

DESIGN & SIMULATION LAB REPORT

Topic : HandWritten Digit Classification using VGG-16

Arkajit Pal

22SP06008

M.Tech 1st Year, SPCOM
IIT Bhubaneswar

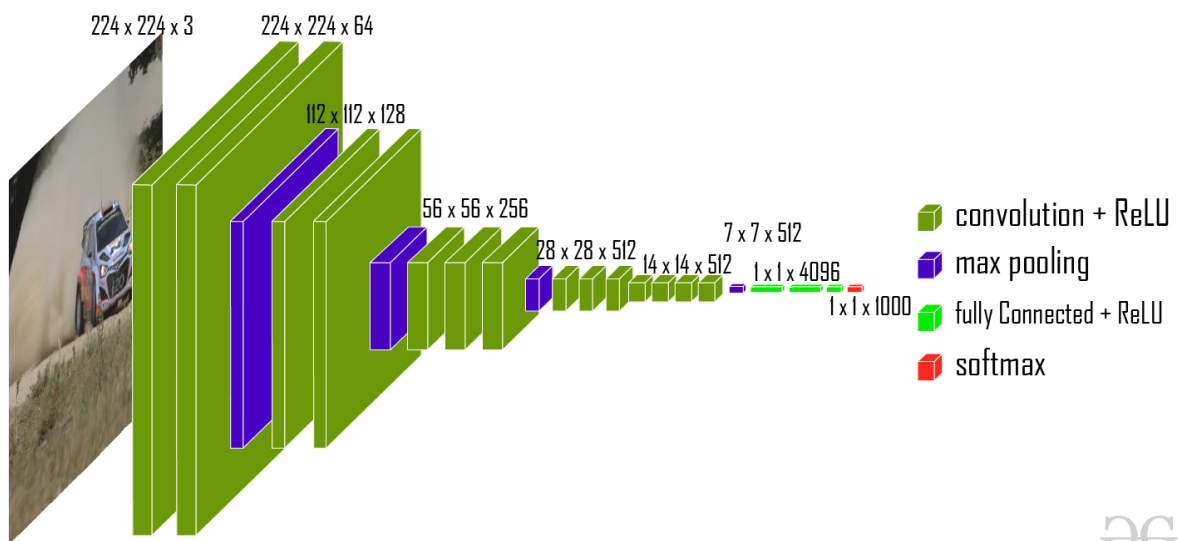


Introduction

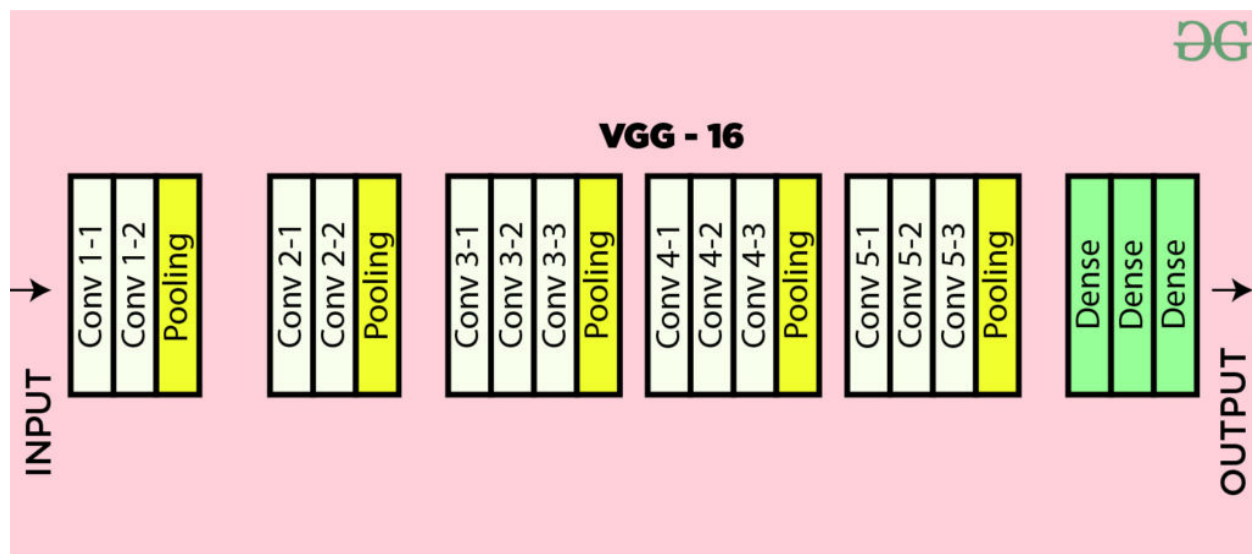
Convolutional neural networks (CNNs) are very effective in perceiving the structure of handwritten characters/words in ways that help in automatic extraction of distinct features and make CNN the most suitable approach for solving handwriting recognition problems. Our aim in the proposed work is to explore the various design options like number of layers, stride size, receptive field, kernel size, padding and dilution for CNN-based handwritten digit recognition. VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date. The 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.


Theory

MNIST Data : The MNIST database (Modified National Institute of Standards and Technology database) is a large collection of handwritten digits. It has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger NIST Special Database 3 (digits written by employees of the United States Census Bureau) and Special Database 1 (digits written by high school students) which contain monochrome images of handwritten digits. The digits have been size-normalized and centered in a fixed-size image. The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. the images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.



VGG Architecture: The input to the network is an image of dimensions (224, 224, 3). The first two layers have 64 channels of a 3*3 filter size and the same padding. Then after a max pool layer of stride (2, 2), two layers have convolution layers of 128 filter size and filter size (3, 3). This is followed by a max-pooling layer of stride (2, 2) which is the same as the previous layer. Then there are 2 convolution layers of filter size (3, 3) and 256 filters. After that, there are 2 sets of 3 convolution layers and a max pool layer. Each has 512 filters of (3, 3) size with the same padding. This image is then passed to the stack of two convolution layers. In these convolution and max-pooling layers, the filters we use are of the size 3*3 instead of 11*11 in AlexNet and 7*7 in ZF-Net. In some of the layers, it also uses 1*1 pixel which is used to manipulate the number of input channels. There is a padding of 1-pixel (same padding) done after each convolution layer to prevent the spatial feature of the image.





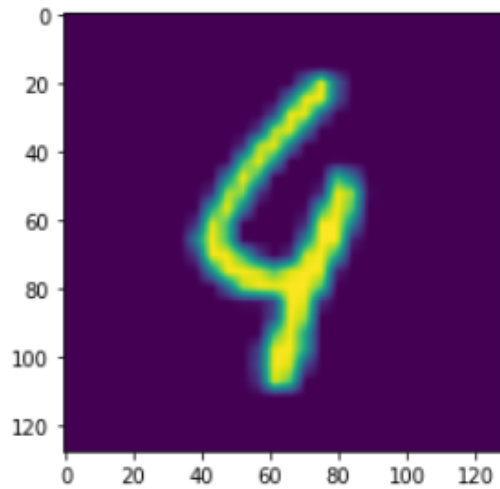
After the stack of convolution and max-pooling layer, we got a $(7, 7, 512)$ feature map. We flatten this output to make it a $(1, 25088)$ feature vector. After this there is 3 *fully* connected layer, the first layer takes input from the last feature vector and outputs a $(1, 4096)$ vector, the second layer also outputs a vector of size $(1, 4096)$ but the third layer output a 1000 channels for 1000 classes of ILSVRC challenge i.e. 3rd fully connected layer is used to implement softmax function to classify 1000 classes. All the hidden layers use ReLU as its activation function. ReLU is more computationally efficient because it results in faster learning and it also decreases the likelihood of vanishing gradient problems.

METHODOLOGY

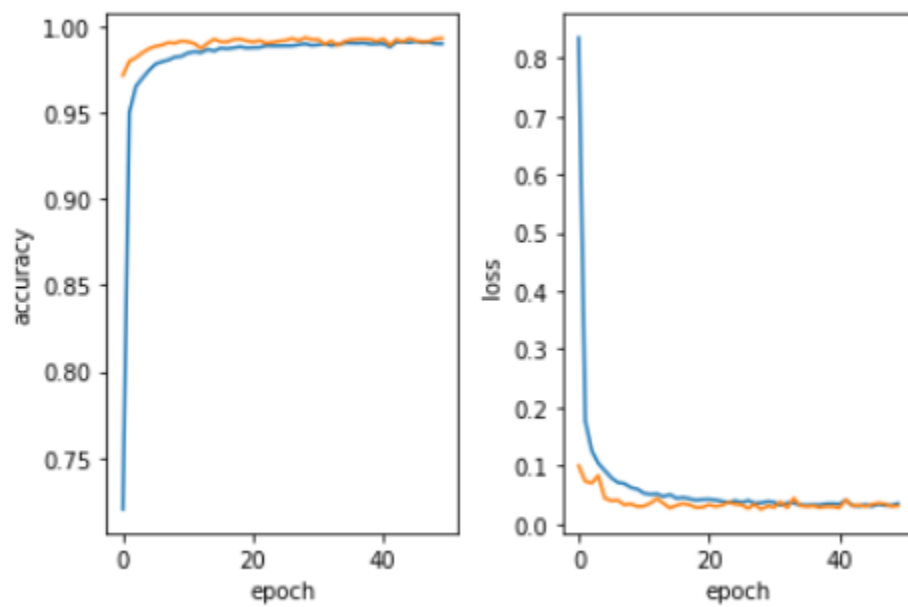
- Using MNIST datasets
- Training of datasets
- Fitting the model
- Evaluating the model
- Prediction matching, Results and Graphs
 - Handwritten dataset pixel size is 28X28.
 - VGG16 model is designed 13 Convolution Layer and 3 Fully Connected Layer.
 - Used Filter is 4,8,16,32,64
 - Used Kernel (3X3)
 - ReLu and Softmax is as Activation Report
 - 'Same' as padding
 - Strides size is - 2x2
 - Dropout

RESULT

› exact value - 4
predicted value - 9



Fig(1) Predicted Value



Fig(2) Training and Loss Curves



DISCUSSION

The model is trained in such a manner that it is showing more than 99% accuracy. Even if accuracy is 99.29%, some of the images are predicted wrong.

CONCLUSION

Accuracy for this model is 99.29% and Loss curve is converging in the same manner.

Score = 99.29%

REFERENCE

1. IIT Madras NPTEL, Dr. Mitesh Khapra Deep Learning
2. Codebasics
3. <https://www.geeksforgeeks.org/vgg-16-cnn-model/>
4. <https://elitedatascience.com/keras-tutorial-deep-learning-in-python>