Object Detection Using Deep Neural Networks

Malay Shah
Institute of Technology, Nirma University
Ahmedabad, India
13bce102@nirmauni.ac.in

Prof. Rupal Kapdi
Institute of Technology, Nirma University
Ahmedabad, India
rupal.kapdi@nirmauni.ac.in

***Abstract*— The problem discussed in this article is object detection using deep neural network especially convolution neural networks. Object detection was previously done using only conventional deep convolution neural network whereas using regional based convolution network [3] increases the accuracy and also decreases the time required to complete the program. The dataset used is PASCAL VOC 2012 which contains 20 labels. The dataset is very popular in image recognition, object detection and other image processing problems. Supervised learning is also possible in implementing the problem using Decision trees or more likely SVM. But neural network work best in image processing because they can handle images well. (*Abstract*)**

Keywords—object detection; neural network

I. INTRODUCTION

Object detection has been a topic for challenge and many methodologies are applied. Object detection is detecting a specific object from an image of multiple and complex lines and shapes. Object detection is used in face detection, object tracking, image retrieval, automated parking systems [12]. The number of the applications are increasing in number. The main use of object detection is image classification or more precisely image retrieval. For understanding the convolution neural network, deep neural network is important. Papers in deep neural network are studied to understand the concepts of convolution neural network. NIPS paper on regional based convolution neural network is also referred for further comparison [3]. Object detection is being used in various other fields like defense, architecture, etc. But it is used the most for medical purposes. One of the examples is detecting tumor in the brain using deep neural network [11].

II. MACHINE LEARNING FUNDAMENTALS

A. Logistic Classifier and softmax function

The most common linear function which is used widely for the training of data is the logistic classifier. Logistic classifier consists of weight and biases which are used to tune the changes in the data. The weight decides the class of the point [9].

$$Wx + B = y$$

Where W = weights, x= inputs, B=biases, y=output

The input of the equation is a vector and the output is a 1d array of values in which the highest value corresponds to the highest likely label. The softmax function changes this output into the probabilities where the addition of probabilities is one and the one with the highest output value has the highest probability.

The output of the softmax is then converted into 1 hot encoding vectors which allow the user to know which class label is predicted. The whole map is shown in the Fig. 1 [9].

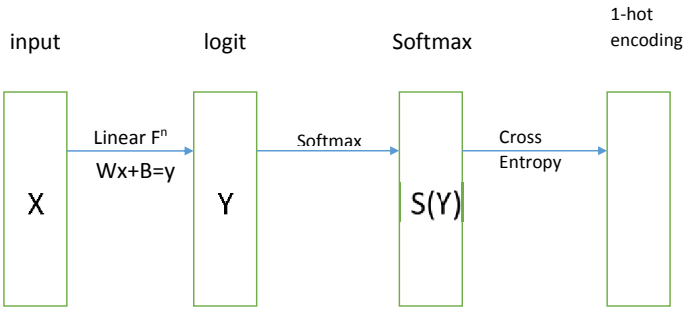

Figure 1: Multinomial logistic classification

III. DEEP NEURAL NETWORKS[6]

Deep neural networks are an extension of the artificial neural networks. Increasing the depth of the ANN increases the depth of the network which is defined as deep neural network. Deep networks are those whose depth is more than 3 or have more than 3 layers. The main advantage of using these techniques is that we can also use non-linearity. Non-linearity can be used by implementing RELU (Rectified Linear Units) in the hidden layers. Deep neural network has total of (n+1) x k parameters where n is the input and k is the output.

Using a linear function has many advantages over using a non-linear function. The differentiation of a linear function is a constant. Matric multiplication in GPU is faster in linear function while in non-linear function. Linear functions are very stable i.e. small changes does affect the output. So while implementing neural network with more layers a linear function can be used.

A. RELU (Rectified Linear Units)

RELU functions are the functions which convert non-linear equation into a linear one. The RELU function is defined as

$$y = \begin{cases} 0 \; if \; x < 0 \\ x \; if \; x \geq 0 \end{cases}$$

The advantage of this function is that the derivative of the function is a constant function. So when adding a hidden layer in a neural network RELU function are used.

B. Regularization

Regularization is a technique in which the number of parameters are decreased implicitly. When large network are used chance of overfitting increases. After the overfitting is done, we try to minimize the overfitting by implementing regularization. The time required to complete the training is dependent on performance and is a near linear curve. After certain period overfitting starts and the training should be terminated at that point. There are 2 ways through which regularization can be done:

1) L2 Regularization:Adding certain function to the loss function reduces the overall number of the iteration. The l2 loss can be calculated by the equation:

$$L^{'} = L + \beta \frac{1}{2} \|W\|_2^2$$

Where $\|w\|_2^2 = {W_1}^2 + {W_2}^2 + {W_3}^2 + \cdots + {W_n}^2$

2) Dropout: Dropout is dropping some of the units from the neural network so that the network is not entirely dependent on those units. It removes the unit temporarily. It prevents overfitting and also provides an efficient way for combining many different neural networks[7].

C. Convolution Neural Networks

Translation variance: For instance, there is a problem of identifying a cat in an image, the position of the cat is not important but the recognition is important. For such examples the weight sharing is needed [8].

For such purposes convolution neural networks are used. Convolution neural network is similar to the basic neural network but for images. It makes an assumption that the input is a collection of pixels which allows us to encode from the properties of the architecture. The convolution Net architecture is a list of layers that transform an image volume into an output volume (holding the class scores). Many functions are used with Convolution Nets (Pooling, RELUs, etc.). The convolution function converts the 3d image into 3d image with different depths.

The convolutions try to summarize all the data available in the image into a single cell of an array such that it can be easily accessed. As an analogy, each pixel is a cell in a matrix and each cell is capable of storing 3 values. As and when we do convolutions we decrease the length and width of the image and increase the depth i.e. each cell is now capable of storing 9 values for instance. After some number of convolutions the length and width will be equal to 1 and the depth will be maximum. At that point, all the data in summarized into a single cell which can now be used as an input to a linear function.

1) Kernel or patch: The matrix which is used to traverse through the whole image

2) Feature Map: Representation of image from where the kernel is selected

3) Stride: the speed at which the kernel is traversed through the image.

4) Padding: whether any extra values are needed

a) Same Padding: the kernel is traversed with every pixel reached same times. The height and the width does not change.

b) Valid Padding: No padding is done. Here the height and the width of the convolution changes

D. Advanced Convolution Neural Network

Convolution Neural network can be made more efficient by using one or more of these three methods[6]

1) Pooling: Pooling is a method where all the neighbors are combined and the result is used as an input to the next layer. There are 2 types of pooling. Max pooling selects only the maximum of the all the neighbors. Average pooling takes average of all the values of the neighbors.

2) 1x1 convolution: Usually 1x1 convolution is absurd but when 1x1 pixel convolution is implemented using a linear function the mathematics is cheaper and also easy to use than when one uses direct convolution.

3) Inception: Inception is using any of the above techniques in any patch i.e. we can use pooling for one patch while conventional convolution for another patch. This decreases the time and the computation required implement convolution on an image. Inception is the most widely used method in convolution neural network for image recognition and detection.

IV. APPRAOCHES

Many approaches have been used for the problem statement. Moreover, ImageNet Large Scale Visual Recognition Competition (ILSVRC) competition is held every year for the same. These are the few nets and approaches used in recent years:

A. AlexNet

The ImageNet dataset was too extensive to implement supervised machine learning until development of AlexNet in the ILSVRC. The ILSVRC is like the Olympics in the image processing field. ALEXNet was the first algorithm to use deep convolution neural network for classifying the images. AlexNet contains 5 convolution layers, dropout layers, 3 pooling layers and 3 fully connected networks. The algorithm used in the fc are the usual stochastic gradient descent. The developers were able to achieve 15.2% top 5 error. The top 5 error is the rate at which, given an image, the model does not output the correct label with its top 5 predictions. Due to the amount of data and the processing power needed, the developers used 2 GTX GPU simultaneously. It took 2 days to train the net [1].

To combat the problem of overfitting, developers implemented dropout layers. Dropout significantly decreases the overfitting as well as some processing. They also used data

augmentation techniques which included image translations, horizontal reflections and patch extraction. ReLU is used for tackling non-linearity functions as tanh function, which were used conventionally, are slower.

B. VGG Net

This neural net was trained with simplicity and depth. It contained 19 layers which only used 3x3 strides with filter and padding of 1, along with a 2x2 max-pooling layers with stride 2. The developers Karen Simonyan and Andrew Zisserman of university of Oxford were able to achieve the top 5 error of 7.3%. The use of 3x3 strides instead of 11x11 as in AlexNet and 7x7 as in ZF-Net because combination of two 3x3 strides layers has an effective 5x5 filter which can give benefits of larger stride while keeping the advantages of the smaller filter. Using smaller stride decreases the number of parameters and in turn the processing power. The developers used caffe for building a model. They also used ReLU layers after each convolution layer and trained with batch gradient descent model [4].

C. GoogLeNet

GoogleNet is developed by Google developers for competing in the ILSVRC. It was able to achieve 6.3% error rate and the winner of the competition. They introduced inception module which is the combination of many convolution in a single image. GoogleNet is a 22 layer model which spreads linearly as well as laterally. Stacking these many layers and huge filter increases the computational cost and memory as well as increases the chances of overfitting [4].

D. Regional based convolution neural networks (RCNN)

RCNN is an advanced convolution neural network which initially divided the image into multiple regions and then apply convolution nets to that region individually. RCNN has been recently used in real time object detection. RCNN is comprised of 3 modules or phases. The first phase includes dividing or categorizing independent regions from the image. Some methods to achieve that are object-ness, selective search, object proposals, constraint parametric min-cuts (CPMC) multi-scale combinatorial grouping etc. The most preferred method is to use selective search. The second phase is the feature extraction where the weights are decided and the number of layers are decided for each convolution net. This is the most important module as high tuning is required. The third and the last module is the training module, where the convolution nets are trained with images with some labels.

The training module is further divided into 3 parts. The first part is the supervised pre-training where the images are trained using conventional convolution neural network where there are only one object in an image. The second part is the Domain specific fine tuning where the layers in convolution neural network only deals in the domain level. The third part is the object category classifiers where the object are placed in their respected classes or categories.

There are few more advanced version of region based convolution neural network, fast RCNN and faster RCNN. They use more refined version of the conventional RCNN for finding regions in an image.

V. Transfer Learning

Training a neural network from scratch takes more time and processing power as it is very difficult to find the dataset of sufficient size and ground truth. Instead, it is no uncommon to pre-train the convolution neural network on the large dataset like the ImageNet and use it for another dataset like PASCAL VOC with some fine-tuning. It nearly takes 2-3 weeks to learn a net on the ImageNet dataset so researchers keep their final checkpoints. Caffe has model zoo for this purpose. There are three major ways in which transfer learning is achieved.

Convolution neural net as a fixed feature extractor. Remove the fully connected layers from a convolution neural net trained on a large dataset and use that as a feature extractor for the new dataset. For AlexNet, a 4096-D vector is generated for each image and their output is then entered into any classifier. These features are called CNN codes. After getting the feature vector we can train the images in a linear classifier for a new dataset.

Fine- Tuning the convolution neural net. The second method is used when data is not in the format we need. Earlier only the classifier is replaces and retrained, but one can even change the parameters used in each layer according to the new dataset. One can only change few layers or all the layers or can only fine-tune high level portion of the network.

When training through a pre trained model, one should keep some things in mind. One cannot remove any layer arbitrarily. One has to keep in mind the architecture of the network [5].

VI. Results

This result was gained from applying the algorithms which are described above to the PASCAL VOC [10] dataset. The dataset contains 20 labels having minimum 1000 images. The images are in the .jpeg format. The size of the images ranges from 300x300 to 500x500. The images have to be converted into the desired format to comply with the net. The labels varies from object such as table, aero-plane, and car to person, dog, cat, etc. The table shows that the RCNN is working best in the present conditions. The same net can result in different result based on the initial weights and parameters.

CNN	**Top 5 error**
AlexNet	20%
VGG Net	18.2%
GoogleNet	15.5%
RCNN	12%

VII. Discussion

From the above result, one can conclude that the region based convolution neural network is more optimized at a very basic level. It is in dispute whether it can be said as the best form of solution to the problem or not. This result is valid

only in certain parameter. Another researcher can engender new parameters and would achieve less error rates than this but one cannot argue that the RCNN is better than the other neural net.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet Classification using deep neural networks," University of Toronto part of Advances in Neural information processing systems 25 (NIPS 2012)

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, " Going deeper with convolution," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature H ierarchies for Accurate Object Detection and Semantic Segmentation," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587

[4] Karen Simonyan, Andrew Zisserman, " Very deep Convolution networks for Large-scale image recognition", arXiv:1409.1556 [cs.CV]

[5] Stanford course, "Transfer Learning", retrieved from http://cs231n.github.io/transfer-learning/ last accessed on 18 April 2017

[6] I. Goodfellow, Y Bengio, A Courville, "Deep Forward Networks" in Deep Learning. MIT Press, 2016, ch 6.

[7] I. Goodfellow, Y Bengio, A Courville, "Regularization for Deep learning" in Deep Learning. MIT Press, 2016, ch 7.

[8] I. Goodfellow, Y Bengio, A Courville, "Convolution Networks" in Deep Learning. MIT Press, 2016, ch 9.

[9] Udacity course, "Deep learning," retrieved from https://www.udacity.com/course/deep-learning--ud730 last accessed on 20 April 2017

[10] "The Pascal VOC Hompage," retrieved from http://host.robots.ox.ac.uk/pascal/VOC/ last accessed on 18 April 20, 2017.

[11] Rupal R. Agravat, and Mehul S. Raval, "Brain Tumor Segmentation,"Computer Society of India , December 2016, p. 31.

[12] I. Goodfellow, Y Bengio, A Courville, "Regularization for Deep learning" in Deep Learning. MIT Press, 2016, ch 12.