Chapter 9

Convolutional Neural Network

Contents

9 Convolutional Neural Network **1**
9.1 How do computers see? . . . 2
9.2 Computer Vision . . . 2
9.2.1 Grayscale Model . . . 2
9.2.2 RGB Color Model . . . 3
9.3 Image Classification . . . 3
9.4 Convolutional Neural Network . . . 3
9.5 Filter . . . 4
9.6 Convolution . . . 5
9.7 MNIST Dataset . . . 5
9.8 Image Analysis . . . 5
9.9 Pooling . . . 5
9.10 Spatial Invariance . . . 5
9.11 Increased Efficiency . . . 5
9.12 Convolution + Activation + Pooling . . . 6
9.13 Feature Maps . . . 6
9.14 Training a CNN . . . 6
9.15 LeNet 5 . . . 6
9.16 VGGNet . . . 6

9.1 How do computers see?

9.2 Computer Vision

9.2.1 Grayscale Model

- Images contain pixels with just one value.
- Can be represented using a 2-D array.
- 0: black, 255: white, 1–254: shades of gray.

9.2.2 RGB Color Model

- Each color channel is stored in 8 bits.
- 8 bits can store 256 values (0–255).
- Also known as 24-bit color (8×3).

9.3 Image Classification

- Can we directly take an image and feed it to a regular fully-connected neural network?
  - Yes, we can, but we will need to first flatten the 2-D image array.
- Issues:
  - No spatial information.
  - Too many parameters.
- Solution:
  - Exploit spatial structure.
  - Each neuron in the hidden layer only respond to a certain set of neurons in the previous layer.
  - Connect the patch in input layer to a single neuron in the subsequent layer.
  - Use a sliding window to define all possible connections.
  - Weighting the connection between the patches and the next layer will allow uss to learn the features.

9.4 Convolutional Neural Network

- CNN or ConvNet is a specialized kind of neural network for processing data that has a known grid-like topology.
  - Image data, which can be thought of as a 2-D grid of pixels.
  - Time-series data, which can be thought of as a 1-D grid taking samples at regular time intervals
- ...
- Convolutional layer performs a transformation called convolution, a specialized king of linear operation on its input.
- In CNN, convolution replaces general matrix multiplication in their convolution layers.
- CNN is specialized for pattern detection.

- Convolutional layer specifies the number of filter kernels each layer must have, and these filters are used to detect patterns.
- Each layer in a convolutional neural network has a 3-D lattice structure.
- Three types of transformations between layers:

  Convolution Apply filters to generate feature maps.

  Activation function To introduce nonlinearity.

  Pooling Downsampling operation on each feature map.
- CNN performs these transformations repeatedly:
  - Higher-order feature detectors after convolution.
  - Lower spatial resolution after pooling.
- In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations.
- In the second stage, each linear activation is run through a nonlinear activation function, such as ReLU. This stage is called the detector stage.
- In the third stage, a pooling function is used to modify the output of the layer further. A pooling function replaces the output of the network at a certain location with a summary statistic of the nearby outputs:
  - The max pooling operation reports the maximum output within a rectangular neighboorhood.
  - Other pooling strategies include average pooling, weighted average pooling, L2 norm, etc.

Spatial locality features at nearby locations in an image are most likely to have joint causes and consequences.

Spatial position homogeneity features deemed significant in one region of an image are likely to be significant in others.

Spatial scale homogeneity locality and position homogeneity should apply across a range of spatial scales.

9.5 Filter

- At every convolutional layer, we specify how many filter kernels we want.
- Filter is a matrix used for blurring, sharpening, embossing, edge detection, and more. The values within this matrix are initialized with random numbers.

9.6 Convolution

- Convolution is a mathematical operation on two functions x and h that produces a third function $x \times h$.
- For CNN, we denote convolution as $s_i = (x \times w)_i$

  x the input

  w the filter

9.7 MNIST Dataset

9.8 Image Analysis

- Assume there is a convolutional layer accepting handwritten digits from the MNIST dataset and trying to classify them correctly.

9.9 Pooling

- Reduces dimensionality of the image progressively as you go deeper into the convolutional network.
- This means every filter now is being slid over a smaller image, thus it captures a larger receptive field from the previous layer.
- Other benefits include spatial invariance and increased efficiency.

9.10 Spatial Invariance

- Pooling helps to make the representation become approximately invariant to small translations of the input.
- Invariance to translation means that if the input is translated by a small amount, the values of most of the pooled outputs do not change.
- Invariance to local translation can be a very useful property if we care more about whether some feature is present than exactly where is is.
- The use of pooling can be viewed as adding a strong prior that the function the layer learns must be invariant to small translations.

9.11 Increased Efficiency

- Pooling units summarize detector units by reporting summary statistics for pooling regions spaced k pixels apart rather than 1 pixel apart.

9.12 Convolution + Activation + Pooling

9.13 Feature Maps

- As you progress through the layers, the feature maps become increasingly complex and abstract.
- Lower-level feature maps detect simple edges and shapes, while deeper feature maps encode high-level concepts, such as object parts or entire objects.
- Feature maps become sparser as we go deeper, meaning the filters detect less features.
- Deeper feature maps contain less information about the image and more about the class of the image.

9.14 Training a CNN

- The same procedure from backpropagation applies here. The error terms from the output layer is passed back to the previous layers, one by one.
- Backpropagation for the pooling layer:
  - Assuming max pooling
  - The backpropagated error is $\delta_{pool} -$
- Backpropagation for the convolutional layer

9.15 LeNet 5

- Designed by LeCun et al. for character recognition in both handwriting and machine printing.

9.16 VGGNet

- Developed by Simonyan and Zisserman at the Visual Geometry Group in Oxford University.
- Main contribution was showing that depth of the network is a critical component for good performance.