Convolutional Neural Networks

S. Dakurah

02/26/20

Outline

Outline

Feed Forward NN

- Softmax Classifier?
  - This is simply[1] a classification function similar to the SVM or logistic.
  - It uniqueness stems from it's ability to output a ranked probability space(i.e. indicating the probability of the input belonging to a given class).
  - The mathematical representation is given as:

$$p(y = j|X) = \frac{e^{x^T w_j}}{\sum_{k=1}^{p} e^{X^T w_k}}$$

- Why a specific set of activation functions?
  - Some reference to the activation properties of biological neurons. Still reading on it.

[1] In essence a normalized exponential function

Outline

Convolutional Neural Networks - Motivation

- Specifically designed to handle image classification.
- Why not use regular FFNN?
  - Images are high-dimensional vectors.
  - It would take a large number of parameters to characterize the network
- Why not flatten[2] the images and pass it in as a vector?
  - Distortion of the spatial features of the image(i.e. a three dimensional object will be compressed into a one dimensional object, huge loss of information).
- How does CNN address these shortfalls?

[2] This will eventually be done, but after we've learnt a lot about the spatial composition of the image.

Sample CNN Architecture

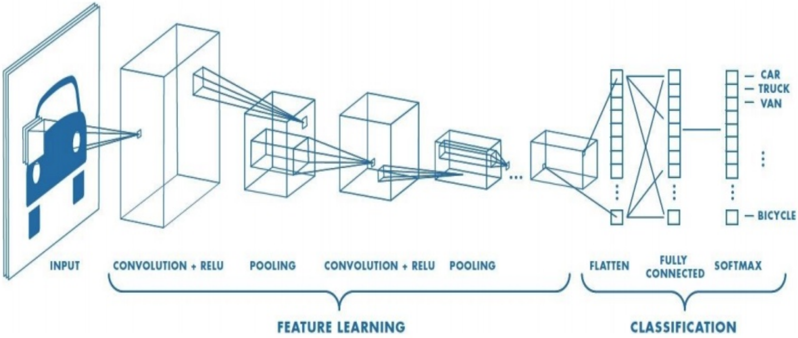

Figure: Sample CNN Architecture. (Source: Google)

Outline

The Input Layer

- An input will simply be a raw pixels values of an image.
- Convert the image to it's matrix representation.

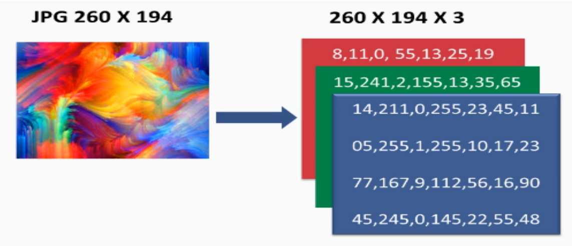

Figure: Sample Image Representation (Source: Google)

- The depth refers to the color channels(in this case 3, RGB), the height and width in pixels is given by the dimensions of the input image.

The Convolution Layer

- Pick a local region within the image.
- Pass the representation of that region to a neuron in the CONV layer.
- Convert the image to it's matrix representation.
- The neuron then apply a set of filters.
- What are filters?
  - These are a set of learnable weights (can be learned via BP).
  - These vector of weights (convolve?) the input.
  - That's, it provides a measure of how close a patch of input is to a feature (e.g. arch, edge).

Example of a Convolved Feature

- Assume we take a local region of the image represented by this 5×7 matrix[3], and we want to extract a feature of size 3×3.

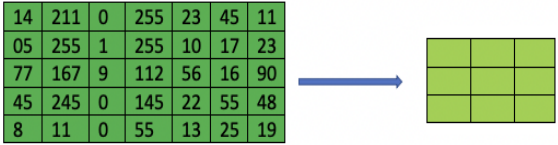

Figure: Initial Representation (1 Channel)

- Then we can define the **stride** size to control how we move or slide around the window. If we choose a stride of size 1, the convolved feature will proceed as follows.

[3] This has to be a square matrix by design to maintain symmetry

Example of a Convolved Feature

This should be included - sixteen

- First slide

14	211	0	255	23	45	11
05	255	1	255	10	17	23
77	167	9	112	56	16	90
45	245	0	145	22	55	48
8	11	0	55	13	25	19

$\longrightarrow$

4		

- Second slide

14	211	0	255	23	45	11
05	255	1	255	10	17	23
77	167	9	112	56	16	90
45	245	0	145	22	55	48
8	11	0	55	13	25	19

$\longrightarrow$

4	13	

- Third slide

14	211	0	255	23	45	11
05	255	1	255	10	17	23
77	167	9	112	56	16	90
45	245	0	145	22	55	48
8	11	0	55	13	25	19

$\longrightarrow$

4	13	0

Example of a Convolved Feature

- We then repeat the same process for the other two channels.
- We can add more filters by defining additional weight matrices.
- Three set of parameters control the output volume.
  - The depth (number of filters)
    - If we assume this is the initial layer, we can represent the filter(weight) matrix of the first channel as $w_0[:, :, :, 0]_{4x3x1}$.
    - Similarly, the weight matrix of the 2nd and 3rd channel will be $w_0[:, :, :, 1]_{4x3x1}$ & $w_0[:, :, :, 2]_{4x3x1}$.
    - The weight matrix of the entire first layer W_0 will be of size $4x3x3$.
  - The stride
  - The zero-padding

Summary of the Convolution Layer

- Accept an input of size $W_i x H_i x D_i$, width, height, and dimension.
- Requires 4 hyperparameters:
  - Filter size K
  - The receptive field F (size of the local region covered at each stride).
  - Stride size (S).
  - The amount of zero padding (P).
- Produce an output volume of size $W_{i+1} x H_{i+1} x D_{i+1}$.
  - $W_{i+1} = (W_i - F + 2P)/S + 1$
  - $H_{i+1} = (H_i - F + 2P)/S + 1$
  - $D_{i+1} = K$

The Activation Layer

- This is basically used to improve the non-linearity of the network without affecting the receptive fields.
- ReLU[4] (and its improved version Leaky ReLU) is often used.
- What exactly is the neuron that is been activated?
  - Anytime you take a dot product of a filter and a local region to produce a 3D volume (in this instance) is a neuron.

[4] Results in faster training

The Pooling Layer

- The feature output from the convolution layer are sensitive to the location of the region in the image. Shifts, change in resolution could lead to different feature mapping.
- A common approach to dealing with this is down sampling (DS).
- DS can be achieved by changing the stride size across the image.
- A more robust[5] method is "pooling".

[5] Does not require retraining of the initial input

The Pooling Layer

- How does pooling work?
  1. Select a feature map, which simply an output from a neuron.
  2. Map a lower dimensional representation of that feature space.
  3. Apply a pooling method to the mapped-out feature space.
- Two common pooling methods:
  - Average Pooling: Simply calculate the average if each feature map.
  - Maximum Pooling: You take the maximum within each feature map.

Example of a pooled feature

- Average Pooling

- Max Pooling

- This produces a lower dimensional representation. This will be repeated for all channels.
- As you move through the network, convolution and pooling can be repeated many times.

Fully Connected Layer

- This is where classification takes place.
- At this layer, you apply regular neural network architecture to the flatten representation of the learned 33D features.
- Classification works the same as in normal neural networks!

Outline

Regularization in CNN and Sample Implementation

- (Next week - 03/04)

Outline

References

1. Pattern Recognition and Machine Learning $\sim$ M. Bishop
   - Chaper 5.5
2. A Gentle Introduction to Backpropagation $\sim$ S. Shashi
   - Click to Access