Image Classification Using Convolutional Neural Nets

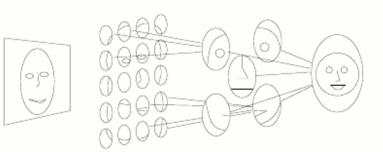
C. Aden, C. Qin, and N. Ulle

University of California, Davis

Introduction to CNN

Convolutional Neural Networks (CNNs) are variants of *multilayer perceptrons*, inspired by the cell arrangement within the visual cortex.

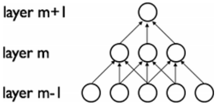
How it roughly works: Extracting different types of *local features* from the input image.



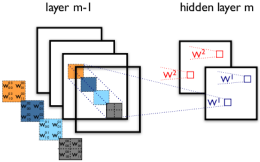
Key feature: Invariance to geometric transformations (*shift, scale and distortion*) of the input. Allows us to detect features regardless of position in a picture.

Elements of CNN

- A **convolutional layer** has the following two components:
  - *Local receptive fields* enable the network to extract elementary visual features, e.g., oriented edges, end-points, and corners.



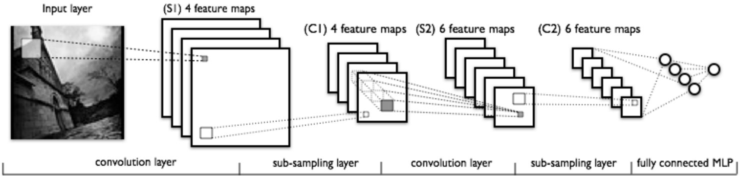
  - *Shared weights* allows for features to be detected regardless of their position in the visual field.



- **Sub-sampling layer** reduces the number of free parameters (spatial resolution) and provides translation invariance.

Architecture of CNN

A CNN consists of *alternating* convolution and sub-sampling layers.



The invariance to translation and distortion of the input is achieved by a progressive increase of the number of feature maps coupled with a progressive reduction of spatial resolution.

The OverFeat Program

- Convolutional Neural Network, pre-trained on the ImageNet image hierarchy database (14.2 million images).
- Written in C++, can be compiled to work on the GPU, or with a tuned BLAS.
- Extracts features from images, to be used in classification.

Our Approach

- Run OverFeat on the training images $(20,000)$ to generate features.
- Max-pool over the feature layers to reduce the space $(1 \times 4096$ for each image)
- Use the $20,000 \times 4096$ features matrix, along with labels to train a support vector machine.
- Run OverFeat on the $5,000$ testing images.
- Using the trained SVM classifier, assign 'cat' or 'dog' to each image and store the probability of that decision.

Results

Results through Cross-Validation on Training Data (5% held-out):

- Classifying based on pixel values with Radial Basis Function SVM: 57% (recall that guessing is about 50%).
- Classifying ImageNet labels from OverFeat: 88.7%.
- ImageNet feature layer with linear-kernel SVM: 95%
- ImageNet feature layer with cubic polynomial-based kernel SVM: 96%
- ImageNet feature layer with Radial Basis Function-based kernel SVM: 98%

Difficulties

- Non-dog/cat images (ex: picture of a rose, image of text, hand-drawn cat, 0KB image)
- Cats and dogs in strange poses
- Very small images (smaller than 100×100). They were enlarged in Python to allow the algorithm work, but classification rate suffered.