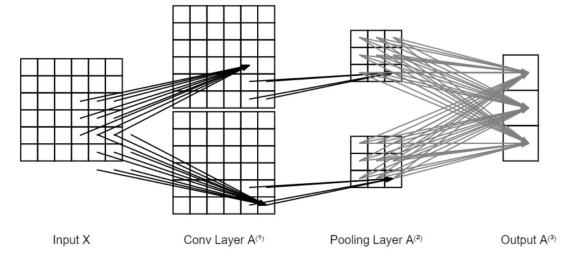
10/10

- Convolution
  - Using pixel intensities as the features assume pixels are independent of their neighbors. This is inappropriate for most of the computer vision tasks.
  - Neighboring pixel intensities can be combined to create one feature that captures the information in the region around the pixel.
  - Linearly combining pixels in a rectangular region is called convolution: Link, Wikipedia.
  - ⇨ The convolution of a vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ with a filter $w = (w_{-k}, w_{-k+1}, \ldots, w_{k-1}, w_k)$ is: $a_i = (a_{i1}, a_{i2}, \ldots, a_{im}) = x \star w$, where $a_{ij} = w_{-k}x_{i(j+k)} + w_{-k+1}x_{i(j+k-1)} + \ldots + w_k x_{i(j-k)}$ for $j = 1, 2, \ldots, m$.
  - ⇨ The convolution of an $m \times m$ matrix X_i with a $(2k+1) \times (2k+1)$ filter W is $A_i = X_i \star W$, where $A_{ij} = W_{-k} \star X_{i(j+k)} + W_{-k+1} \star X_{i(j+k-1)} + \ldots + W_k \star X_{i(j-k)}$ for rows $j = 1, 2, \ldots, m$.
  - ⇨ 3D convolution can be defined in a similar way.
  - The convolution filter is also called "kernel", but different from the kernel matrix for SVMs.
  - We add zero paddings around the image so that it is larger
  - Remember to flip the kernel before use
- Traditional computer vision algorithms
  - Traditional computer vision algorithms use engineered features for images based on image gradients.
  - ⇨ Image gradients are changes in pixel intensity due to the change in the location of the pixel: Wikipedia.
  - ⇨ Histogram of Gradients (HOG) can be used as a features for images and is often combined with SVM for face detection and recognition tasks: Wikipedia.
  - ▶ Math Note (Optional)
  - ▼ Math Note (Optional)
  - For HOG features: in every 8 by 8 pixel region of an image, the gradient vectors $\begin{bmatrix} \nabla_x I(s,t) \\ \nabla_y I(s,t) \end{bmatrix}$ are put into 9 orientation bins, for example, $\left[0, \frac{2}{9}\pi\right], \left[\frac{2}{9}\pi, \frac{4}{9}\pi\right], \ldots, \left[\frac{16}{9}\pi, 2\pi\right]$, and the histogram count is used as the HOG features.
  - The resulting bins are normalized within a block of 2 by 2 regions.
  - Scale Invariant Feature Transform (SIFT) produces similar feature representation using histogram of oriented gradients: Wikipedia.
  - ⇨ It is location invariant.
  - ⇨ It is scale invariant: images at different scales are used to compute the features.
  - ⇨ It is orientation invariant: dominant orientation in a larger region is calculated and all gradients in the region are rotated by the dominant orientation.
  - ⇨ It is illumination and contrast invariant: feature vectors are normalized so that they sum up to 1, and thresholded (values smaller than a threshold, for example 0.2, are made 0).
- Deep learning convolution
  - The features can be engineered using computer vision techniques such as HOG or SIFT.
  - They can also be learned as hidden units in a neural network. These neural networks are call convolutional neural networks (CNN): Link, Link, Wikipedia.
  - Instead of activation units $a = g\left(w^\top x + b\right)$ or $a = g\left(w \cdot x + b\right)$, the dot product can be replaced by convolution (usually cross-correlation in practice, which is convolution without flipping the filters). The resulting matrix of activation units is called an activation map computed as $A = g\left(W \star x + b\right)$.
  - ⇨ For filters in CNN, zero padding means adding zeros around the image pixel matrices so that the activation maps have the same size as the image; and no padding means not adding zeros around the image so the activation maps will be $2k$ pixels smaller than the image in each dimension.
  - ⇨ Filters with a stride of s means the skipping $s-1$ pixels when moving the filter around when computing the convolution: a stride of 1 is the standard convolution; and a stride of $2k+1$ (filter size) is also called non-overlapping and each pixel is only used once in the computation of the convolution.
- Conv and pooling layers
  - 

    - In the conv layer, there are 2 filters with size 3, we need to train 2 * 3 * 3 = 18 weights (possibly 2 biases)
    - In the pooling layer, there are 2 filters, but none of the weights need to be trained
    - In the output layer, there are 18 units in the previous layer, fully connected with 3 output units, 54 weights and 3 biases

10/17

- Google net / inception net
    - 1*1 convolution
        - mlp on each pixel
        - Shrink the size of the output
    - Multiple softmax halfway
        - Fix vanishing gradient problem