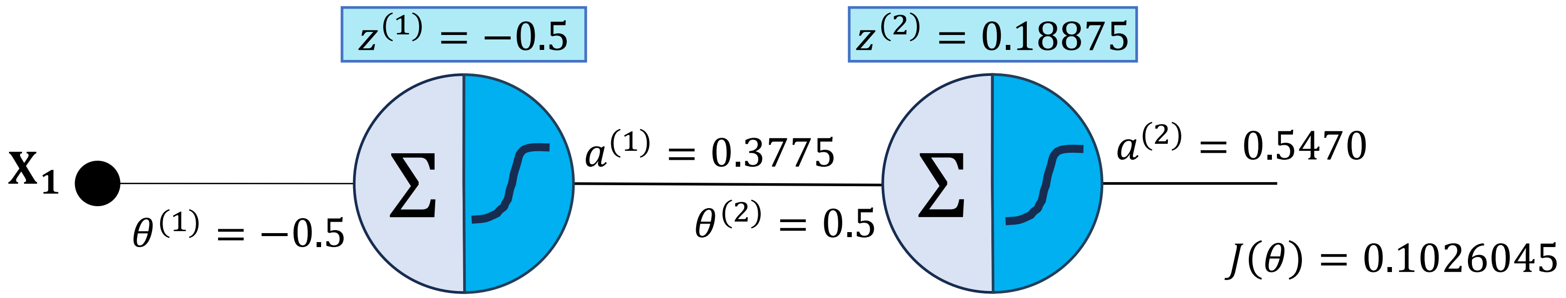


$$\theta_t = \theta_{t-1} - \eta \nabla J$$

Problema: Como computar $\frac{\partial J}{\partial \theta^{(2)}}$ e $\frac{\partial J}{\partial \theta^{(1)}}$?

A função $J(\theta)$ é uma função composta

Treinando um MLP



$$\theta_t = \theta_{t-1} - \eta \nabla J$$

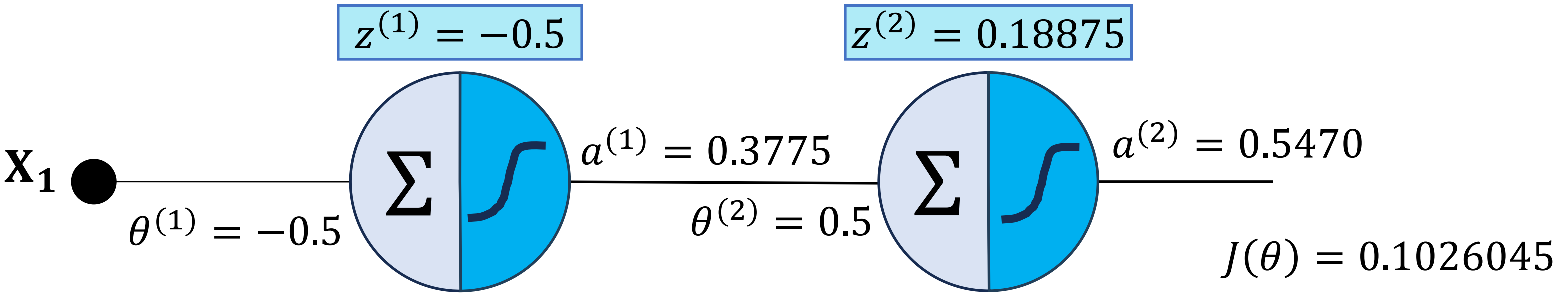
Problema: Como computar $\frac{\partial J}{\partial \theta^{(2)}}$ e $\frac{\partial J}{\partial \theta^{(1)}}$?

A função $J(\theta)$ é uma função composta

Solução: Regra da cadeia

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\partial J}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\partial J}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

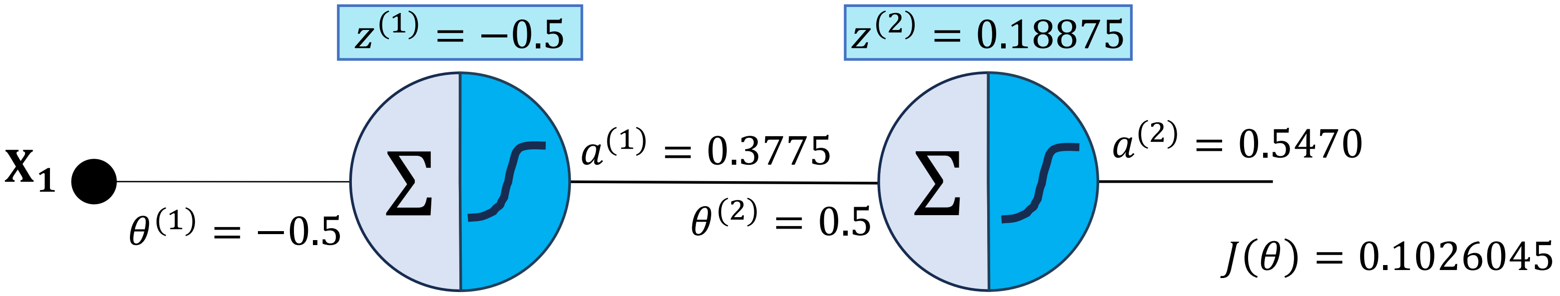


$$\frac{\partial J}{\partial \theta^{(2)}} = \boxed{\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \boxed{\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

$$\boxed{\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)}$$

$$\theta_t = \theta_{t-1} - \eta \nabla J$$



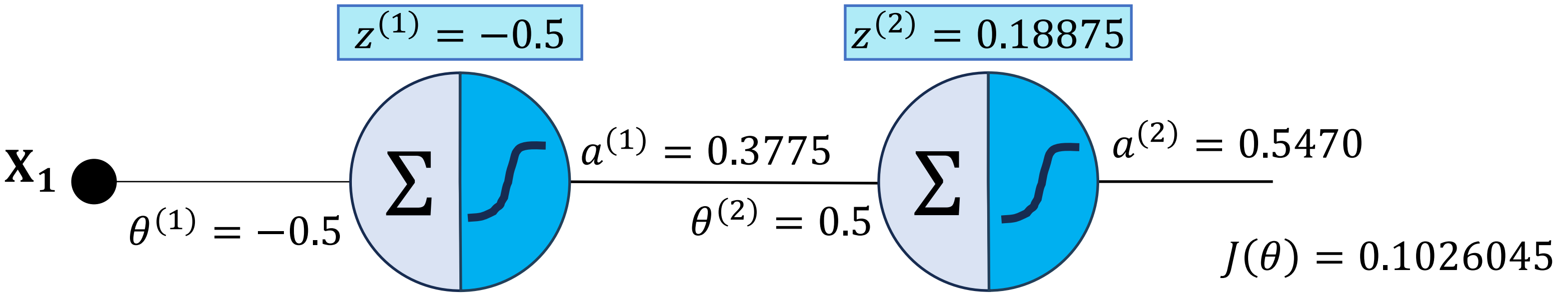
$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \boxed{\frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \boxed{\frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

$$\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)$$

$$\boxed{\frac{\partial a^{(2)}}{\partial z^{(2)}} = a^{(2)}\left(1 - a^{(2)}\right)}$$

$$\theta_t = \theta_{t-1} - \eta \nabla J$$



$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \boxed{\frac{\partial z^{(2)}}{\partial \theta^{(2)}}}$$

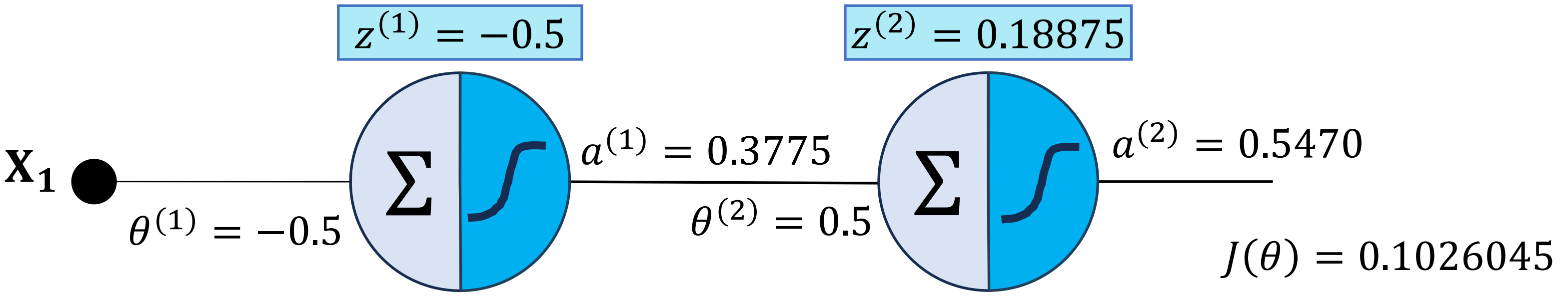
$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

$$\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)$$

$$\frac{\partial a^{(2)}}{\partial z^{(2)}} = a^{(2)}\left(1 - a^{(2)}\right)$$

$$\boxed{\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = a^{(1)}}$$

$$\theta_t = \theta_{t-1} - \eta \nabla J$$



$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \boxed{\frac{\partial z^{(2)}}{\partial a^{(1)}}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

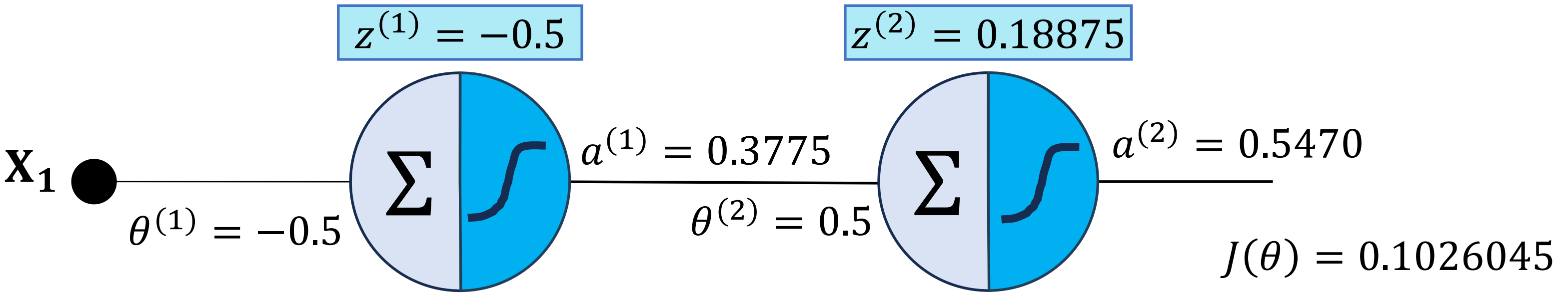
$$\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)$$

$$\boxed{\frac{\partial z^{(2)}}{\partial a^{(1)}} = \theta^{(2)}}$$

$$\frac{\partial a^{(2)}}{\partial z^{(2)}} = a^{(2)}\left(1 - a^{(2)}\right)$$

$$\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = a^{(1)}$$

$$\theta_t = \theta_{t-1} - \eta \nabla J$$



$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \boxed{\frac{\partial a^{(1)}}{\partial z^{(1)}}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

$$\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)$$

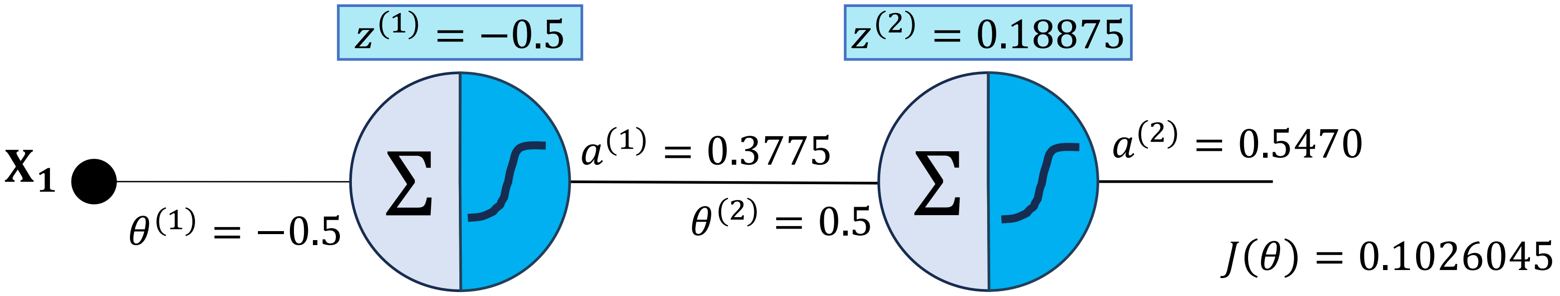
$$\frac{\partial z^{(2)}}{\partial a^{(1)}} = \theta^{(2)}$$

$$\frac{\partial a^{(2)}}{\partial z^{(2)}} = a^{(2)}\left(1 - a^{(2)}\right)$$

$$\boxed{\frac{\partial a^{(1)}}{\partial z^{(1)}} = a^{(1)}\left(1 - a^{(1)}\right)}$$

$$\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = a^{(1)}$$

$$\theta_t = \theta_{t-1} - \eta \nabla J$$



$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \boxed{\frac{\partial z^{(1)}}{\partial \theta^{(1)}}}$$

$$\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)$$

$$\frac{\partial a^{(2)}}{\partial z^{(2)}} = a^{(2)}\left(1 - a^{(2)}\right)$$

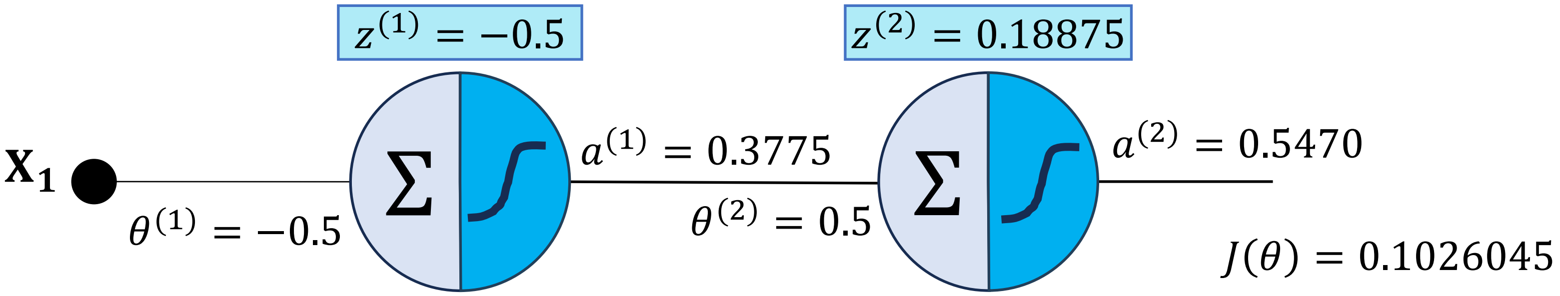
$$\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = a^{(1)}$$

$$\frac{\partial z^{(2)}}{\partial a^{(1)}} = \theta^{(2)}$$

$$\frac{\partial a^{(1)}}{\partial z^{(1)}} = a^{(1)}\left(1 - a^{(1)}\right)$$

$$\boxed{\frac{\partial z^{(1)}}{\partial \theta^{(1)}} = X_1}$$

$$\theta_t = \theta_{t-1} - \eta \nabla J$$



$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

$$\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)$$

$$\frac{\partial z^{(2)}}{\partial a^{(1)}} = \theta^{(2)}$$

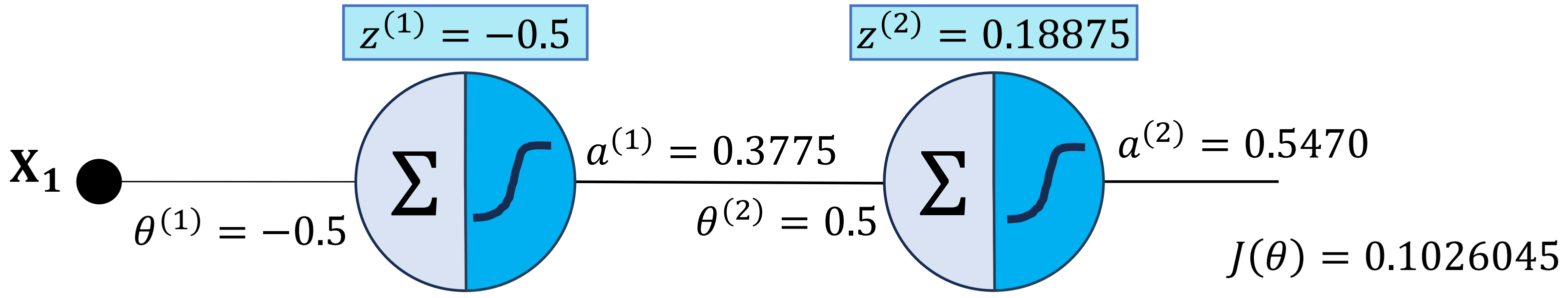
$$\frac{\partial a^{(2)}}{\partial z^{(2)}} = a^{(2)}\left(1 - a^{(2)}\right)$$

$$\frac{\partial a^{(1)}}{\partial z^{(1)}} = a^{(1)}\left(1 - a^{(1)}\right)$$

$$\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = a^{(1)}$$

$$\frac{\partial z^{(1)}}{\partial \theta^{(1)}} = X_1$$

$$\theta_t = \theta_{t-1} - \eta \nabla J$$



$$\frac{\partial J}{\partial \theta^{(2)}} = \left(a^{(2)} - y\right)a^{(2)}\left(1 - a^{(2)}\right)a^{(1)} = (0.5470 - 1)0.5470(1 - 0.5470)0.3775$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \left(a^{(2)} - y\right)a^{(2)}\left(1 - a^{(2)}\right)\theta^{(2)}a^{(1)}\left(1 - a^{(1)}\right)X_1 = (0.5470 - 1)0.5470(1 - 0.5470)(0.5)0.3775(1 - 0.3775)1$$

$$\frac{\partial J}{\partial a^{(2)}} = \frac{\partial \frac{1}{2}\left(y - a^{(2)}\right)^2}{\partial a^{(2)}} = \left(a^{(2)} - y\right)$$

$$\frac{\partial a^{(2)}}{\partial z^{(2)}} = a^{(2)}\left(1 - a^{(2)}\right)$$

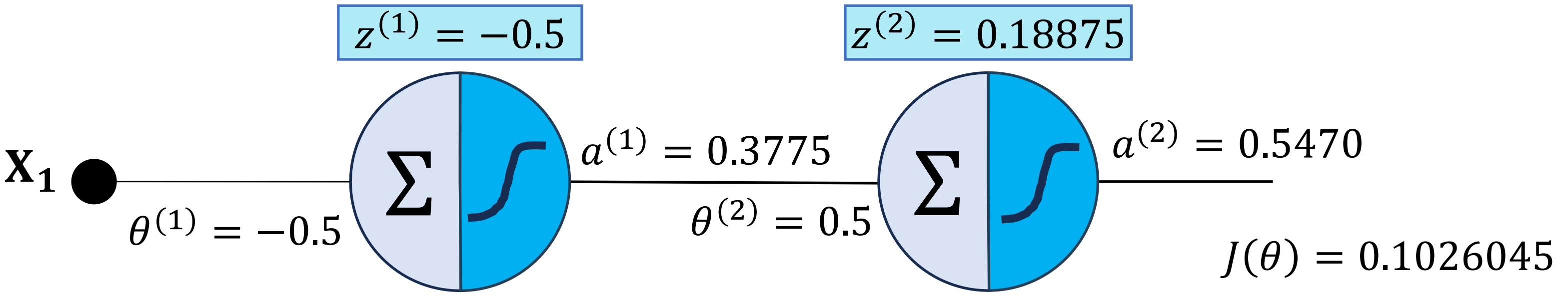
$$\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = a^{(1)}$$

$$\frac{\partial z^{(2)}}{\partial a^{(1)}} = \theta^{(2)}$$

$$\frac{\partial a^{(1)}}{\partial z^{(1)}} = a^{(1)}\left(1 - a^{(1)}\right)$$

$$\frac{\partial z^{(1)}}{\partial \theta^{(1)}} = X_1$$

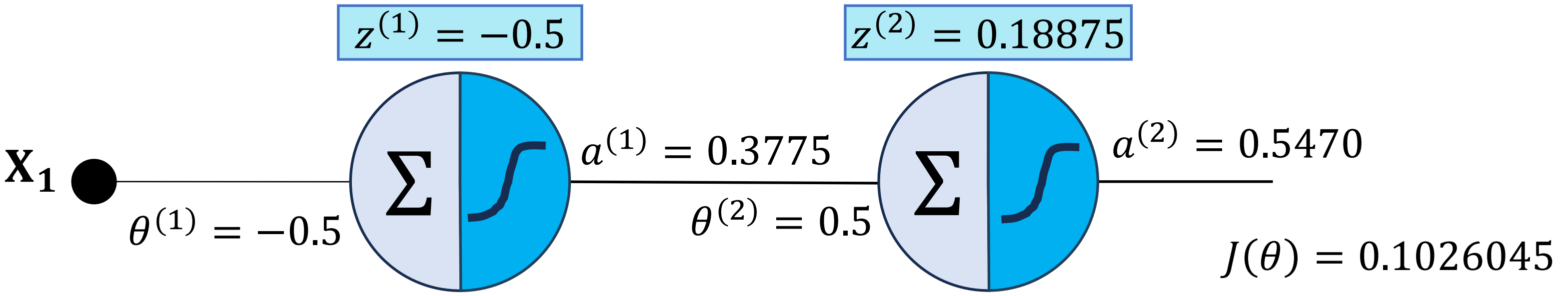
$$\theta_t = \theta_{t-1} - \eta \nabla J$$



$$\frac{\partial J}{\partial \theta^{(2)}} = -0.0423741194325$$
$$\frac{\partial J}{\partial \theta^{(1)}} = -0.013188944673365625$$

Este é o *Backward Pass*

$$\theta_t = \theta_{t-1} - \eta \nabla J$$

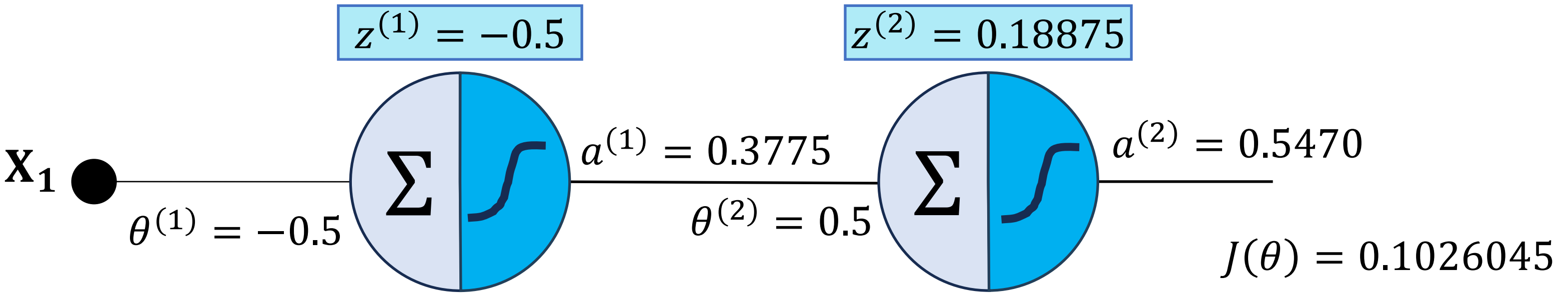


$$\frac{\partial J}{\partial \theta^{(2)}} = -0.0423741194325$$
$$\frac{\partial J}{\partial \theta^{(1)}} = -0.013188944673365625$$

Agora podemos usar a regra de atualização da descida de gradiente para computar os novos pesos

$$\theta_t = \theta_{t-1} - \eta \nabla J$$

$$\theta_t^{(2)} = 0.5 - 0.1(-0.042374)$$
$$\theta_t^{(1)} = -0.5 - 0.1(-0.013189)$$



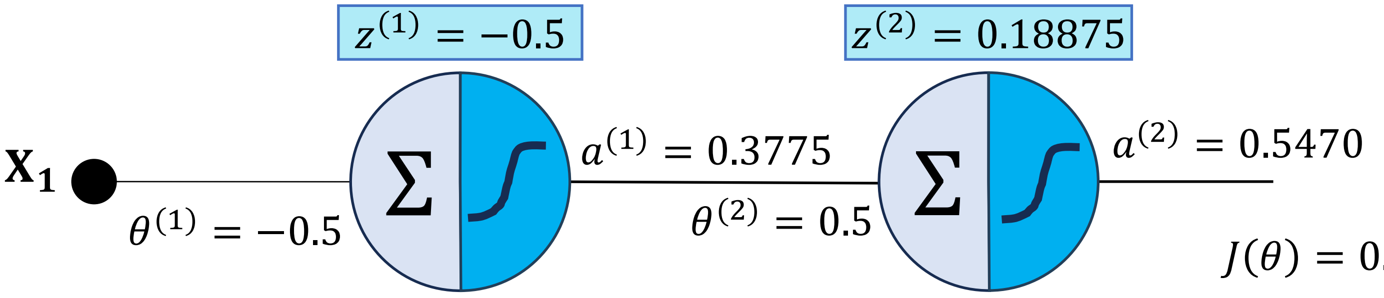
$$\frac{\partial J}{\partial \theta^{(2)}} = -0.0423741194325$$
$$\frac{\partial J}{\partial \theta^{(1)}} = -0.013188944673365625$$

Agora podemos usar a regra de atualização da descida de gradiente para computar os novos pesos

$$\theta_t = \theta_{t-1} - \eta \nabla J$$

$$\theta_t^{(2)} = \quad 0.5042374$$
$$\theta_t^{(1)} = -0.4986811$$

Essa é a atualização dos pesos (*Optimizer* SGD)



$J(\theta) = 0.1026045$

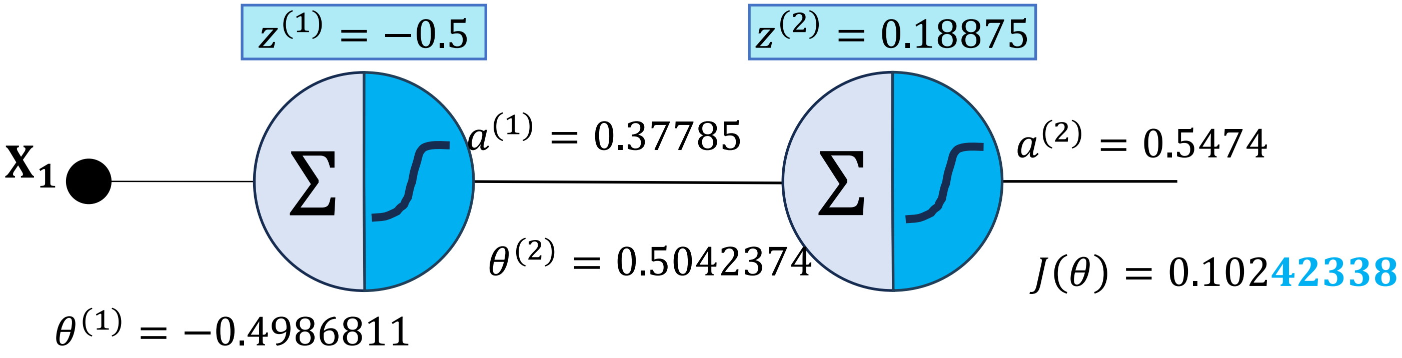
$$\frac{\partial J}{\partial \theta^{(2)}} = -0.0423741194325$$
$$\frac{\partial J}{\partial \theta^{(1)}} = -0.013188944673365625$$

Com os novos parâmetros, podemos computar um novo valor para a função de custo

$$\theta_t^{(2)} = 0.5042374$$
$$\theta_t^{(1)} = -0.4986811$$

$$\hat{y}_t = \sigma\big(0.5042374 * \sigma(1 * (-0.4986811))\big) = 0.547\mathbf{4}$$
$$J(\theta_t) = \frac{1}{2}(1 - 0.5474)^2 = 0.102\mathbf{42338}$$

Comparando com PyTorch



$$\frac{\partial J}{\partial \theta^{(2)}} = -0.0423741194325$$

$$\frac{\partial J}{\partial \theta^{(1)}} = -0.0131889446733625$$

```python
import torch
from torch.optim import SGD
```

```python
class RedeSimples(torch.nn.Module):
    def __init__(self) -> None:
        super().__init__()
        self.theta_1 = torch.nn.parameter.Parameter(
            torch.tensor(-0.5), requires_grad=True)
        self.theta_2 = torch.nn.parameter.Parameter(
            torch.tensor(0.5), requires_grad=True)

    def forward(self, x):
        x = torch.sigmoid(x * self.theta_1)
        x = torch.sigmoid(x * self.theta_2)
        return x
```

```python
optimizer = SGD(model.parameters(), lr=0.1)
x = torch.tensor(1.0)
y = torch.tensor(1.0)

model.train()
optimizer.zero_grad()
y_hat = model(x)
loss = (y - y_hat)**2/2
loss.backward()
print(f'Grad: {model.theta_2.grad}, {model.theta_1.grad}')
optimizer.step()
print(f'Pesos: {model.theta_2.data}, {model.theta_1.data}')
```

```
Grad: -0.04237288236618042, -0.01318769808858633
Pesos: 0.5042372941970825, -0.4986812174320221
```

As etapas do treinamento de uma rede neural

- 1º Computar todas as saídas da rede (*Forward Pass*)

As etapas do treinamento de uma rede neural

- 1º Computar todas as saídas da rede (*Forward Pass*)
- 2º Computar o quão diferente as saídas são em relação a variável alvo (*Loss Function*)
  - Não existe apenas uma forma de avaliar a divergência entre a saída e a variável alvo
  - Variam conforme a tarefa
  - Podem ter um ou mais termos com objetivo de regularização

As etapas do treinamento de uma rede neural

- 1º Computar todas as saídas da rede (*Forward Pass*)
- 2º Computar o quão diferente as saídas são em relação a variável alvo (*Loss Function*)
- 3º Computar o gradiente do erro em relação aos parâmetros da rede (*Backward Pass*)
  - O gradiente pode ser analítico ("*exato*") ou numérico ("*aproximado*")
  - Usualmente computado via grafo computacional

As etapas do treinamento de uma rede neural

- 1º Computar todas as saídas da rede (*Forward Pass*)
- 2º Computar o quão diferente as saídas são em relação a variável alvo (*Loss Function*)
- 3º Computar o gradiente do erro em relação aos parâmetros da rede (*Backward Pass*)
- 4º Atualizar os pesos (*Optimizer*)
    - Usamos a versão *vanilla* no exemplo anterior, mas vamos ver variações
    - *Batch, mini-Batch ou estocástico*
    - *Com ou sem momentum*

As etapas do treinamento de uma rede neural

- 1º Computar todas as saídas da rede (*Forward Pass*)
- 2º Computar o quão diferente as saídas são em relação a variável alvo (*Loss Function*)
- 3º Computar o gradiente do erro em relação aos parâmetros da rede (*Backward Pass*)
- 4º Atualizar os pesos (*Optimizer*)

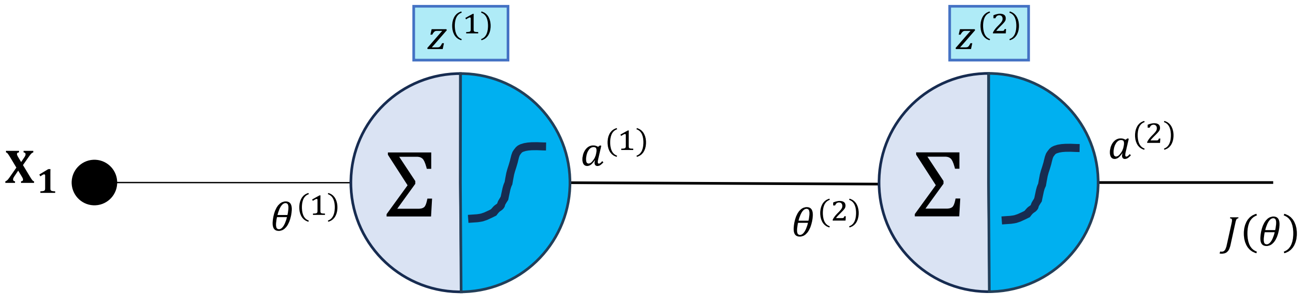
```
optimizer = SGD(model.parameters(), lr=0.1)
x = torch.tensor(1.0)
y = torch.tensor(1.0)

model.train()
optimizer.zero_grad()
y_hat = model(x)
loss = (y - y_hat)**2/2
loss.backward()
optimizer.step()
```

Generalizando o treinamento

- Repare que o gradiente em relação a cada peso $\theta^{(i)}$ depende de:
  - Um gradiente global $\delta^{(i)} = \frac{\partial J}{\partial z^{(i)}}$
  - Um gradiente local $\frac{\partial z^{(i)}}{\partial \theta^{(i)}} = a^{(i-1)}$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$



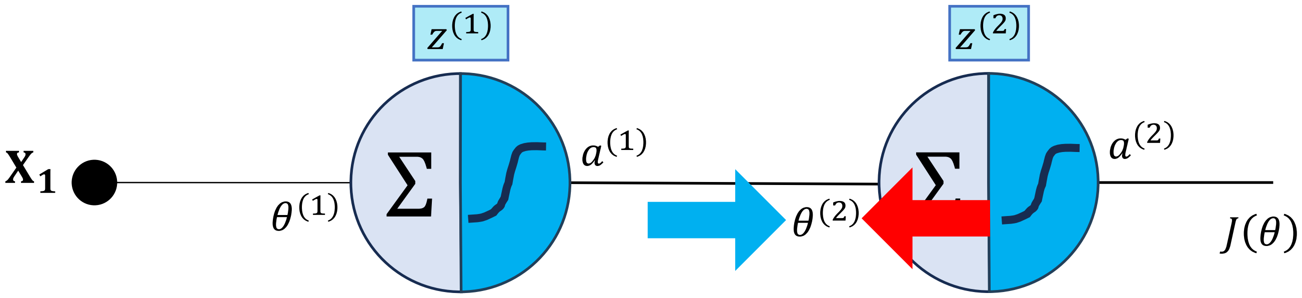
Generalizando o treinamento

- Repare que o gradiente em relação a cada peso $\theta^{(i)}$ depende de:
  - Um gradiente global $\boxed{\delta^{(i)} = \frac{\partial J}{\partial z^{(i)}}}$
  - Um gradiente local $\boxed{\frac{\partial z^{(i)}}{\partial \theta^{(i)}} = a^{(i-1)}}$

$i = 2$

$$\frac{\partial J}{\partial \theta^{(2)}} = \boxed{\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}}} \boxed{\frac{\partial z^{(2)}}{\partial \theta^{(2)}}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$



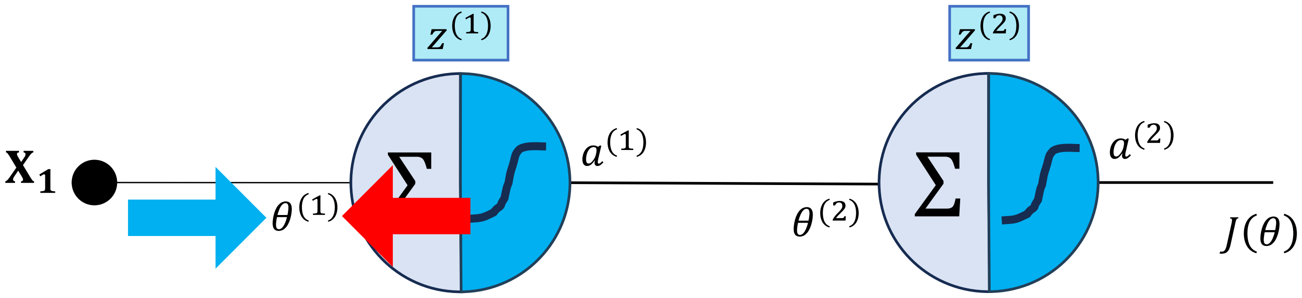
Generalizando o treinamento

- Repare que o gradiente em relação a cada peso $\theta^{(i)}$ depende de:
  - Um gradiente global $\boxed{\delta^{(i)} = \frac{\partial J}{\partial z^{(i)}}}$
  - Um gradiente local $\boxed{\frac{\partial z^{(i)}}{\partial \theta^{(i)}} = a^{(i-1)}}$

$i = 1$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \boxed{\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{a}^{(2)}} \frac{\partial \boldsymbol{a}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}}} \boxed{\frac{\partial z^{(1)}}{\partial \theta^{(1)}}}$$



Generalizando o treinamento

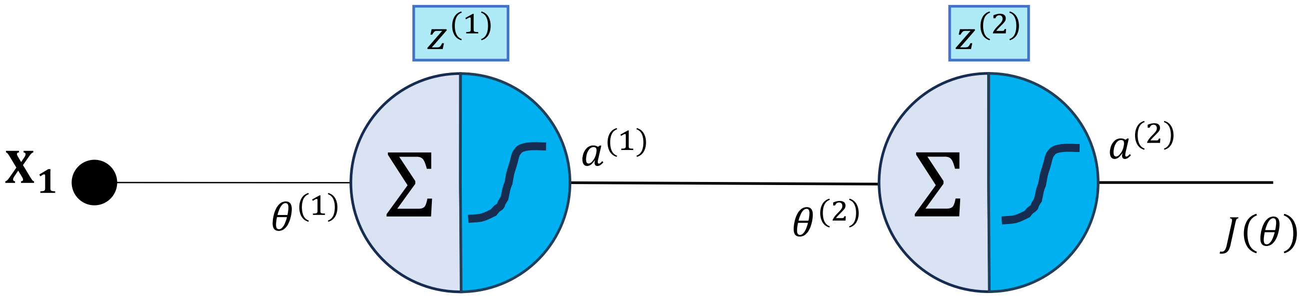
- Repare que o gradiente em relação a cada peso $\theta^{(i)}$ depende de:
  - Um gradiente global $\delta^{(i)} = \frac{\partial J}{\partial z^{(i)}}$
  - Um gradiente local $\frac{\partial z^{(i)}}{\partial \theta^{(i)}} = a^{(i-1)}$

1

$$\boxed{\frac{\partial J}{\partial \theta^{(i)}} = \delta^{(i)} a^{(i-1)}}$$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

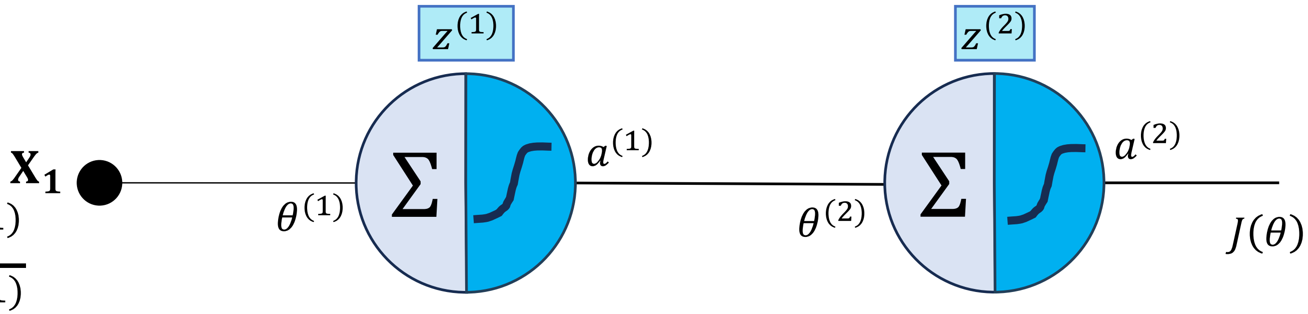


Generalizando o treinamento

- Repare que o gradiente em relação a cada peso $\theta^{(i)}$ depende de:
  - Um gradiente global $\delta^{(i)} = \frac{\partial J}{\partial z^{(i)}}$
  - Um gradiente local $\frac{\partial z^{(i)}}{\partial \theta^{(i)}} = a^{(i-1)}$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}} = \delta^{(2)} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}} = \delta^{(1)} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

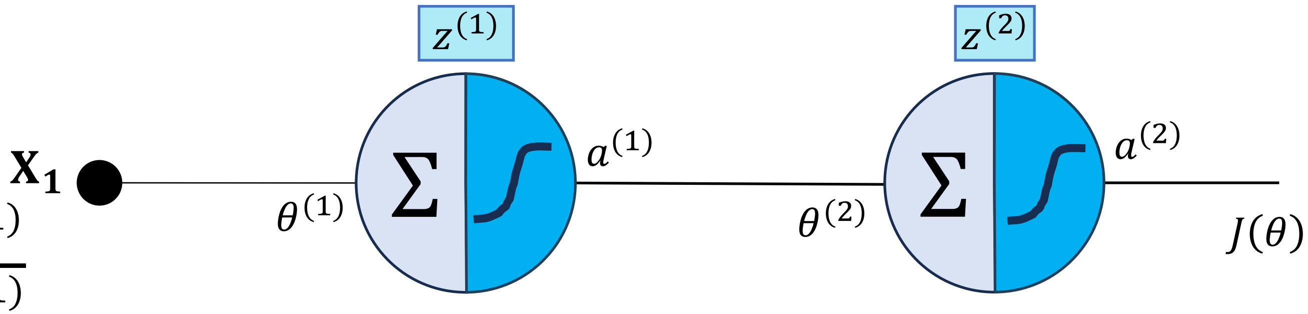


Generalizando o treinamento

- Repare que o gradiente global em uma camada $\delta^{(l)}$ depende do gradiente global da camada posterior $\delta^{(l+1)}$:

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a}^{(2)}} \frac{\boldsymbol{\partial a}^{(2)}}{\boldsymbol{\partial z}^{(2)}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}} = \delta^{(2)} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a}^{(2)}} \frac{\boldsymbol{\partial a}^{(2)}}{\boldsymbol{\partial z}^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}} = \delta^{(1)} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

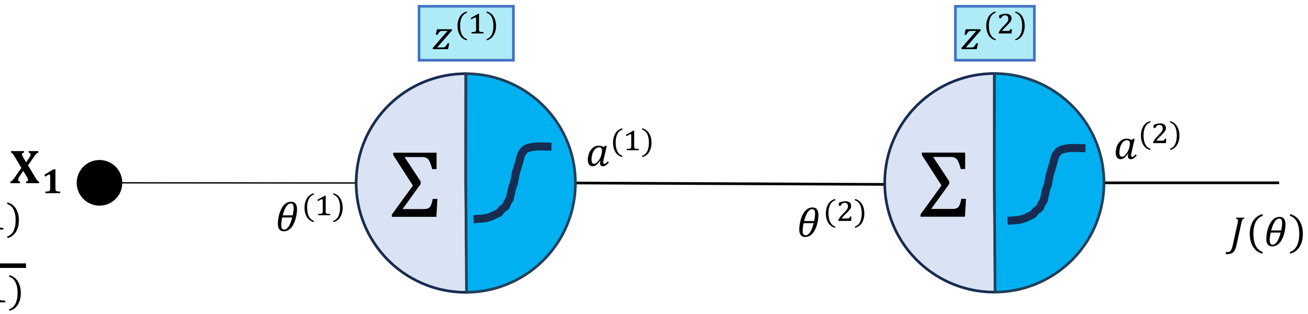


Generalizando o treinamento

- Repare que o gradiente global em uma camada $\delta^{(l)}$ depende do gradiente global da camada posterior $\delta^{(l+1)}$:
- $\delta^{(1)} = \delta^{(2)} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}}$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}} = \delta^{(2)} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}} = \delta^{(1)} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

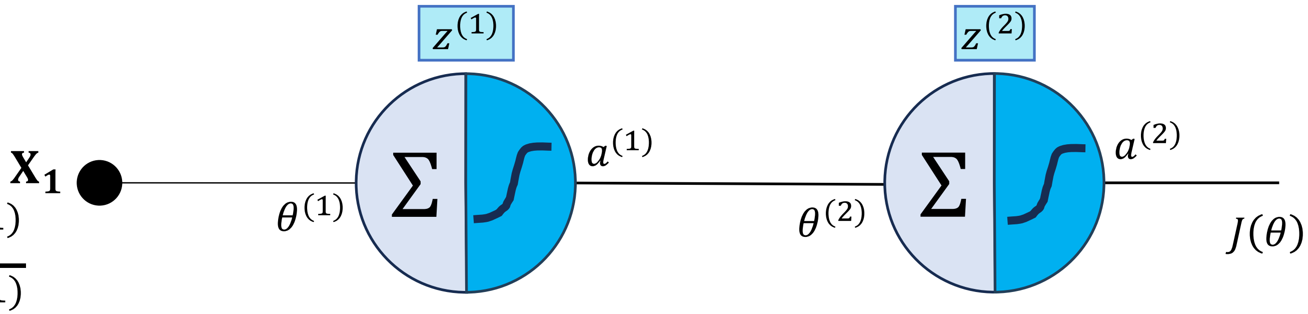


Generalizando o treinamento

- Repare que o gradiente global em uma camada $\delta^{(l)}$ depende do gradiente global da camada posterior $\delta^{(l+1)}$:
- $\delta^{(1)} = \delta^{(2)} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}}$
  - $\frac{\partial z^{(2)}}{\partial a^{(1)}} = \theta^{(2)}$
  - $\frac{\partial a^{(1)}}{\partial z^{(1)}} = a'^{(1)}$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}} = \delta^{(2)} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}} = \delta^{(1)} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$

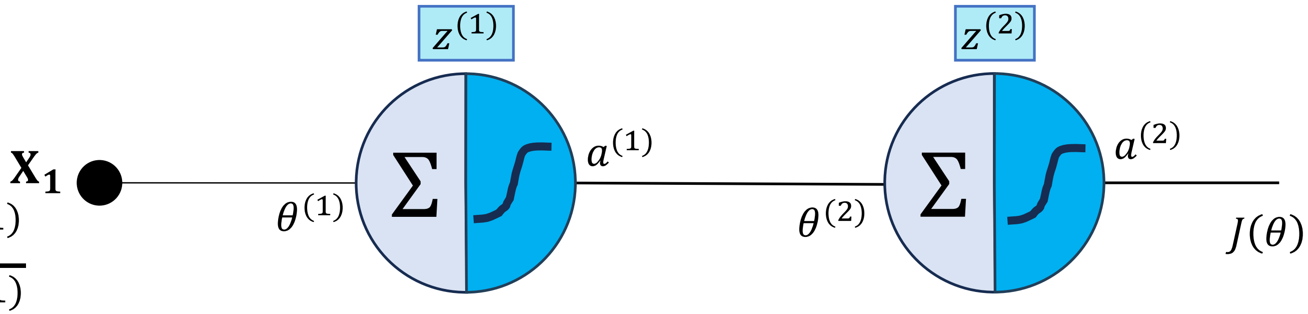


Generalizando o treinamento

- Repare que o gradiente global em uma camada $\delta^{(l)}$ depende do gradiente global da camada posterior $\delta^{(l+1)}$:
- $\delta^{(1)} = \delta^{(2)}\theta^{(2)}a'^{(1)}$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}}\frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}}\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = \delta^{(2)}\frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}}\frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}}\frac{\partial z^{(2)}}{\partial a^{(1)}}\frac{\partial a^{(1)}}{\partial z^{(1)}}\frac{\partial z^{(1)}}{\partial \theta^{(1)}} = \delta^{(1)}\frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$



Generalizando o treinamento

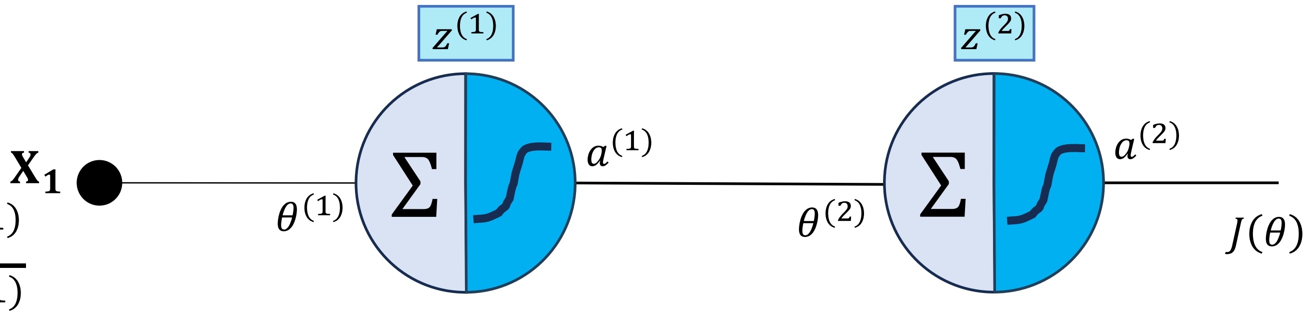
- Repare que o gradiente global em uma camada $\delta^{(l)}$ depende do gradiente global da camada posterior $\delta^{(l+1)}$:
- $\delta^{(1)} = \delta^{(2)}\theta^{(2)}a'^{(1)}$

2

$$\boxed{\delta^{(l)} = \left(\delta^{(l+1)}\theta^{(l+1)}\right)a'^{(l)}}$$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}}\frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}}\frac{\partial z^{(2)}}{\partial \theta^{(2)}} = \delta^{(2)}\frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}}\frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}}\frac{\partial z^{(2)}}{\partial a^{(1)}}\frac{\partial a^{(1)}}{\partial z^{(1)}}\frac{\partial z^{(1)}}{\partial \theta^{(1)}} = \delta^{(1)}\frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$



Generalizando o treinamento

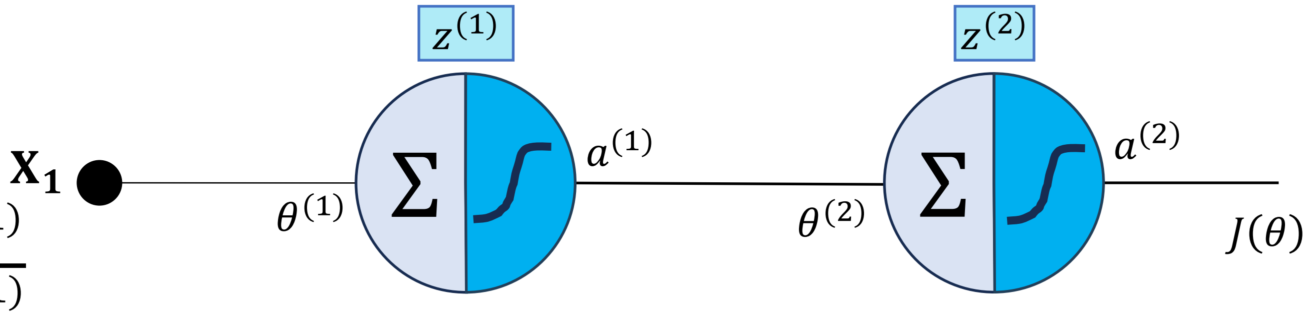
- Repare que o gradiente da última camada $\delta^{(L)}$ é igual a:
- $\delta^{(L)} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(L)}}} \frac{\boldsymbol{\partial a^{(L)}}}{\boldsymbol{\partial z^{(L)}}}$

3

$$\boxed{\delta^{(L)} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(L)}}} \frac{\boldsymbol{\partial a^{(L)}}}{\boldsymbol{\partial z^{(L)}}}}$$

$$\frac{\partial J}{\partial \theta^{(2)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial \theta^{(2)}} = \delta^{(2)} \frac{\partial z^{(2)}}{\partial \theta^{(2)}}$$

$$\frac{\partial J}{\partial \theta^{(1)}} = \frac{\boldsymbol{\partial J}}{\boldsymbol{\partial a^{(2)}}} \frac{\boldsymbol{\partial a^{(2)}}}{\boldsymbol{\partial z^{(2)}}} \frac{\partial z^{(2)}}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial \theta^{(1)}} = \delta^{(1)} \frac{\partial z^{(1)}}{\partial \theta^{(1)}}$$



Generalizando o treinamento

- Temos três componentes para implementar o *backpropagation*
- Vamos revisar algumas questões que podem ter ficado de fora

1

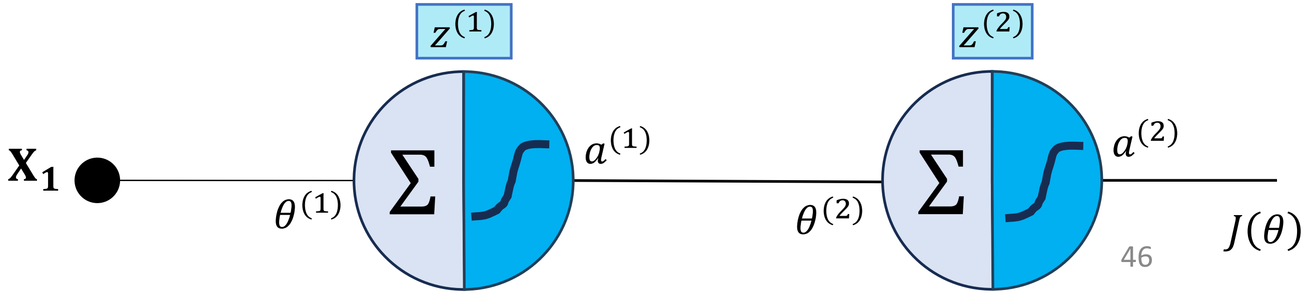
$$\boxed{\frac{\partial J}{\partial \theta^{(i)}} = \delta^{(i)} a^{(i-1)}}$$

2

$$\boxed{\delta^{(l)} = \left(\delta^{(l+1)} \theta^{(l+1)}\right) a'^{(l)}}$$

3

$$\boxed{\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}}}$$



Generalizando o treinamento

- Temos três componentes para implementar o *backpropagation*
- Vamos revisar algumas questões que podem ter ficado de fora
  - *bias* – ou o termo livre

1

$$\frac{\partial J}{\partial \theta^{(i)}} = \delta^{(i)} \boxed{a^{(i-1)}}$$

2

$$\delta^{(l)} = \left(\delta^{(l+1)}\theta^{(l+1)}\right)a'^{(l)}$$

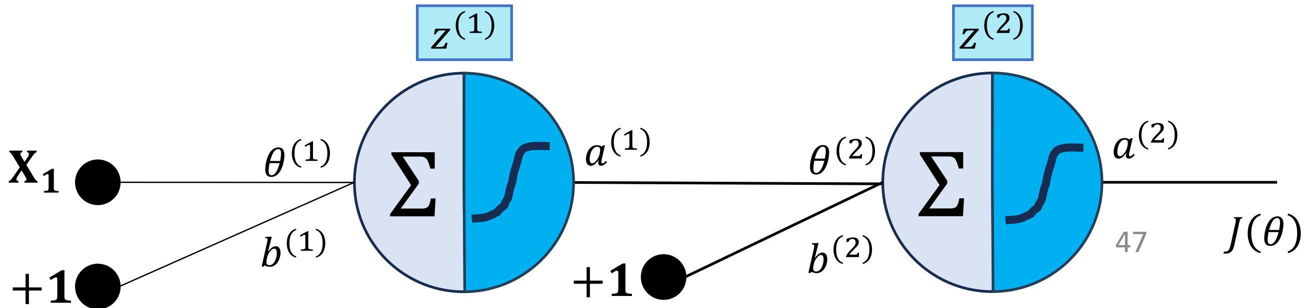
3

$$\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}}$$

1.1

$$\frac{\partial J}{\partial b^{(i)}} = \delta^{(i)}$$

Termo constante em +1



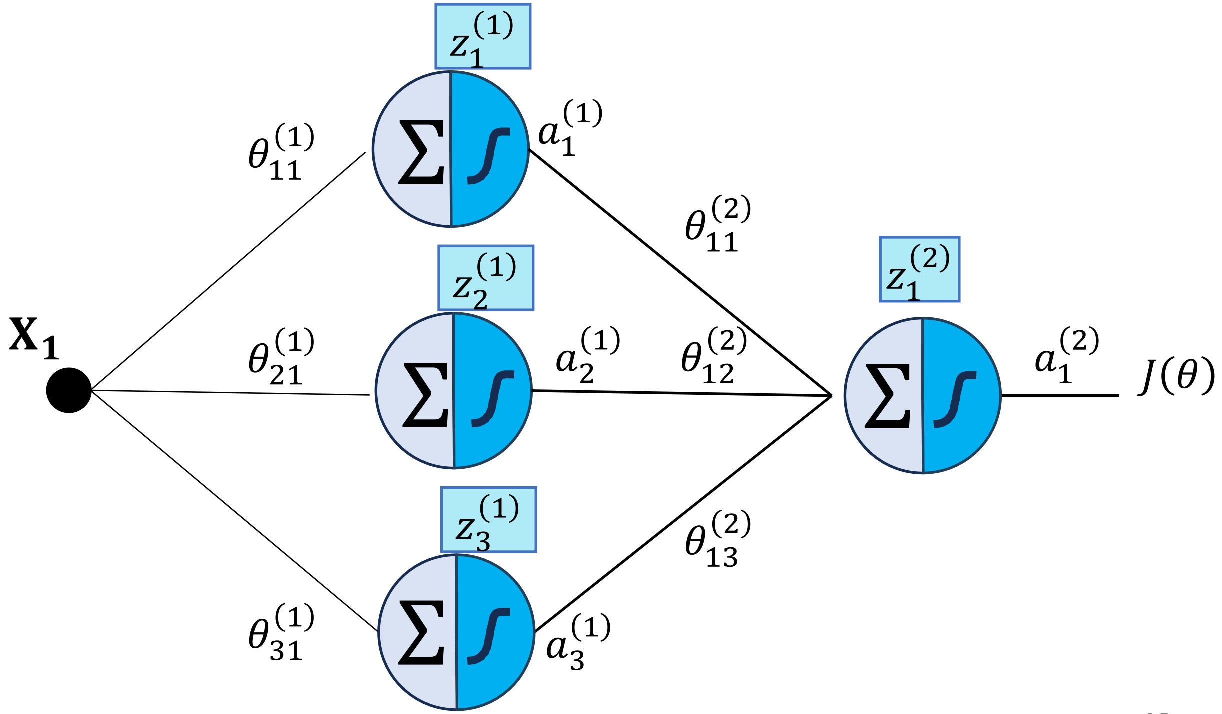
Generalizando o treinamento

- Temos três componentes para implementar o *backpropagation*
- Vamos revisar algumas questões que podem ter ficado de fora
  - *bias* – ou o termo livre
  - *Número de unidades*

1
$$\frac{\partial J}{\partial \theta^{(i)}} = \delta^{(i)} a^{(i-1)}$$

2
$$\delta^{(l)} = \left(\delta^{(l+1)} \theta^{(l+1)}\right) a'^{(l)}$$

3
$$\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}}$$

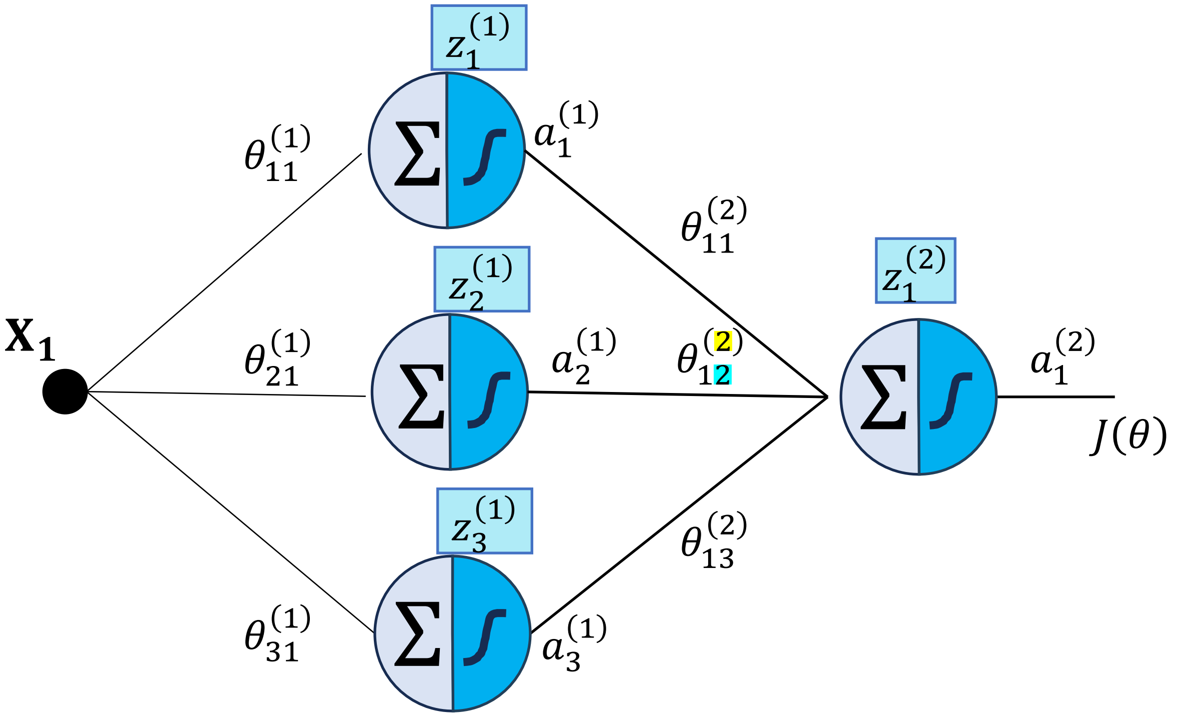


Compreendendo a notação

$$\theta_{jk}^{(l)}$$

- l: número da camada
- j: saída
- k: entrada

Por exemplo: $\theta_{12}^{(2)}$ é o peso da camada 2 que multiplica a ativação da unidade 2 da camada 1 e é usado como entrada na unidade 1 da camada 2



Generalizando o treinamento

- Vamos revisar algumas questões que podem ter ficado de fora
  - *bias* – ou o termo livre
  - *Número de unidades:* Multiplicações viram produto de Hadamard

1

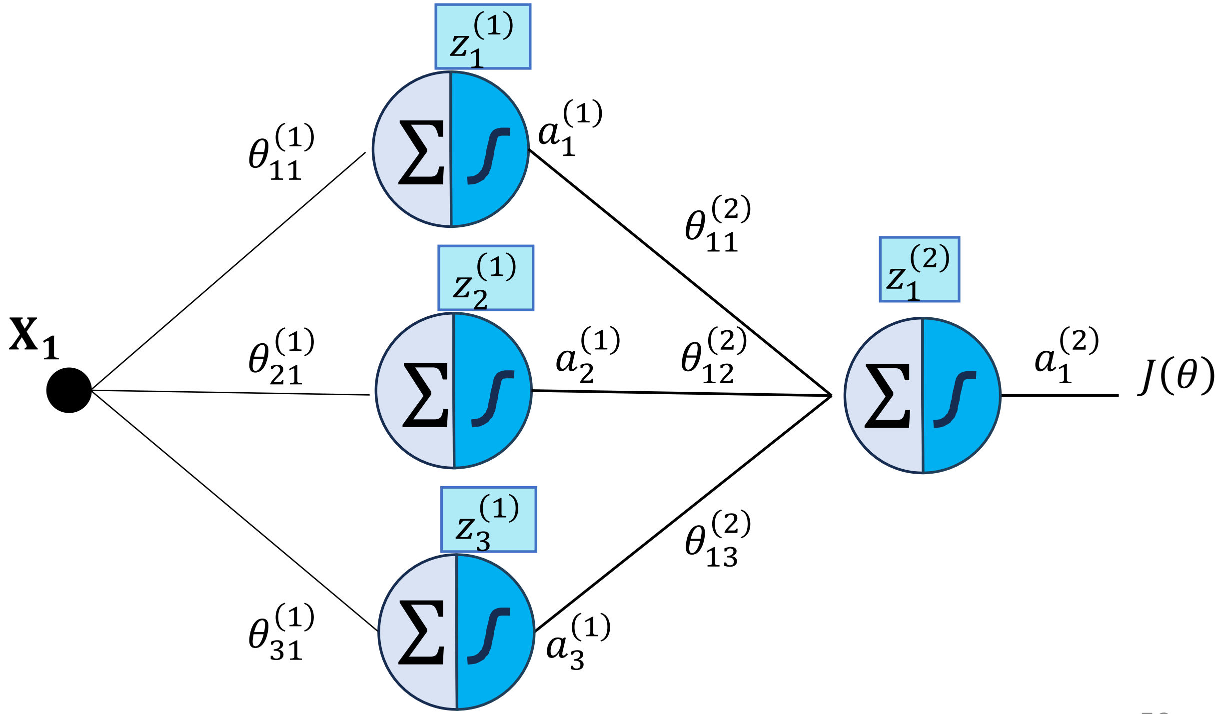
$$\frac{\partial J}{\partial \theta^{(i)}} = {\delta^{(i)}}^T a^{(i-1)}$$

2

$$\delta^{(l)} = \left(\delta^{(l+1)} \theta^{(l+1)}\right) \odot a'^{(l)}$$

3

$$\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \odot \frac{\partial a^{(L)}}{\partial z^{(L)}}$$



Generalizando o treinamento

- Todas as componentes do treinamento podem ser representadas e implementadas em operações envolvendo matrizes! ☺

1

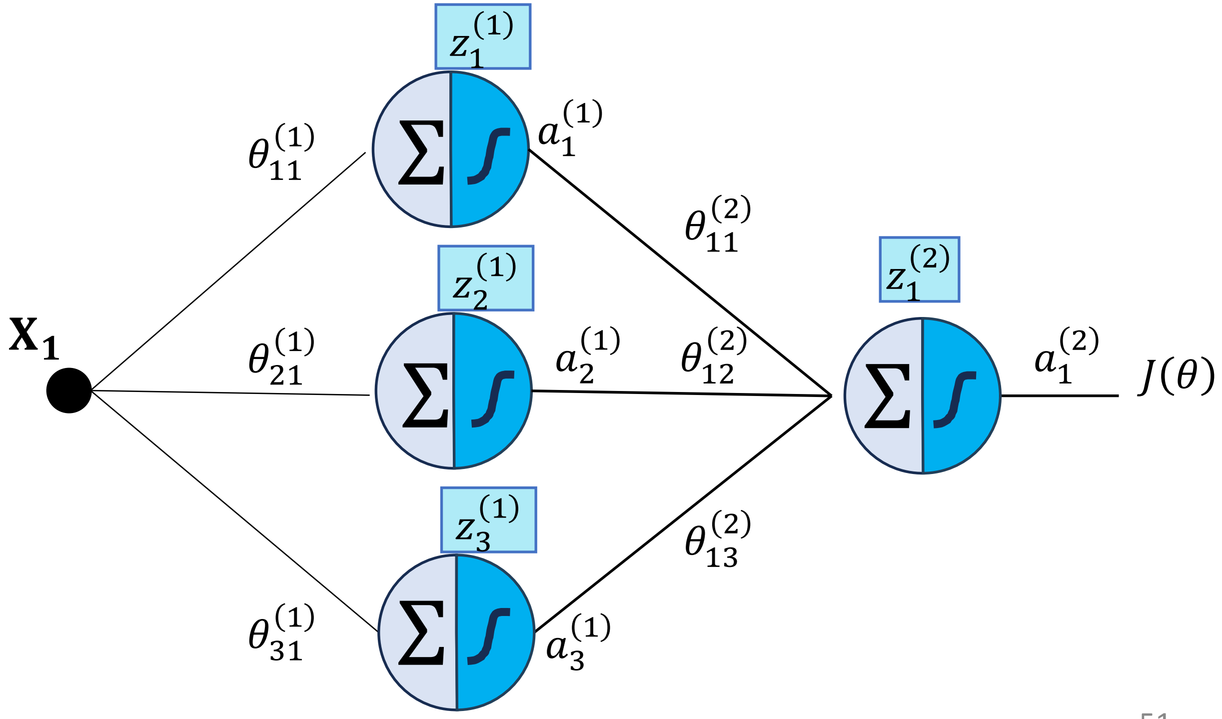
$$\boxed{\frac{\partial J}{\partial \theta^{(i)}} = \delta^{(i)^T} a^{(i-1)}}$$

2

$$\boxed{\delta^{(l)} = \left(\delta^{(l+1)} \theta^{(l+1)}\right) \odot a'^{(l)}}$$

3

$$\boxed{\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \odot \frac{\partial a^{(L)}}{\partial z^{(L)}}}$$

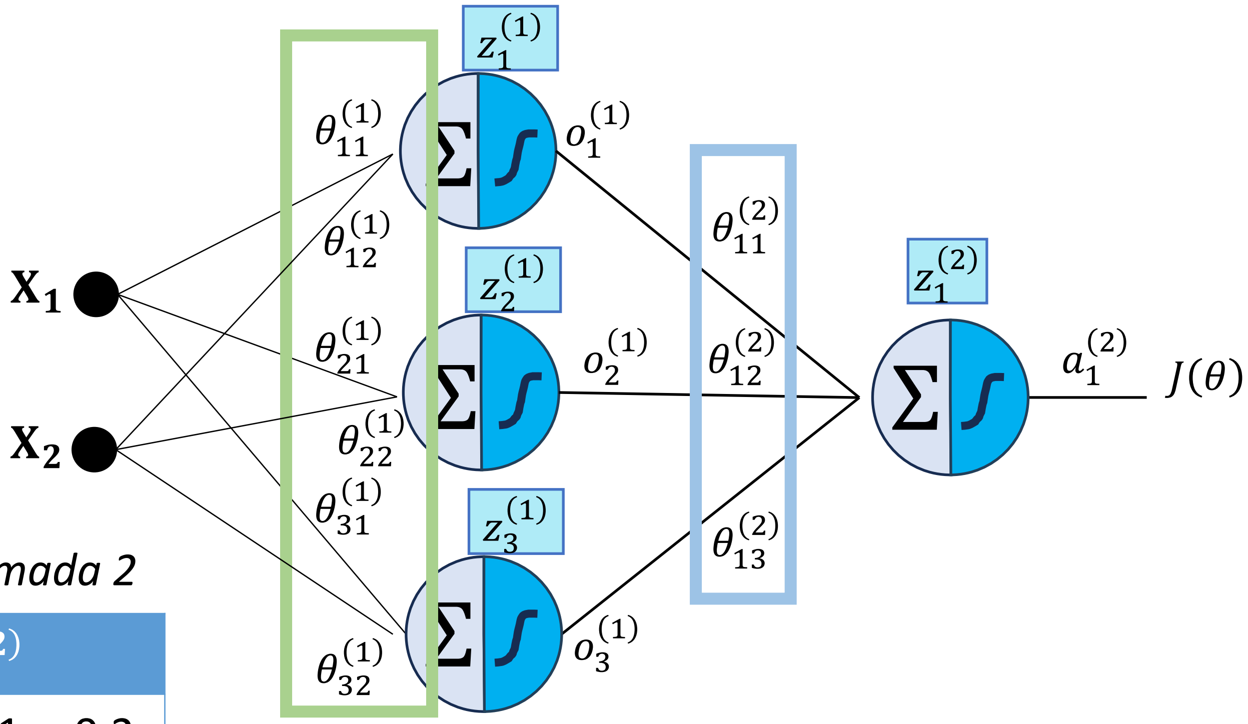


Um exemplo vetorizado

$$\boxed{\frac{\partial J}{\partial \theta^{(i)}} = {\delta^{(i)}}^T a^{(i-1)}} \quad 1$$

$$\boxed{\delta^{(l)} = \left(\delta^{(l+1)} \theta^{(l+1)}\right) \odot a'^{(l)}} \quad 2$$

$$\boxed{\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \odot \frac{\partial a^{(L)}}{\partial z^{(L)}}} \quad 3$$



Dataset

X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

Forward Pass

X_1	X_2
0.5	0.1
0.2	0.6

2x2

·

Pesos Camada 1

$\theta^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

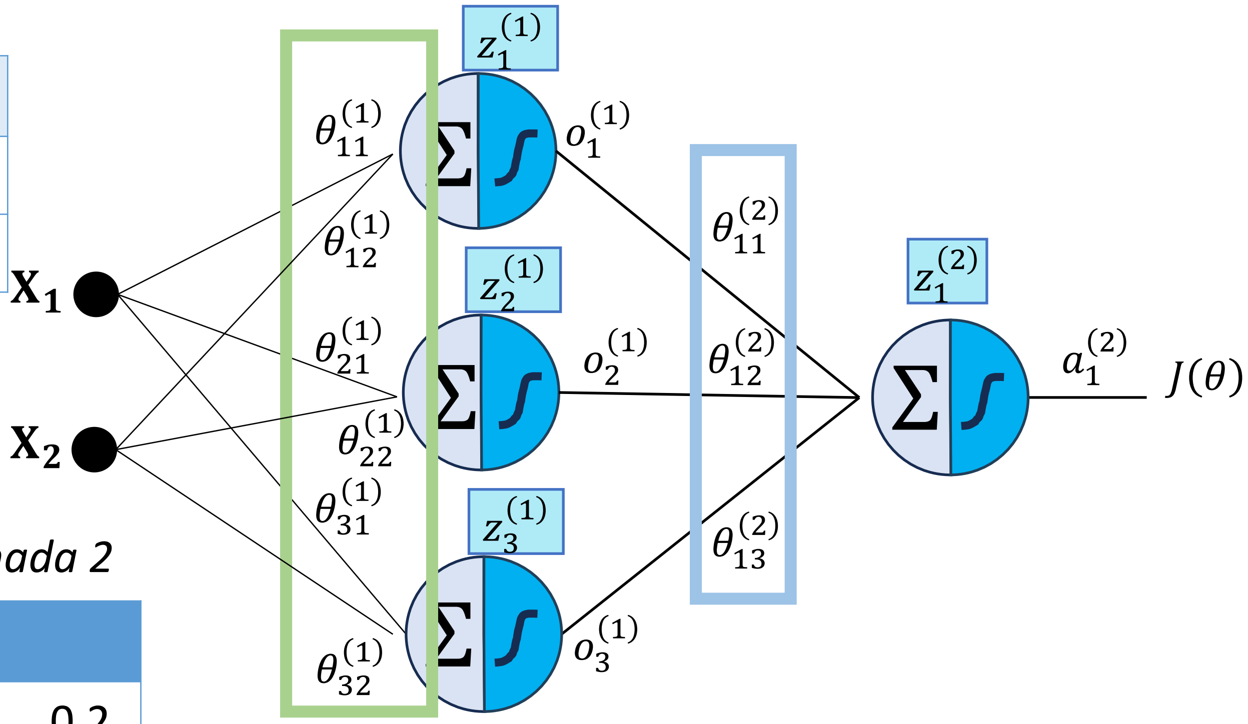
3x2

T

=

$z^{(1)}$		
0.27	0.29	-0.23
0.22	0.06	-0.26

2x3



Pesos Camada 2

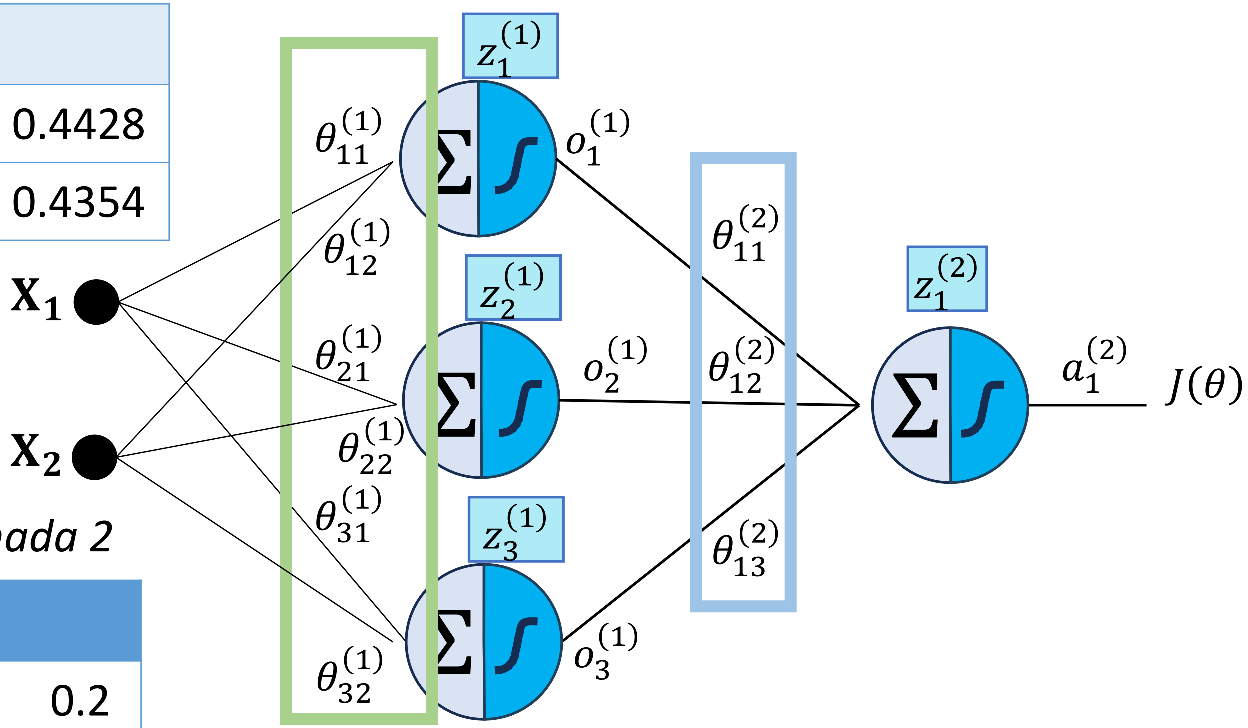
$\theta^{(2)}$		
0.7	-0.1	0.2

X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Forward Pass

$$\mathit{sigmoid}\left(\begin{array}{ccc} \multicolumn{3}{c}{\boldsymbol{z}^{(1)}} \\ \hline 0.27 & 0.29 & -0.23 \\ 0.22 & 0.06 & -0.26 \end{array}\right) = \begin{array}{ccc} \multicolumn{3}{c}{\boldsymbol{a}^{(1)}} \\ \hline 0.5671 & 0.5720 & 0.4428 \\ 0.5548 & 0.5150 & 0.4354 \end{array}$$

2x3



X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

Forward Pass

$\boldsymbol{a}^{(1)}$		
0.5671	0.5720	0.4428
0.5548	0.5150	0.4354

2x3

$\cdot$ *Pesos Camada 2*

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

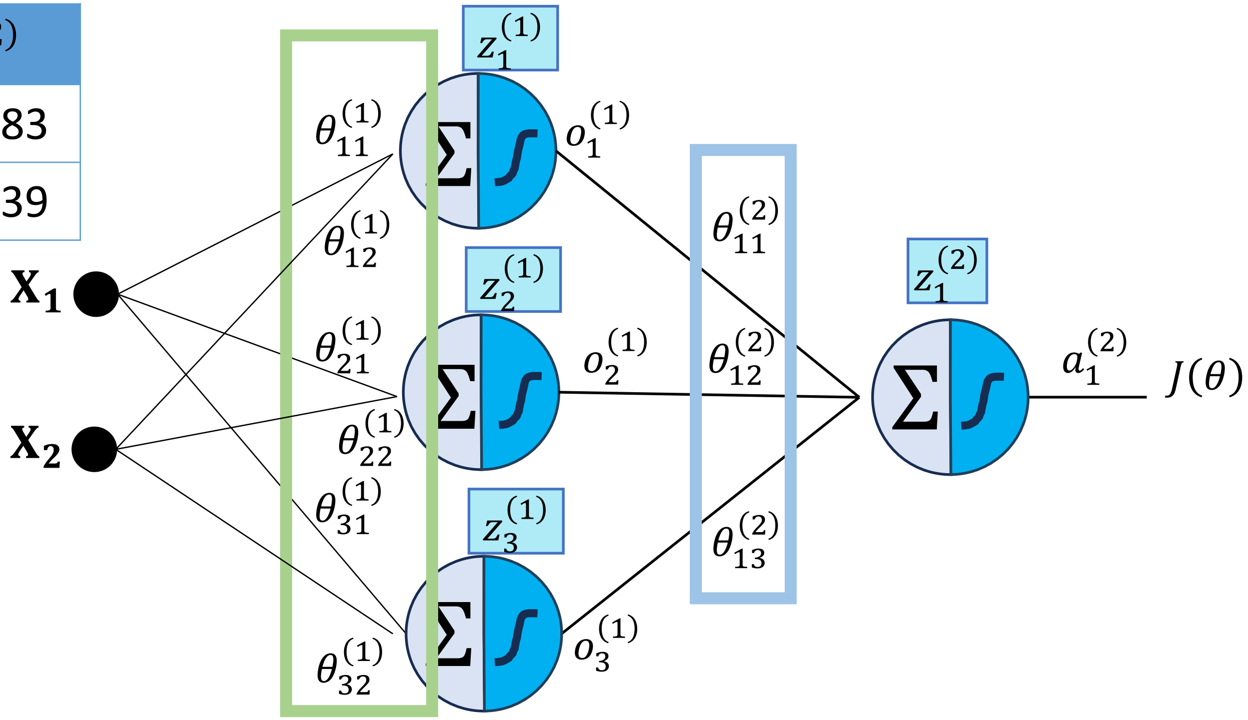
$^{\mathbf{T}}$

1x3

$=$

$\mathbf{z}^{(2)}$
0.4283
0.4239

2x1



X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

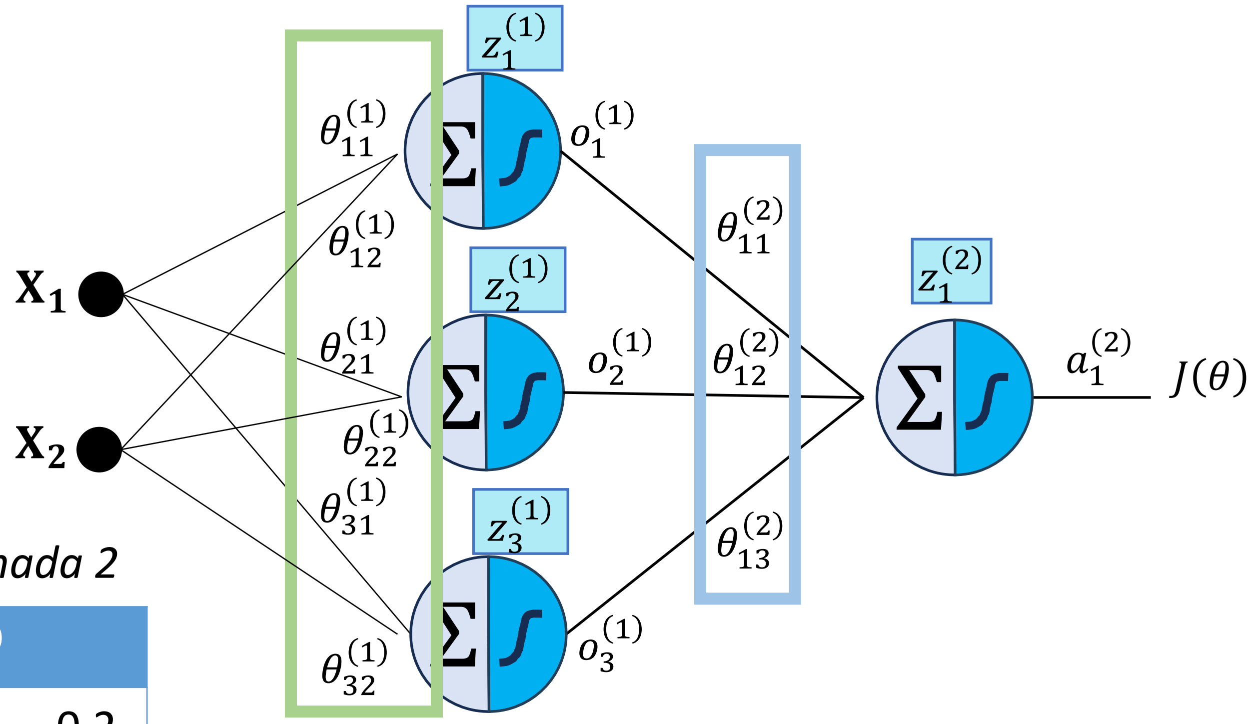
Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Forward Pass

$$sigmoid\left(\begin{array}{c}\mathbf{z}^{(2)}\\ \hline 0.4283\\ 0.4239\end{array}\right) = \begin{array}{c}\mathbf{a}^{(2)}\\ \hline 0.6055\\ 0.6044\end{array}$$

2x1 2x1



X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

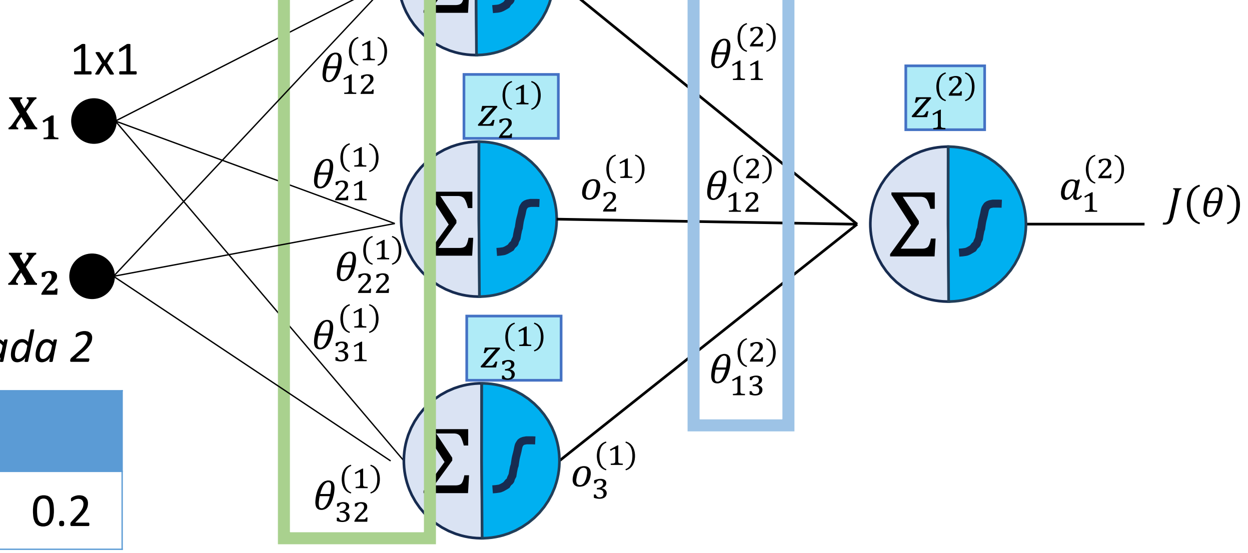
$\theta^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\theta^{(2)}$		
0.7	-0.1	0.2

Loss Function

$$\frac{1}{2}\left(\underset{2\text{x}1}{\overset{\mathbf{a}^{(2)}}{\begin{bmatrix}0.6055\\0.6044\end{bmatrix}}} - \underset{2\text{x}1}{\overset{\boldsymbol{y}}{\begin{bmatrix}0.7\\0.8\end{bmatrix}}}\right)^{\mathbf{T}} \cdot \left(\underset{2\text{x}1}{\overset{\mathbf{a}^{(2)}}{\begin{bmatrix}0.6055\\0.6044\end{bmatrix}}} - \underset{2\text{x}1}{\overset{\boldsymbol{y}}{\begin{bmatrix}0.7\\0.8\end{bmatrix}}}\right) = \underset{1\text{x}1}{\overset{\boldsymbol{J}}{\begin{bmatrix}0.02359\end{bmatrix}}}$$



X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

Backward Pass

3

$$\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \odot \frac{\partial a^{(L)}}{\partial z^{(L)}}$$

$a_1^{(2)}$
0.605
0.604

2x1

\-

y
0.7
0.9

2x1

=

$\frac{\partial J}{\partial a^{(L)}}$
-0.095
-0.196

2x1

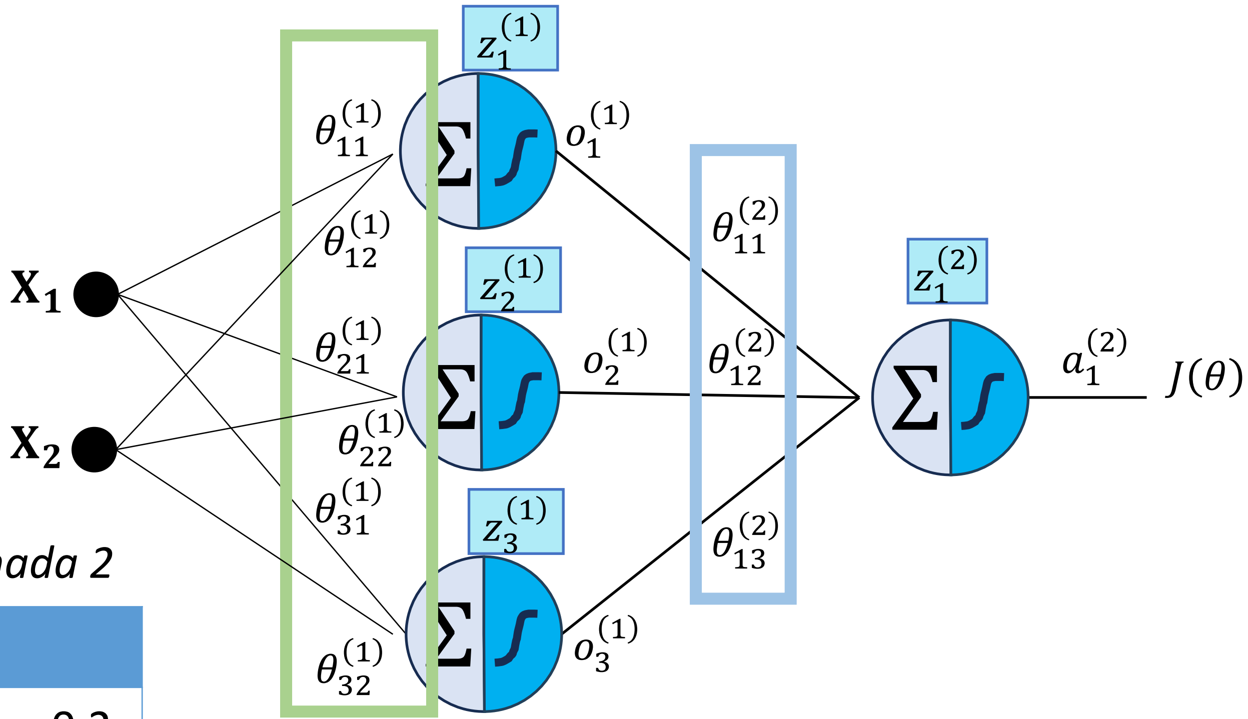
X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\theta^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\theta^{(2)}$		
0.7	-0.1	0.2



J
0.02359

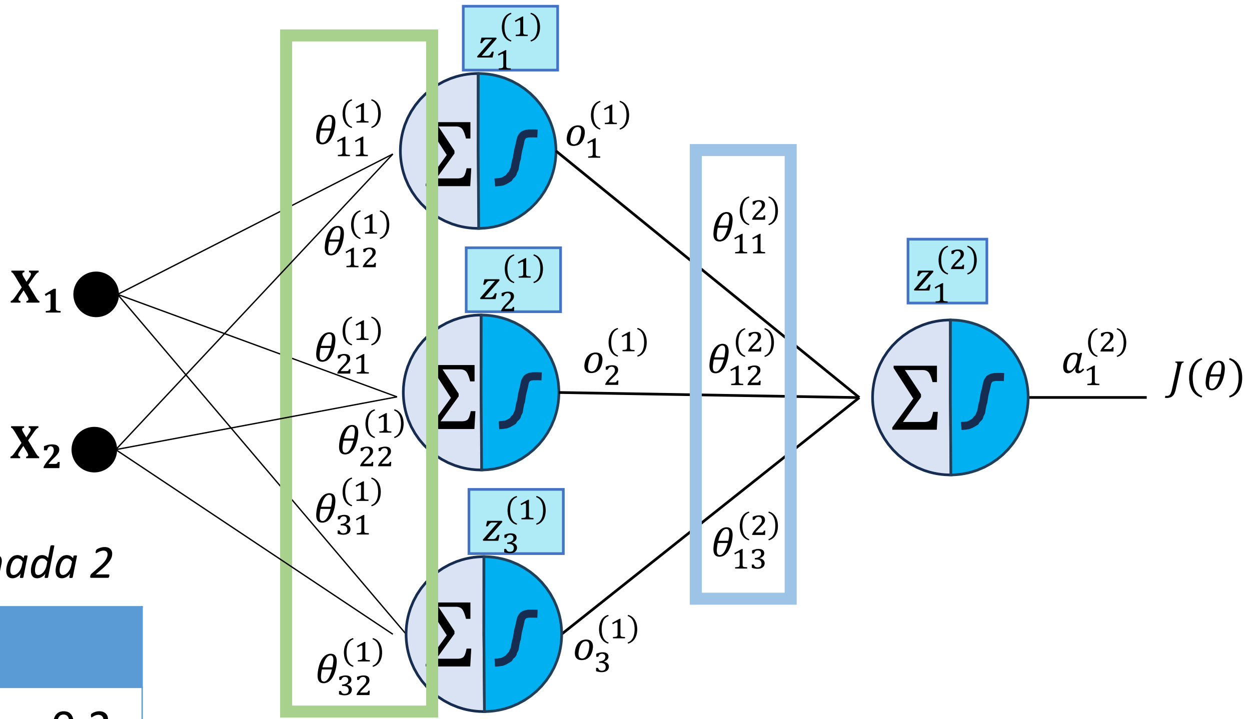
Backward Pass

$$\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \odot \frac{\partial a^{(L)}}{\partial z^{(L)}}$$

3

$\frac{\partial J}{\partial a^{(L)}}$
-0.095
-0.196

$\boldsymbol{a_1^{(2)}}$	·	1 -	$\boldsymbol{a_1^{(2)}}$	=	$\frac{\partial o^{(L)}}{\partial z^{(L)}}$
0.605			0.605		0.238975
0.604			0.604		0.239096



X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta^{(1)}}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\boldsymbol{\theta^{(2)}}$		
0.7	-0.1	0.2

$\boldsymbol{J}$
0.02359

Backward Pass

$$\delta^{(L)} = \frac{\partial J}{\partial a^{(L)}} \odot \frac{\partial a^{(L)}}{\partial z^{(L)}}$$ 3

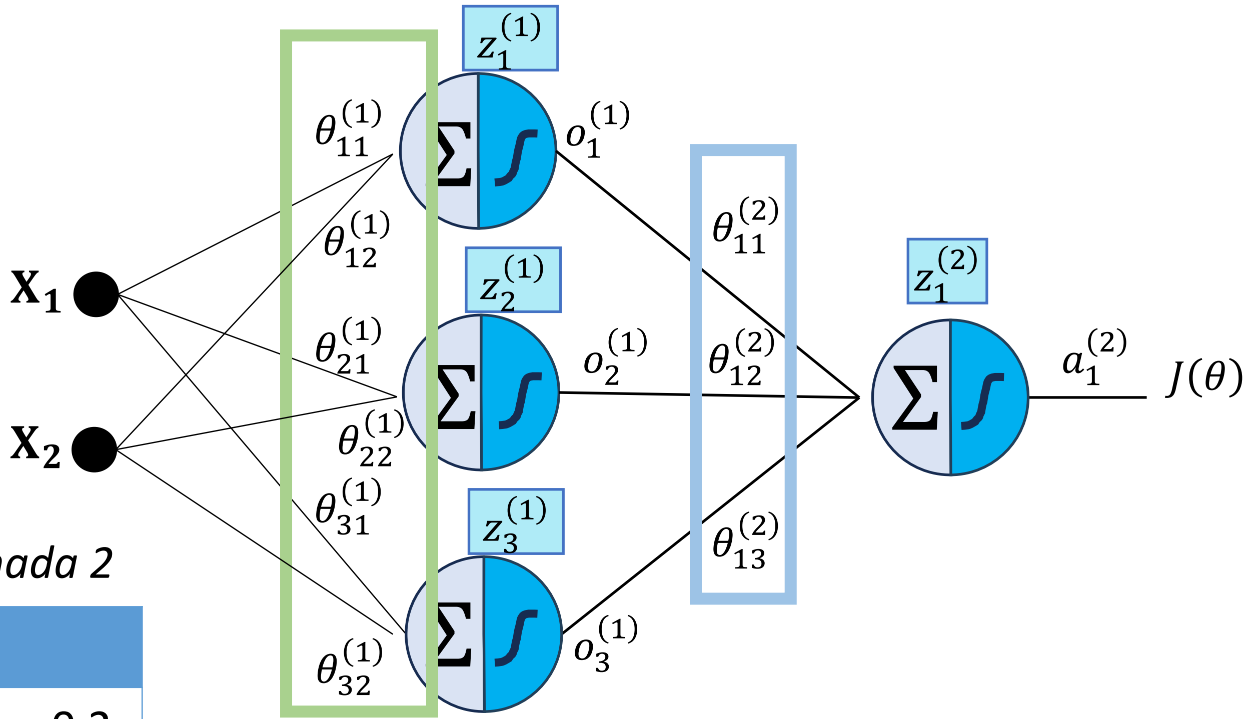
$\frac{\partial J}{\partial a^{(L)}}$
-0.095
-0.196

$\odot$

$\frac{\partial o^{(L)}}{\partial s^{(L)}}$
0.238975
0.239096

=

$\delta^{(L)}$
-0.02258
-0.04676



X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

J
0.02359

Backward Pass

$$\boxed{\delta^{(l)} = \left(\delta^{(l+1)}\theta^{(l+1)}\right)\odot a'^{(l)}} \quad ②$$

$\delta^{(L)}$
-0.02258
-0.04676

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

$\odot$

$\mathbf{a}'^{(1)}$		
0.2455	0.2448	0.2467
0.2470	0.2798	0.2458

$=$

$\delta^{(1)}$		
-0.0039	0.0006	-0.0011
-0.0080	0.0012	-0.0023

X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

J
0.02359

Backward Pass

$$\boxed{\frac{\partial J}{\partial \theta^{(i)}} = \delta^{(i)^T} a^{(i-1)}}$$ 1

$\left(\delta^{(1)}\right)^T$

$\delta^{(1)}$		
-0.0039	0.0006	-0.0011
-0.0080	0.0012	-0.0023

X_1	X_2
0.5	0.1
0.2	0.6

=

$\frac{\partial J}{\partial \theta^{(1)}}$	
-0.00356	-0.00524
0.00051	0.00075
-0.00102	-0.00149

$\left(\delta^{(L)}\right)^T$

$\delta^{(L)}$
-0.02258
-0.04676

$a^{(1)}$		
0.5671	0.5720	0.4428
0.5548	0.5150	0.4354

=

$\frac{\partial J}{\partial \theta^{(2)}}$		
-0.0387	-0.0370	-0.0304

Optimizer

J
0.02359

$$\boxed{\theta_t^{(l)} = \theta_{t-1}^{(l)} - \eta \nabla_\theta J}$$

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

-0.1

$\frac{\partial J}{\partial \theta^{(1)}}$	
-0.00356	-0.00524
0.00051	0.00075
-0.00102	-0.00149

=

$\boldsymbol{\theta}^{(1)}$	
0.5004	0.2005
0.5999	-0.10007
-0.3999	-0.2999

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

-0.1

$\frac{\partial J}{\partial \theta^{(2)}}$		
-0.0387	-0.0370	-0.0304

=

$\boldsymbol{\theta}^{(2)}$		
0.7039	-0.0963	0.2030

Forward Pass

X_1	X_2
0.5	0.1
0.2	0.6

2x2

·

Pesos Camada 1

$\theta^{(1)}$	
0.5004	0.2005
0.5999	-0.10007
-0.3999	-0.2999

T

3x2

=

$z^{(1)}$		
0.2702	0.28997	-0.2299
0.2204	0.0599	-0.2599

2x3

X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 2

$\theta^{(2)}$		
0.7039	-0.0963	0.2030

Forward Pass

$$sigmoid\left(\begin{array}{ccc} & z^{(1)} & \\ 0.2702 & 0.28997 & -0.2299 \\ 0.2204 & 0.0599 & -0.2599 \end{array}\right) = \begin{array}{ccc} & a^{(1)} & \\ 0.5671 & 0.5720 & 0.4428 \\ 0.5549 & 0.5150 & 0.4354 \end{array}$$

2x3

2x3

X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\theta^{(1)}$	
0.50019	0.20004
0.59997	-0.10001
-0.39994	-0.29998

Pesos Camada 2

$\theta^{(2)}$		
0.70129	-0.09870	0.20100

Forward Pass

$a^{(1)}$		
0.5671	0.5720	0.4428
0.5549	0.5150	0.4354

2x3

$\cdot$

Pesos Camada 2

$\theta^{(2)}$		
0.70129	-0.09870	0.20100

1x3

T

=

$z^{(2)}$
0.4340
0.4294

2x1

X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

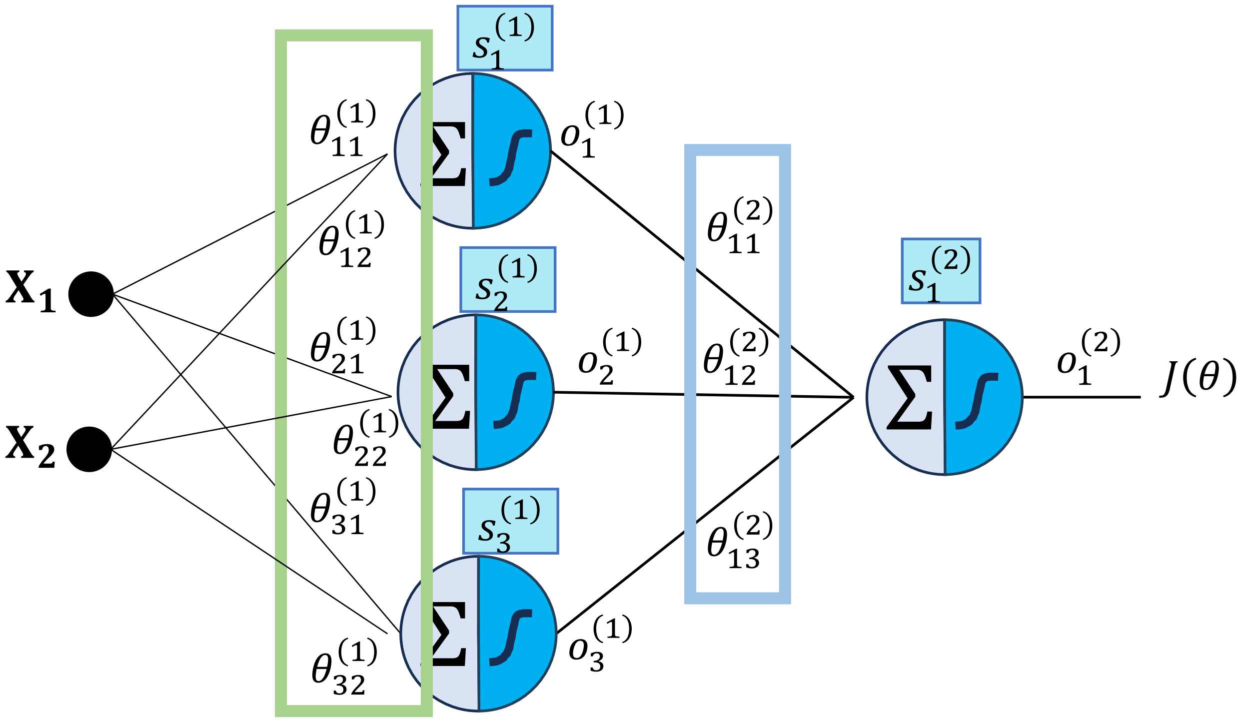
Pesos Camada 1

$\theta^{(1)}$	
0.50019	0.20004
0.59997	-0.10001
-0.39994	-0.29998

Forward Pass

$$sigmoid\left(\begin{array}{c}\mathbf{z}^{(2)} \\ \hline 0.4340 \\ 0.4294\end{array}\right) = \begin{array}{c}\boldsymbol{a}_1^{(2)} \\ \hline 0.6068 \\ 0.6057\end{array}$$

2x1 2x1



X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.50019	0.20004
0.59997	-0.10001
-0.39994	-0.29998

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.70129	-0.09870	0.20100

Loss Function

$$\left(\begin{array}{c} \mathbf{a}^{(2)} \\ \hline 0.6068 \\ 0.6057 \end{array} - \begin{array}{c} \boldsymbol{y} \\ \hline 0.7 \\ 0.8 \end{array} \right)^{T} \cdot \left(\begin{array}{c} \mathbf{a}^{(2)} \\ \hline 0.6068 \\ 0.6057 \end{array} - \begin{array}{c} \boldsymbol{y} \\ \hline 0.7 \\ 0.8 \end{array} \right) = \begin{array}{c} \boldsymbol{J} \\ \hline 0.02321 \end{array}$$

2x1 2x1 2x1 2x1 1x1

X_1	X_2	y
0.5	0.1	0.6
0.2	0.6	0.8

Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.50019	0.20004
0.59997	-0.10001
-0.39994	-0.29998

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.70129	-0.09870	0.20100

O resultado

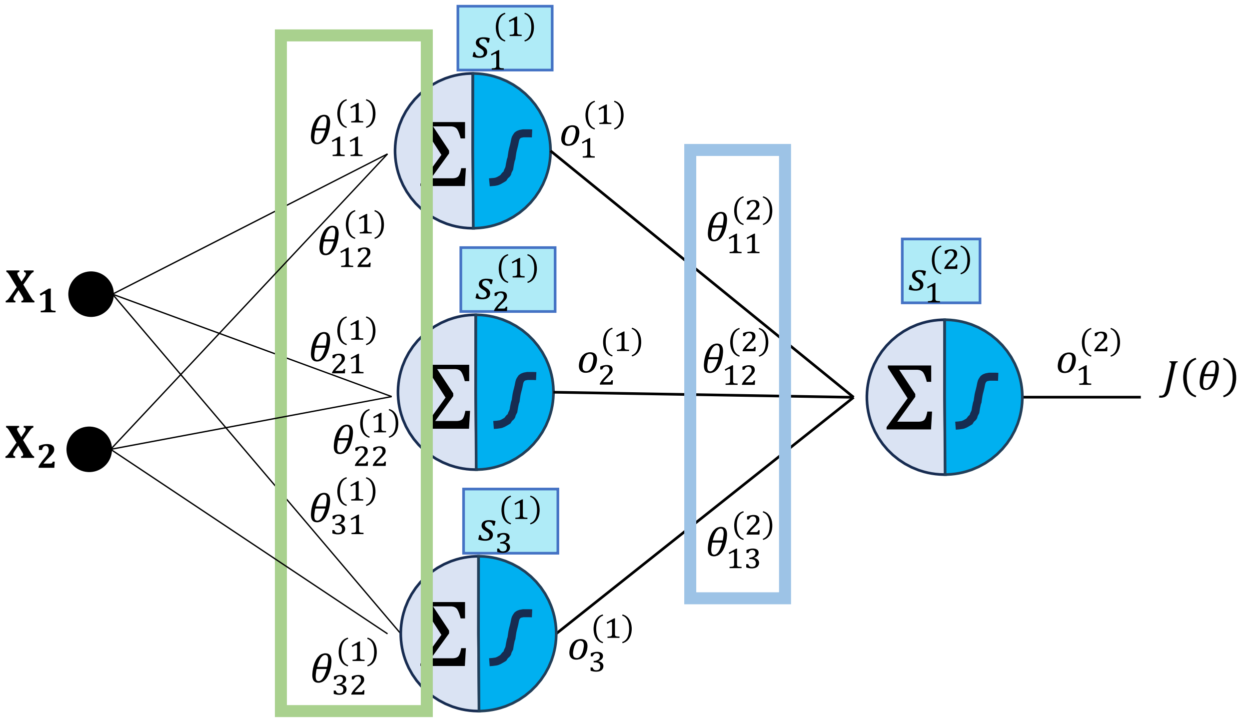
Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.5	0.2
0.6	-0.1
-0.4	-0.3

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.7	-0.1	0.2

$\boldsymbol{J}$
0.02359



Pesos Camada 1

$\boldsymbol{\theta}^{(1)}$	
0.50019	0.20004
0.59997	-0.10001
-0.39994	-0.29998

Pesos Camada 2

$\boldsymbol{\theta}^{(2)}$		
0.70129	-0.09870	0.20100

$\boldsymbol{J}$
0.02321

Resumindo

- Podemos treinar redes neurais de múltiplas camadas usando descida de gradiente
- O processo de treinamento envolve 4 etapas
  - 1º Computar todas as saídas da rede (*Forward Pass*)
  - 2º Computar o quão diferente as saídas são em relação a variável alvo (*Loss Function*)
  - 3º Computar o gradiente do erro em relação aos parâmetros da rede (*Backward Pass*)
  - 4º Atualizar os pesos (*Optimizer*)
- Todo o treinamento pode ser implementado de forma vetorizada

Referências:

Sugere-se fortemente a leitura de:

- Capítulo 2 - NIELSEN, Michael A. Neural networks and deep learning. San Francisco, CA, USA: Determination press, 2015. (http://neuralnetworksanddeeplearning.com/index.html)
- Primeira publicação sobre o Backpropagation - RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. nature, v. 323, n. 6088, p. 533-536, 1986.