

# Inteligência Artificial

## Tipos de Redes Neurais

Felipe Augusto Lima Reis

[felipe.reis@ifmg.edu.br](mailto:felipe.reis@ifmg.edu.br)



**INSTITUTO  
FEDERAL**  
Minas Gerais

# Sumário

- 1 Tipos de Aprendizado
- 2 R. Recorrentes
- 3 R. Não Supervisionadas
- 4 R. Convolucionais

# TIPOS DE APRENDIZADO

# Tipos de Aprendizado

- Tradicionalmente redes neurais aprendem de forma supervisionada
  - Esse método exige que os dados sejam previamente classificados;
  - Os resultados gerados são comparados aos desejados;
- No entanto, existem ainda outras formas de aprendizado
  - Alguns algoritmos possibilitam que as redes aprendam sem classificação prévia das informações;
  - Outros, fornecem apenas informações parciais, deixando a rede descobrir sozinha novas características;
  - Redes, ainda, podem, aprender com base em premiações e punições.

# Aprendizado supervisionado

- A rede aprende utilizando dados de treinamento pré-classificados (classificação manual humana);
- A rede modifica os pesos para aumento de sua precisão;
- O algoritmo de treinamento supervisiona a diferença entre as saídas da rede em relação às respostas esperadas
  - A diferença de valores é utilizada para adequação dos pesos;

# Aprendizado supervisionado

- A rede é considerada treinada quando (uma ou mais opções):
  - Atingir um número pré-determinado de épocas de treinamento;
  - Não melhorar seu desempenho por um determinado número de épocas (*early stopping*).
  - Atingir uma condição pré-estabelecida<sup>1</sup>.

---

<sup>1</sup>Nível de um perda pequeno, acurácia/precisão alta, entre outras métricas

# Aprendizado não supervisionado

- Ocorre sem intervenção humana;
- Rede deve se auto-organizar em relação às similaridades existentes entre os elementos;
- Pesos das conexões são ajustados pelo algoritmo de aprendizado;
- Útil em situações onde dados precisam ser classificados ou *clusterizados* em classes ainda indefinidas
  - Desse modo, não requer classificação humana das amostras.

# Aprendizado semi supervisionado



- Intermediário entre o aprendizado supervisionado e o não supervisionado;
- São fornecidas informações incompletas para treinamento, cabendo ao software a finalização do conjunto de treinamento;
- Utilizada em detecção de fraudes, classificação de vídeos, definição de *clusters* (grupos) de clientes, etc.



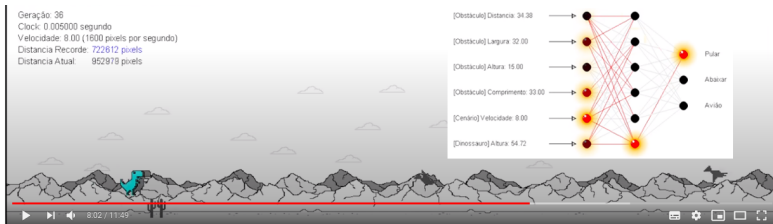
# Aprendizado por reforço



- Treinamento destinado a redes que interagem com um ambiente dinâmico, com objetivo de desempenhar uma ação;
- A rede recebe uma recompensa (ou premiação), caso atinja o objetivo, ou uma punição, caso falhe;
- O objetivo da rede é obter o maior valor de recompensa possível.

# Aprendizado por reforço

- Utilizada, por exemplo, para ensinar redes neurais a executarem tarefas, como jogar jogos eletrônicos<sup>2</sup>.



Fonte: Adaptado de [Universo Programado, 2019] - <https://www.youtube.com/watch?v=NZ1IYr1s1Ak><sup>3</sup>

<sup>2</sup> Pode ser necessário programar (simular) o ambiente, para interação.

<sup>3</sup> O procedimento descrito no vídeo inclui passos não estritamente relacionados ao aprendizado por reforço.

# REDES RECORRENTES

# Redes Recorrentes

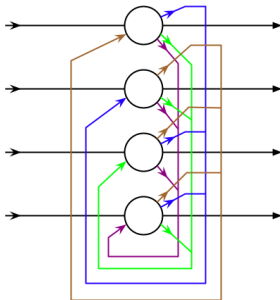
- Redes com alimentação para frente são acíclicas
  - Os dados somente trafegam das entradas para a saída da rede;
  - Uma vez treinada, seu estado é fixo, mesmo que novas entradas sejam apresentadas à rede;
  - Isso indica que a rede não possui memória [Coppin, 2004];
- Redes recorrentes possuem conexões de nós de saída para nós de entrada;
  - Podem ser possíveis, inclusive, conexões arbitrárias, entre quaisquer nós.

# Redes Recorrentes

- **Redes recorrentes** são utilizadas em situações nas quais a saída depende tanto dos dados de entrada quanto das entradas anteriores
  - Exemplo: controle sistemas dinâmicos, previsão climática, predição de ações, etc.;
- As redes recorrentes buscam manter o equilíbrio e a estabilidade [Coppin, 2004];
- Os valores de estabilidade da rede são conhecidos como memórias fundamentais [Coppin, 2004].

# Redes de Hopfield

- **Redes de Hopfield** são um modelo de rede recorrente criado por John Hopfield em 1982;
- Essas redes nem sempre são estáveis, podendo oscilar entre valores estáveis de saída [Coppin, 2004].



Fonte: [Wikipedia contributors, 2020c]

# Redes de Hopfield

- São consideradas *attractor networks*<sup>4</sup>, uma vez que seu resultado é atraído por certos valores estáveis [Coppin, 2004]
  - Valores estáveis são conhecidos como memórias fundamentais;
  - A rede é capaz de aprender conjuntos de estados.
- Redes de Hopfield, originalmente, utilizam a seguinte função de ativação, que permite os valores de -1, 0 e 1:

$$\text{Sign}(X) = \begin{cases} +1 & \text{for } X > 0 \\ -1 & \text{for } X < 0 \end{cases}$$

Fonte: [Coppin, 2004]

---

<sup>4</sup>Tradução direta: Rede de atratores

# Redes de Hopfield

- Os pesos da rede são normalmente definidos em forma de matriz, de acordo com a fórmula:

$$W = \sum_{i=1}^N X_i X_i^t - NI$$

onde,

- $X_i$ : vetor de entrada, representando  $m$  valores de entrada da rede;
- $X_i^t$ : matrix transposta de  $X_i$ ;
- $I$ : matriz identidade  $m \times m$ ;
- $N$ : número de estados  $X_i$  que serão aprendidos;



# Redes de Hopfield



- Os valores de saída da rede são dados por:

$$Y_i = \text{Sign}(WX_i - \theta)$$

onde,

- $X_i$ : vetor de entrada, representando  $m$  valores de entrada da rede;
- $\theta$ : matriz correspondente ao limiar de ativação;

# REDES NÃO SUPERVISIONADAS

# Redes Não Supervisionadas



- Utilizam treinamento não supervisionado
  - Não existe uma classificação prévia esperada, ou seja, não há indicação de quais valores são adequados ou não;
- As redes devem descobrir por si só padrões, características, correlações ou categorias a partir dos dados de entrada
  - Essas características devem ser reproduzidas na saída da rede [Barreto, 1998].

# Aplicações



- Segundo [Barreto, 1998], as redes não supervisionadas podem ter as seguintes aplicações:
  - **Agrupamento (*clustering*)**: agrupa as entradas em classes, de acordo com a similaridade (retorna apenas uma classe);
  - **Prototipação**: semelhante ao anterior, porém gera um protótipo (exemplo) da classe apropriada;
  - **Codificação**: gera uma versão codificada da entrada<sup>5</sup>, mantendo a informação mais relevante;
  - **Mapas auto-organizáveis**: organiza dimensionalmente dados complexos em grupos, de forma a manter a representação com propriedades relevantes a partir da entrada.

---

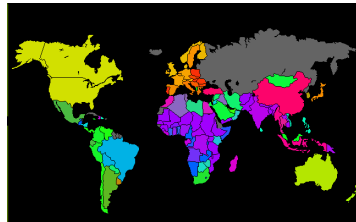
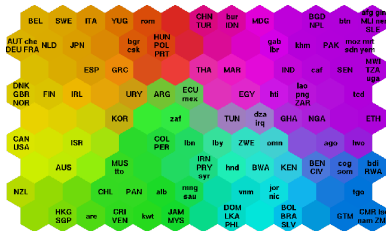
<sup>5</sup>Possivelmente com menos bits.

# Aprendizagem Competitiva

- Neurônios de cada camada competem pelo direito de responder a uma determinada entrada [Coppin, 2004] [Barreto, 1998];
- A resposta é dada pelo neurônio que tiver o maior valor de ativação [Coppin, 2004]
  - Modelo “vencedor-leva-tudo” (*the winner-take-all*, WTA);
  - A unidade vencedora tem sua saída fixada em 1, enquanto as demais são fixadas em 0 [Barreto, 1998];
- Problema comum: unidades mortas (*dead units*)
  - Unidades mal inicializadas podem nunca ganhar competições;
  - Para solução, podem ser criados mecanismos para limitar a vitória de neurônios [Barreto, 1998].

# Mapa Auto Organizáveis

- Os mapas auto-organizáveis (SOM) podem ser utilizados para inferir correlações e prover redução de dimensionalidade;
- Nas figuras abaixo, 39 indicadores que descrevem qualidade de vida<sup>6</sup> foram resumidos, utilizando mapas auto-organizáveis<sup>7</sup>.



Fonte: [Laboratory of Computer and Information Science (CIS), 1997]

<sup>6</sup>Indicadores incluem estado de saúde, nutrição, serviços educacionais, etc.

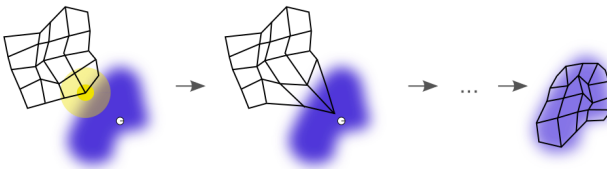
<sup>7</sup>Os dados utilizados para criação das imagens são oriundos de estatísticas do Banco Mundial, em 1992.

# Mapa de Kohonen

- **Mapas de Kohonen** correspondem a uma arquitetura de rede proposta por Teuvo Kohonen em 1982 [Coppin, 2004];
  - Também chamado de mapa de *features* (características) auto-organizado;
  - Utiliza o algoritmo “Vencedor-leva-tudo”;
  - Busca subdividir as entradas em *clusters*;
  - A rede é alimentada com vetores de exemplo e a distância Euclidiana entre eles é calculada e os pesos são computadas;
- A rede possui, basicamente, 2 camadas: [Coppin, 2004]
  - Camada de entrada;
  - Camada de *clusterização* (ou Kohonen), que serve à saída.

# Mapa de Kohonen - Treinamento

- O neurônio cujo vetor de pesos é mais parecido (semelhante) à entrada é chamado de melhor unidade de “melhor unidade correspondente” (*Best Matching Unit*, BMU);
- Os pesos do BMU e dos neurônios são ajustados com base no vetor de entrada.



Fonte: [Wikipedia contributors, 2020e]

[Link: versão animada do Mapa de Kohonen](#) [Wikipedia contributors, 2020b]

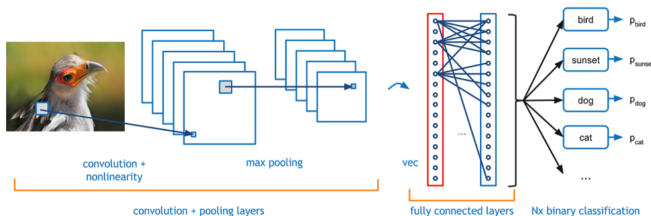
[Link: animação do treinamento Mapa de Kohonen](#) [Wikipedia contributors, 2020b]



# REDES CONVOLUCIONAIS

# Redes Convolucionais

- Redes convolucionais (CNNs) são compostas por camadas com convoluções, operações de *pooling* e funções de ativação ReLU (nas camadas intermediárias);
- Dependendo do objetivo, podem ser utilizadas camadas totalmente conectadas no final da rede [Li et al., 2021].



Fonte: [Florindo, 2018]

# Redes Convolucionais

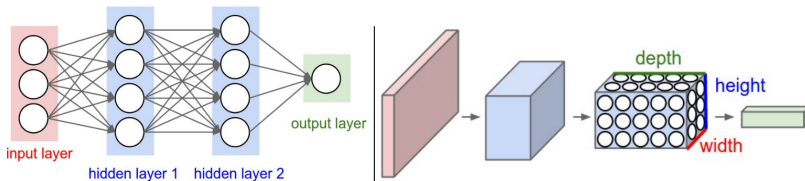
- Em geral, as redes convolucionais trabalham com imagens
  - Com isso, a arquitetura pode conter algumas restrições não existentes em redes tradicionais;
- Ao contrário de redes convencionais, redes convolucionais possuem neurônios organizados em 3 dimensões:
  - As dimensões são: largura, altura e profundidade<sup>8</sup>;
  - Essa característica é especialmente útil no processamento de imagens;
  - Imagens possuem altura, largura e, em geral, 3 canais de cores [Li et al., 2021].

---

<sup>8</sup>Tradução do inglês: *width*, *height* e *depth*.

# Redes Convolucionais

- Ao contrário das redes totalmente conectadas:
  - Os neurônios em uma camada são conectados apenas a uma pequena região da camada anterior;
  - A rede transforma a entrada 3D em um “volume” 3D de ativações de neurônios;
  - No final de algumas redes existem camadas de achatamento (*flatten*) e/ou camadas totalmente conectadas, para produzir previsões [Li et al., 2021].



Fonte: [Li et al., 2021]

# CONVOLUÇÃO

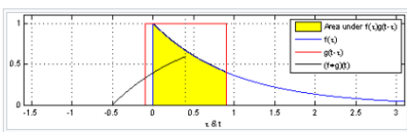
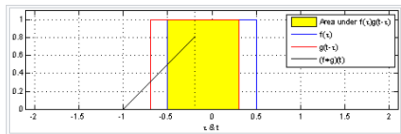
# Operação de Convolução

- A convolução é a operação mais importante em uma rede convolucional;
- Consiste em um operador linear para duas funções  $f$  e  $g$ ;
- Resulta em um terceiro valor  $s$ , que mede a soma do produto ao longo da região subentendida pela superposição de  $g$ , quando deslocada sobre uma função  $f$  [Weisstein, 2018].

$$(f * g)(t) = s(t) = \int_{-\infty}^{\infty} f(u) \cdot g(t - u) du$$

# Operação de Convolução

- A convolução pode ser entendida como a forma com que um sistema opera sobre um sinal de entrada [Smith, 1997].



Fonte: Adaptado de [Wikipedia contributors, 2020a]

[Link: animação de uma convolução 1](#) [Wikipedia contributors, 2020a]

[Link: animação de uma convolução 2](#) [Wikipedia contributors, 2020a]

# Operação de Convolução

- Na terminologia de redes convolucionais, a função  $f$  corresponde a **entrada**,  $g$  representa o **kernel** e a saída  $s$  corresponde ao mapa de saída (**feature map**);
- Segundo [Goodfellow et al., 2016], as convoluções são utilizadas para aprimorar resultados nos seguintes campos:
  - Interações esparsas: possibilitam o armazenamento de características importantes com baixo número de parâmetros;
  - Compartilhamento de parâmetros: possibilita armazenamento apenas de alguns valores, que geram os demais pesos da rede;
  - Representações equivariantes: alterações realizadas na entrada produzem efeitos correspondentes nas saídas.



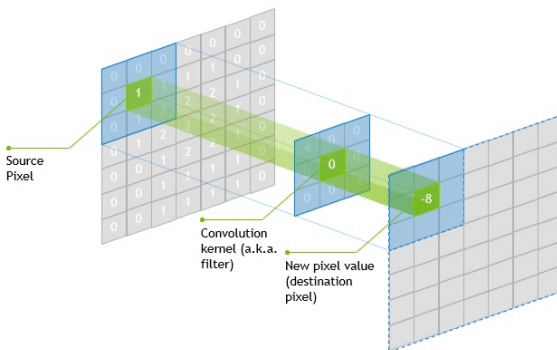
# Camada Convolucional



- Principal camada das redes convolucionais, responsável pela maior parte dos cálculos [Li et al., 2021];
- Convoluções utilizam múltiplos filtros (*kernels*), que podem ter diferentes tamanhos (ex.:  $5 \times 5 \times 3$ ,  $32 \times 32 \times 3$ , etc);
- Durante o *forward pass*, é feita a convolução de cada filtro na largura e altura do volume de entrada;
- Em seguida, são calculados os produtos escalares entre as entradas do filtro e a entrada em qualquer posição [Li et al., 2021].

# Camada Convolutiva

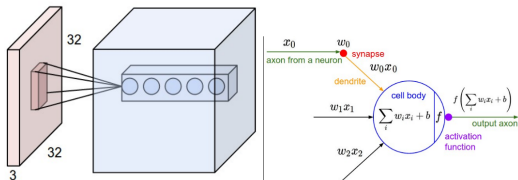
- Os filtros são “deslizados” sobre largura e altura do volume;



Fonte: [Martin, 2018]

# Camada Convolutional

- Produz-se um mapa de ativação bidimensional que fornece as respostas desse filtro em cada posição espacial;
- A rede aprende quais filtros ativar a partir das entradas;
- Os filtros ativados são empilhados na camada de profundidade para produção do volume de saída [Li et al., 2021].

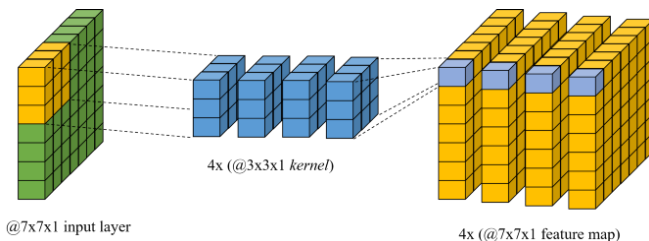


Fonte: [Li et al., 2021]

[Link: Animação com o funcionamento de uma convolução](#) [Li et al., 2021].

# Camada Convolucional

- A aplicação de múltiplos filtros (*kernels*) convolucionais dá origem a uma profundidade diferente na camada seguinte [Li et al., 2021].



Fonte: [Yunus, 2020]

[Link: Animação com o funcionamento de uma convolução](#) [Li et al., 2021].

# Profundidade, *Stride* e *Zero-padding*

- O tamanho do volume, é definido por 3 parâmetros:
  - *Profundidade (depth)*: hiperparâmetro correspondente ao número de *kernels* que serão utilizados para criação de mapas de características
    - Cada filtro pode aprender características diferentes, como cores, texturas e variações de tonalidade;
  - *Stride*: taxa de “deslizamento” dos *kernels*, em *pixels*;
  - *Zero-padding*<sup>9</sup>: corresponde ao à quantidade de zeros adicionados às bordas da imagem para melhor ajuste de tamanho (*pad*) [Li et al., 2021] [IBM, 2020].

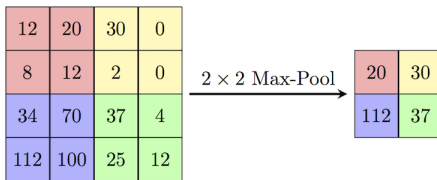
---

<sup>9</sup>Frequentemente denominado apenas como *padding*, pode ser traduzido como preenchimento de zeros.

# POOLING

# Pooling

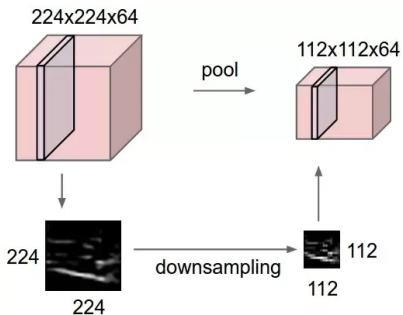
- A função de *pooling* é utilizada para prover informação estatística a respeito das saídas próximas;
- Tem como objetivo maximizar as vantagens da saída e tornar-se invariante a pequenos ruídos;
- Podem também ser utilizadas para redução da quantidade de neurônios entre camadas da rede, agrupando resultados na camada seguinte [Goodfellow et al., 2016].



Fonte: [Computer Science Wiki, 2020]

# Pooling

- A operação de *pooling* é executada em cada camada do volume, promovendo redução de dimensionalidade;



Fonte: [Computer Science Wiki, 2020]



# Pooling



- Os tipos de pooling mais comuns são:
  - **Max-pooling:** utiliza operação de máximo no filtro de *pooling*
    - Tipo mais utilizado de *pooling*, presente em redes como a VGGNet e AlexNet;
    - Capaz de selecionar os *pixels* mais ativados localmente, descartando *pixels* com valores baixos;
    - Capaz de representar uma região, por meio das informações mais importantes;
  - **Avg-pooling:** utiliza a operação de média no filtro de *pooling*
    - Capaz de representar uma região, por meio de informações médias;
    - Menos comum que *max-pooling*, mas presente na rede ResNet.

## FUNÇÃO DE ATIVAÇÃO RELU

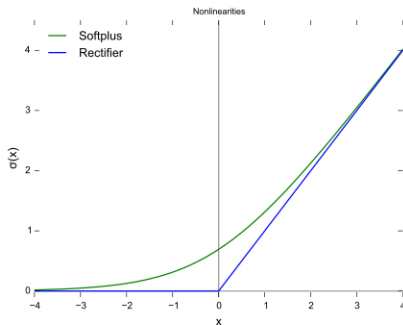
# ReLU

- ReLU: *Rectified linear unit* (Unidade linear retificada);
- Função de ativação empregada em redes neurais convolucionais, principalmente em camadas intermediárias;
- Vantagens:
  - Melhor propagação de gradientes: diminuição da quantidade de gradientes que tendem a zero devido ao acúmulo de camadas e pequenos valores, quando comparado a função sigmoide;
  - Invariante a escala:  $\max(0, ax) = a \max(0, x)$  para  $a \geq 0$ ;
  - Ativação esparsa: em redes inicializadas aleatoriamente, somente 50% das camadas intermediárias são ativadas;

# ReLU

- A função ReLU é definida formalmente como:

$$f(x) = \max(0, x)$$



Fonte: [Wikipedia contributors, 2020d]

# Referências I



Barreto, G. d. A. (1998).

Redes neurais não-supervisionadas para processamento de sequências temporais.

[Online]; acessado em 07 de Setembro de 2020. Disponível em: [https://teses.usp.br/teses/disponiveis/18/18133/tde-25112015-111953/publico/Dissert\\_Barreto\\_GuilhermeA.pdf](https://teses.usp.br/teses/disponiveis/18/18133/tde-25112015-111953/publico/Dissert_Barreto_GuilhermeA.pdf).



Computer Science Wiki (2020).

Max-pooling / pooling.

[Online]; acessado em 08 de Setembro de 2020. Disponível em: [https://computersciencewiki.org/index.php/Max-pooling/\\_Pooling](https://computersciencewiki.org/index.php/Max-pooling/_Pooling).



Coppin, B. (2004).

Artificial intelligence illuminated.

Jones and Bartlett illuminated series. Jones and Bartlett Publishers, 1 edition.



da Silva, D. M. (2014).

Inteligência Artificial - Slides de Aula.

IFMG - Instituto Federal de Minas Gerais, Campus Formiga.



Florindo, J. a. B. (2018).

Redes neurais convolucionais.

[Online]; acessado em 08 de Setembro de 2020. Disponível em: <https://www.ime.unicamp.br/~jbflorindo/Teaching/2018/MT530/T10.pdf>.



Goodfellow, I., Bengio, Y., and Courville, A. (2016).

Deep Learning.

MIT Press.

<http://www.deeplearningbook.org>.

# Referências II



IBM (2020).

Convolutional neural networks.

[Online]; acessado em 26 de Janeiro de 2021. Disponível em:

<https://www.ibm.com/cloud/learn/convolutional-neural-networks>.



Laboratory of Computer and Information Science (CIS) (1997).

World poverty map.

[Online]; acessado em 01 de Junho de 2021. Disponível em:

<http://www.cis.hut.fi/research/som-research/worldmap.html>.



Li, F.-F., Krishna, R., and Xu, D. (2021).

Convolutional neural networks (cnns / convnets).

[Online]; acessado em 26 de Janeiro de 2021. Disponível em:

<https://cs231n.github.io/convolutional-networks/>.



Martin, S. (2018).

What's the difference between a cnn and an rnn?

[Online]; acessado em 26 de Janeiro de 2021. Disponível em:

<https://blogs.nvidia.com/blog/2018/09/05/whats-the-difference-between-a-cnn-and-an-rnn/>.



Russel, S. and Norvig, P. (2013).

Inteligência artificial.

Campus - Elsevier, 3 edition.

# Referências III



Smith, S. W. (1997).

The Scientist and Engineer's Guide to Digital Signal Processing.  
California Technical Publishing, San Diego, CA, USA.  
<http://www.dspguide.com>.



Universo Programado (2019).

Ai destroying the google's dinosaur game!

[Online]; acessado em 01 de Setembro de 2020. Disponível em:  
<https://www.youtube.com/watch?v=NZlIYr1slAk>.



Weisstein, E. W. (2018).

Convolution.  
<http://mathworld.wolfram.com/Convolution.html>.



Wikipedia contributors (2020a).

Convolution.  
[Online]; acessado em 08 de Setembro de 2020. Disponível em:  
<https://en.wikipedia.org/wiki/Convolution>.



Wikipedia contributors (2020b).

Mapas de kohonen.

[Online]; acessado em 07 de Setembro de 2020. Disponível em:  
[https://pt.wikipedia.org/wiki/Mapas\\_de\\_Kohonen](https://pt.wikipedia.org/wiki/Mapas_de_Kohonen).

# Referências IV



Wikipedia contributors (2020c).

Modelo de hopfield.

[Online]; acessado em 08 de Setembro de 2020. Disponível em:  
[https://pt.wikipedia.org/wiki/Modelo\\_de\\_Hopfield](https://pt.wikipedia.org/wiki/Modelo_de_Hopfield).



Wikipedia contributors (2020d).

Rectifier (neural networks).

[Online]; acessado em 08 de Setembro de 2020. Disponível em:  
[https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)).



Wikipedia contributors (2020e).

Self-organizing map.

[Online]; acessado em 07 de Setembro de 2020. Disponível em:  
[https://en.wikipedia.org/wiki/Self-organizing\\_map](https://en.wikipedia.org/wiki/Self-organizing_map).



Yunus, M. (2020).

11 artificial neural network (ann) â part 6 konsep dasar convolutional neural network (cnn).

[Online]; acessado em 26 de Janeiro de 2021. Disponível em: <https://yunusmuhammad007.medium.com/11-artificial-neural-network-ann-part-6-konsep-dasar-convolutional-neural-network-cnn-3cc10fd9cf68>