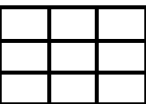
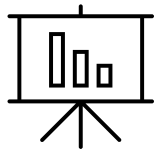
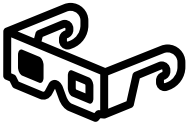
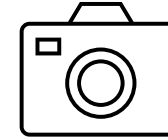
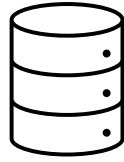


**Báo cáo
nhận diện
động vật
bằng CNN**

Báo cáo nhận diện động vật bằng CNN



Thành Viên Nhóm



01

• Nguyễn Công Hiệp-B20DCCN239

02

• Lương Xuân Khôi-B20DCCN383

03

• Nguyễn Quốc Vương-B20DCCN751

04

• Nguyễn Phú Tiến -B20DCCN596

05

• Nguyễn Công Uẩn – B20DCC715

06

• Nguyễn Đồng Hoàng-B20DCCN276

Nội dung chính



01

- Giới thiệu về đề tài

02

- Lý do chọn đề tài và ý nghĩa

03

- Tổng quan về CNN

04

- Khái niệm cơ bản

05

- Thu thập và xử lý dữ liệu

Nội dung chính



06

- Kiến trúc mạng CNN

07

- Huấn luyện mô hình

08

- Đánh giá kết quả

09

- Ứng dụng và tiềm năng phát triển



01

Giới thiệu Dự án Nhận diện Động vật bằng CNN



1

Giới thiệu Dự án Nhận diện Động vật bằng CNN

Tổng quan

Trong thời đại công nghệ 4.0, trí tuệ nhân tạo (AI) đã trở thành một phần không thể thiếu trong nhiều lĩnh vực, từ y tế, giao thông, đến giáo dục và bảo vệ môi trường. Một trong những ứng dụng nổi bật của AI là nhận diện hình ảnh, đặc biệt là nhận diện động vật. Dự án "Nhận Diện Động Vật bằng Mạng Nơ-ron Tích Chập (CNN)" nhằm xây dựng một hệ thống có khả năng tự động nhận diện và phân loại các loài động vật từ hình ảnh, hỗ trợ cho nhiều mục đích nghiên cứu khoa học và bảo tồn thiên nhiên.

Mục tiêu

Phát triển hệ thống nhận diện động vật chính xác cao.
Nhận diện nhiều loài động vật khác nhau (hoang dã, nuôi, lưỡng cư,...).
Ứng dụng trong bảo tồn động vật, nông nghiệp, an ninh, giáo dục,...



2

• Lý do chọn đề tài và ý nghĩa

• Lý do chọn đề tài:



Tính cấp thiết:

Nhận diện động vật đóng vai trò quan trọng trong nhiều lĩnh vực như bảo tồn động vật hoang dã, nông nghiệp, an ninh, giáo dục. Tuy nhiên, các phương pháp truyền thống thường tốn thời gian, công sức và độ chính xác chưa cao. Do đó, việc ứng dụng trí tuệ nhân tạo (AI), đặc biệt là mạng nơ-ron tích chập (CNN) vào nhận diện động vật là hướng đi tiềm năng để giải quyết bài toán này.



Tính khả thi

CNN đã chứng minh hiệu quả vượt trội trong xử lý hình ảnh, nhận diện vật thể. Việc áp dụng CNN vào nhận diện động vật là hoàn toàn khả thi với nguồn dữ liệu ngày càng phong phú và công nghệ tính toán phát triển mạnh mẽ.



Tính sáng tạo

Dự án có thể đề xuất các giải pháp mới, sáng tạo cho bài toán nhận diện động vật, góp phần nâng cao hiệu quả và mở rộng phạm vi ứng dụng.



Sở thích cá nhân

Đề tài phù hợp với sở thích, định hướng nghiên cứu của bản thân, tạo động lực để nghiên cứu và học hỏi sâu hơn.



02

• Ý nghĩa thực tiễn của đề tài

Bảo tồn động vật hoang dã

Hệ thống nhận diện động vật tự động giúp theo dõi, giám sát số lượng, di cư của các loài động vật, hỗ trợ công tác bảo tồn hiệu quả hơn.

01

An ninh

Phát hiện động vật xâm nhập khu vực cấm, bảo vệ an ninh cho con người và tài sản

02

Giáo dục

Giúp học sinh, sinh viên học tập về các loài động vật một cách sinh động, trực quan.

03

Nghiên cứu khoa học

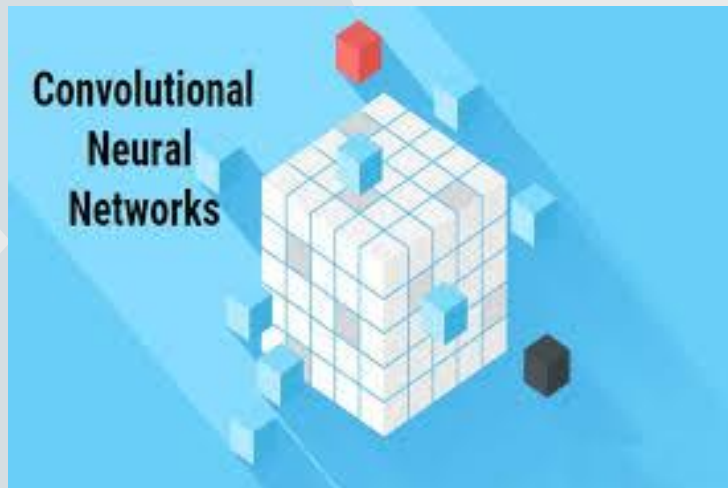
Cung cấp dữ liệu, thông tin cần thiết cho nghiên cứu khoa học về hành vi sinh thái của động vật

04



03

Giới thiệu tổng quan về CNN



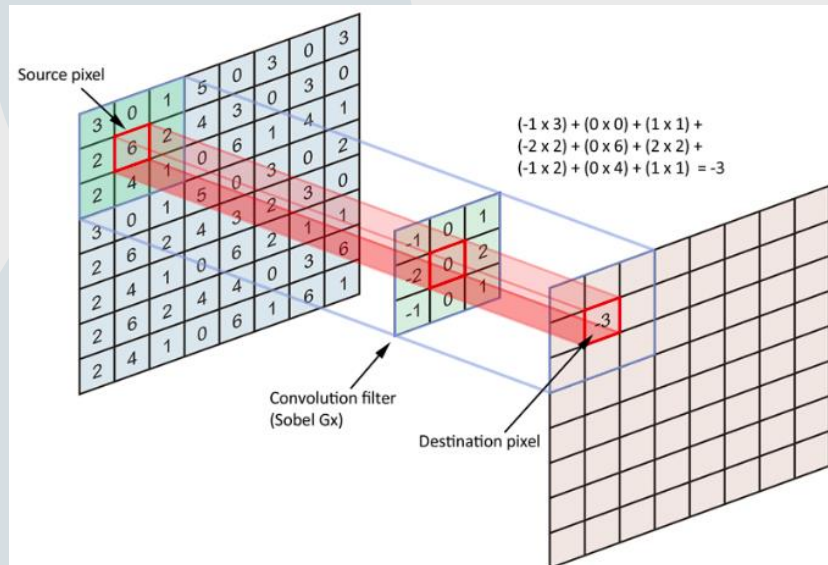
Convolutional Neural Network (CNN), hay còn gọi là ConvNet, là một loại mạng nơ-ron tích tụ được sử dụng phổ biến trong lĩnh vực xử lý hình ảnh.

Thuật toán CNN là một phần quan trọng của Deep Learning- tập hợp các thuật toán nhằm xây dựng mô hình dữ liệu trừu tượng thông qua việc sử dụng nhiều lớp xử lý cấu trúc phức tạp.



03

Cách hoạt động của thuật toán CNN



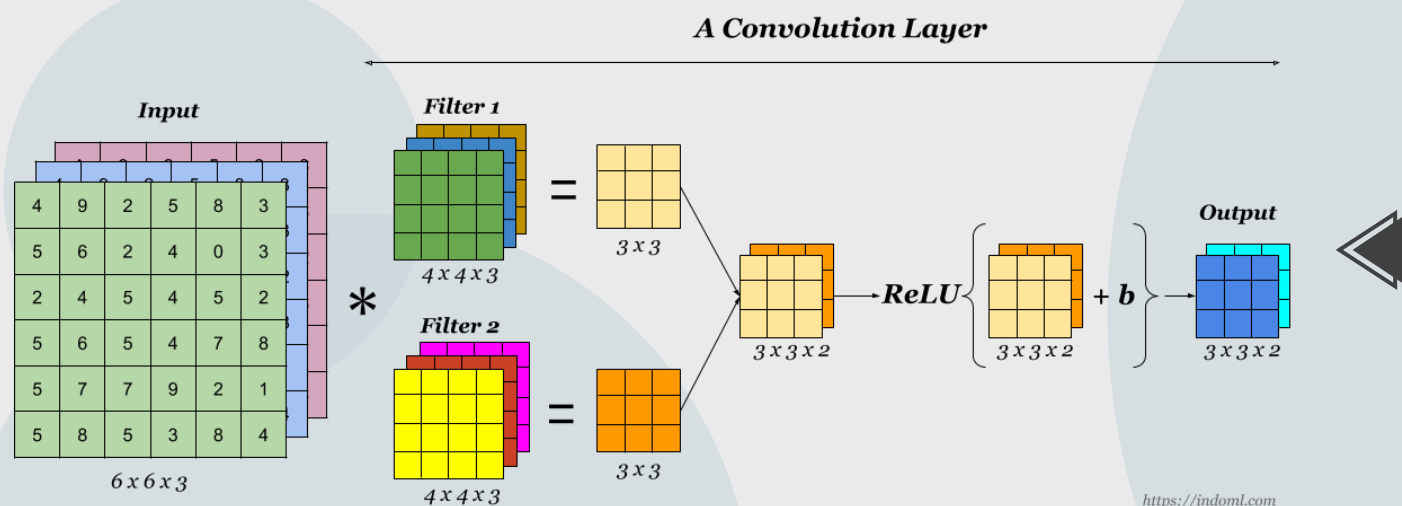
CNN hoạt động dựa trên dữ liệu ảnh và được sử dụng rộng rãi trong các ứng dụng như nhận dạng hình ảnh và khuôn mặt

Khác với các mạng nơ-ron thông thường, thuật toán CNN nhận đầu vào dưới dạng một mảng hai chiều và xử lý trực tiếp trên hình ảnh, giúp nó trích xuất các đặc trưng quan trọng từ dữ liệu hình ảnh một cách hiệu quả



04

• Khái niệm cơ bản về CNN



Convolutional layers

- *kernel là 1 ma trận sẽ quét ma trận dữ liệu gõ vào, sau đó đưa qua hàm kích hoạt(activation function) thường là hàm ReLu để thu được ma trận mới gọi là feature map*
- *Convolutional layers là một lớp ẩn chứa các feature map và mỗi feature map này sẽ là đặc trưng của ngõ vào*

<https://indoml.com>

Công thức tính kích thước ảnh bản đầu khi qua bộ lọc

$$O = \frac{I + 2p - k}{s} + 1$$

O là kích thước ảnh đầu vào

I là kích thước ảnh đầu ra

P là kích thước khoảng trắng bao gồm các bit 0 ngoài viền ảnh gốc(padding)

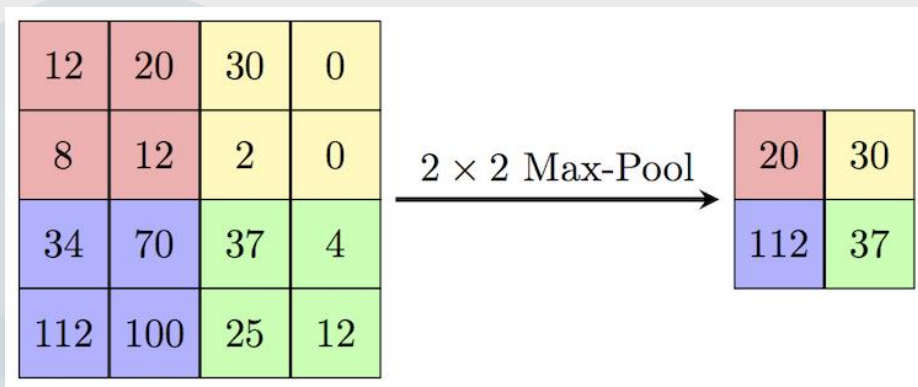
K là kích thước bộ lọc

S là bước trượt của bộ lọc(stride)



04

• Khái niệm cơ bản về CNN



Max pooling: Lấy ra điểm ảnh lớn nhất khi ma trận trượt qua



Average pooling: Lấy ra trung bình cộng tất cả điểm ảnh trượt qua

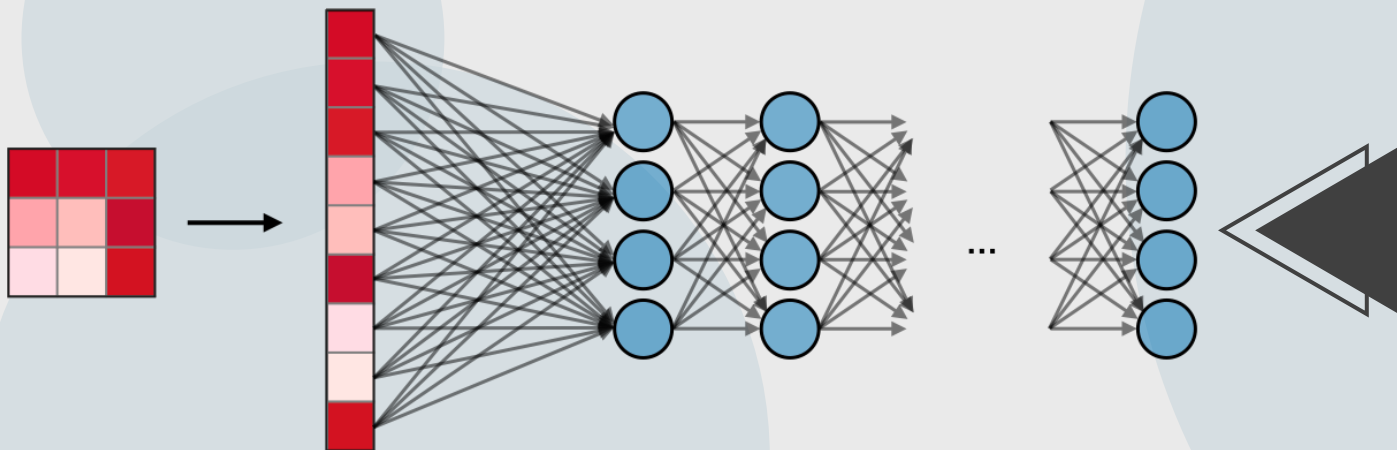
Pooling

- Là lớp được sử dụng giữa các tích chập để giảm kích thước của feature map nhưng vẫn giữ nguyên đặc trưng
- Có 2 loại pooling phổ biến: max pooling và average pooling



04

• Khái niệm cơ bản về CNN



Fully connected

- Sau convolutional layers, pooling layers sẽ đến lớp kết đủ(fully connected)
- Lớp này dùng để phân ra các lớp có số neural tùy thuộc vào số dữ liệu ngõ vào



05

• Thu thập và xử lý dữ liệu

- Tập dữ liệu là dữ liệu về hình ảnh của 8 loài động vật khác nhau. Mỗi loại động vật có từ 1300 - 2100 bức ảnh khác nhau.
- Dữ liệu tổng cộng có
 - ❑ bộ train gồm: chiếm khoảng 80%
 - ❑ bộ test gồm: chiếm khoảng 12 %
 - ❑ bộ validation gồm: chiếm khoảng 8%
- Tiền xử lý dữ liệu
 - ❑ Dữ liệu được xử lý gán nhãn theo mỗi loài tương ứng
 - ❑ sử dụng ImageDataGenerator để thực hiện các biến đổi hình ảnh trên hình ảnh gốc, giúp tăng cường dữ liệu huấn luyện và cải thiện khả năng tổng quát hóa của mô hình.
 - ❑ Chuẩn hoá các giá trị trong ảnh về giá trị trong khoảng từ 0 - 1, giúp làm giảm độ lớn của dữ liệu và giúp mô hình học tốt hơn.
 - ❑ kích thước ảnh được đưa về 200 x 200 trước khi được đưa vào mô hình



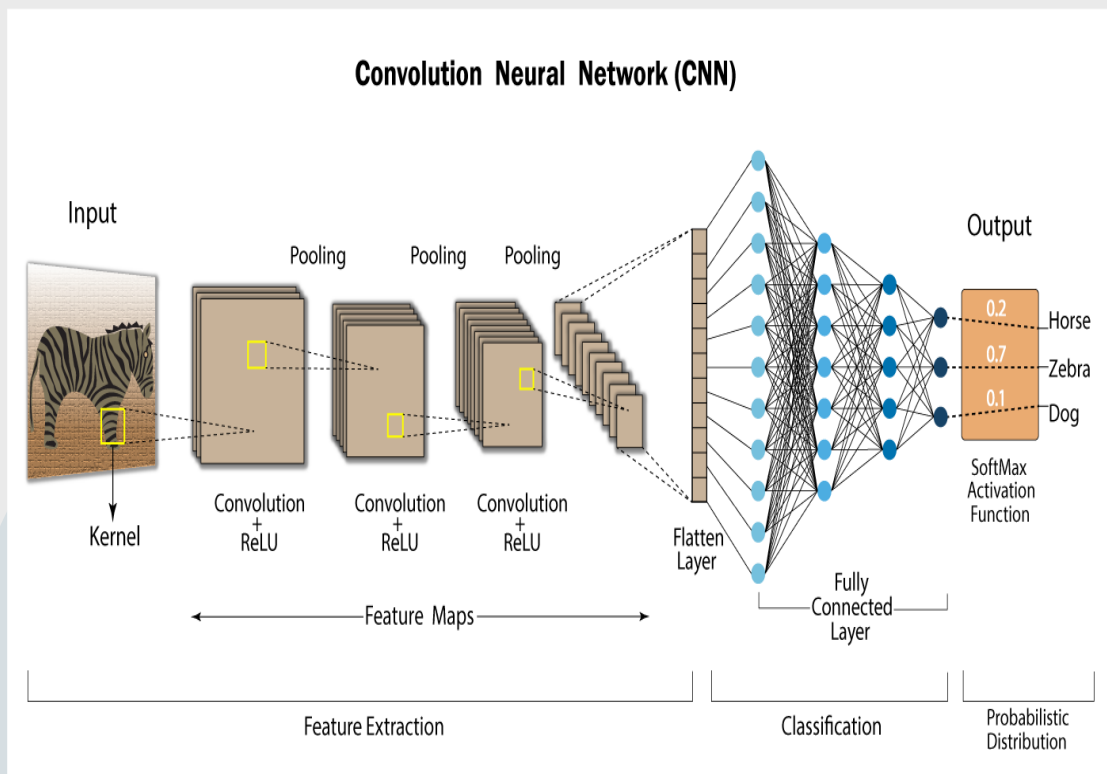


06

Kiến trúc mạng CNN

**06**

• Kiến trúc mạng CNN



Mạng CNN là một tập hợp các lớp tích chập chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó

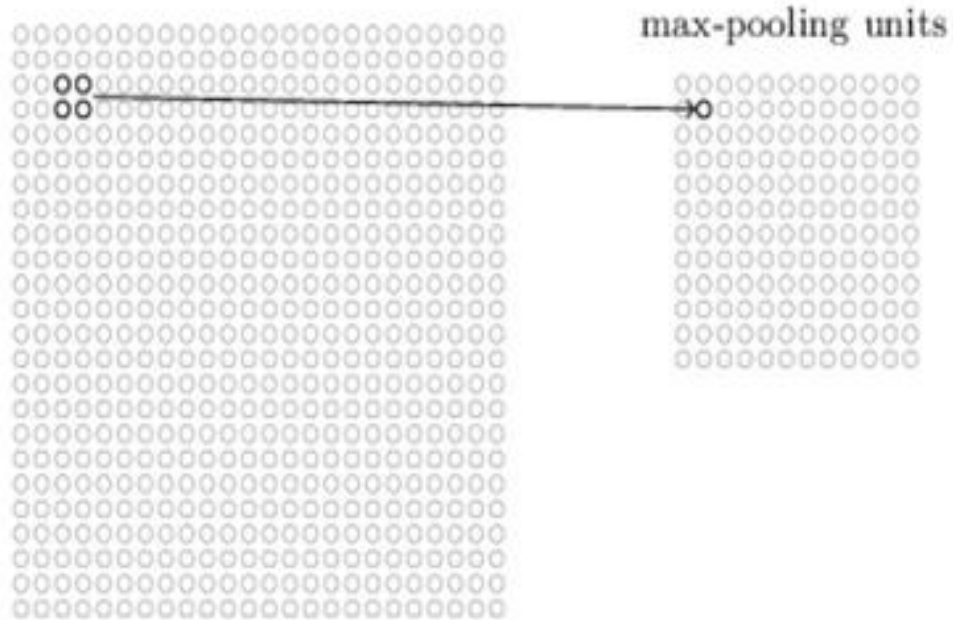
Mỗi một lớp được sử dụng các filter khác nhau



06

• Kiến trúc mạng CNN

hidden neurons (output from feature map)



Mạng CNN sử dụng 3 ý tưởng cơ bản:

- Các trường tiếp nhận cục bộ (local receptive field)
- Trọng số chia sẻ (shared weights)
- Tổng hợp (pooling).

Lớp tổng hợp (pooling layer): Lớp pooling thường được sử dụng ngay sau lớp convolutional để đơn giản hóa thông tin đầu ra để giảm bớt số lượng neuron. Thủ tục pooling phổ biến là max-pooling, thủ tục này chọn giá trị lớn nhất trong vùng đầu vào 2×2 qua lớp Max Pooling thì số lượng neuron giảm đi phân nửa.

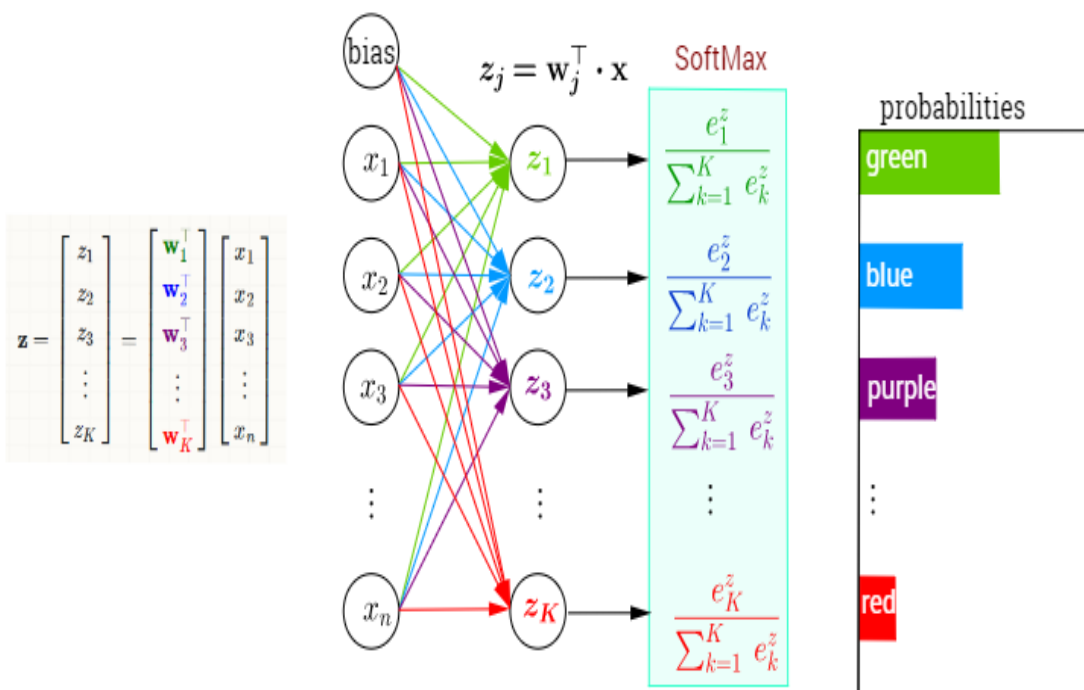
2 lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối (fully connected layer). Lớp này nối mọi neuron từ lớp max pooled tới mọi neuron của tầng ra.



06

• Kiến trúc mạng CNN

Multi-Class Classification with NN and SoftMax Function



Softmax activation function

tín hiệu từ các lớp của Neural Network có thể là các giá trị âm. Vậy ta cần một hàm số *mượt* luôn trả về giá trị dương để dễ dàng cho quá trình tính toán. Softmax activation function là 1 hàm một hàm đồng biến để có thể đánh giá chính xác giá trị của neural đó

$$[a_i = \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)}]$$

C là số lượng các lớp cần được phân loại. Hàm số này được gọi là hàm Softmax

Trong trường hợp các tham số quá lớn, có thể dẫn tới tràn số khi đưa qua hàm Mũ, ta có thể trừ chúng đi một lượng α nhất định.

$$[a_i = \frac{\exp(z_i - \alpha)}{\sum_{j=1}^C \exp(z_j - \alpha)}]$$



07

Huấn luyện mô hình CNN

**07**

• Huấn luyện mô hình CNN

```
model.compile(  
    optimizer=RMSprop(learning_rate=0.001),  
    #optimizer="adam",  
    loss='categorical_crossentropy',  
    metrics=['accuracy']  
)
```

Metrics:

Dùng để tính toán tần suất các dự đoán của mô hình khớp với các nhãn thực tế. Nó đưa ra ý tưởng về hiệu suất của mô hình trong quá trình đào tạo và đánh giá.

Thiết lập các tham số để chuẩn bị cho quá trình huấn luyện mô hình. Việc này giúp xác định cách mô hình sẽ được tối ưu hóa, hàm mất mát (loss function) nào sẽ được sử dụng và các tiêu chí đánh giá (metrics) nào sẽ được theo dõi trong quá trình huấn luyện và kiểm tra.

Optimizer:

RMSprop(learning_rate=0.001)

RMSprop (Root Mean Square Propagation): điều chỉnh tốc độ học tập cho từng tham số riêng lẻ learning_rate=0,001 đặt tốc độ học tập ban đầu cho trình tối ưu hóa.

Loss Function:

categorical_crossentropy

Được sử dụng cho các vấn đề phân loại nhiều lớp trong đó mỗi đầu vào thuộc về một trong một số danh mục. Nó so sánh phân phối xác suất dự đoán (từ lớp đầu ra softmax) với phân phối thực tế (one-hot encoded labels) và tính toán hàm loss.



08

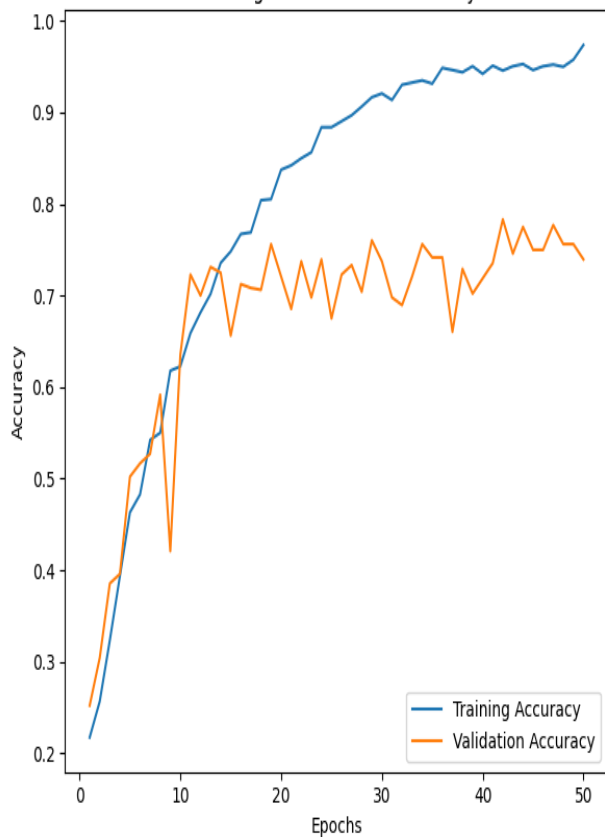
Kết quả và đánh giá



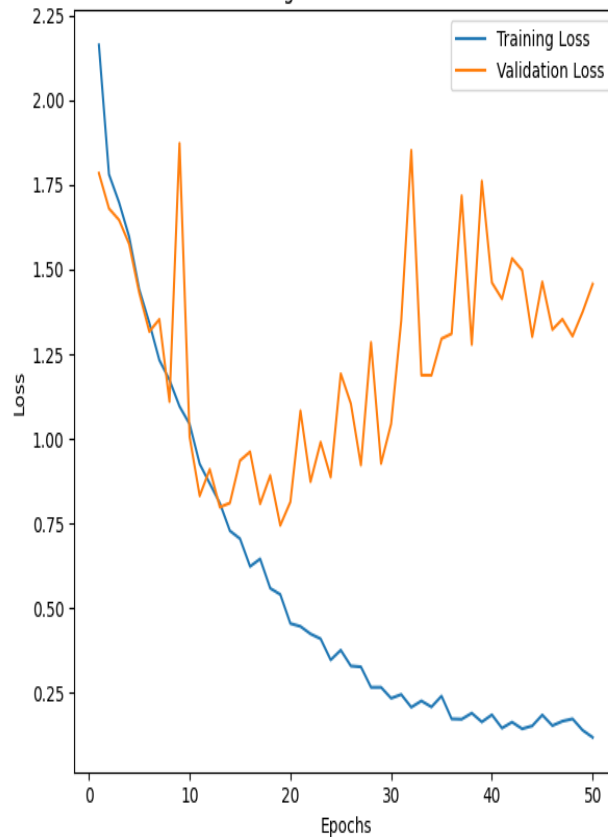
08

• Kết quả và đánh giá

Training and Validation Accuracy



Training and Validation Loss



Kết quả của độ chính xác chung

- Độ chính xác của tập huấn luyện (Training Accuracy) tăng đều đặn và đạt khoảng 0.9 vào cuối quá trình huấn luyện.
- Độ chính xác của tập xác thực (Validation Accuracy) dao động nhiều hơn và không tăng đều đặn, duy trì xung quanh 0.7 đến 0.8 sau khoảng epoch thứ 10.
=> Sự khác biệt giữa hai đường này cho thấy mô hình có thể đang bị overfitting, tức là nó hoạt động tốt trên tập huấn luyện nhưng không tốt trên tập xác thực.
- Mất mát của tập huấn luyện (Training Loss) giảm đều đặn và đạt dưới 0.25 vào cuối quá trình huấn luyện, cho thấy mô hình đang học tốt từ tập huấn luyện.
- Mất mát của tập xác thực (Validation Loss) dao động nhiều và không giảm đều đặn, thường dao động giữa 0.75 và 1.5 sau khoảng epoch thứ 10.
=> Sự dao động mạnh của Validation Loss và không giảm đều đặn cho thấy mô hình có thể đang gặp vấn đề về generalization, tức là nó không tổng quát hóa tốt từ tập huấn luyện sang tập xác thực.



08

• Kết quả và đánh giá

Kết quả





Hạn chế:

Overfitting: Có dấu hiệu rõ ràng của overfitting khi mô hình học rất tốt từ tập huấn luyện (độ chính xác cao và mất mát thấp) nhưng không đạt được kết quả tương tự trên tập xác thực (độ chính xác dao động và mất mát không giảm).

Nguyên nhân:

Mô hình phức tạp:

- Số lượng layers hoặc neurons quá nhiều.
- Mô hình có quá nhiều tham số so với số lượng dữ liệu huấn luyện.

Dữ liệu huấn luyện không đủ hoặc đa dạng:

- Số lượng mẫu trong tập huấn luyện quá ít.
- Dữ liệu không đủ đa dạng để mô hình học được các đặc trưng tổng quát.

Thiếu các kỹ thuật regularization:

- Không sử dụng các kỹ thuật giảm overfitting như dropout, L2 regularization.



08

• Kết quả và đánh giá

Phương pháp khắc phục:

Thêm Dropout Layers:

- Dropout là một kỹ thuật regularization phổ biến, giúp giảm overfitting bằng cách "tắt" ngẫu nhiên một tỷ lệ các neurons trong quá trình huấn luyện.

```
model.add(Dropout(0.5)) # Tắt ngẫu nhiên 50% các neurons
```

Sử dụng Regularization (L2 Regularization):

- Thêm regularization vào các layers để giảm overfitting.

```
from keras.regularizers import l2

model.add(Dense(256, activation='relu', kernel_regularizer=l2(0.01)))
```

Tăng cường dữ liệu (Data Augmentation):

- Tạo thêm dữ liệu huấn luyện bằng cách sử dụng các kỹ thuật tăng cường dữ liệu như xoay, dịch chuyển, zoom, vv.

```
train_data_gen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2
```



08

• Kết quả và đánh giá

Phương pháp khắc phục:

Sử dụng Early Stopping:

- Dropout là một kỹ thuật regularization phổ biến, giúp giảm overfitting bằng cách "tắt" ngẫu nhiên một tỷ lệ các neurons trong quá trình huấn luyện.

Giảm độ phức tạp của mô hình:

- Giảm số lượng layers hoặc số lượng neurons trong mỗi layer để tránh mô hình học quá nhiều chi tiết không cần thiết.

Ngoài ra còn các cách như thu thập thêm dữ liệu huấn luyện hoặc sử dụng các bộ dữ liệu lớn hơn nếu có thể hay kết hợp nhiều mô hình khác nhau để giảm sự lệch và nhiễu.

```
from keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss', patience=10,
                               history = model.fit(
                                   x=train_generator,
                                   validation_data=validation_generator,
                                   batch_size=32,
                                   epochs=50,
                                   callbacks=[early_stopping]
                               )
```

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(200, 200, 3)))
model.add(MaxPooling2D(2, 2))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(8, activation='softmax'))
```



09

- Ứng dụng và tiềm năng phát triển



09

• Ứng dụng và tiềm năng phát triển



Tiềm năng phát triển:

- Giám sát môi trường và bảo tồn động vật hoang dã: giám sát và theo dõi các loài động vật hoang dã trong tự nhiên thông qua các hệ thống camera. giúp các nhà nghiên cứu và quản lý môi trường thu thập dữ liệu về sự phân bố và hành vi của các loài

Ứng dụng trong nghiên cứu sinh học và hành vi động vật: phân tích hành vi và tương tác giữa các loài động vật trong tự nhiên

Phát hiện động vật gây hại: nhận diện các loại động vật gây hại trong các khu vực dân cư hoặc nông nghiệp



Hướng phát triển: Cải thiện độ chính xác:

- Tăng kích thước bộ dữ liệu, tăng chất lượng ảnh, số lượng loài động vật
- Sử dụng kỹ thuật augmentation như xoay, phóng to, thu nhỏ, lật ảnh, áp dụng các hiệu ứng màu sắc để tạo ra nhiều biến thể của ảnh động vật, từ đó tăng sự đa dạng và cải thiện khả năng tổng quát hóa của mô hình.
- Sử dụng dữ liệu về những lĩnh vực khác, thay đổi thông số huấn luyện mô hình như: loại bỏ các lớp, thay đổi kích thước bộ lọc, và thay đổi số lượng neural trong các lớp ẩn



Thanks for listening

Cảm ơn vì đã lắng nghe

