PSIT College of engineering Kanpur

Project Report (3rd Semester)

Image Recognition using convolutional neural network

In partial fulfilment of requirement of degree of **Bachelor of Technology (B.Tech)**

By

Ravi Kumar Maurya (1834810076)

Rasik Mishra (1834810075)

Ritik Kumar (1834810078)

Rishabh Kushwaha (1834810077)

Under the supervision of

Mr. Amit Kumar

Introduction

In deep learning, a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as **shift invariant** or **space invariant artificial neural networks** (**SIANN**), based on their shared-weights architecture and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing
CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

Objective

Objective of this project is to make an Model that Recognises the image of a an input image(cat and dog) .based on convolutional neural network.

Our project fully describes the concept of convolutional neural network:

Software and Hardware Used
Library : keras and numpy
Programming Language: python
IDE: spyder(from anaconda navigator)
Operating System: Windows 8 and above

Implementation

We are going to implement a Model that Recognises and image and predict the output based on convolutional neural network. First of we will convert the image into a feature map and the apply convolutional neural network first to train the and then to test the data based on that feature map.

This project has a vast arena of development, notably the Sixth Sense project which completely revolutionizes the digital world. The code can be used again to predict other images and can help the humanity in many ways like if we replace the images of brain and train our neural network to predict the tumour in brain. Then brain tumour can easily be detected. Likewise several other useful projects can be made by this convolutional neural network model

Python modules used description

1. Keras

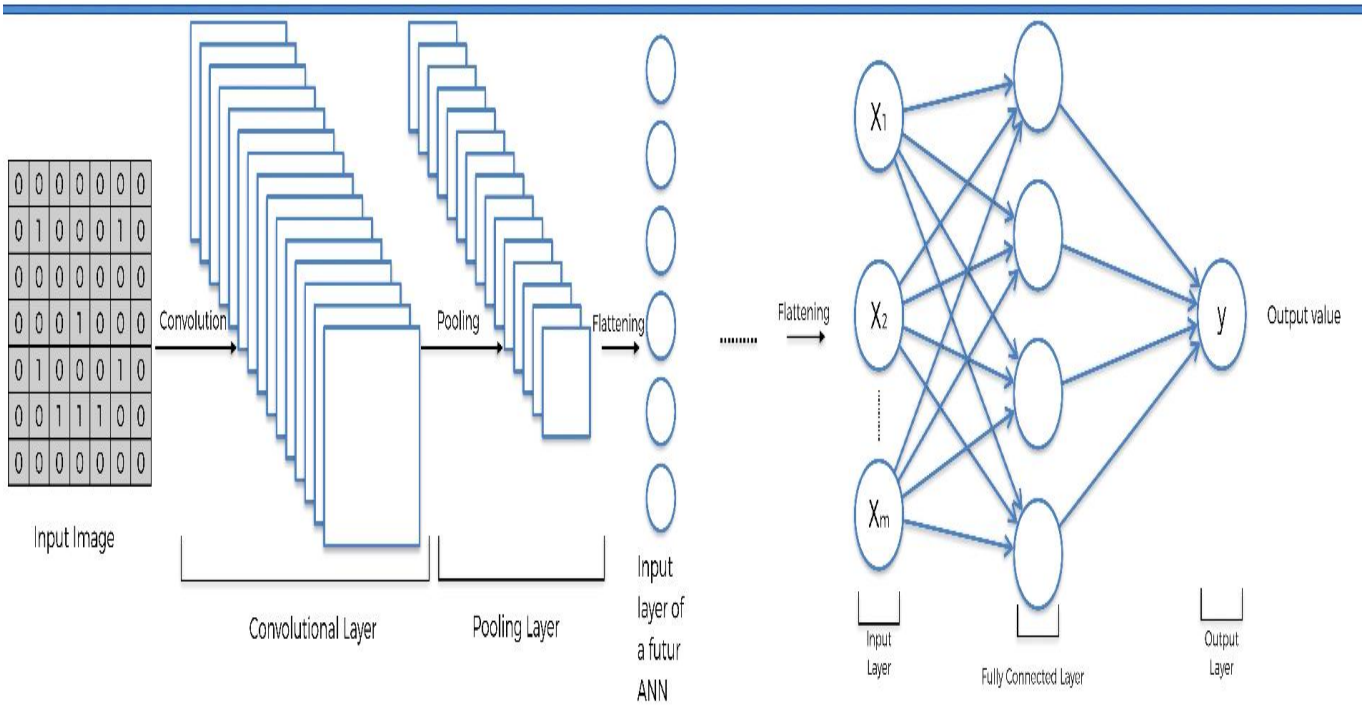
Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

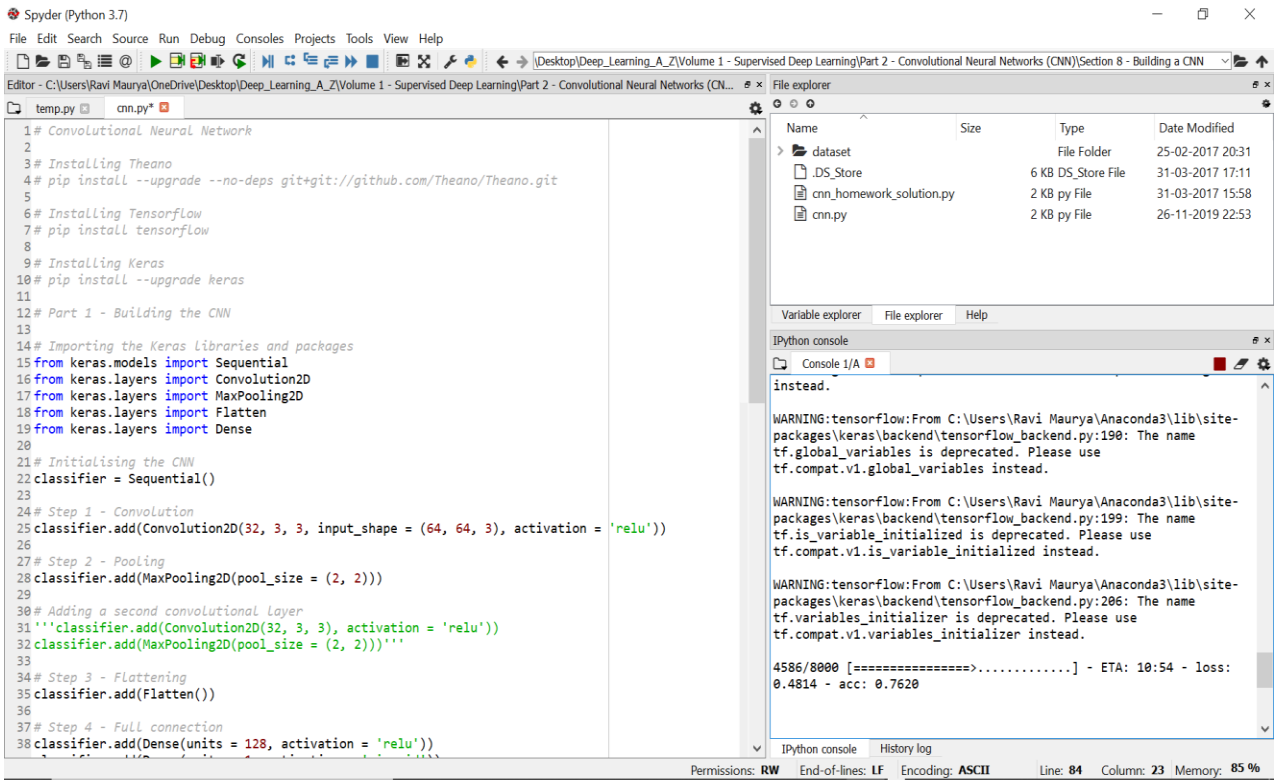
Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing Deep Neural Network code.

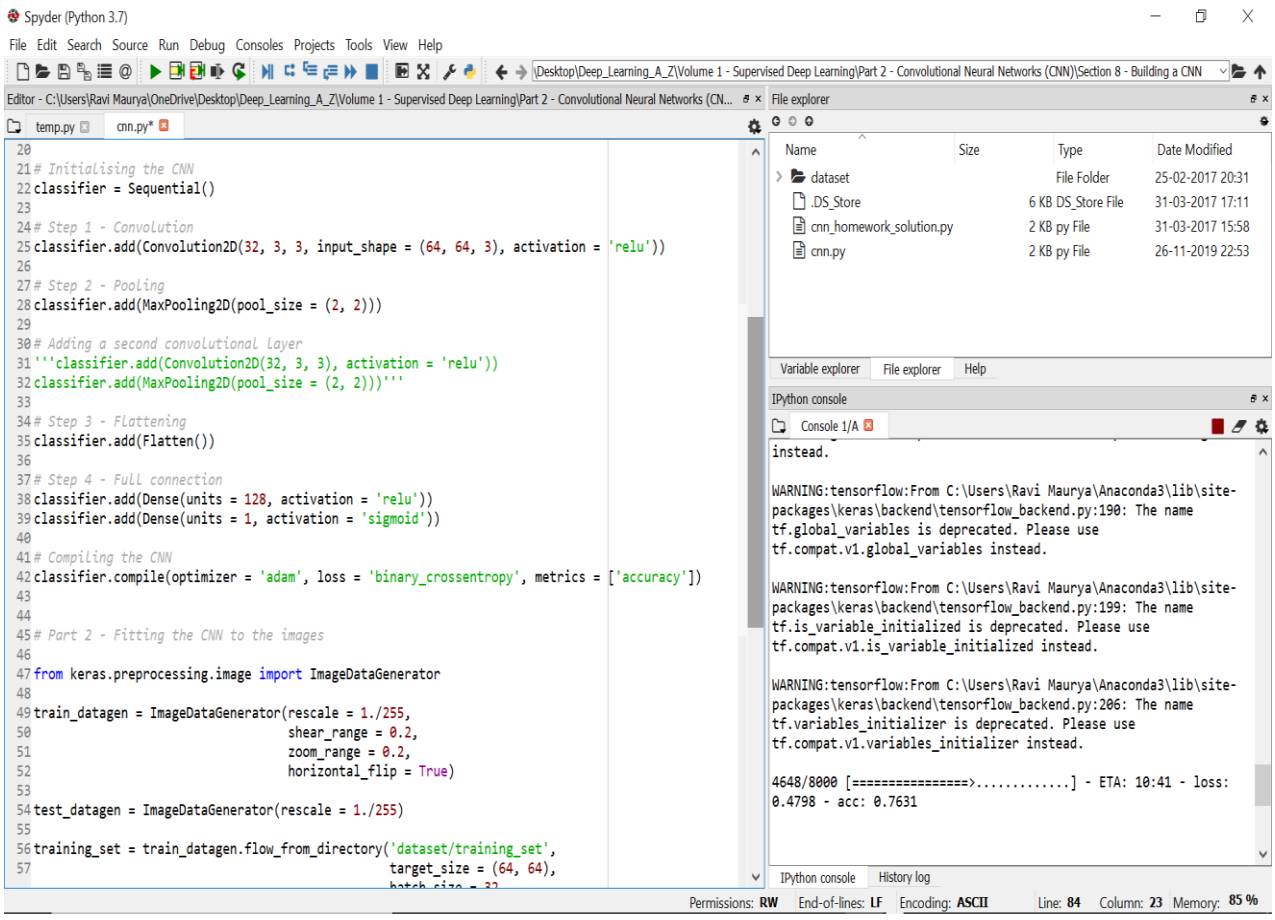
2 .Numpy

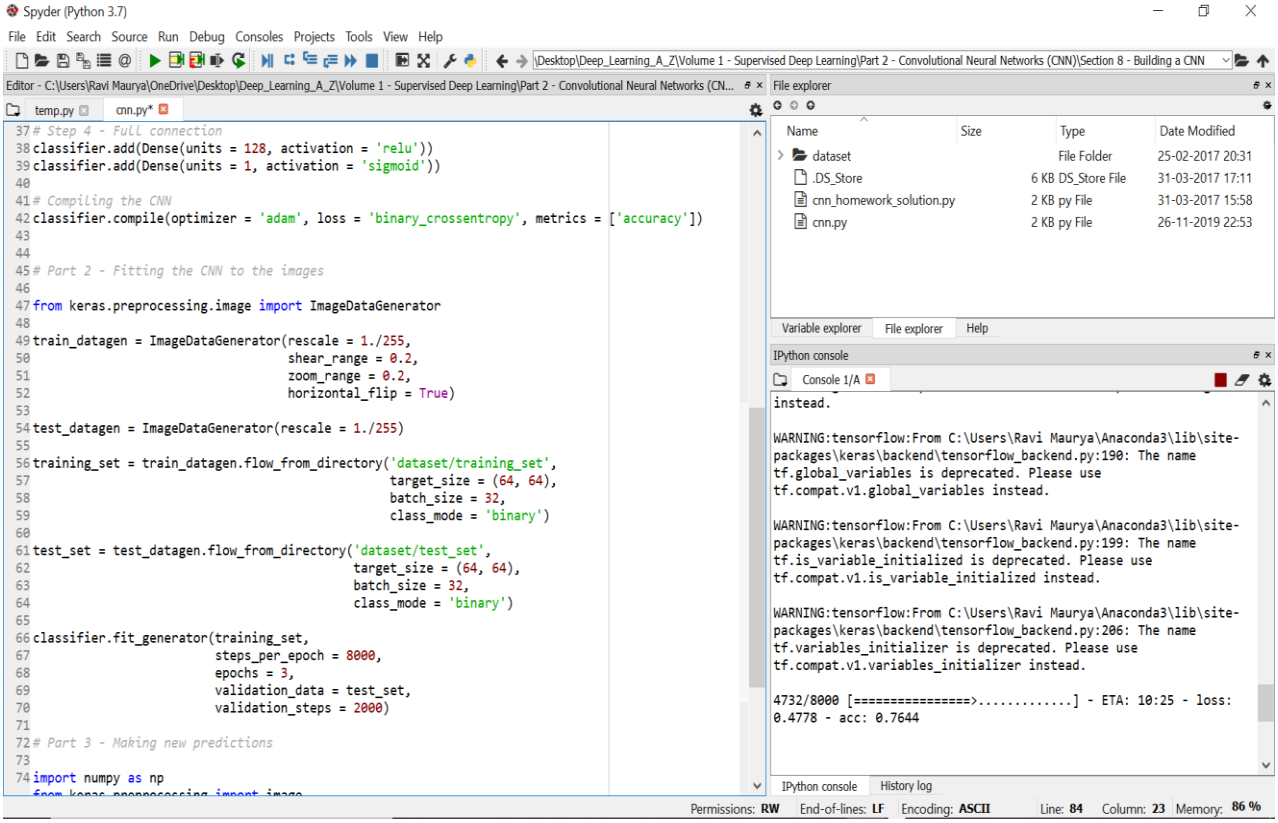
NumPy (pronounced (NUM-py) or sometimes (NUM-pee)) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors

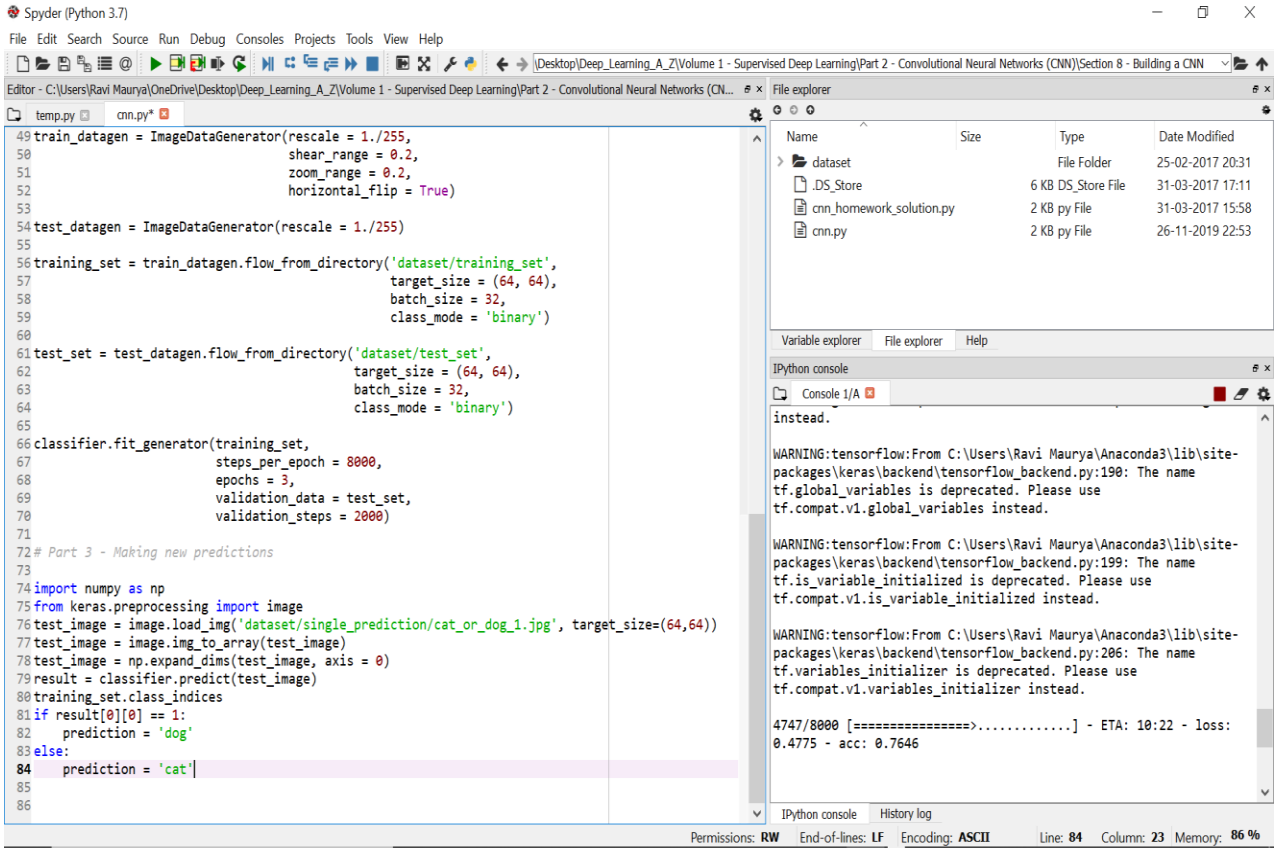
Blueprint diagram of Model




Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
\Desktop\Deep_Learning_A_Z\Volume 1 - Supervised Deep Learning\Part 2 - Convolutional Neural Networks (CNN)\Section 8 - Building a CNN
Editor - C:\Users\Ravi Maurya\OneDrive\Desktop\Deep_Learning_A_Z\Volume 1 - Supervised Deep Learning\Part 2 - Convolutional Neural Networks (CN...
temp.py
cnn.py*
# Convolutional Neural Network
# Installing Theano
# pip install --upgrade --no-deps git+git://github.com/Theano/Theano.git
# Installing Tensorflow
# pip install tensorflow
# Installing Keras
# pip install --upgrade keras
# Part 1 - Building the CNN
# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
# Initialising the CNN
classifier = Sequential()
# Step 1 - Convolution
classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Adding a second convolutional Layer
'''classifier.add(Convolution2D(32, 3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))'''
# Step 3 - Flattening
classifier.add(Flatten())
# Step 4 - Full connection
classifier.add(Dense(units = 128, activation = 'relu'))
File explorer
Name Size Type Date Modified
dataset File Folder 25-02-2017 20:31
.DS_Store 6 KB DS_Store File 31-03-2017 17:11
cnn_homework_solution.py 2 KB py File 31-03-2017 15:58
cnn.py 2 KB py File 26-11-2019 22:53
Variable explorer File explorer Help
IPython console
Console 1/A
instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:190: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:199: The name tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:206: The name tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer instead.
4586/8000 [==========>.....] - ETA: 10:54 - loss: 0.4814 - acc: 0.7620
IPython console History log
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 84 Column: 23 Memory: 85 %


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
\Desktop\Deep_Learning_A_Z\Volume 1 - Supervised Deep Learning\Part 2 - Convolutional Neural Networks (CNN)\Section 8 - Building a CNN
Editor - C:\Users\Ravi Maurya\OneDrive\Desktop\Deep_Learning_A_Z\Volume 1 - Supervised Deep Learning\Part 2 - Convolutional Neural Networks (CN...
temp.py
cnn.py*
# Initialising the CNN
classifier = Sequential()
# Step 1 - Convolution
classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Adding a second convolutional Layer
'''classifier.add(Convolution2D(32, 3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))'''
# Step 3 - Flattening
classifier.add(Flatten())
# Step 4 - Full connection
classifier.add(Dense(units = 128, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
# Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
# Part 2 - Fitting the CNN to the 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)
training_set = train_datagen.flow_from_directory('dataset/training_set',
target_size = (64, 64),
File explorer
Name Size Type Date Modified
dataset File Folder 25-02-2017 20:31
.DS_Store 6 KB DS_Store File 31-03-2017 17:11
cnn_homework_solution.py 2 KB py File 31-03-2017 15:58
cnn.py 2 KB py File 26-11-2019 22:53
Variable explorer File explorer Help
IPython console
Console 1/A
instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:190: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:199: The name tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:206: The name tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer instead.
4648/8000 [==========>.....] - ETA: 10:41 - loss: 0.4798 - acc: 0.7631
IPython console History log
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 84 Column: 23 Memory: 85 %


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:\Users\Ravi Maurya\OneDrive\Desktop\Deep_Learning_A_Z\Volume 1 - Supervised Deep Learning\Part 2 - Convolutional Neural Networks (CN...
temp.py
cnn.py*
37 # Step 4 - Full connection
38 classifier.add(Dense(units = 128, activation = 'relu'))
39 classifier.add(Dense(units = 1, activation = 'sigmoid'))
40
41 # Compiling the CNN
42 classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
43
44
45 # Part 2 - Fitting the CNN to the images
46
47 from keras.preprocessing.image import ImageDataGenerator
48
49 train_datagen = ImageDataGenerator(rescale = 1./255,
50 shear_range = 0.2,
51 zoom_range = 0.2,
52 horizontal_flip = True)
53
54 test_datagen = ImageDataGenerator(rescale = 1./255)
55
56 training_set = train_datagen.flow_from_directory('dataset/training_set',
57 target_size = (64, 64),
58 batch_size = 32,
59 class_mode = 'binary')
60
61 test_set = test_datagen.flow_from_directory('dataset/test_set',
62 target_size = (64, 64),
63 batch_size = 32,
64 class_mode = 'binary')
65
66 classifier.fit_generator(training_set,
67 steps_per_epoch = 8000,
68 epochs = 3,
69 validation_data = test_set,
70 validation_steps = 2000)
71
72 # Part 3 - Making new predictions
73
74 import numpy as np
File explorer
Name Size Type Date Modified
dataset File Folder 25-02-2017 20:31
.DS_Store 6 KB DS_Store File 31-03-2017 17:11
cnn_homework_solution.py 2 KB py File 31-03-2017 15:58
cnn.py 2 KB py File 26-11-2019 22:53
Variable explorer File explorer Help
IPython console
Console 1/A
instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:190: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:199: The name tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:206: The name tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer instead.
4732/8000 [===============>.....] - ETA: 10:25 - loss: 0.4778 - acc: 0.7644
IPython console History log
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 84 Column: 23 Memory: 86 %


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:\Users\Ravi Maurya\OneDrive\Desktop\Deep_Learning_A_Z\Volume 1 - Supervised Deep Learning\Part 2 - Convolutional Neural Networks (CN...
temp.py
cnn.py*
49 train_datagen = ImageDataGenerator(rescale = 1./255,
50 shear_range = 0.2,
51 zoom_range = 0.2,
52 horizontal_flip = True)
53
54 test_datagen = ImageDataGenerator(rescale = 1./255)
55
56 training_set = train_datagen.flow_from_directory('dataset/training_set',
57 target_size = (64, 64),
58 batch_size = 32,
59 class_mode = 'binary')
60
61 test_set = test_datagen.flow_from_directory('dataset/test_set',
62 target_size = (64, 64),
63 batch_size = 32,
64 class_mode = 'binary')
65
66 classifier.fit_generator(training_set,
67 steps_per_epoch = 8000,
68 epochs = 3,
69 validation_data = test_set,
70 validation_steps = 2000)
71
72 # Part 3 - Making new predictions
73
74 import numpy as np
75 from keras.preprocessing import image
76 test_image = image.load_img('dataset/single_prediction/cat_or_dog_1.jpg', target_size=(64,64))
77 test_image = image.img_to_array(test_image)
78 test_image = np.expand_dims(test_image, axis = 0)
79 result = classifier.predict(test_image)
80 training_set.class_indices
81 if result[0][0] == 1:
82 prediction = 'dog'
83 else:
84 prediction = 'cat'
85
86
File explorer
Name Size Type Date Modified
dataset File Folder 25-02-2017 20:31
.DS_Store 6 KB DS_Store File 31-03-2017 17:11
cnn_homework_solution.py 2 KB py File 31-03-2017 15:58
cnn.py 2 KB py File 26-11-2019 22:53
Variable explorer File explorer Help
IPython console
Console 1/A
instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:190: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:199: The name tf.is_variable_initialized is deprecated. Please use tf.compat.v1.is_variable_initialized instead.
WARNING:tensorflow:From C:\Users\Ravi Maurya\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:206: The name tf.variables_initializer is deprecated. Please use tf.compat.v1.variables_initializer instead.
4747/8000 [===============>.....] - ETA: 10:22 - loss: 0.4775 - acc: 0.7646
IPython console History log
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 84 Column: 23 Memory: 86 %

Conclusion

Convolutional neural network provides a better way to feature an image with the help of python keras library one can easily process the image. And can easily implement the concept of deep learning i.e convolutional neural network. Images one featured they are pooled according to max function hence called max pooling. Since the input layers on our neural network better work with 1D flattend data so we use flattend process in third step to flatten our data once the data fatten we connect it to neural networks and done our image processing after training our neural network we can accurately predict the output of an image keeping in mind the loss function. We should train our model to such an extent such that there is very less loss should present in the output with accuracy above 75 percent

References

www.udemy.com

www.superdatascience.com

www.keras.io

Research papers