Mô tả cấu trúc mạng CNN phân loại ảnh chó và mèo.

Source code CNN

Link github: <https://github.com/coolioasjulio/Cat-Dog-CNN-Classifier>

Input vào là 1 tấm hình bất kỳ của chó hoặc mèo 🡪 Output là nhận diện chó hoặc mèo.

Tập dữ liệu dataset là 10 000 tấm hình . trong đó tập train là 8 000 tấm ( hình chó và mèo mỗi loại 4 000) , tập test là 2 000 tấm hình ( hình chó và mèo mỗi loại 1 000) . Với tỉ lệ train và test là 80 % : 20 %.

Nguồn ảnh:

-Train : <https://github.com/coolioasjulio/Cat-Dog-CNN-Classifier/tree/master/dataset/training_set>

-Test: <https://github.com/coolioasjulio/Cat-Dog-CNN-Classifier/tree/master/dataset/test_set>

Kích thước ảnh : bất kỳ .

Thư viện : Keras

Keras là một library được phát triển vào năm 2015 bởi François Chollet, là một kỹ sư nghiên cứu deep learning tại google. Nó là một open source cho neural network được viết bởi ngôn ngữ python. keras là một API bậc cao có thể sử dụng chung với các thư viện deep learning nổi tiếng như tensorflow.

Source code :

# Image Classification

# Import libraries

from keras.models import Sequential

from keras.layers import Conv2D

from keras.layers import Conv3D

from keras.layers import MaxPooling2D

from keras.layers import MaxPooling3D

from keras.layers import GlobalAveragePooling2D

from keras.layers import Dense

# Initalize CNN

classifier = Sequential()

# Add 2 convolution layers

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), input\_shape=(64, 64, 3), activation='relu'))

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

# Add pooling layer

classifier.add(MaxPooling2D(pool\_size=(2,2)))

# Add 2 more convolution layers

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

# Add max pooling layer

classifier.add(MaxPooling2D(pool\_size=(2,2)))

# Add 2 more convolution layers

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

# Add max pooling layer

classifier.add(MaxPooling2D(pool\_size=(2,2)))

# Add global average pooling layer

classifier.add(GlobalAveragePooling2D())

# Add full connection

classifier.add(Dense(units=2, activation='softmax'))

# Compiling the ANN

classifier.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Fit CNN to images

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(

rescale=1./255,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True)

test\_datagen = ImageDataGenerator(rescale=1./255)

train\_set = train\_datagen.flow\_from\_directory(

'dataset/training\_set',

target\_size=(64, 64),

batch\_size=32,

class\_mode='categorical')

test\_set = test\_datagen.flow\_from\_directory(

'dataset/test\_set',

target\_size=(64, 64),

batch\_size=32,

class\_mode='categorical')

classifier.fit\_generator(

train\_set,

steps\_per\_epoch=8000,

epochs=25,

validation\_data=test\_set,

validation\_steps=2000)

classifier.save('model.h5')

# Test accuracy of classifier

def test\_accuracy(classifier, test\_set, steps):

num\_correct = 0

num\_guesses = 0

for i in range(steps):

a = test\_set.next()

guesses = classifier.predict(a[0])

correct = a[1]

for index in range(len(guesses)):

num\_guesses += 1

if round(guesses[index][0]) == correct[index]:

num\_correct += 1

return num\_correct, num\_guesses

Code train model có 10 lớp ( không tính Input và Output)

* 2 lớp đầu:

# Add 2 convolution layers

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), input\_shape=(64, 64, 3), activation='relu'))

* Scan của input ban đầu là (64,64,3)
* Dựa vào Convolution Filter =32
* Ma trận kernel (3,3), pad=0, Stride=1
* Đưa qua activation funciton relu
* Kết quả sẽ là một là 1 ma trận (62,62,32).

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

* Kết quả sẽ là một là 1 ma trận (60,60,32).

# Add max pooling layer

classifier.add(MaxPooling2D(pool\_size=(2,2)))

* Kết quả sẽ là một là 1 ma trận (30,30,32) với pad=0, Stride=2

# Add 2 more convolution layers

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

* Kết quả sẽ là một là 1 ma trận (28,28,32), pad=0, Stride=1
* Kết quả sẽ là một là 1 ma trận (26,26,32), pad=0, Stride=1

# Add max pooling layer

classifier.add(MaxPooling2D(pool\_size=(2,2)))

* Kết quả sẽ là một là 1 ma trận (13,13,32), pad=0, Stride=2

# Add 2 more convolution layers

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

classifier.add(Conv2D(filters=32, kernel\_size=(3,3), activation='relu'))

* Kết quả sẽ là một là 1 ma trận (11,11,32), pad=0, Stride=1
* Kết quả sẽ là một là 1 ma trận (9,9,32), pad=0, Stride=1

# Add max pooling layer

classifier.add(MaxPooling2D(pool\_size=(2,2)))

* Kết quả sẽ là một là 1 ma trận (5,5,32), pad=0, Stride=2

# Add global average pooling layer

classifier.add(GlobalAveragePooling2D())

* Kết quả sẽ là một là 1 ma trận (3,3,32), pad=0, Stride=2