Advanced Convolutional Neural Network Architecture : A Detailed Review

[1]Suresh Arunachalam T, [2]Shahana R, [3]Kavitha T
[1,2]B.Tech(CSE) - Final Year, [3]Assistant Professor, CSE
Periyar Maniammai Institute of Science & Technology,
Thanjavur, India

***Abstract* - *Deep Learning is a part of Machine Learning algorithm. It looks and works like a brain, it is also known as Artificial Neural Networks and it is trained from a large amount of data. Now-a-days deep learning technology plays a vital role in many fields such as an object detection, classification, image segmentation and recognition. In this paper, we discussed benefits, issues and applications of advanced architecture available in deep learning networks. Generally, deep learning architecture consists of three layers namely, input layer, hidden layers (Convolutional layers, Pooling layers and activation layers), and output layer. The learning can be achieved by any of the supervised, unsupervised, reinforcement algorithms. The common issues in the architectures fall under the requirement for larger dataset to train the neural network and it consumes more time to train the network.*

***Keywords* - *Deep Learning, Advanced Architecture, CNN.*

I. INTRODUCTION

Deep learning is the emerging technology and it is the part of machine learning which is powered by artificial neural networks. It plays a vital role in various fields such as search engine,
machine translation, medical field, voice recognition, computer vision, etc. From the word deep learning, the "deep" represent the number of hidden layers in neural networks. Machine Learning uses automated algorithms to process the input and generate the output which can either be a classification or binary value whereas Deep Learning uses neural network to process the input and produce the output in various forms such as text, audio, video, image, binary value. It support GPU, it is used to train enormous amount of data and reduces the time complexity. The architectures are differs from the parameters such as processing speed, filters, number of layers and activation functions.

II. RELATED WORKS

Convolutional Neural Networks

Artificial Neural Networks are applied in many classification tasks using audio, image, and text. Neural networks are applied in many fields. For classifying the tex,t Recurrent Neural Networks are used. CNNs are mainly used for image classification. Generally CNN consists of three layers namely input layer, hidden layers and an output layer.

Input Layer – This is the initial layer where the inputs are given. The more number of neurons helps to extract the more number of features.
Hidden Layer - It is an intermediate between the input layer and the output layer. It can have many numbers of hidden layers. It purely depends upon the model and the data set. By convolving the input image matrix, it generates the output in each layer with the help of previous layer.

[1] Convolution Layer – This layer extracts the features from the input layer by using filters.
[2] Activation function – The activation functions like Rectified Linear Unit(ReLU), sigmoid and tanh are used to train the network much faster and improves the accuracy. It replaces the negative value by '0'.
[3] Max Pool Layer – This layer reduces the size of the image by extracting the maximum pixel value from the activation layer.
[4] Fully connected layer - It is used for classifying the objects into different classes. In this layer each and every neuron is connected with the next layer.

Output layer – The output of the fully connected layer is the input to this layer. It generates the output with confidence value.

Pseudocode for Activation function

```
//It detect and classify the objects
//Input: Extracted features from convolution layer
//Output: Elimination of negative value
1:  activation_function←lamda
                              y:1.00/(1.00+cn.exp(-y)
2:  input_func←cn.random.func(2,3)
3:  layer1←activation_layer(cn.dot(K1,
                input_func)+a1)
4:  layer2←activation_layer(cn.dot(K2,
                layer1)+a2)
5:  output←cn.dot(K3,layer2)+a3
```

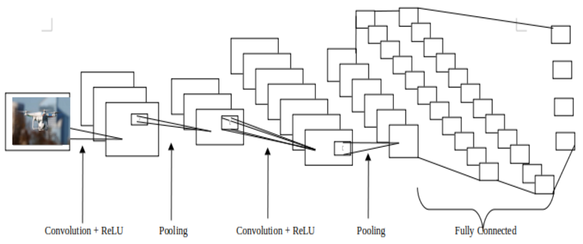

Fig. 1 CNN Architecture

Deep Network Architectures

A. VGG Net

The researchers at Visual Graphics Group developed the VGG Network at Oxford in the year of 2014. The shape of the network is in pyramid, where bottom layers are closer to the image which are wide and the top layers are deep. It consists of 19 Layers as maximum. The Subsequent convolutional layers in VGG Net are followed by pooling layers. Layers are narrower because of pooling layer. It took 138 million features to train the neural networks. It is a best approach for analysing a particular task. It is used in various applications because of its pre-trained networks that are freely available on internet. It takes more time to train the network even on GPU, that is more than a week to train. It consists of various weighted parameters, it makes the model more heavier.

Pseudocode for VGGNET

```
//It detect and classify the objects
//Input: It can be a video or image
//Output: Detecting object using bounding box
1:  if vgg_py is None
2:    location ← inspect.getfile(videopath)
3:    location← os.location.abspath
                (os.path.join(path, os.pardir))
4:    location ← os.path.join(location,
                "video.npy")
5:    vgg_py ← location
6:    self.data_dict ← np.load(vgg_py,
                encoding←'latin1').item()
7:    start_time ← time.time()
8:    rgb_scaled ← rgb * 255.0
9:    rgb ←  tf.split(num←3, value←rgb_scaled)
```

B. GoogleNet

GoogleNet is also known as Inception Network. It was introduced by researchers at Google. It is a robust modelwhich provides solution for classification and detection challenges. It is more deeper than VGG because it consists of additionally 3 layers. In this architecture, varied features are extracted within a single layer. This helps the neural network for better performance. During training, it has more choice for solving the task. It can either convolve the input or pool it immediately. There are many inception modules present in final architecture and they are arranged in sequential order. Comparing with Inception Network, the training is slightly dissimilar to GoogleNet. GoogleNet trains the neural network quickly than Visual Graphics Group. When compared to VGG, the scope of the pre-trained GoogleNet is compact. The size of the GoogleNet is 96MB whereas VGG is 500MB. To extract the characteristics at various sizes from the input, they have used 1x1 , 3x3, and 5x5 kernels for each cell. Convolutions with high spatial kernels like 5x5 and 7x7 are used to extract high level features present in the network but the drawback of the model is intensive computations. To improve the overall performance of a network, 2 auxiliary outputs are connected throughout the neural network. It was found that primary auxiliary output had no observable effect on the end quality of neural network. The additional auxiliary outputs discovered the performance of the system. Without using an auxiliary branch converging results in better performance in same neural network architecture.Version-1 of GoogleNet took 5 million features whereas version-3 consists of 23 million features to train the neural networks. In dense connection architecture, every output neuron is connected with every input neuron. GoogleNet found that the activation in a deep network are either redundant or unwanted because of correlations between them. The effective architecture of deep neural networks has a sparse connection between the activations. GoogleNet won ILSVRC competition in 2014 from Google. It successfully brought out top 5 flaws at the rate of 6.67%. In

ImageNet competition, and it attained 93.3% top-5 accuracy.

Pseudocode for Google Net

```
//It detect and classify the objects
//Input: It can be a video or image
//Output: Detecting object using bounding box
1:  c ← a.shape
2:  half ← self.na // 2
3:  sqr1 ← L.sqr(c)
4:  more_channel ← L.alloc(0., b, ch + 2*half+c)
5:  sqr1 ← L.set_subtensor(more_channel,sqr1)
6:  size ← self.k
7:  alpha ← self.alpha / self.na
8:  for i in range(self.na):
9:  size← alpha * sqr(na)
10: size ← size ** self.na
11:  y ← y / size
12:  return y
```

C. You Only Look Once (YOLO)

YOLO is used for object detection. The initial step in YOLO is to generate bounding boxes by splitting the image and then it executes the YOLO algorithm for every boxes to recognize the object class. After finding the classes, it will merge all the boxes to form a best bounding box around the object. For real time detection, all these processes are done in parallel which helps to detect the objects in real time. It can able to process 45 frames per second. The advantage of the YOLO is that it is applicable for real time. Advanced version of YOLO is called Fast YOLO, it processes 155 images per second but it is not able to produce accurate result when compared to YOLO method. The limitation of YOLO is, each pixel predicts 2 boxes and have 1 class. YOLO is unable to detect small objects that occur in the groups.

Pseudocode for YOLO

```
//It detects and classifies the objects
//Input: It can be a video or image
//Output: Detecting object using bounding box
1:  for out1 to outs1
2:    for detection1 to out1
3:        mark ← detection1[6:]
4:        class_uniq_id ← np.argmax(mark)
5:        conf ← scores[class_id]
6:     if conf > 0.5
7:          x ← int(detection[0] * Width)
8:          y ← int(detection[1] * Height)
9:          w ← int(detection[2] * Width)
10:         h ← int(detection[3] * Height)
11:         x1 ← x - w / 2
12:         y1 ← y - h / 2
13:         class2.append(class_uniq_id)
14:         conf.append(float(conf))
15:         boxes.append([x, y, w, h])
```

D. ResNet

Residual Networks is the short form of ResNet. It contains multiple succeeding residual modules, which are primary structure of this architecture. It has two choices, either it can ignore step or it can implement a group of functions. Now it is equivalent to GoogleNet because the residual elements are arranged in sequential order to form a complete network. It consists of 1000s of residual layers which are used to design and train the network. Deep Residual Networks or ResNet become more familiar in 2015, which is extensively used by many AI researchers and data scientists. CNN is used for visual recognition and image classification. The tasks become too complex. It is difficult to train a neural network so deep layers are needed to enhance and compute the accuracy of the system. ResNet provides solution to this issue. The intensity of a ResNet can change – image classification issue had 152 layers which was introduced by Microsoft researchers. It took 25 million parameters to train the neural networks. The result of the ResNet is accurate and needs less weight than RNN and LSTM. Deeper networks are able to learn complex task which leads to high performance. The drawback of ResNet is if the layers in Residual Network are more deep then it is hard to detect the flaws and it is unable to propagate back correctly and quickly. If the layers are narrowed, it is not capable for efficient learning and it leads to low performance.

Pseudocode for ResNet

```
//It detect and classify the objects
//Input: It can be a video or image
//Output: Detecting object using bounding box
1:   if in_tensor is None
2:      input = Input(shape ← input)
3:   else
4:         if not K.is_keras_tensor(input):
5:              input ← Input(tensor←input_tensor,
                shape←input_shape)
6:          else
                input ← input_tensor
7:    if K.image_data_format() == 'channels_last'
             bn_axis ← 3
```

```
8:  else:
9:      bn_axis ← 1
```

E. Recurrent Neural Networks

RNN is a neural network used in the field of Natural Language Processing because of its favourable outcome. The applications of RNN contains two approaches, either it can predict the text and correct the flaw in the prediction period or need to train a system in particular domain to produce equivalent text. R represent recurrent because the same task is performed for every individual element in a continuous order, with the dependent output on preceding computations.

- Deep RNN – This kind of RNN has more number of layers which produces accurate results.
- Bidirectional RNN – In this type, the output of the RNN depends not only on previous but also the succeeding channel.

The advantage of RNN is that it shares the similar parameters to all stages. This minimizes the number of features that require to learn. With the help of CNN, it can able to generate exact details of an image which is unlabeled. The drawback of RNN is that it is not suitable to track long- standing dependencies like paragraphs or long sentences. RNN is unable to arrange into deeper system because of activation functions used in RNN which makes the gradient deteriorates over multiple layers.

Pseudocode for RNN

```
//It detect and classify the objects
//Input: It can be a video or image
//Output: Detecting object using bounding box

1:  tot_training_length ← 50
2:  for seq1 to sequences1
3:    for i to range(tot_training_length,len(seq1))
4:        extract ← seq[i - training_length:i + 1]
5:        features.append(extract[:-1])
6:        labels.append(extract[-1])
7:  features ← np.array(features)
```

Comparison between Architectures

Architecture	**Frames / Sec**	**Memory Size**	**Advantages**	**Limitations**
VGG	20	528 MB	Pre-trained network	Training consumes more time
Google Net	36	96 MB	Faster to train	--
YOLO	45	More than 500 MB	Accurate Result	Faster version unable to produce accurate result
ResNet	50	More than 2 GB	Accurate result and less weight	The layers in Residual Network are more deep then it is hard to detect the flaws.
RNN	17	More than 400 MB	It shares the similar parameters to all stages. This minimizes the number of features that requires to learn.	RNN is not suitable to track long-standing dependencies like paragraphs or long sentences.

IV. SUMMARY

Object detection, recognition and classification are powered by deep neural architectures. There are several architectures are available such as VGGNet, GoogleNet, YOLO, ResNet and RNN. The architectures are differs from each others in terms of parameters such as activation functions, filters, number of layers and processing speed.

CONCLUSION

Deep learning architectures helps in improving the performance of computer vision tasks. In this paper, we have surveyed various deep learning architectures such as VGGNet, GoogleNet, YOLO, ResNet and RNN. From this we like to conclude that, GoogleNet is most preferable architecture because it generates accurate result by processing 36 frames/sec and it consumes less memory (96MB) while comparing with other architectures.

REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). CVIU, 110(3):346–359, 2008.

[2] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

[3] Moving Object Detection and Locating Based on Region Shrinking Algorithm, Zhihui Li, Haibo Liu and Di Sun, IEEE 2012.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[6] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. (Dec. 2015). ``Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385

[8] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, ``Deep fully-connected networks for video compressive sensing," Digit. Signal Process., vol. 72, pp. 9_18, Jan. 2018.

[9] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, ``Long short term memory recurrent neural network classifier for intrusion detection," in Proc. Int. Conf. Platform Technol. Service (PlatCon), Feb. 2016, pp. 1_5.