CPE 490 590: Machine Learning for Engineering Applications

14 Learning on Sequence Data

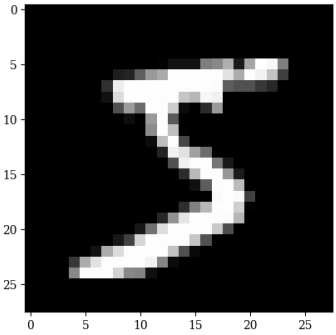
Rahul Bhadani
Electrical & Computer Engineering, The University of Alabama in Huntsville

Outline

Convolutional Neural Network

Motivation

- Image can be considered as a form of a matrix.



Motivation

- In order to train with Artificial Neural Network (ANN), we need to convert them 1-D array (that is flattening).

$$\begin{array}{|c|c|c|}\hline a_{11} & a_{12} & a_{13} \\ \hline a_{21} & a_{22} & a_{23} \\ \hline \end{array} \rightarrow \begin{array}{|c|c|c|c|c|c|}\hline b_1 = a_{11} & b_2 = a_{12} & b_3 = a_{13} & b_4 = a_{21} & b_5 = a_{22} & b_6 = a_{23} \\ \hline \end{array}$$

- In doing so, we are losing pixel-to-pixel relationship.
- In addition, working with color images would mean we have at least three channels R, G, B.
- So if we have 200×200 image size, and if we flatten it, then we will have $200 \times 200 \times 3 = 120000$ features.
- And if we have just 100 images, that's $120000 \times 100 = 12000000$ features.

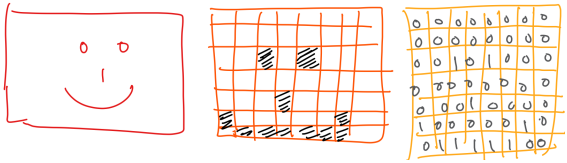
Limitations of ANN for Images

In that sense, ANN suffers from two conditions:

- Unable to preserve spatial information
- Curse of dimensionality

To preserve the pixel-to-pixel spatial relationship, we include two additional operations in the neural network:

1. Convolution
2. Pooling

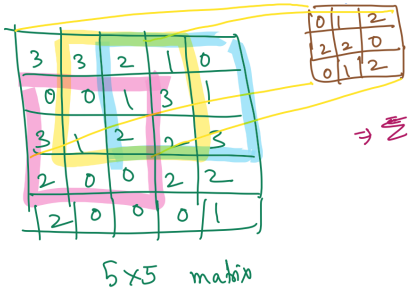


Convolution

Convolution is an operation to reduce the number of pixels through some operation that in turn reduces the number of weights to learn.

In convolution, the reduction in number of features happens by sliding a kernel across an image performing the dot product.

Kernel: It is an $n \times n$ matrix of some numbers whose dimension is usually smaller than the image size.



Kernels

A particular kernel usually extracts certain features from an image, or basically it performs some sort of image processing.

Examples of Kernels

$$\text{Edges: } \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\text{Sharpen: } \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\text{Box Blur: } \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Edge-detection Example

Original Image

Edge Detection Filtered Image

Filter

The window we use in kernel is called a filter. The window of a certain size is slid across the image. We can apply pooling to perform operations such as minimum, median, maximum.

- Images can be represented as matrices, where each element corresponds to a pixel
- A filter is just a small matrix that is convolved with same-sized sections of the image matrix

Convolutional Filters

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 2 & 4 & 4 & 2 & 0 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 0 & 1 & 2 & 3 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \\ & \\ & \\ & \end{bmatrix}$$

$$(0 * 0) + (0 * 1) + (0 * 0) +$$
$$(0 * 1) + (1 * -4) + (2 * 1) +$$
$$(0 * 0) + (2 * 1) + (4 * 0) = 0$$

Convolutional Filters

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 2 & 4 & 4 & 2 & 0 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 0 & 1 & 2 & 3 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 & 0 \\ -2 & -5 & -5 & -2 \\ 2 & -2 & -1 & 3 \\ -1 & 0 & -5 & 0 \end{bmatrix}$$

Pooling

Pooling does a mathematical operation to the output of the convolution image. But you can also apply pooling directly to the image.

Pooling: Downsampling

Combine multiple adjacent nodes into a single node: max-pooling

$$\begin{bmatrix} 0 & -1 & -1 & 0 \\ -2 & -5 & -5 & -2 \\ 2 & -2 & -1 & 3 \\ -1 & 0 & -5 & 0 \end{bmatrix} \rightarrow \textbf{max} \rightarrow \begin{bmatrix} 0 & \\ & \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & -1 & -0 \\ -2 & -5 & -5 & -2 \\ 2 & -2 & -1 & 3 \\ -1 & 0 & -5 & 0 \end{bmatrix} \rightarrow \textbf{max} \rightarrow \begin{bmatrix} 0 & 0 \\ & \end{bmatrix}$$

- Reduces the dimensionality of the input to subsequent layers and thus, the number of weights to be learned
- Further, it protects the network from (slightly) noisy inputs

Downsampling with Stride

- Only apply the convolution to some subset of the image
- e.g., every other column and row = a stride of 2

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 2 & 4 & 4 & 2 & 0 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 0 & 1 & 2 & 3 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} -2 & \\ & \end{bmatrix}$$

Downsampling with Stride

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 2 & 4 & 4 & 2 & 0 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 0 & 1 & 2 & 3 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ & \end{bmatrix}$$

Downsampling with Stride

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 2 & 4 & 4 & 2 & 0 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 0 & 1 & 2 & 3 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} -2 & -2 & 1 \\ & & \end{bmatrix}$$

Downsampling with Stride

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 2 & 1 & 0 \\ 0 & 2 & 4 & 4 & 2 & 0 \\ 0 & 1 & 3 & 3 & 1 & 0 \\ 0 & 1 & 2 & 3 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} -2 & -2 & 1 \\ 0 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix}$$

Operation after Pooling

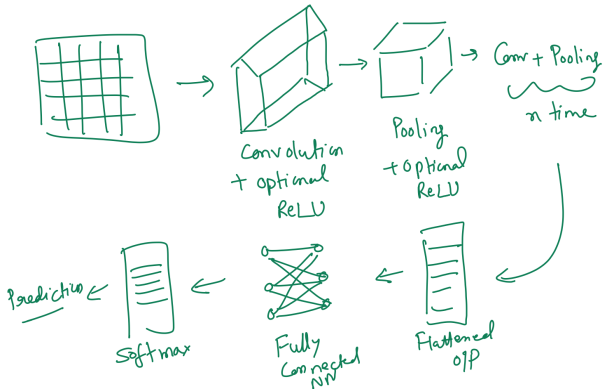
After applying convolution and pooling, we apply a non-linear activation, ReLU.

Then after that, you have more Convolution, more Pooling layers.

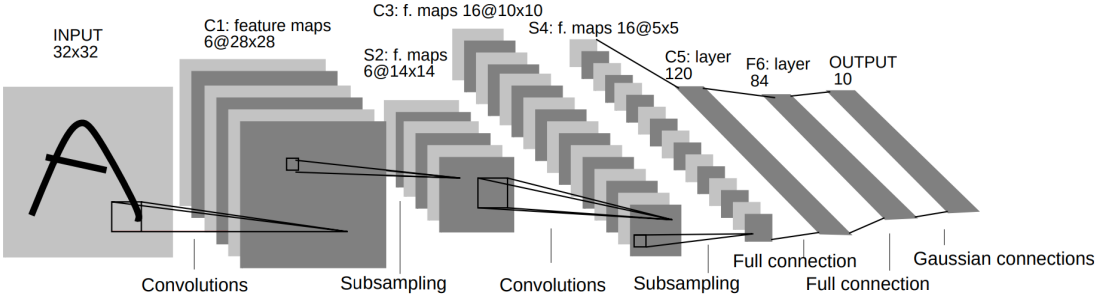
Finally, after several sequences of Convolution + Pooling + ReLU, we flatten the output from the previous layer and then we feed it to the fully connected layer (ANN).

Image Classification

Then, we apply a softmax activation if our goal is to do supervised image classification.



Lenet



Source: http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

- One of the earliest, most famous deep learning models – achieved remarkable performance at handwritten digit recognition ($<$ 1% test error rate on MNIST dataset)
- Used sigmoid (or logistic) activation functions between layers and mean-pooling, both of which are pretty uncommon in modern architectures

A note on multi-dimensionality and vectorization:

- Images are high-dimensional datasets. This is true even more for colored images.
- Let's say $X \in \mathbb{R}^{H \times W}$ is a matrix (for a grayscale image of height H and width W).
- After flattening, an image can be represented as a column vector $X \in \mathbb{R}^{D}$ with $D = H \times W$ elements. For a color image with 3 channels, $D = H \times W \times 3$.

Tensor

We need a concept of higher-order matrices which essentially called **Tensor**.

- $X \in \mathbb{R}^{H \times W \times D}$ is an order-3 tensor, or third-order Tensor. Another way to consider an order 3 tensor as D channels of matrices. $D = 1$ reduces it to a matrix.
- A scalar value is order-0 tensor.
- A vector is order-1 tensor.
- A matrix is order-3 tensor.

We can also have 4 dimensions if we consider opacity of an image (say in PNG).

Tensor to Vector

Given a tensor, we can arrange all numbers inside it into a long vector following a prespecified order.

Example (Matrix Vectorization - Column First Order):

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \xrightarrow{\text{In MATLAB } A(:)} \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix}$$

In order to vectorize an order 3 tensor, we would first vectorize the first channel (which is a matrix), then the second channel, and so on. So we see that it is a recursion or a recursive process. The same can be applied to tensors of order > 3.

Tensor in CNN

In CNN, the convolution kernel will form an order 4 tensor.

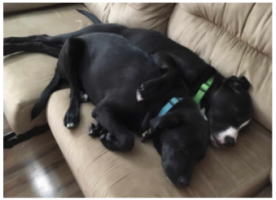
CNN Input

1. Depth of input: Number of channels in a colored image.
2. Number of filters: We can have more than one filter, each responsible for learning a specific feature.
3. Spatial dimensions of an image: $H \times W$.

Tensor Orders (Examples):

- Order 3 tensor: A batch of grayscale images (or a black & white movie).
- Order 4 tensor: A batch of colored images or a single colored movie.
- Order 5 tensor: A batch of movies.

Tensors



Example:
$3 \times 4 \times 6$ tensor

4	1	2	16	3	6
1	5	2	6	14	15	8
5	26	4	6	8	9	5	3
0	0	16	5	2	8	2	1
	7	5	2	14	11	7	8
		15	2	5	0	9	8

Source: https://www.cs.cmu.edu/~mgormley/courses/10601//slides/lecture17-cnn.pdf

Further Reading

- https://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf
- http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf