

O método Adam e outros tipos de camadas



- Até agora, estudamos Redes Neurais imaginando que o treinamento dos seus parâmetros é feito usando uma aplicação tradicional do Método do Gradiente, tal como, por exemplo,

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

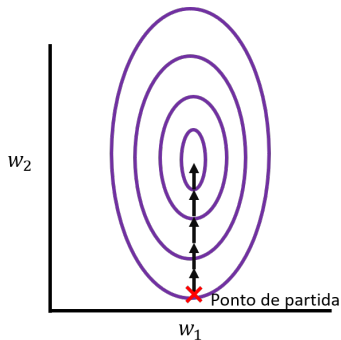
onde α é um valor fixo > 0 (taxa de aprendizado).

- Entretanto, hoje em dia existem métodos de otimização avançada aplicada à redes neurais que vão ajustando α automaticamente durante o processo de treinamento. Nesse contexto insere-se o **Método Adam**

Método do Gradiente versus Método Adam

Método do Gradiente *versus* Método Adam

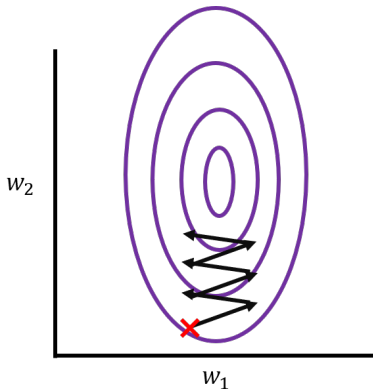
Primeiro caso: Utilizando um valor fixo e pequeno para α , o Método do Gradiente tenderá a dar vários pequenos passos em direção ao mínimo de $J(\vec{w}, b)$, conforme ilustrado abaixo:



- Para um exemplo como esse, o Método Adam perceberá que é possível **aumentar** o valor de α para agilizar o processo de chegada até o mínimo de $J(\vec{w}, b)$.

Método do Gradiente versus Método Adam

Segundo caso: utilizando um valor fixo e muito grande para α , o Método do Gradiente tenderá a dar passos maiores, porém poderá ocorrer o que está ilustrado abaixo:



- Nesse caso, o Método Adam perceberá que é melhor **reduzir** o valor de α para suavizar o processo de chegada até o mínimo de $J(\vec{w}, b)$.

Características do Método Adam

- A sigla 'Adam' abrevia a expressão **A**daptive **M**oment estimation
- Ele utiliza uma taxa de aprendizado α diferente para cada parâmetro que está sendo treinado, conforme exemplo abaixo:

$$\begin{aligned}w_1 &= w_1 - \alpha_1 \frac{\partial}{\partial w_1} J(\vec{w}, b) \\&\dots \\w_{12} &= w_{12} - \alpha_{12} \frac{\partial}{\partial w_{12}} J(\vec{w}, b)\end{aligned}$$

Características do Método Adam

- A sigla 'Adam' abrevia a expressão **Adaptive Moment estimation**
- Ele utiliza uma taxa de aprendizado α diferente para cada parâmetro que está sendo treinado, conforme exemplo abaixo:

$$\begin{aligned}w_1 &= w_1 - \alpha_1 \frac{\partial}{\partial w_1} J(\vec{w}, b) \\&\dots \\w_{12} &= w_{12} - \alpha_{12} \frac{\partial}{\partial w_{12}} J(\vec{w}, b)\end{aligned}$$

Ideia principal:

- Se w_j (ou b_j) está se movendo de forma constante, com pouca variação em termos de direção, aumentar α_j (Primeiro caso que vimos)
- Se w_j (ou b_j) está oscilando muito em termos de direção, diminuir α_j (Segundo caso que vimos)

Características do Método Adam

- A sigla 'Adam' abrevia a expressão **Adaptive Moment estimation**
- Ele utiliza uma taxa de aprendizado α diferente para cada parâmetro que está sendo treinado, conforme exemplo abaixo:

$$\begin{aligned}w_1 &= w_1 - \alpha_1 \frac{\partial}{\partial w_1} J(\vec{w}, b) \\&\dots \\w_{12} &= w_{12} - \alpha_{12} \frac{\partial}{\partial w_{12}} J(\vec{w}, b)\end{aligned}$$

Ideia principal:

- Se w_j (ou b_j) está se movendo de forma constante, com pouca variação em termos de direção, aumentar α_j (Primeiro caso que vimos)
- Se w_j (ou b_j) está oscilando muito em termos de direção, diminuir α_j (Segundo caso que vimos)

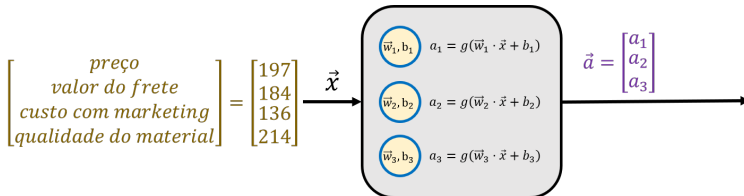
Observação:

- Nós já estamos utilizando o Método Adam nos nossos códigos, entregando a ele um palpite inicial acerca da taxa de aprendizado α .
- Ainda vale a pena **mantermos a estratégia** de testar diferentes valores para α para verificar se a convergência torna-se mais rápida ou não, apesar do método já ter uma estratégia automatizada para seleção desse parâmetro ao longo das iterações.

Outros tipos de camadas para as nossas redes neurais

Outros tipos de camadas para as nossas redes neurais

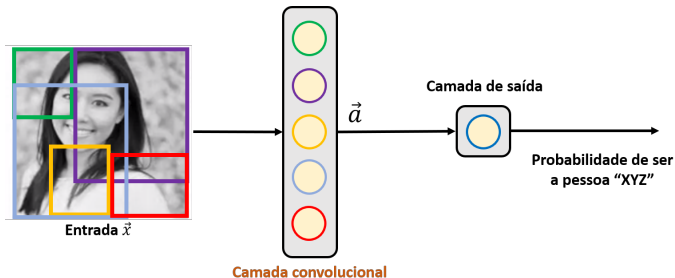
- Conforme ilustrado abaixo, em camadas do tipo **Dense**, a saída de cada neurônio é uma função de **todas as ativações** de saída vindas da camada anterior.
- No exemplo, por se tratar da Camada 1, note que as ativações de saída vindas da camada anterior são o próprio $\vec{x} = \vec{a}^{[0]}$ (características de entrada).



A seguir, veremos um exemplo de um outro tipo de camada que é possível de ser utilizada numa rede neural

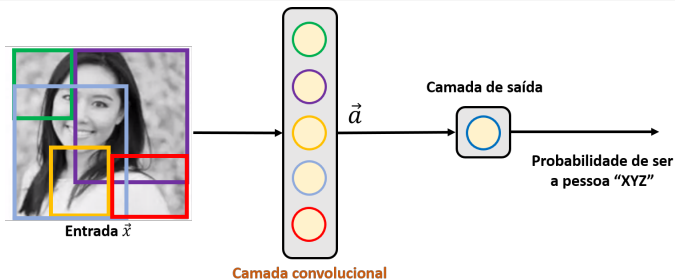
Camadas convolucionais

- Conforme ilustrado abaixo, em camadas do tipo **Convolucionais**, a saída de cada neurônio é uma função apenas de **parte das ativações** de saída vindas da camada anterior.



Camadas convolucionais

- Conforme ilustrado abaixo, em camadas do tipo **Convolucionais**, a saída de cada neurônio é uma função apenas de **parte das ativações** de saída vindas da camada anterior.



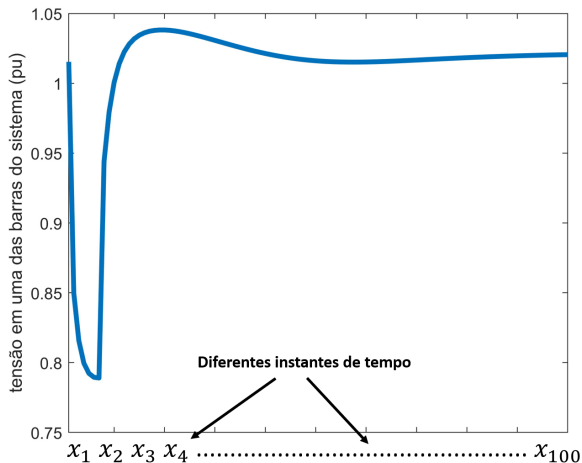
Qual é a vantagem de fazermos isso?

- O processamento torna-se mais rápido.
- Menos dados de treinamento são necessários, já que menos parâmetros estarão presentes no modelo (tendência menor de ocorrer *overfitting*)

Observação

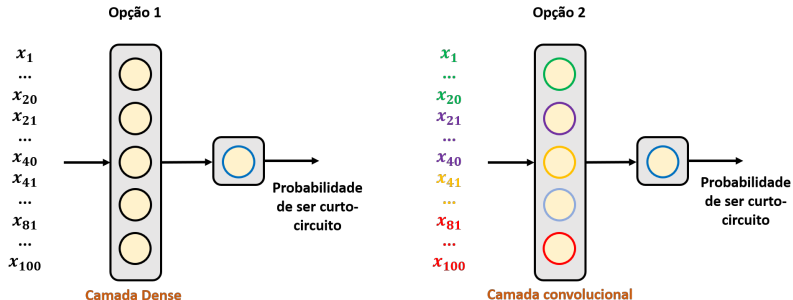
- Se a sua rede neural possui muitas camadas convolucionais, então ela pode também ser chamada de **Rede neural convolucional**

Presença ou não de curto-circuito em um SEP



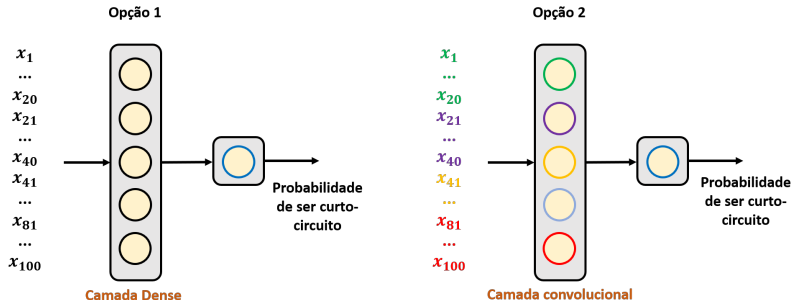
Qual é a probabilidade desse sinal demonstrar a presença de um curto-circuito no SEP? Na sua opinião, está ocorrendo um curto?

Presença ou não de curto-circuito em um SEP



- Na **Opção 1**, todos os neurônios da camada densa recebem todas as características x_1, \dots, x_{100} (toda a janela temporal de dados).
- Na **Opção 2**, note que houve uma distribuição temporal seletiva de dados para os neurônios (janelamento temporal).

Presença ou não de curto-circuito em um SEP



- Na **Opção 1**, todos os neurônios da camada densa recebem todas as características x_1, \dots, x_{100} (toda a janela temporal de dados).
- Na **Opção 2**, note que houve uma distribuição temporal seletiva de dados para os neurônios (janelamento temporal).

Observação final: É também possível inserir mais camadas convolucionais intermediárias. Seguindo a ideia de camada convolucional, cada neurônio dessas camadas receberá apenas **parte das ativações** de saída vindas da camada anterior

Iremos agora verificar como usar o Tensorflow para treinar um modelo de reconhecimento de dígitos de 0 a 9 escritos à mão.

OBS: Continuaremos a utilizar o Método Adam, mas não usaremos nesse código camadas convolucionais.

Nome do arquivo que trabalharemos agora:

código - reconhecendo dígitos de 0 e 9 escritos a mão.ipynb

Parte 1

Rode todo o "codigo - reconhecendo digitos de 0 e 9 escritos a mao.ipynb" sem fazer qualquer tipo de alteração. Certifique-se de que você o compreendeu.

Parte 2

- 1 Verifique se o aumento de número de épocas aumenta a taxa de acerto do modelo.