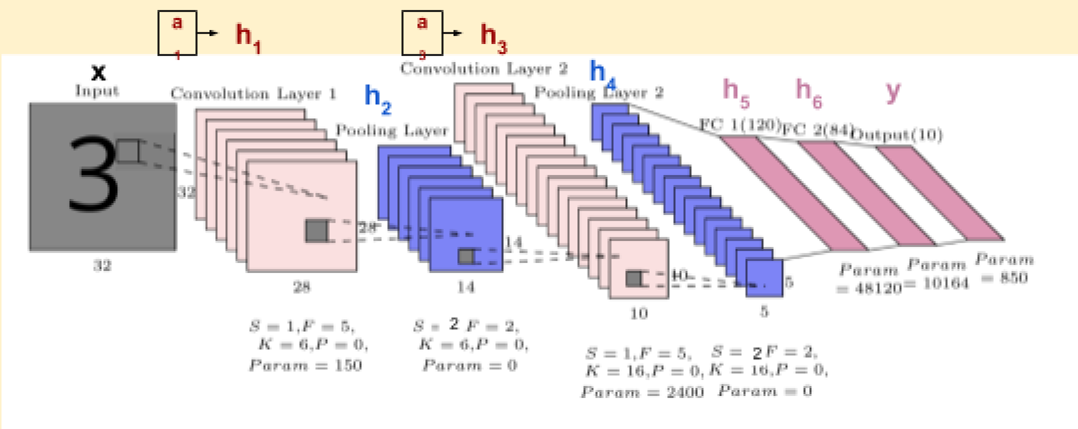
PadhAI: From Convolution Operation to Neural Network

One Fourth Labs

Our First Convolutional Neural Network (CNN)

How to use a convolutional neural network for image classification?

1. The following diagram illustrates the configuration and working of a Convolutional Neural Network. It follows the **LeNet** architecture, created by Yann LeCun



2. Let us sequentially break down the various layers in this CNN
3. **Input**:
   a. The image takes 32x32 pixel inputs.
   b. There is no depth component because the images are in black & white.
4. **Convolution Layer 1**:
   a. Here, the filter size F = 5, and the central cell is the pixel of interest
   b. Stride length S = 1
   c. We use a total of 6 filters, i.e. K = 6
   d. No padding is used, i.e. P = 0
   e. Each of the filters generate 28x28 output (calculated using W_O, H_O formula).
   f. Our hidden representation at this layer is $\mathbf{a_1}$ **= 28x28x6** (D_O = K).
   g. Non-linearity like tanh or ReLU(preferred for CNN) is applied to $\mathbf{a_1}$ making it $\mathbf{h_1}$
   h. If we were to proceed as a Fully Connected Network, we would have an extremely large number of parameters (32x32 x 28x28x6 = 4,816,896 parameters).
   i. However in this sparsely connected network, each of the 6 filters is of size 5x5x1. So the number of parameters would be much more manageable (6 x 5x5x1 = 150 parameters).
   j. This is significantly smaller than in a fully connected network, thereby reducing the chance of overfitting.
   k. Here, the values F, S, K, P etc are all counted as hyperparameters.
5. **Max Pooling Layer 1**:
   a. The hyperparameters are as follows
   b. Filter size F = 2
   c. Stride length S = 2
   d. No. of filters K = 6
   e. Padding P = 0
   f. Here, from a 2x2 filter, we select only 1 value. Therefore for a stride of 2, the output dimensions are half of the input(h_1) dimensions, i.e 14x14
   g. We apply the max pooling independently to all 6 of the h_1 layers, giving us $\mathbf{h_2}$ **= 14x14x6**
   h. No parameters for this layer as we are simply choosing the largest value in the filter and not applying any weights to it.

6. **Convolutional Layer 2**:
    a. The hyperparameters are as follows
    b. Filter size F = 5
    c. Stride length S = 1
    d. No. of filters K = 16
    e. Padding P = 0
    f. Thus, the filter dimensions are 5x5x6
    g. Here, 16 filters are applied to the input h_2, thereby giving us an output depth of D_O = 16
    h. Calculating W_O and H_O using the formula, we get 10x10
    i. Our hidden representation at this layer is $\mathbf{a_3}$ **= 10x10x16**
    j. Non-linearity like tanh or ReLU(preferred for CNN) is applied to $\mathbf{a_3}$ making it $\mathbf{h_3}$
    k. The number of parameters for the filters(16 x 5x5x6) is 2400 parameters
    l. This is much smaller than what we would have had in a fully connected network
7. **Max Pooling Layer 2**:
    a. The hyperparameters are as follows
    b. Filter size F = 2
    c. Stride length S = 1
    d. No. of filters K = 16
    e. Padding P = 0
    f. Here, from a 2x2 filter, we select only 1 value. Therefore for a stride of 2, the output dimensions are half of the input(h_3) dimensions, i.e 14x14
    g. We apply the max pooling independently to all 16 of the h_1 layers, giving us $\mathbf{h_4}$ **= 5x5x16**
    h. No params for this layer as we are simply choosing the largest value in the filter.
8. **Fully connected layer 1**:
    a. Number of neurons: 120
    b. Input is $\mathbf{h_4}$ flattened, i.e. 5x5x16 = 400
    c. No. of parameters in $\mathbf{h_5}$ = 120x400 + 120-bias = 48120 parameters
9. **Fully connected layer 2**:
    a. Number of neurons: 84
    b. Input is number of neurons in $\mathbf{h_5}$ = 120
    c. No. of parameters in $\mathbf{h_6}$ = 84x120 + 84-bias = 10164 parameters
10. **Output layer**:
    a. Number of neurons: 10
    b. Input is number of neurons in $\mathbf{h_6}$ = 84
    c. No. of parameters in **y** = 10x84 + 10-bias = 850 parameters
11. Overall, this combination of Convolutional and fully-connected layers is much more efficient than an entirely fully connected network. It has a significantly lower number of parameters but still is able to estimate functions of very high complexity.