International Journal on Applications in Information and Communication Engineering
Volume 9 : Issue 3 : August 2023, pp 25– 29 www.aetsjournal.com ISSN (Online) : 2394 - 6237

An Analysis of Convolutional Neural Networks and its applications

Dr.R.Pushpalakshmi
Professor & Head
Department of Cyber Security
PSNA College of Engineering and Technology
India
Mrs. M. S. Vinu
Assistant Professor ,
Department of Computer Science and Engineering
Nehru Institute of Engineering and Technology
India

Abstract— Convolutional Neural Network (CNN) is one of the most significant networks in the deep learning field. Since CNN made impressive achievements in many areas, including but not limited to computer vision and natural language processing, it attracted much attention both of industry and academia in the past few years. The existing reviews mainly focus on the applications of CNN in different scenarios without considering CNN from a general perspective, and some novel ideas proposed recently are not covered. In this review, we aim to provide novel ideas and prospects in this fast-growing field as much as possible. Besides, not only two-dimensional convolution but also one-dimensional and multi-dimensional ones are involved. First, this review starts with a brief introduction to the history of CNN. Second, we provide an overview of CNN. Third, classic and advanced CNN models are introduced, especially those key points making them reach state-of-the-art results. Fourth, through experimental analysis, we draw some conclusions and provide several rules of thumb for function selection. Fifth, the applications of onedimensional, two-dimensional, and multi-dimensional convolution are covered. Finally, some open issues and promising directions for CNN are discussed to serve as guidelines for future work.

Keywords— Deep learning, convolutional neural networks, deep neural networks, computer vision etc

I. Introduction

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data.

When it comes to Machine Learning, Artificial Neural Networks perform really well. Neural Networks are used in various datasets like images, audio, and text. Different types of Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural networks. In this blog, we are going to build a basic building block for CNN.

In a regular Neural Network there are three types of layers:

Input Layers: It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).

Hidden Layer: The input from the Input layer is then fed into the hidden layer. There can be many hidden layers depending on our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of the output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.

Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

The data is fed into the model and output from each layer is obtained from the above step is called feedforward, we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we backpropagate into the model by calculating the derivatives. This step is called Backpropagation which basically is used to minimize the loss.

II. Convolution Neural Network

Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominantly used to extract the feature from the grid-like matrix dataset. For example visual datasets like images or videos where data patterns play an extensive role.

CNN architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

CNN architecture

Simple CNN architecture

The Convolutional layer applies filters to the input image to extract features, the Pooling layer downsamples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

How Convolutional Layers works

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (i.e the channel as images generally have red, green, and blue channels).

Image Channel - Geeksforgeeks

Now imagine taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different widths, heights, and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called Convolution. If the patch size is the same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weights.

Convolution operation in CNN

Image source: Deep Learning Udacity

Now let’s talk about a bit of mathematics that is involved in the whole convolution process.

Convolution layers consist of a set of learnable filters (or kernels) having small widths and heights and the same depth as that of input volume (3 if the input layer is image input).

For example, if we have to run convolution on an image with dimensions 34x34x3. The possible size of filters can be axax3, where ‘a’ can be anything like 3, 5, or 7 but smaller as compared to the image dimension.

During the forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have a value of 2, 3, or even 4 for high-dimensional images) and compute the dot product between the kernel weights and patch from input volume.

As we slide our filters we’ll get a 2-D output for each filter and we’ll stack them together as a result, we’ll get output volume having a depth equal to the number of filters. The network will learn all the filters.

Layers used to build ConvNets

A complete Convolution Neural Networks architecture is also known as covnets. A covnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

III. Types of layers: datasets

Let’s take an example by running a covnets on of image of dimension 32 x 32 x 3.

Input Layers: It’s the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3.

Convolutional Layers: This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually 2×2, 3×3, or 5×5 shape. it slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred ad feature maps. Suppose we use a total of 12 filters for this layer we’ll get an output volume of dimension 32 x 32 x 12.

Activation Layer: By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimensions 32 x 32 x 12.

Pooling layer: This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.

Flattening: The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.

Fully Connected Layers: It takes the input from the previous layer and computes the final classification or regression task.

Image source: cs231n.stanford.edu

Output Layer: The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.

Artificial Intelligence has been witnessing monumental growth in bridging the gap between the capabilities of humans and machines. Researchers and enthusiasts alike, work on numerous aspects of the field to make amazing things happen. One of many such areas is the domain of Computer Vision.

The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner, and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning have been constructed and perfected with time, primarily over one particular algorithm — a Convolutional Neural Network.

Ready to try out your own convolutional neural nets? Check out Saturn Cloud for free compute (including free GPUs).

IV. A CNN SEQUENCE TO CLASSIFY HANDWRITTEN DIGITS

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

Why ConvNets over Feed-Forward Neural Nets?

Flattening of a 3x3 image matrix into a 9x1 vector

An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really.

In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and the reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Input Image

4x4x3 RGB Image

In the figure, we have an RGB image that has been separated by its three color planes — Red, Green, and Blue. There are a number of such color spaces in which images exist — Grayscale, RGB, HSV, CMYK, etc.

You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (7680×4320). The role of ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction. This is important when we are to design an architecture that is not only good at learning features but also scalable to massive datasets.

V. CONVOLUTION LAYER — THE KERNEL

Convoluting a 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)

In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in color yellow. We have selected K as a 3x3x1 matrix.

Kernel/Filter, K =

1 0 1
0 1 0
1 0 1

The Kernel shifts 9 times because of Stride Length = 1 (Non-Strided), every time performing an elementwise multiplication operation (Hadamard Product) between K and the portion P of the image over which the kernel is hovering.

VI. MOVEMENT OF THE KERNEL

The filter moves to the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed.

Convolution operation on a MxNx3 image matrix with a 3x3x3 Kernel

In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between Kn and In stack ([K1, I1]; [K2, I2]; [K3, I3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output.

Convolution Operation with Stride Length = 2

The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network that has a wholesome understanding of images in the dataset, similar to how we would.

There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying Valid Padding in the case of the former, or Same Padding in the case of the latter.

SAME padding: 5x5x1 image is padded with 0s to create a 6x6x1 image

When we augment the 5x5x1 image into a 6x6x1 image and then apply the 3x3x1 kernel over it, we find that the convolved matrix turns out to be of dimensions 5x5x1. Hence the name — Same Padding.

On the other hand, if we perform the same operation without padding, we are presented with a matrix that has dimensions of the Kernel (3x3x1) itself — Valid Padding.

The following repository houses many such GIFs which would help you get a better understanding of how Padding and Stride Length work together to achieve results relevant to our needs.

vdumoulin/conv_arithmetic

A technical report on convolution arithmetic in the context of deep learning - vdumoulin/conv_arithmetic

github.com

VII. Pooling Layer

3x3 pooling over 5x5 convolved feature

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise-suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

Types of Pooling

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-level details even further, but at the cost of more computational power.

After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

Classification — Fully Connected Layer (FC Layer)

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation is applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

There are various architectures of CNNs available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future.

VIII. Conclusion

Due to the advantages of convolutional neural networks, such as local connection, weight sharing, and down-sampling dimensionality reduction, they have been widely deployed in both research and industry projects. This paper provides a detailed survey on CNN, including common building blocks, classic networks, related functions, applications, and prospects. First, we discuss basic building blocks of CNN and present how to construct a CNN-based model from scratch. Second, some excellent networks are expounded. From them, we obtain some guidelines for devising novel networks from the perspective of accuracy and speed. More specifically, in terms of accuracy, deeper and wider neural structures are able to learn better representation than shallow ones. Besides,residual connections can be leveraged to build extremely deep neural networks, which can increase the ability to handle complex tasks. In terms of speed, dimension reduction and lowrank approximation are very handy tools. Third, we introduce activation functions, loss functions, and optimizers for CNN. Through experimental analysis, several conclusions are reached. Also, we offer some rules of thumb for the selection of these functions. Fourth, we discuss some typical applications of CNN.Different dimensional convolutions should be designed for various problems. Other than the most frequently-used twodimensional CNN used for image-related tasks, onedimensional and multi-dimensional CNN are harnessed in lots of scenarios as well. Finally, even though convolutions possess many benefits and have been widely used, we reckon that it can be refined further in terms of model size, security, and easy hyperparameters selection. Moreover, there are lots of problems that convolution is hard to handle, such as low generalization ability, lack of equivariance, and poor crowded-scene results, so that several promising directions are pointed.

IX. References

[1] W. S. Mcculloch, and W. H. Pitts, "A logical Calculus of Ideas Immanent in Nervous Activity," The Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1942.

[2] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review, pp 368-408, 1958.

[3] C. V. D. Malsburg, "Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms."

[4] Davd. Rumhar, Geoffrey. Hinton, and RonadJ. Wams, "Learning representations by back-propagating errors."

[5] A. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," IEEE Transactions on Acoustics Speech & Signal Processing, vol. 37, no. 3, pp. 328-339, 1989.

[6] W. Zhang, "Shift-invariant pattern recognition neural network and its optical architecture," in Proceedings of annual conference of the Japan Society of Applied Physics, 1988.

[7] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," Neural Computation, vol. 1, no. 4, pp. 541- 551.

[8] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," Physics Letters A, vol. 144, no. 6-7, pp. 333-340.

[9] Specht, and D.F., "A general regression neural network," IEEE Transactions on Neural Networks, vol. 2, no. 6, pp. 568-576.

[10] B. L. Lecun Y , Bengio Y , et al., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in neural information processing systems, vol. 25, no. 2, 2012.

[12] N. Aloysius, and M. Geetha, "A review on deep convolutional neural networks," Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017. pp. 588-592.

[13] A. Dhillon, and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," Progress in Artificial Intelligence, 2019/12/20, 2019.

[14] W. Rawat, and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," Neural Computation, pp. 1-98.

[15] Q. Liu, N. Zhang, W. Yang, S. Wang, Z. Cui, X. Chen, and L. Chen, "A Review of Image Recognition with Deep Convolutional Neural Network."

[16] S. Rehman, H. Ajmal, U. Farooq, Q. U. Ain, and A. Hassan, "Convolutional neural network based image segmentation: a review."