Convolutional Neural Network

VASHIEGARAN MANOGARAN

October 19 , 2020

Convolutional neural network (CNN) is a biologically-inspired network, inspiration from the visual cortex [20]. CNN is one of the main categories to do images recognition, images classifications. Objects detections and recognition faces are some of the areas where CNN is widely used. Technically, the CNN model's objective to train to recognize patterns, for an example, edges of an image. To do so, the image input will pass through a series of convolution layers with filers or also known as kernels, polling, fully connected layers and apply a SoftMax function to label an object with probabilistic values between 0 and 1.

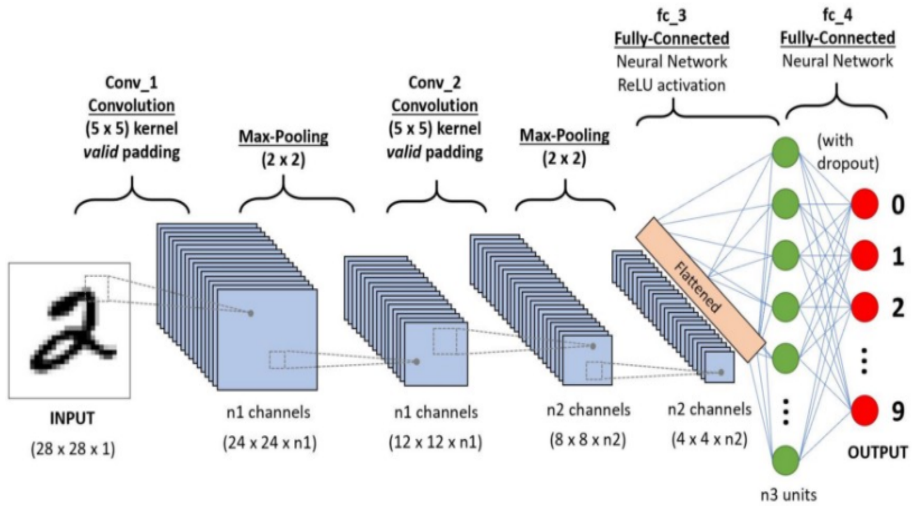

Figure 3.2: Example of CNN architecture. This image is borrowed from [21]

0.0.1 Convolutional layer

Convolutional layers consist of three parts mainly: Convolution, activation function and pooling

Figure 3.3: A visualization of 5 x 5 sized filter convolving around an input volume and producing an activation map, this image copied from the book Neural Networks and Deep Learning by Michael Nielsen [22].

0.0.2 Activation Function

Convolutional layers consist of three parts mainly: Convolution, activation function and pooling

Function 1 *Sigmoid*	Function 2 *Softplus*	Function 3 *Tanh*	Function 4 *ReLU*	Function 5 *lReLU*
$\frac{1}{e^{-x}}$	$log(e^x + 1)$	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$	$max(x, 0)$	$x > 0 \Rightarrow x$ $else \Rightarrow \alpha * x$

Table 3: Activation Functions considered in this thesis.

0.0.3 Pooling

Pooling is used to reduce the dimension of each activation map, to improve the processing speed of neural networks. The most commonly used pooling are max pooling and average pooling. Normally a 2x2 sized windows sliders over the matrix of data and in max pooling, the highest value in the 2x2 windows is stored. Average pooling will calculate the average of all value in the 2x2 sized window.

0.0.4 Loss Function

The loss functions compare a ground truth (GT) answer against a generated output. The loss function decides how well the model is achieving its objective, during the training the loss function is minimized each iteration to improve the deep learning model. A few commonly used loss functions that will be considered are the Cross-entropy loss, Mean Squared Error (MSE), L1, and L2.

0.1 Batch Normalization

In order to understand batch normalization, we need to first understand what data normalization is in general, and we learned about this concept in the episode on dataset normalization.

When we normalize a dataset, we are normalizing the input data that will be passed to the network, and when we add batch normalization to our network, we are normalizing the data again after it has passed through one or more layers.

Why normalize again if the input is already normalized?

Well, as the data begins moving though layers, the values will begin to shift as the layer transformations are preformed. Normalizing the outputs from a layer ensures that the scale stays in a specific range as the data flows though the network from input to output.

The specific normalization technique that is typically used is called standardization. This is where we calculate a z-score using the mean and standard deviation.

$z = \frac{x - mean}{std}$

How Batch Norm Works

$y = \frac{x - mean}{std} * \gamma + \beta$

When using batch norm, the mean and standard deviation values are calculated with respect to the batch at the time normalization is applied. This is opposed to the entire dataset, like we saw with dataset normalization.

Note that the scaling given by γ corresponds to the multiplication operation, and the sifting given by β corresponds to the addition operation.

The Scale and sift operations sound fancy, but they simply mean multiply and add.

These learnable parameters give the distribution of values more freedom to move around, adjusting to the right fit.

The scale and sift values can be thought of as the slope and y-intercept values of a line, both which allow the line to be adjusted to fit various locations on the

2D plane.