Neural networks

Computer vision - parameter sharing
Topics: computer vision

• We can design neural networks that are specifically adapted for such problems
  ‣ must deal with very high-dimensional inputs
    - 150 x 150 pixels = 22500 inputs, or 3 x 22500 if RGB pixels
  ‣ can exploit the 2D topology of pixels (or 3D for video data)
  ‣ can build in invariance to certain variations we can expect
    - translations, illumination, etc.

• Convolutional networks leverage these ideas
  ‣ local connectivity
  ‣ parameter sharing
  ‣ pooling / subsampling hidden units
**Topics:** parameter sharing

- Second idea: share matrix of parameters across certain units
  - units organized into the same “feature map” share parameters
  - hidden units within a feature map cover different positions in the image

Same color = same matrix of connections
Topics: parameter sharing

• Second idea: share matrix of parameters across certain units
  ‣ units organized into the same “feature map” share parameters
  ‣ hidden units within a feature map cover different positions in the image

same color = same matrix of connections
Topics: parameter sharing

- Second idea: share matrix of parameters across certain units
  - units organized into the same “feature map” share parameters
  - hidden units within a feature map cover different positions in the image

same color = same matrix of connections
Topics: parameter sharing

- Second idea: share matrix of parameters across certain units
  - units organized into the same “feature map” share parameters
  - hidden units within a feature map cover different positions in the image
Topics: parameter sharing

- Second idea: share matrix of parameters across certain units
  - units organized into the same “feature map” share parameters
  - hidden units within a feature map cover different positions in the image

\[ W_{ij} \text{ is the matrix connecting the } i^{th} \text{ input channel with the } j^{th} \text{ feature map} \]
Topics: parameter sharing

- Solves the following problems:
  - reduces even more the number of parameters
  - will extract the same features at every position (features are “equivariant”)

\[
W_{ij} \text{ is the matrix connecting the } i^{th} \text{ input channel with the } j^{th} \text{ feature map}
\]
Each feature map forms a 2D grid of features:

- can be computed with a discrete convolution (*) of a kernel matrix $k_{ij}$ which is the hidden weights matrix $W_{ij}$ with its rows and columns flipped.

The tanh-squashed filter bank is a completion layer and by an average down-sampling employed by architectures [28, 25], which alternate tanh-squashed filter banks with max-pooling layers.

This is the basic building block of transforms used in computer vision.

For instance, a filter bank layer $F_{ij}$ is defined as a uniform weighting over the columns of the $5 \times 5$ input, and connects input feature map.

Max-Pooling and Subsampling Layer -

Two such stages, followed by a classifier, generally a multinomial logistic regression.

This module simply applies parameter sharing.

Given a particular architecture, a number of training procedures are used to find parameters.

Random Features and Supervised Classifier -

At each stage, a learning rule is used to learn the parameters that optimize the output (could have added a bias).

The $F_{ij}$ is a trainable scalar coefficient (G).

In the following, superscripts are used to identify the pose of this layer is to build robustness to small distortions, denoted by a superscript, so that, an average pooling while the spatial resolution is decreased.

Disregarding the spatial resolution is decreased. Each output feature map is then performed local subtractive and divisive normalizations, e.g.,

$$\sum_{i,j,k} \text{denominator is the weighted standard deviation of all features (would have a bias) normalized so that}$$

$$a_{G \text{ a u s s i a n} \text{ window (of size } 9 \times 9 \text{ into each window)}}$$

A negative normalization operation for a given site is performed, including the positive part, and produced similar results to other architectures [24, 20].

The tanh-squashed filter bank is a typical building block of transform transforms and by an average downsampling. Recognition architectures are composed of one or additional convolutional networks, alternating tanh-squashed filter banks with max-pooling layers.

Each convolutional layer is identified by a letter.