

Image Classification using Deep Learning

Sejal Bhat

Dept. of Computer Science Engineering, USICT, Guru Gobind Singh Indraprastha University, Delhi, India

Abstract - Deep Learning aims to work on complex data and achieve accuracy. It works on AI-based domains like Natural Language Processing and Computer vision[1]. In deep learning, the computer model learns to perform classification of task from images, text or sound. Image classification is the task of extracting essential features from given input image that are required to predict the correct classification. The objective is to build a Convolution Neural Network model that can correctly predict and classify the input image as Dog or Cat. The classification is done by extracting specific features of the input image. The CNN Model consists of various layers like Convolution layer, ReLU layer, Pooling layer, etc. The model is trained well with training data. At last, the CNN model is tested for accuracy in image classification with the help of some test images.

Key Words: Deep Learning, Convolution Neural Network, ReLU, Maxpooling, Keras.

1. INTRODUCTION

Deep Learning is used on complex and large data. It basically makes the computer learn to filter inputs through layers so that the correct prediction and classification takes place. The libraries used for classification are Tensorflow, Keras, OpenCV and Tensorboard. Tensorflow is a free and open-source python library used for fast numerical computations and to create deep learning model. Keras is also a free and open-source library that acts as an interface for Tensorflow. OpenCV is an open-source computer vision library that is used to perform operations on image. Tensorboard is basically used to visualize the training model over time.

The dataset used for the image classification is Cats and Dogs dataset which is taken from Kaggle. The dataset consists of colored images of dogs and cats. The model used to implement this classification is CNN. The most beneficial aspect of CNN is to reduce parameters in neural network [2]. CNN are widely used in areas related to image classification and recognition [3]. The model is trained and then it is tested on various images by loading the dataset to Google Collab. The next section describes the methodology of CNN. The third section consists of implementation related to the work. The fourth section consists of results obtained and in the last section, conclusion has been discussed.

2. METHODOLOGY

The model used for Image Classification using Deep Learning is CNN. Convolution Neural Network consists of various layers in order to predict and classify the test input image.

- i. **Convolution Layer:** - This layer is the basic unit of CNN. It consists of a filter that is applied on the input data so that the features are extracted from it for classification purpose. In this work, we used 64 units and 3x3 window size. The 3x3 is the feature detector used to extract the feature map. CNN takes a 2D image as an input. Feature detector help in identify important features present in image like edges, vertical lines, etc. The feature map values are calculated using equation-

$$G[m, n] = (f * h)[m, n] = \sum_j \sum_k h[j, k] f[m-j, n-k]$$

Here, f is the input image and h is the filter or feature detector. G[m, n] is the result matrix where m and n are the indexes of rows and columns. Asterisk * sign denotes the convolution between the image and the feature detector.

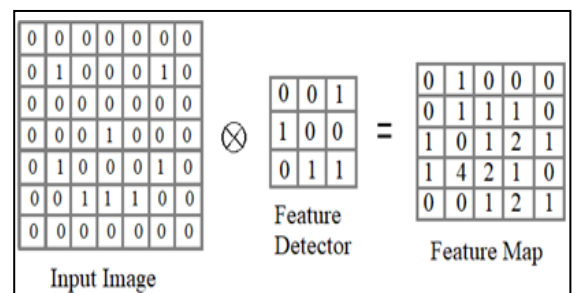


Fig -1: Working of Convolution Layer

In fig, the Dimension of image- (m, n) = (7, 7)

Dimension of Feature detector- (j, k) = (3, 3)

Dimension of Feature Map-

$((m-j+1), (n-k+1)) = (5, 5)$

- ii. **Rectified Linear Unit:** - The aim of using this layer is to apply non-linearity so that the neural network is capable to perform tasks like object classification and text prediction. This layer helps the Conv2D layer in extracting the features from the input image to obtain the feature map. This activation function is better than Tanh and Sigmoid function as it is faster in terms of speed. RELU is used to remove the negative values from feature map. In convolution layer, the resultant feature maps may consist of some negative value.

So ReLU is used to remove negative values and add non-linearity.

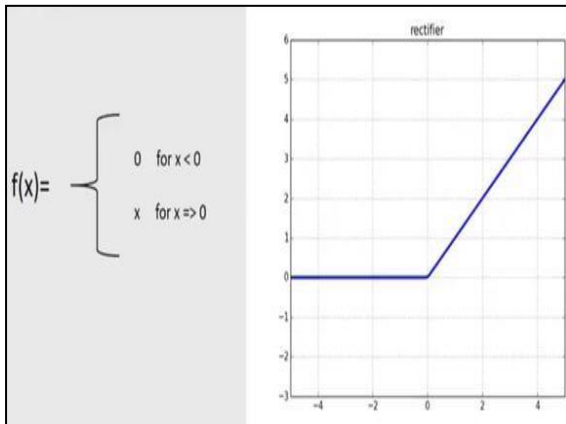


Fig -2: Working of ReLU Layer

iii. **Pooling Layer:-** This Layer is used to reduce the dimensions of a feature map using Max pooling technique. In Maxpooling layer, only the most important features of image are extracted. In this work, we used 2x2 window size for Maxpooling. In the below image, the stride taken for maxpooling is 2. The maximum value present in the stride 2 over 1 feature map of (5,5) is selected and at last we get the resultant pooled feature map with reduced size of (3,3).

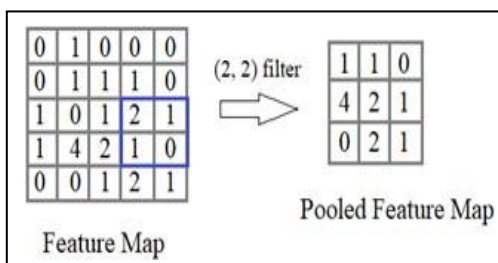


Fig -3: Working of Pooling Layer

iv. **Flattening Layer:-** This layer is mainly used to convert the output of convolution layer or Max Pooling layer to 1-D feature vector so that it can be easily used to predict the object. Flattening is basically done to store only the pixel value of

image and not its spatial structure. The input 2-D image is flattened to 1-D feature vector so that it is easily sent to fully connected layer for classification purpose. All of these individual values in 1-D feature vector are treated as separate features that represent the image.

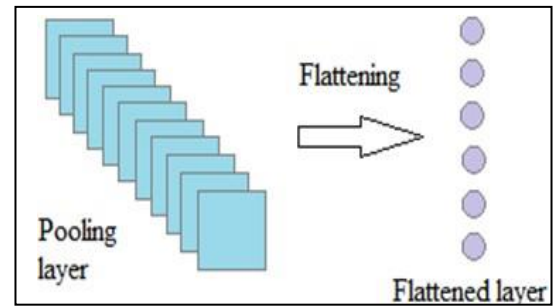


Fig -4: Working of Flattening Layer

v. **Fully-connected Layer:-** The main aim of this layer is to classify the input image into a label for object classification problem. This is the last layer of CNN model that gives the output. The fully connected layer performs two operations that are Linear transformation and a Non-Linear transformation. The Linear Transformation is performed using equation:-

$$Z = W^T \cdot A + b$$

Here, A is the input, W is randomly initialized weight matrix and b is the bias matrix.

The Non-Linear Transformation in binary classification is performed using Sigmoid function. The range of a Sigmoid function is in range (0,1). The equation of Sigmoid activation function is:-

$$f(x) = 1/(1+e^{-x})$$

After the flattening layer, the dense layer uses ReLU activation function for the hidden layers. Then, dense layer uses Sigmoid function for output layer. The dense layer is used to classify the input image.

3. IMPLEMENTATION

For implementation, the dataset named "Pets Dataset" is taken from Kaggle. The training dataset consists images of 1100 Cats and 1100 Dogs. The Cat and dog dataset is loaded to Google Collab through google drive. Google Collab is a free and open-source service provided by google. It is used to develop various machine learning and deep learning models. It also supports GPU (Graphical Processing Unit) for faster training of models. In Google Collab, we imported some libraries like Numpy, Matplotlib, OS and CV2. Numpy is used to do various array operations. Matplotlib is used to show the image. The image is shown in the form of OS is used to iterate through directories and join paths. CV2 is used to do some image operations. For implementation, Cat and dog dataset is loaded to Google Collab through google drive. For

classification, CATEGORIES list consists of category Cat and Dog. The images in the given dataset are named in numbers, so we will convert image to array. The dataset consists of colored images of Cats and Dogs. For faster classification, we will convert colored images to Grayscale. The classification is recommended to be done in grayscale because colored image is made of RGB combination and each color in RGB (Red, Green, Blue) has its own feature map. Images in dataset are in either landscape form or in portrait form. So in order to simplify, we will use one type of size. Once the IMG_SIZE is chosen, we will resize the images.

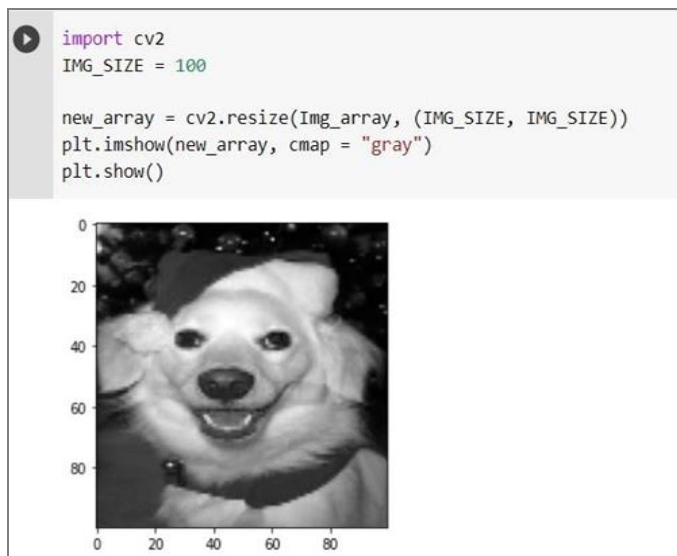


Fig -5:- Resized Grayscale Image

After this, the training set is made. The label is to be set for classification as we cannot map to names like dog and cat. Some index needs to be created for better classification like 0 for Cat and 1 for Dog. Now, again resizing is done. Resizing is done for simplification. The index and the resized images are added to training set. For better classification, dataset should be balanced.

For classification purpose, we need one feature set and one target set (labels). For training set, X array consists of the features and y array consists of labels. Now we will import pickle, so that our images in training set are mapped to required label and then saving it. Pickle is used to save the model. It is used to serialize the objects of model.

Once the training set is ready, import Tensorflow to create the deep learning model. Keras is also used to provide interface to Tensorflow. In this project, CNN model is used to develop the deep learning model. We will import Convolution, Activation, Maxpooling, Flatten, Dropout and Dense layers.

The data is normalized before being feed to the neural network. The most easiest way is to scale the data. Convolution layer is used to fetch the features of input image. In this implementation, we used 64 units and 3x3

window size. After Convolution, Activation is applied to model. The Activation used in this work is Rectified Linear Unit (ReLU). ReLU is added for adding non-linearity for classification. Maxpooling is done to extract the most specific feature of image. In this work, we used 2x2 window size for maxpooling. Flatten layer is used flatten the output of Convolution layer or Pooling layer to 1-D feature vector. Dropout layer is used to drop random set of neurons. Dense layer is the fully connected layer that is used to classify the image into a label for classification. The total number of trainable parameters are 9,835,265.

| Layer (type) | Output Shape | Param # |
|------------------------------|--------------------|---------|
| conv2d (Conv2D) | (None, 98, 98, 64) | 640 |
| activation (Activation) | (None, 98, 98, 64) | 0 |
| max_pooling2d (MaxPooling2D) | (None, 49, 49, 64) | 0 |
| flatten (Flatten) | (None, 153664) | 0 |
| dense (Dense) | (None, 64) | 9834560 |
| activation_1 (Activation) | (None, 64) | 0 |
| dense_1 (Dense) | (None, 1) | 65 |
| activation_2 (Activation) | (None, 1) | 0 |
| Total params: 9,835,265 | | |
| Trainable params: 9,835,265 | | |
| Non-trainable params: 0 | | |

Fig -6:- CNN Model

The type of model used in this work is Sequential Model. The Tensorflow and Keras library are used to build this model. It is termed sequential because it is the easiest way to build a deep learning model in Keras. The model is built layer by layer. The layers of CNN model are stacked well in sequential order from input to output. The activation_1 function uses the ReLU activation function and the activation_2 uses the Sigmoid activation function as binary classification problem is considered. This CNN Model is the trained model. The accuracy of this model for generalized data will be tested using some test data.

4. RESULTS

The CNN model for Image classification of Dog and Cat is implemented. Some test images are used for testing the accuracy of classification given by CNN Model. The Test image1 named "dog_small.jpg" is used as test input image for classification. It is correctly classified by CNN Model as Dog.



Fig -7: Test Image1

Dog is taken as a test data. The trained CNN model will extract the features of Dog using Convolution layer with the help of 3x3 feature detector window. We will obtain the required feature map of input image. The negative values of the Feature map are removed with the help of ReLU layer. This layer adds some non-linearity.

The Maxpooling layer is used to extract the special and important features of Dog in 2x2 window. The Flattening layer will be used to convert the 2x2 Maxpooling window to 1-D Feature vector. All of these individual values in 1-D feature vector are basically the separate features that represent the image. At last, the Dense layer will use the ReLU and Sigmoid Activation function for classification of the correct category.

```
prediction = model.predict([prepare('dog_small.jpg')])
print(CATEGORIES[int(prediction[0][0])])
```

Dog

Fig -8: Classification of Test Image1

Here, the prediction is done on the basis of indexes allotted to the label of Cat or Dog. The index 0 is for Cat and 1 is for Dog. The index is mapped to the label that is defined in the CATEGORIES list.

5. CONCLUSION

The CNN model for Image classification using deep learning is made. The model consists of various layers like convolution layer, ReLU layer, Maxpooling layer, Flattened layer, Dropout layer and Dense layer. The model is tested by providing some test input images to check whether the correct output is obtained or not. The essential features of image are extracted by Convolution and Pooling layer. Some index value is assigned to each label for easy classification. The Dense layer in the model will label the image as Dog or Cat. The input test image will be classified either as dog or cat.

REFERENCES

- [1] Jia, X. (2017, May). Image recognition method based on deep learning. In 2017 29th Chinese Control And Decision Conference (CCDC) (pp. 4730-4735). IEEE.
- [2] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolution neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). IEEE.
- [3] Toleubay, Y., & James, A. P. (2020). Getting started with TensorFlow deep learning. In Deep Learning Classifiers with Memristive Networks (pp. 57-67). Springer, Cham.
- [4] Om, P., & Vijay, G. (2018). Classification of vegetables using TensorFlow. International Journal for Research in Applied Science and Engineering Technology, 6(4), 2926-2934.
- [5] Haykin, S. (2010). Neural Networks and Learning Machines, 3/E. Pearson Education India.
- [6] Sultana, F., Sufian, A., & Dutta, P. (2018). Advancements in Image Classification using Convolutional Neural Network. 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks.