CS540 Introduction to Artificial Intelligence
Lecture 8

Young Wu
Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

July 1, 2021

SIFT and HOG Features

Motivation

- SIFT and HOG features are expensive to compute.
- Simpler features should be used for real-time face detection tasks.

Real-Time Face Detection

Motivation

- Each image contains 10000 to 500000 locations and scales.
- Faces occur in 0 to 50 per image.
- Want a very small number of false positives.

Features

Motivation

- There should be lots of very simple features.
- Each feature can define a weak classifier.
- Weak classifiers are easy to create and they are okay if they are at least slightly better than random guessing.
- Use boosting to combine the weak classifiers. This is called an ensemble classifier.

Face Features

Motivation

- For the specific task of face detection, domain knowledge can be used to construct the features.

1. The eye region is darker than the forehead or the upper cheeks.
2. The nose bridge region is brighter than the eyes.
3. The mouth is darker than the chin.

Haar Features Diagram

Motivation

Haar Features

Definition

- Haar features are differences between sums of pixel intensities in rectangular regions. Some examples include convolution with the following filters.

$$\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \dots$$

Integral Image

Definition

- Haar features are easy to compute because integral images can be used.
- An integral image of an image I is the sum of all pixels above and to the left of the pixel (s, t) in the image.

$$II(s, t) = \sum_{s' < s, t' < t} I(s', t')$$

- It can be efficiently computed using the following formula.

$$II(s, t) = I(s, t) + II(s - 1, t) + II(s, t - 1) - II(s - 1, t - 1)$$

Haar Feature Computation

Definition

- The sum of pixel intensities in any rectangular block can be computed in constant time given the integral image.
- For a rectangle with the top left corner at (s, t), top right corner at (s', t), bottom left corner at (s, t'), bottom right corner at (s', t'), the sum of pixel intensities can be computed using the following formula (instead of summing up the elements in the rectangle).

$$II\left(s', t'\right) + II\left(s, t\right) - II\left(s', t\right) - II\left(s, t'\right)$$

Weak Classifiers

Definition

- Each weak classifier is a decision stump (decision tree with only one split) using one Haar feature x.

$$f(x) = \mathbb{1}_{\{x>\theta\}}$$

- Finding the threshold by comparing the information gain from all possible splits is too expensive, so θ is usually computed as the average of the mean values of the feature for each class.

$$\theta = \frac{1}{2}\left(\frac{1}{n_0}\sum_{i:y_i=0} x_i + \frac{1}{n_1}\sum_{i:y_i=1} x_i\right)$$

Strong Classifiers

Definition

- The weak classifiers are trained sequentially using ensemble methods such as AdaBoost.
- A sequence of T weak classifiers is called aT -strong classifier.
- Multiple T -strong classifiers can be trained for different values of T and combined into a cascaded classifier.

Cascaded Classifiers

Definition

- Start with aT -strong classifier with small T, and use it reject obviously negative regions (regions with no faces).
- Train and use aT -strong classifier with larger T on only the regions that are not rejected.
- Repeat this process with stronger classifiers.

Cascading

Definition

- For example, at $T = 1$, the classifier achieves $a100$ percent detection rate and $a50$ percent false-positive rate.
- At $T = 5$, the classifier achieves $a100$ percent detection rate and $a40$ percent false-positive rate.
- At $T = 20$, the classifier achieves $a100$ percent detection rate and $a10$ percent false-positive rate.
- The result is a cascaded classifier with 100 percent detection rate and $0.5 \cdot 0.4 \cdot 0.1 = 2$ percent false positive rate.

Viola-Jones

Discussion

- Each classifier operates on a 24 by 24 region of the image.
- Multiple scales of the image with a scaling factor of 1.25 are used. The classifiers can be scaled instead in practice so that the integral image only needs to be calculated once.
- The detector is moved around the image with stride 1.
- Nearby detections of faces are combined into a single detection.

Viola-Jones Diagram

Discussion

Learning Convolution

Motivation

- The convolution filters used to obtain the features can be learned in a neural network. Such networks are called convolutional neural networks and they usually contain multiple convolutional layers with fully connected and softmax layers near the end.

Description of Algorithm

Description

- Convolve the input image with a filter.
- Pool the output of convolution.
- Feed the output of pooling into a neural network.

Convolutional Layers

Definition

- In the (fully connected) neural networks discussed previously, each input unit is associated with a different weight.

$$a = g\left(w^T x + b\right)$$

- In the convolutional layers, one single filter (a multi-dimensional array of weights) is used for all units (arranged in an array the same size as the filter).

$$A = g\left(W * X + b\right)$$

Inputs and Outputs of a Layer

Definition

- The output of a convolution layer is called a feature map.
- There can be multiple feature maps in a single convolutional layer. Each feature map is found by a convolution between the same input and a different filter (with a different bias).
- The output of one convolutional layer can be either used as the input of another convolutional layer or flattened to a vector and used as the input of a fully connected or softmax layer.

$2D$ Convolutional Layer Diagram

Definition

$3D$ Convolutional Layer Diagram

Definition

Pooling

Definition

- Combine the output of the convolution by max pooling,

$$a = \max\left\{x_1 \ldots x_m\right\}$$

- Combine the output of the convolution by average pooling,

$$a = \frac{1}{m}\sum_{j=1}^{m} x_j$$

Pooling Diagram

Definition

Training Convolutional Neural Networks, Part I

Discussion

- The training is done by gradient descent.
- The gradient for the convolutional layers with respect to the filter weights is the convolution between the inputs to that layer and the output gradient from the next layer.

$$\frac{\partial C}{\partial W} = X * \frac{\partial C}{\partial O}$$

- The gradient for the convolutional layers with respect to the inputs is the convolution between the 180 degrees rotated filter and the output gradient from the next layer.

$$\frac{\partial C}{\partial X} = \text{ rot } W * \frac{\partial C}{\partial O}$$

Training Convolutional Neural Networks, Part II

Discussion

- There are usually no weights in the pooling layers.
- The gradient for the max-pooling layers is 1 for the maximum input unit and 0 for all other units.
- The gradient for the average pooling layers is $\frac{1}{m}$ for each of the m units.

LeNet Diagram and Demo

Discussion

AlexNet Diagram

Discussion

VGG, GoogleNet, ResNet

Discussion