Carolyn McAughren
558 755 906
Instructor, Professor Huizhu Liu

November 8, 2021

Convolutional Neural Networks

CSCI 479 Advanced Topics in Artificial Intelligence: Term Paper

Introduction

The Convolutional Neural Network is a class of Neural Networks designed to analyze visual imagery (Bhardwaj, et al., 2018, p. 94). The traditional, fully connected, Neural Network is not ideal for Computer Vision tasks as it doesn't "scale well for visual information mining" (Zemmari & Benois-Pineau, 2020, p. 49). In a Neural Network, each neuron in each layer is connected to each neuron in the previous layer, and each of those connections have a weight which must be learned during training. To process an image, each neuron in the first layer will be connected to each pixel. If we consider, for example, a small colored image of 200 pixels by 200 pixels, one neuron in the first layer will have 120,000 weights to learn. The number of weights quickly grows to an unmanageable amount (Zemmari & Benois-Pineau, 2020, p. 49).

Another issue with using a traditional Neural Network for image processing, other than the scalability issues, is that when you feed an image into a Neural Network, the pixels of the image are flattened into a vector. This means the system cannot derive any spatial structure and can have no knowledge of how the pixels are arranged. This is a missed opportunity, because in vision and images, pixels which are closer together, similar in color or grouped into some shape are more likely to be of the same object (Mohit, et al., 2018, p. 33).

Background

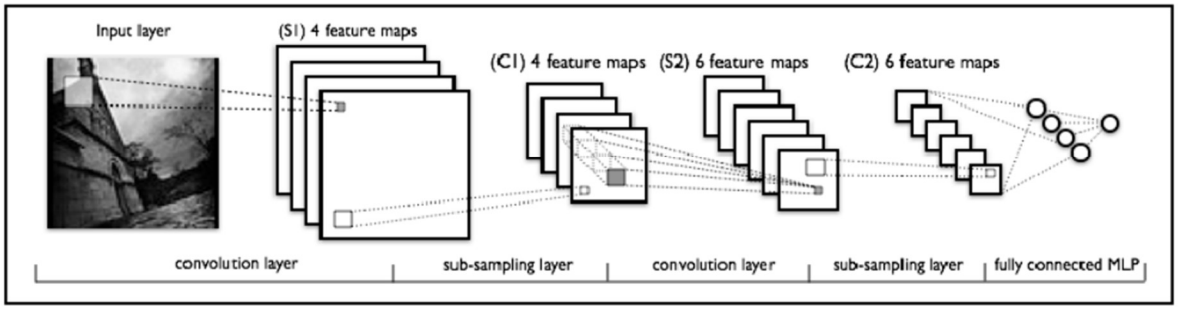
The convolutional neural network was designed with computer vision tasks in mind. It was first introduced by Kunihiko Fukushima in 1980. His "Neocognitron" introduced a neural network with additional layers, and it was designed to learn handwritten digits (Bhardwaj, et al., 2018, p. 92). He drew inspiration from research done by Hubel and Wiesel, who published data about the organization of the animal visual cortex (Zemmari & Benois-Pineau, 2020, p. 50). Individual cortical neurons are "sensitive only to limited parts of the global visual field" (Zemmari & Benois-Pineau, 2020, p. 50). Those small portions of the visual field are called receptive fields and they partially overlap to cover the entire visual field. Fukushima mimicked this is his design, but it wasn't until later research into backpropagation by Geoffrey Hinton in 1986 that the CNN as we know it was made possible (Bhardwaj, et al., 2018, p. 93). LeNet, designed to recognize handwritten digits on cheques (LeCun, 1989), is commonly known as the first Convolutional Neural Network. The initial draft was developed in 1989 by Yann LeCun, the first to apply backpropagation to a practical application. The CNN didn't gain its world fame until a decade later though, when Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton from the University of Toronto won the 2012 ImageNet competition with their convolutional neural network "AlexNet" (Zemmari & Benois-Pineau, 2020, p. 55).

ImageNet is a prestigious yearly Computer Vision competition, began in year 2007 by Princeton. Competitors design algorithms for object detection and image classification, and in 2012, AlexNet won by unprecedented margin and changed the face of Computer Vision forever. AlexNet used 5 convolutional layers, 3 max pooling layers, and 3 fully-connected layers – they won with an error margin of 15.3% - more than 10.8% percentage points lower than the runner up (Zemmari & Benois-Pineau 2020, p. 55). Since then, the convolutional neural network has

been the base of most mainstream image processing algorithms, and most competitors in ImageNet use some unique variation of it (Bhardwaj, et al., 2018, p. 28).

Overview

The convolutional Neural network was the solution to the scalability and spatial data issues found when using a traditional Neural Network for image processing. The most obvious difference from a traditional Neural Network and the Convolutional Neural Network is that the Neurons of a CNN are arranged in "three dimensions (height, width and depth) so as to match the geometric shape of data (images can be seen as cuboids of pixels)" (Zemmari & Benois-Pineau, 2020, p. 49). A color image will be considered 3D as it has a depth of 3, but a grey scale image is considered 2D as its depth is only 1. The second large difference about the CNN is that, unlike a traditional Neural Network which is Fully-Connected, most layers of the CNN are only locally connected (Zemmari & Benois-Pineau, 2020, p. 50). This reduces the number of weights which need to be trained significantly, and it also allows for spatial relationships between pixels in an image to be retained- something that was impossible with a traditional neural network.



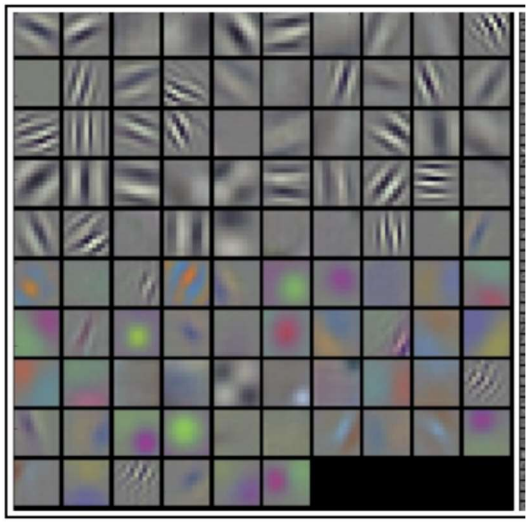
The Layers of a Convolutional Neural Network (Bhardwaj, et al., 2018, p. 98)

The Convolutional Neural Network is made up of 3 main components: The Convolutional Layers, which detect regional patterns in an image (Mohit, et al., p. 41). The Pooling Layers, which come after the Convolutional Layers and help reduce dimensionality (Mohit, et al., p. 41), and the Fully-Connected Layers which come last, and essentially function as a traditional Neural Network does (IBM Cloud Education, 2020, para 3). There are usually multiple iterations of paired Convolution and Pooling Layers, but the Fully-Connected Layers always come at the end.

Convolutional Layer

The Convolutional Layers look for spatial connections between pixels in the image. The process is like sliding a small window over the input image and considering only a portion of its entire data at a time – Instead of considering the entire image at once, we process it in smaller pieces, similar to how the cortical neurons for the eye work. The window is called a 'filter', and it is a weighted, trainable matrix (Zemmari & Benois-Pineau, 2020, pp. 50-51). Each Convolutional Layer has several different filters to run over the image, and each filter will be ran over each layer of the image separately. The separate versions of a certain filter for each of those layers are called kernels. In the first Convolutional layer, with the original image there will be 3 layers for

an RBG image, so there will be 3 kernels for each filter, one for each layer. These kernels are not nessecarily identical, the weights of their matrices may be different to pick out differences in the separate color channels. They will be initially set to some values based on the Programmers research and discretion, but they will be trained by the system using backpropagation to improve (Bhardwaj, et al., 2018, p. 99).

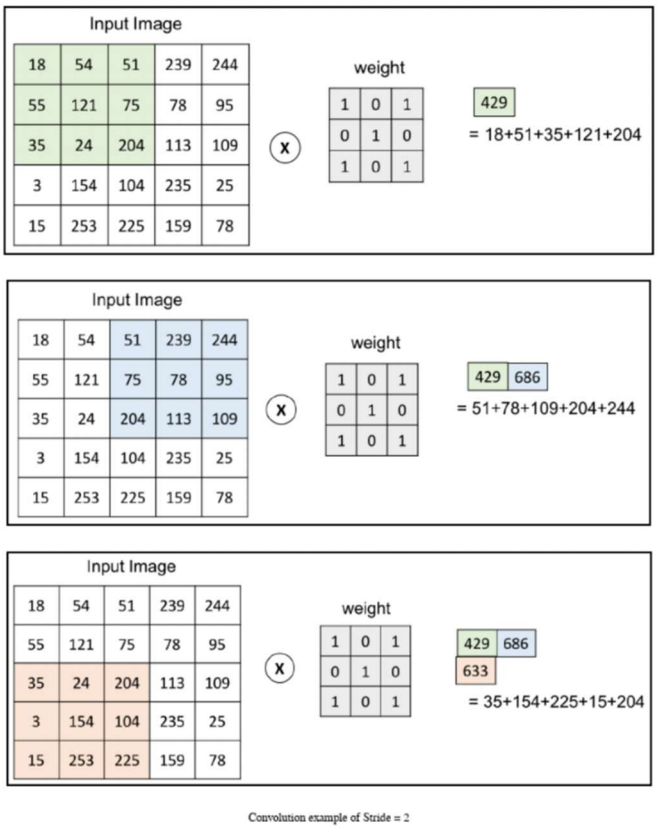


Filter examples, visualized. (Bhardwaj, et al., 2018, p. 108)

In the earlier Convolutions, the filters are designed to pick up low level spatial connections between pixels- they look for edges, similiar colors or shapes. In later convolutions, the filters will look for more complex features, like eyes on a face or tires on a car (IBM Cloud Education, 2020, para. 3). The filter slides from left to right and top to bottom, and the number of pixels that they slide over for each movement is called the Stride.

We can control the behavior of the Convolutional Layer by specifying the size and number of these filters. To increase the number of neurons in a layer, we can increase the number of filters, and to look for larger patterns in the image we can make the filter windows larger (Mohit, et al., p. 40). The number of filters, their size, and the stride length used are hyper parameters, so they are not trained by the system and are a design decision for the programmers (IBM Cloud Education, 2020, para. 7).

Whatever the filter format, at each movement of a kernel as it passes across a layer of the image, a dot multiplication is performed between the kernel and the section of the image it is currently processing. Each pixel value in the kernel is multiplied by the corresponding pixel values found in the original image, and then they are added together to get one value for that movement of the kernel on that layer (Bhardwaj, et al., 2018, pp. 72-73). The resulting value from that movement is placed in its respective position in an output map, which has several names: ‘Activation Map’, ‘Convolved Feature’ or a ‘Feature Map’ (IBM Cloud Education, 2020, para. 5). Let us call it the Convolved Feature going forward for consistency.



Filter movements (Bhardwaj, et al., 2018, p.73)

There will be one Convolved Feature per filter, so the values for the kernels of that filter (which run over each layer of the image) are summed together along with a bias (also trainable) to produce one Convolved Feature per filter (Bhardwaj, et al., 2018, p. 99). There are many filters per Convolutional Layer, so there will be many Convolved Features. We can think of each Convolved Feature as a neuron in a Convolutional Layer. These Convolved Features in the Convolutional Layer will be smaller than the original image was, by a factor of the stride length. For a stride of 2, the Convolved Feature will be half the size of the original image, and as we proceed along and do more Convolutional Layers, the size will get smaller still (Mohit, et al., 2018, p. 40). Results are more accurate if the size of the image is preserved, and this can be accomplished by padding the original image. A common padding technique is same-padding, adding a row of 0's along the border of an image, but there are other techniques (Thomas, 2019, para. 7).

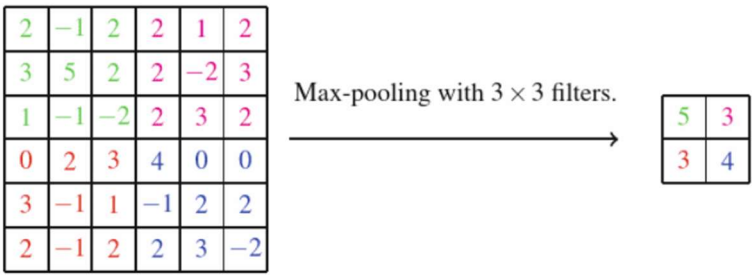
These Convolved Features are stored stacked together, so the more Filters the deeper the output of each Convolutional Layer, and again with each subsequent Convolutional Layer after that. (Zemmari & Benois-Pineau, 2020, pp. 50-51)

After a Convolution Layer completes, a non-linearity activation function is applied to each Convolved Feature. There are many activation functions, but the most common is the Rectified Linear Unit (ReLU) as it has been found to be superior over the competition. The ReLU activation function is applied to each node output in the Convolved Feature, and it essentially

sets any value below 0 to 0 (Bhardwaj, et al., 2018, p. 64). The activation function is the same as what is seen in a traditional Neural Network- if the value is over the threshold (0 in this case) the neuron will fire, otherwise it will be turned off.

Pooling Layer

The Pooling Layer comes after a Convolutional + ReLU layer pair, or sometimes multiple Convolutional + ReLU layers will repeat a few times before a Pooling Layer is performed, depending on the specific Convolutional Neural Network algorithm used. The Pooling Layer helps reduce dimensionality of the Convolved Features (Mohit, et al., 2018, p. 41), which reduces the computational complexity for the upper layers. The Pooling Layer summaries the outputs of neighboring groups of nodes in the Convolved Features, allowing the overall size of that Convolved Feature to be shrank, meaning there are less parameters to be learned (Zemmari & Benois-Pineau, 2020, pp. 52-53).



An illustration of a max-pooling operation: the size of the image is reduced

(Zemmari & Benois-Pineau, 2020, p. 53)

There are mutliple ways to Pool, but the most effective and most common method is “Max Pooling”. To conduct Max Pooling, another window-like kernel will be used, with some size and stride determined by the programmer. It slides across each Convolved Feature as it does in the Convolutional Layers, but it performs a different operation at each movement across the layer. Max pooling selects the largest value in each window to save and ignores the rest. Another Pooling method called Average Pooling takes the average of all the values in the kernel for that movement, but it is less effective that taking the Maximum (Zemmari & Benois-Pineau, 2020, p. 53). Max Pooling also serves to eliminate the noisy activations, which is why it performs better than Average Pooling. By discarding so many of the activations the system is more effective and helps to avoid overfitting (Bhardwaj, et al., 2018, p. 100). We must be careful not to Pool too often however, because we are discarding information from the image, and it could affect results of the classification if done too much.

Connectivity between the layers

Earlier it was mentioned that there would be 3 kernels per filter for the first Convolutional Layer if we have a color image- one for the Red, Blue, and Green layers of the initial image. In later Convolution layers, the output of previous layers will be fed into the new Convolutional Layer,

so each filter could have more than 3 kernels- it will have one for each Convolved Feature from the previous layer. The number of filters for each Convolutional Layer can be different from the other layers, but each filter will have a kernel to cover each layer of the depth of the Convolved Feature from the previous layer. The number of filters and their sizes in each layer are design decisions which need to be made while constructing the network (Thomas, 2019, para. 15).

Fully-Connected Layer

After however many iterations of Convolutional and Pooling Layers, the final layers of the CNN come in to classify the image. First, the last Convolved Feature with its many layers is flattened into 1D vector. It is then fed into a Fully-Connected (Dense) Layer Neural Network to classify the image using softmax function (Bhardwaj, et al., 2018, p. 65). The softmax function outputs a probability from 0 to 1 for each of the classification labels the model is trying to predict. The outputted results are then compared against the known results for the training data: based on the success or failure, the weights for the neurons and biases of the Fully-Connected layers, the weights of the filters (the values of the matrices for their kernels) and the biases used from the Convolutional Layers are all tweaked based on feedback from back-propogation (IBM Cloud Education, 2020, para. 9).

Conclusion

The Convolutional Neural Network is the leading algorithm in computer vision for object detection. It consists of some pattern of Convolutional Layers (likely paired with the ReLU activation function), Pooling Layers (likely Max-Pooling) and Fully-Connected Layers to classify the image. A Convolutional Neural Network is superior for image classification because it retains spatial information relevant to the classification of objects along the depth of the image as it is passed through the network, but it eliminates unnecessary activation noise from the data through Pooling. With less parameters, the system can be deeper without being significantly more computationally expensive.

Most leaders in Computer Vision use some form of the Convolutional Neural Network in their architecture, but the technology is not perfected yet. It can identify objects but not the nuance as a human can, it can fail to detect images under new lighting or new angles and can be prone to overfitting (as is a traditional Neural Network). But with every year, we see new versions of the CNN being designed, and its performance it only improving.

References

Bhardwaj, A., Di, W., & Wei, J. (2018, January 30). *Deep learning essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing, Limited.

IBM Cloud Education. (2020, October 20). *Convolutional Neural Networks.* IBM. https://www.ibm.com/cloud/learn/convolutional-neural-networks

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*, 1097-1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D. (1989, December). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation, 1*(4), 541-551. doi: 10.1162/neco.1989.1.4.541.

Mohit Sewak, Md. Rezaul Karim, & Pradeep Pujari. (2018). *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python.* Packt Publishing.

Thomas, C. (2019, May 27). *An introduction to Convolutional Neural Networks.* Towards Data Science. https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7

Zemmari, A., & Benois-Pineau, J. (2020, January 23). *Deep learning in mining of visual content.* Springer International Publishing AG.