Introduction to Machine Learning

Lecture 13 - Convolutional Neural Networks (CNN)

Guang Bing Yang, PhD

Convolutional Neural Networks

- Introduction to Convolutional Neural Networks.
- Case Study

What is Convolutional Neural Networks

- Convolutional Neural Networks is a class of deep neural networks.
- It is a major technique to solve the computer vision problem.
- It arms to provide a richer class of density models than the single one.
- It employs a mathematical operation called convolution, which defines as the integral of the product of the two functions after one is reversed and shifted
- CNNs are regularized versions of multilayer perceptrons
- CNNs are fully connected networks

CNNs

- CNNs have been widely applied in:
  - image and video recognition,
  - recommender systems,
  - image classification,
  - Image segmentation,
  - medical image analysis,
  - natural language processing,
  - brain-computer interfaces,
  - and financial time series.

CNNs: Architecture

- Overview:
  - Regular neural networks consist of input layer, hidden layer(s), and output layer.
  - They are not good enough to full images
  - e.g., in CIFAR-10, images are size 32x32x3 (wide x high x color channels),
  - so a single fully-connected unit in a first hidden layer of a fully connected Neural Network would have 32*32*3 = 3072 weights
  - For any regular size image (e.g., 1920 x 1024 high-resolution image, the number of weights is more than 3 millions!
  - Need a better architecture of the NN to process the images

CNNs: Architecture

- A convolutional neural network consists of an input layer, hidden layers and an output layer
- The convolution process completes at the hidden layers
- Overall architecture is:
  - **Input layer**—hold the raw image data e.g., 32x32x3
  - **convolution layer** (abs. Cone-layer) — computing a dot product between their weights and a small region connected to in the input
  - **RELU layer** — an elementwise activation function, e.g., Relu function
  - **POOL layer** or **MAX POOL layer** — downsampling operation along the spatial dimensions
  - **Full Connection Layer** (or **FC**) — compute the class scores.

CNNs: Convolution Example

- Example of 6x6 matrix convolved with 3x3 filter/kernel gives us a 4x4 matrix

→ 3×1 + 1×1 + 2×1 + 0×0 + 5×0 + 7×0 + 1×-1 + 8×-1 + 2×-1 = -5

"convolution"

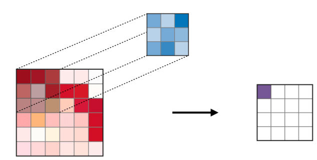
$$\underbrace{\begin{bmatrix} 3 & 0 & 1 & 2 & 7 & 4 \\ 1 & 5 & 8 & 9 & 3 & 1 \\ 2 & 7 & 2 & 5 & 1 & 3 \\ 0 & 1 & 3 & 1 & 7 & 8 \\ 4 & 2 & 1 & 6 & 2 & 8 \\ 2 & 4 & 5 & 2 & 3 & 9 \end{bmatrix}}_{6\times 6} * \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}}_{3\times 3} = \underbrace{\begin{bmatrix} -5 & -4 & 0 & 8 \\ -10 & -2 & 2 & 3 \\ 0 & -2 & -4 & -7 \\ -3 & -2 & -3 & -16 \end{bmatrix}}_{4\times 4}$$

→ filter kernel

Image was copied from Prof. Andrew Ng ML course

CNNs: Architecture

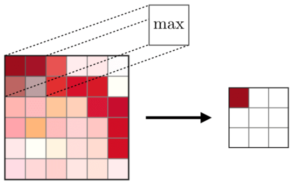
- Convolution layer (**CONV**)
  - filters convolute as scanning the input I w.r.t. its dimensions
  - hyperparameters: the filter size F and stride S
  - The output O is called ***feature map*** or ***activation map***.



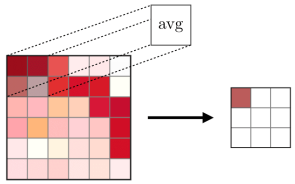
CONV cited from Stanford DeepLearning

CNNs: Architecture

- Pooling (POOL) — downsampling operation,
  - applied after CONV-layer
  - two types: max-pooling and average-pooling



Max-pooling from Stanford DeepLearning

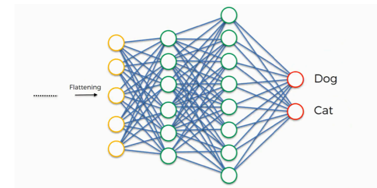


Average-pooling from Stanford DeepLearning

yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: Architecture

- **Fully Connected (FC):**
  - the fully connected layer (FC) operates on a flattened input where each input is connected to all neurons.
  - usually at the end of CNN architectures
  - to optimize class scores



FC layer from SuperDataScience

yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: hyperparameters

- Hyperparameters of CNNs are:
  - receptive field size F
  - stride size S, and
  - zero padding P
- Relations with Input and Output dimensions:
  - Input volume I $\in W \times H \times D$, width, height, and depth
  - Output of CONV as O $\in O_W \times O_H \times K$
  - $O_W = (W - F + 2P)/S + 1$
  - $O_H = (H - F + 2P)/S + 1$

yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: Hyperparameters Example

- Hyperparameters of CNNs example:
  - receptive field size F = 11
  - stride size S = 4
  - zero padding P = 0
- Relations with Input and Output dimensions:
  - Input volume in ImageNet, I $\in 227 \times 227 \times 3$, width, height, and depth
  - Output of CONV as O $\in 55 \times 55 \times 96$, there are 96 filters total
  - $O_W = (227 - 11 + 2 * 0)/4 + 1 = 55$
  - $O_H = (227 - 11 + 2 * 0)/4 + 1 = 55$

yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: hyperparameters

- In general, the formula to calculate convolution output dimensions given the hyperparameters are:
- $O = \dfrac{I - F + P_{start} + P_{end}}{S} + 1,$
- usually, $P_{start} = P_{end} = P$, so
- $O = \dfrac{I - F + 2P}{S} + 1$

yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: Activation Functions

- Commonly used activation function in CNNs:
  - ReLU — Rectified Linear Unit
    - $g(z) = \max(0, z),$
    - Non-linear
  - Leaky ReLU:
    - $g(z) = \max(\epsilon z, z), \quad \epsilon < < 1$
  - ELU:
    - $g(z) = \max(\alpha(e^z - 1), z), \quad \alpha < < 1$

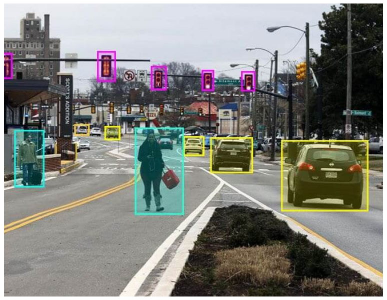
yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: Applications

- Object detection:
  - three main types:
    - Image classification
    - Classification w. Localization
    - Detection
      - Bounding box detection
      - Landmark detection

yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: Applications-Object Detection



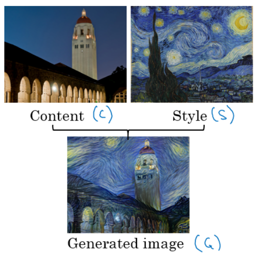
Bounding box detection

Landmark detection

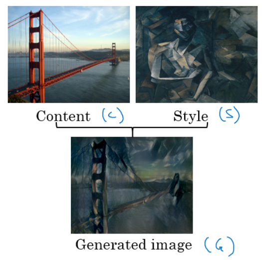
yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

CNNs: Applications

- Neural style transfer:
  - takes a content image C and a style image S and generates the content image G with the style of style image



Neural style transfer cited from Prof. Andrew Ng DeepLearning



Neural style transfer cited from Prof. Andrew Ng DeepLearning

Recap

- A Convolution Neural Network architecture is a set of neural network layers that transform the image volume into an output volume (e.g. holding the class scores)
- There are several types of Layers: CONV, POOL, RELU, and FC
- Each layer accepts an input 3D volume and transforms it to an output 3D volume through a differentiable function
- Layer CONV and FC have parameters, RELU and POOL have no parameters.
- Layer CONV, POOL, and FC have hyperparameters, but RELU has no.
- Output dimensions of CONV layer follow the constraints of stride and zero padding
- Relationship between input dimension, filters and output are given by a formula:
- $O = \dfrac{I - F + 2P}{S} + 1$

yguangbing@gmail.com, Guang.B@chula.ac.th Apr 23th, 2021

Questions?