

# Algoritmos de busca - Inteligência Artificial

Andrei Massaini  
andrei.massaini@hotmail.com  
ICEI - PUC-MG  
Belo Horizonte, Minas Gerais, Brasil

## Abstract

Neste relatório, são apresentados os resultados de testes realizados utilizando uma rede neural convolucional (CNN) em duas bases de dados distintas. O objetivo desses experimentos foi avaliar o desempenho da CNN em diferentes cenários e analisar o impacto de alterações na estrutura da rede.

**Keywords:** Artificial Intelligence, CNN, Redes Neurais

## ACM Reference Format:

Andrei Massaini. 2023. Algoritmos de busca - Inteligência Artificial. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/none>

## 1 Introdução

Nos últimos anos, o avanço da tecnologia e o aumento massivo de dados visuais impulsionaram a utilização das Redes Neurais Convolucionais (CNNs) como uma das abordagens mais eficazes para a classificação de imagens. As CNNs são um tipo especializado de rede neural, projetadas para lidar especificamente com dados de imagens e explorar suas características espaciais.

Diferentemente das redes neurais tradicionais, as CNNs são capazes de aprender automaticamente características relevantes das imagens em diferentes níveis de abstração. Essa capacidade é alcançada por meio das camadas de convolução, que aplicam filtros em pequenas regiões das imagens para detectar padrões visuais específicos. Com o uso de camadas de pooling, que reduzem a dimensionalidade dos dados, as CNNs podem aprender representações hierárquicas, capturando características mais complexas à medida que avançam nas camadas.

A popularidade das CNNs na classificação de imagens pode ser atribuída a sua notável capacidade de generalização, permitindo que sejam treinadas em grandes conjuntos de dados e aplicadas a novas imagens com bons resultados. Essa característica é especialmente valiosa em problemas de classificação, onde é necessário distinguir e categorizar imagens de diferentes classes.

As aplicações das CNNs em classificação de imagens são amplas e abrangem diversas áreas, como reconhecimento de objetos, diagnóstico médico, veículos autônomos, segurança, entre outras. Com seu poderoso poder de aprendizado,

as CNNs têm superado os métodos tradicionais de processamento de imagem, possibilitando o desenvolvimento de sistemas mais eficientes e precisos.

## 2 Bases de dados testadas:

### 2.1 Cachorro e gato

**2.1.1 Definição dos dados:** Os dados foram divididos em 3 conjuntos distintos. Sendo eles:

- **Conjunto de treino:** contém 12500 imagens de cães e 12500 de gatos, que serão utilizadas para treinar o modelo.
- **Conjunto de validação:** contém 12500 imagens de cães e gatos, que será utilizado para validar o desempenho do modelo durante o seu treinamento.
- **Conjunto de validação:** contém 1000 imagens de cães e gatos, que será utilizado para validar o desempenho do modelo após o seu treinamento.

### 2.2 Implementação

A rede neural fornecida foi construída da seguinte forma:

- **Primeira Camada de Convolução:**
  - Número de filtros: 32
  - Tamanho do filtro: 3x3
  - Função de ativação: ReLU
- **Segunda Camada de Convolução:**
  - Número de filtros: 32
  - Tamanho do filtro: 3x3
  - Função de ativação: ReLU
- **Camada de Pooling:**
  - Tamanho do pool: 2x2
- **Camada Totalmente Conectada:**
  - Número de neurônios: 128
  - Função de ativação: ReLU
- **Camada de Saída:**
  - Número de neurônios: 1
  - Função de ativação: Sigmoid

Ao realizar o treinamento do modelo, especificando 5 épocas como parâmetro, obtivemos os seguintes resultados no que tange a acurácia:

```

Epoch 1/5
782/782 [=====] - 156s 199ms/step - loss: 0.6161 - accuracy: 0.6613
Epoch 2/5
782/782 [=====] - 181s 129ms/step - loss: 0.5106 - accuracy: 0.7496
Epoch 3/5
782/782 [=====] - 95s 121ms/step - loss: 0.4761 - accuracy: 0.7781
Epoch 4/5
782/782 [=====] - 99s 126ms/step - loss: 0.4511 - accuracy: 0.7872
Epoch 5/5
782/782 [=====] - 118s 140ms/step - loss: 0.4329 - accuracy: 0.7954

```

Figure 1. Acúrcia durante o treinamento

### 2.2.1 Alterando a CNN.

Agora, para fins de experimentação Foram realizadas algumas alterações na estrutura da rede neural convolucional (CNN):

- Aumento do número de filtros nas camadas de convolução para 64:
  - Essa modificação permite que a rede neural extraia características mais complexas e abstratas das imagens, capturando detalhes e padrões relevantes para a classificação.
- Adição de uma terceira camada de convolução:
  - A inclusão de uma terceira camada de convolução aumenta a capacidade de aprendizado da rede, permitindo mais transformações e detecções de padrões nas imagens.
- Aumento do número de unidades na camada densa para 256:
  - Aumentar o número de unidades na camada densa fornece mais parâmetros para aprendizado, ajudando a capturar relações complexas entre as características extraídas pelas camadas de convolução.
- Utilização da função de ativação ReLU:
  - A função ReLU introduz não-linearidades na rede neural, auxiliando na propagação dos gradientes durante o treinamento e melhorando sua capacidade de aprendizado.

A princípio, essas alterações foram feitas com o objetivo de aumentar a capacidade de aprendizado e melhorar o desempenho do modelo. Após essas alterações, obtivemos os seguintes resultados de acurácia:

```

Epoch 1/5
782/782 [=====] - 306s 390ms/step - loss: 0.6572 - accuracy: 0.6198
Epoch 2/5
782/782 [=====] - 285s 364ms/step - loss: 0.5909 - accuracy: 0.6858
Epoch 3/5
782/782 [=====] - 349s 447ms/step - loss: 0.5228 - accuracy: 0.7422
Epoch 4/5
782/782 [=====] - 333s 425ms/step - loss: 0.4684 - accuracy: 0.7880
Epoch 5/5
782/782 [=====] - 361s 461ms/step - loss: 0.4365 - accuracy: 0.7946

```

Figure 2. Acuracia apos alterações no modelo

### 2.3 Conclusões sobre a alteração

Após as alterações, pode-se concluir o modelo não se beneficiou das alterações realizadas, tendo-se em vista que a acurácia diminuiu na ultima época.

## 2.4 The Simpsons (Bart and Homer)

**2.4.1 Definição da base:** A base de dados descrita acima, contempla duas pastas, de treino e teste que possuem imagens dos personagens da série animada *The Simpsons*.

Como esta base de dados possui um número muito pequeno de instâncias (imagens), pude realizar o treinamento da rede neural estipulando 100 épocas, como parâmetro. Os resultados obtidos foram:

```

epoch 94/100
7/7 [=====] - 3s 378ms/step - loss: 0.8466 - accuracy: 0.9897 - val_loss: 0.6512 - val_accuracy: 0.8714
epoch 95/100
7/7 [=====] - 3s 457ms/step - loss: 0.8845 - accuracy: 0.9744 - val_loss: 0.6879 - val_accuracy: 0.8880
epoch 96/100
7/7 [=====] - 3s 431ms/step - loss: 0.8218 - accuracy: 0.9846 - val_loss: 0.7775 - val_accuracy: 0.8429
epoch 97/100
7/7 [=====] - 3s 354ms/step - loss: 0.8323 - accuracy: 0.9949 - val_loss: 0.4407 - val_accuracy: 0.9143
epoch 98/100
7/7 [=====] - 3s 352ms/step - loss: 0.8408 - accuracy: 0.9897 - val_loss: 0.3228 - val_accuracy: 0.9429
epoch 99/100
7/7 [=====] - 3s 402ms/step - loss: 0.8131 - accuracy: 1.0000 - val_loss: 0.4430 - val_accuracy: 0.9143
epoch 100/100
7/7 [=====] - 3s 354ms/step - loss: 0.8867 - accuracy: 1.0000 - val_loss: 0.4627 - val_accuracy: 0.9143
1/1 [=====] - 0s 75ms/step

```

Figure 3. Acúrcia após 100 épocas

## 2.5 Conclusões

Devido ao conjunto de dados ser pequeno, podemos observar que após várias épocas, o modelo foi capaz de prever com 100 por cento de acurácia, a classe correta das imagens. Dito isso, seria proveitoso aumentar o conjunto de dados afim de melhorar ainda mais sua generalização, e de ter mais parâmetros quando comparado á um conjunto maior de dados.

## 3 Código e executável

O código fonte contendo as implementações pode ser encontrado no repositório a seguir: [https://github.com/Andreixzc/neural\\_network\\_img\\_classifierr](https://github.com/Andreixzc/neural_network_img_classifierr)

## References