Semantic Segmentation using Convolutional Neural Network

|  |  |
| --- | --- |
| Eashwar Sathyamurthy | Dr. Mohammed Charifa |
| Robotics Engineer | Supervising Faculty |
| A. James Clark School of Engineering | Maryland Applied Graduate Engineering |
| University of Maryland | University of Maryland |
| College Park, USA | College Park, USA |
| eashwar@umd.edu | scharifa@umd.edu |

*Abstract*—Human beings are easily able to identify and differentiate objects based of their cognitive skills. Cognitive skills like category formation, pattern recognition, working memory etc helps to segment and process visual information obtained through eyes. A machine process images in the form of matrix containing whole numbers which are known as pixels. Each pixel indicates the brightness levels at that particular location. As the machine does not posses cognitive capability to segment images, a convolution neural net is designed and built through which the machine learns to segment images. The segmentation performed is called semantic segmentation as it groups every pixel of the image into classes based on labels. The report explains in detail Convolution Neural Net architecture and how it can be used for semantic segmentation.

1. INTRODUCTION

Over the years convolution neural net (CNN) has increased its horizon and made significant impact on pattern recognition and machine learning fields. CNN is one type of neural network which has one input layer, one output layer and two or more hidden layers. The hidden layers is a combination of convolution layers, pooling layers, activation functions and fully connected layers. The main distinction between a CNN and a regular neural net is that in CNN neurons handles input and output in three dimensions. An example of CNN is shown in the figure below[1]:

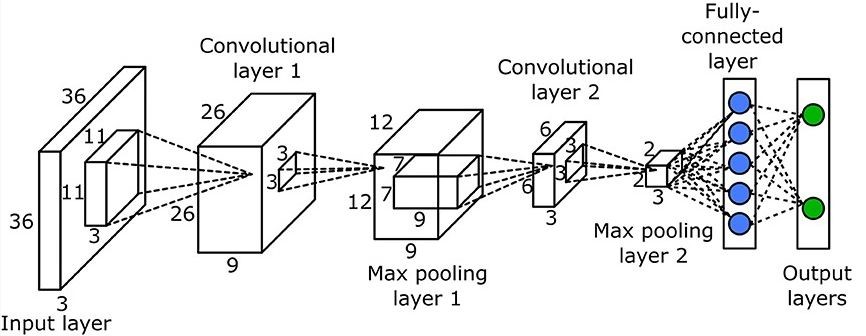


Figure 1: CNN Example

1. DEFINITIONS

In this section, different layers of a CNN are explained in detail.

1. *Input Layer*

The input to the input layer of the CNN is generally a color image in pixel form which is a three dimensional matrix. The dimensions are length, width and number of channels (R,G,B) of the input image. Generally, the input image is resized to the desired dimensions accepted by the CNN.

1. *Convolutional Layer*

In the convolutional layer, convolution operation happens between the kernel and the input image. Kernel is a 2D array whose elements are obtained from Gaussian distribution. An example of 3x3 Guassian Kernel is given below[2]:

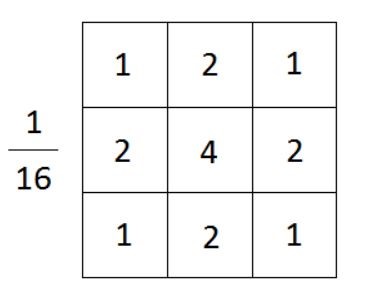
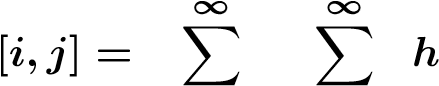


Figure 2: 3x3 Guassian kernel

Mathematically, 2D convolution is represented as follows:

*y***[***m,n***]***.x***[***i − m,j − n***]**

*m***=***−∞n***=***−∞*

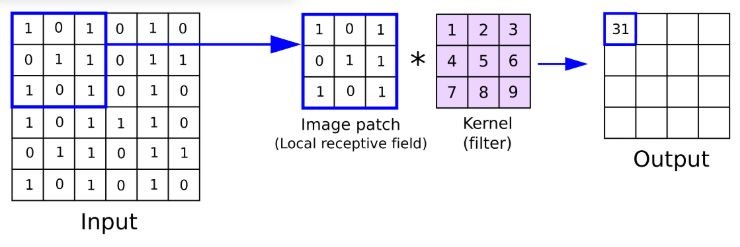
In convolution layer, the convolution operation is performed on all the pixels by sliding the kernel over the input image. The following image shows the convolution operation between the image and the kernel[3]:

Figure 3: Convolving kernel over the image

Typically, in most neural networks the dimensions of the image after performing convolution are maintained same. In order to do that it is important to understand how convolution operation affects the dimensions of the image. The relation is given as follows:

# *IW − KW* + 2 *∗ P*

*OutputWidth* **= + 1** *S*

where *IW* is the image width

*KW* is the kernel width

P is number of padding bits

S = stride = number of pixels shift over the input image

# *IH− KH* + 2 *∗ P*

*OutputHeight* **= + 1** *S*

where *IW* is the image width

*KW* is the kernel width

P is number of padding bits

*C. Pooling Layers:*

Pooling layers in a CNN is added after the convolutional layer. It reduces the size of the image by retaining only the important features in the image and neglecting the rest. Pooling layers have the following advantages:

1. It reduces the number of parameters to train in a CNN.
2. It prevents the neural net from over-fitting problem. Over fitting is a situation where the neural net models the training data very well but fails to generalize the results on the testing data. This usually occurs due large number of parameters which lets the neural net detect all the patterns in an image but fails to learn from the patterns to generalize on the testing image. This can be observed when the neural net give more than 95% accuracy in training data but gives less than 50% accuracy in testing data. Commonly used Pooling Layers: Average pooling and max pooling are the two most commonly used pooling layers in neural network.

a) Average Pooling: In average pooling, the average value is selected from each patch of the feature map to obtained a reduced map size. The following image gives an example of average pooling[4]:

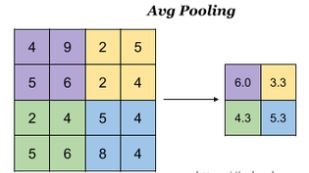
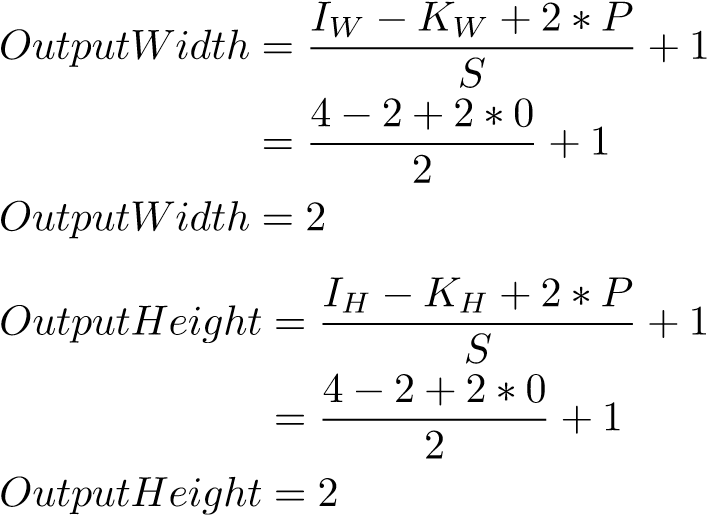


Figure 4: Average Pooling Example

In the above example, the feature map is of size 4x4 and the patch is of size 2x2 and stride = 2 and there are no padding bits. So, if we apply the formula for resulting dimension of the feature map, we get the following:



So, the output feature map will be of size 2x2. Let consider the first (violet) patch and calculate the average pooling output:

4+9+5+6

*Outputvalue* = = 6*.*0

4

This process is repeated over the entire feature map to obtain the reduced 2x2 feature map output.

b) Max Pooling: In max pooling, the maximum value is selected from each patch of the feature map to obtained a reduced map size. Max pooling is the most commonly used pooling layers. The following image gives an example of max pooling[4]:

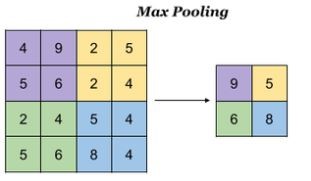


Figure 5: Max Pooling Example

As this is the same example used in average pooling, the output feature map size is 2x2. The following i s the max pooling output for the first (violet) patch of the feature map:

*Outputvalue* = *max*(4*,*9*,*5*,*6) = 9*.*

*D. Activation Functions:*

As the name suggests, an activation function decides whether to activate a neuron or not based on calculating weighted sum and adding bias with it. In the case of a CNN, weighted sum is the output image obtained by convolving the kernel over the input image. Bias is a fixed value which is added pixel wise to the obtained output image. Activation function is applied to this output image which decides whether to select a particular pixel value and pass on to the next layer or not. Need for Activation Functions: In neural networks, the weights (kernel in the case of CNN) and bias values constantly gets updated based on the error at the output. The values are updated using back-propagation algorithm. Activation functions makes the back propagation possible as the gradients are supplied along with errors to update the weight and bias values[5].

Different Activation Functions: There are different activation functions used in neural networks based on its applications. They are:

1) Step Function: Step function is a threshold based activation function which outputs 1 for the values greater than a specified threshold or outputs 0 otherwise. Hence, 1 means the neuron gets activated and 0 means neuron gets deactivated. The following

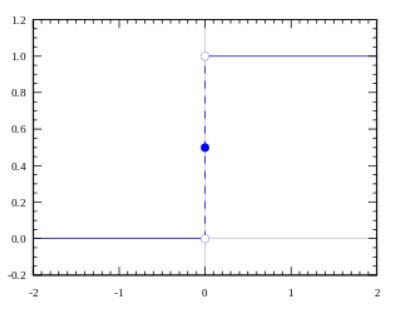


Figure 6: Step Function conditional equations defines step function:

Activation Function *A* = 1 if *y* ≥ *threshold*

= 0 otherwise

Step function can be used as an activation function if the neural net is being used for binary classification where the output is just ”YES” or ”NO”. Typical example of such classification is determining the picture is of a dog or a cat.

But, step functions cannot be used as activation functions for multiple(more than 2) classifications as the neural net might give multiple 1’s and 0’s as output.

2) Linear Function: Linear function means a straight line functions which is proportional to the input. The below figure shows the example: The following

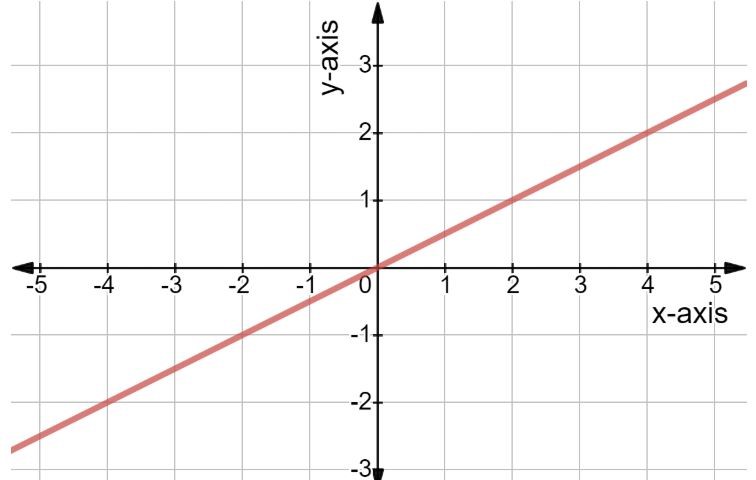


Figure 7: Linear Function

conditional equations defines linear function:

Activation Function *A* = *c* ∗ *X*

Linear function eliminates the multiple classification problem faced by employing step function as the activation function. As the activation function varies with input values, it can take many values not only 0 and 1. Hence, it can classify multiple classes. The main disadvantage of linear function is the derivative becomes constant. This means that the gradient no longer varies with input. Hence, if there was an error in detecting the output, since the gradient is constant there will be constant change in the weights and bias values irrespective of inputs. This means that the neural net fails to learn. Another disadvantage occurs due to linearity property. By, employing linear functions as activation functions to all hidden layers, the output will be a linear combination of hidden layers. Hence, multiple hidden layers can be combined into 1 hidden layer. Hence, by employing linear function only 1-hidden layer neural nets are possible.

3) Sigmoid Function: The below picture shows the sigmoid function.

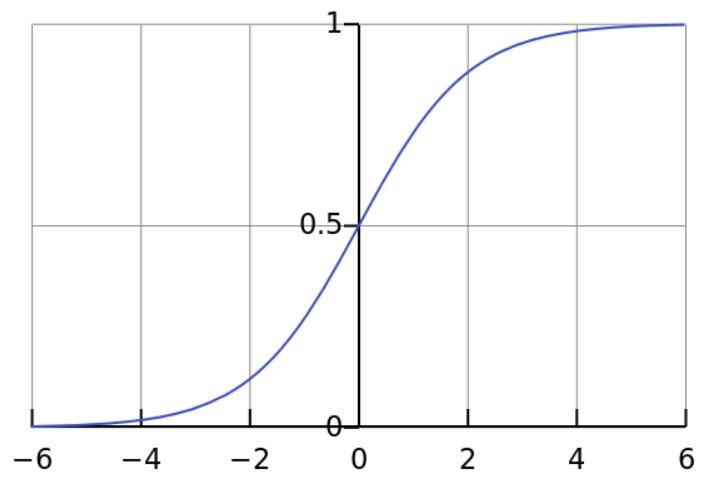
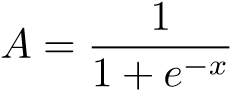


Figure 8: Linear Function

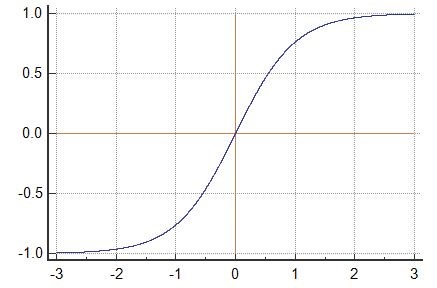
The following conditional equations defines sigmoid function:

Activation Function

The sigmoid function is essentially a smoothened version of step function. This makes it far more useful than step function. Firstly, sigmoid function is non-linear which eliminates the linearity problem. Secondly, it can take many values between 0 and 1 which makes it useful in multiple classification problems.

The only disadvantage of sigmoid function is that gradient of the points farther from origin approaches to zero which gives rise to vanishing gradients problem. This makes the neural net to learn very slowly and in worst case scenarios not learn at all.

1. Tanh Function: The below picture shows the tanh function.

from https://indoml.com/2018/03/07/student-notesconvolutional-neural-networks-cnn-introduction/

1. Sakshi TiwariCheck out this Author’s contributed articles., Sakshi Tiwari, amp; Check out this Author’s contributed articles. (2018, February 06). Activation functions in Neural Networks. Retrieved June 26, 2020, from

https://www.geeksforgeeks.org/activation-functionsneural-networks/

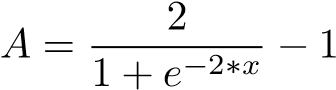
1. Intro to Linear vs. Nonlinear Functions. (n.d.). Retrieved June 26, 2020, from

https://www.expii.com/t/intro-to-linear-vs-

Figure 9: Tanh Function

nonlinear-functions-4318

The following conditional equations defines tanh function:

Activation Function 

Tanh function is a scaled version of sigmoid function. The gradient of the points farther from origin approaches slowly to zero than sigmoid function. But, tanh function also faces the problem of vanishing gradient.

5) Relu: Relu stands for Rectified linear unit. It is one of the most commonly used activation function.

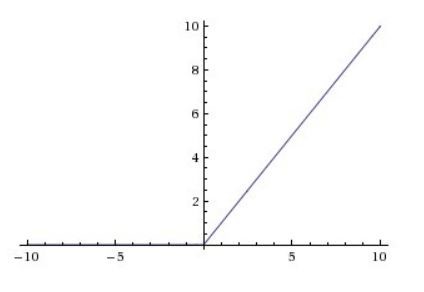


Figure 10: Relu Function

III. REFERENCES:

1. Layers of a Convolutional Neural Network. (2020, March 02). Retrieved from https://mc.ai/layers-of-aconvolutional-neural-network/
2. MACMAC 51644 silver badges1414 bronze badges.

(1969, October 01). Understanding Gaussian Filter and how to plot it in 1-D. Retrieved from

https://stackoverflow.com/questions/60462388/understandinggaussian-filter-and-how-to-plot-it-in-1-d

1. Anh H. Reynolds. (2017, October 15). Convolutional Neural Networks (CNNs). Retrieved from

https://anhreynolds.com/blogs/cnn.html

1. Diterbitkan oleh Benny Prijono Lihat semua pos dari Benny Prijono, Prijono, D., Prijono, B., Lihat semua pos dari Benny Prijono, 14, P., Berkata:, P., . . . (wajib), N. (2018, April 03).

Student Notes: Convolutional Neural Networks

(CNN) Introduction. Retrieved June 26, 2020,