Pooling and Invariance in Convolutional Neural Networks
General Neural Networks

Compositions of linear maps and component-wise non-linearities
Neural Networks
Common Non-Linearities

Rectifier Linear Unit

Sigmoid

Hyperbolic tangent
Biological Inspiration

Neurons diagram

Rectified Linear Unit
Sparse Connections

Not required to be fully connected
Backpropagation

A backpropagation network trains with a two-step procedure. The activity from the input pattern flows forward through the network, and the error signal flows backward to adjust the weights.
Representations Learnt
Representation Learning

Images far apart in Euclidean space

Need to find representation such that members of same class are mapped to similar values
Convolutional Neural Network

Compositions of:

Convolutions by Linear Filter
Thresholding Non-Linearities
Spatial Pooling
Convolution by Linear Filter

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

Convolution kernel (emboss)

New pixel value (destination pixel)
Convolution by Linear Filter
Example
Convolutional Neural Networks

1) Convolution by Linear Filter

2) Apply non-linearity

3) Pooling
Convolutional Neural Network
Convolutional Neural Networks
Imagenet Network
Successes

Computer vision
Speech
Chemistry
Object Classification
Segmentation
Object Detection
Speech
Physical Chemistry

Successfully predict atomization energy, polarizability, frontier orbital eigenvalues, ionization potential, electron affinity and excitation energies from molecular structure.
Visualization of First Layer
Standard Pooling Mechanisms

Ave pooling

Max pooling
Example
Heterogenous Pooling

Some filters passed to Ave pooling

Others filters passed to Max pooling
Pooling Continuum

\[\text{ave} \iff l_1\]

\[\text{max} \iff l_\infty\]

Accordingly, LeCun et al. 2012 ran experiment with variety of “p” values.
Results along spectrum

Optimal for this SVHN dataset was $p = 4$. 
L_p learnt pooling

Why not learn optimal p for each filter map?
Stochastic Pooling

- a) Image
- b) Filter
- c) Rectified Linear
- d) Activations, $a_i$
- e) Probabilities, $p_i$
- f) Sampled Activation, $s$
Stochastic Pooling

Expectation at Test Time

\[ s_j = \sum_{i \in R_j} \rho_i \cdot a_i \equiv \frac{l_2 \cdot l_2}{l_1} \]
Entropy Pooling

Extend to variable $p$

In particular, $\frac{l_{p+1} \ast l_{p+1}}{l_p}$

Alternative: $\frac{l_\infty \ast l_\infty}{l_1}$
Max-Out pooling

Pooling across filters

Substantial Improvement in performance and allowed depth
Example
Compete Neurons

Neurons can suppress other responses

(a) max-pooling

(b) LWTA
Visualizations of Filters

Early Layers
Visualizations
Visualizations
Invariance under Rigid Motion

Goodfellow et al. 2009 demonstrated the CNN are invariant under $D_n$

Indeed, depth of NN critical to establishing such invariance
Unstable under Deformation

Szegedy et al.
Lipshitz Bounds for Layers

Max and ReLU are contractive

FC Layers: usual linear operator norm

Conv Layers: Parseval’s and DFT yield explicit formula

\[ \|f(x + r) - f(x)\| \leq C \cdot \|r\| \]
Solutions?

Regularize Lipshitz operator
Coding Symmetry

Convolutional Wavelet Networks
Architecture

Wavelet convolutions composed with modulus operator
Gabor Wavelets

Trigonometric function in Gaussian Envelope

\[ f(x) = e^{-(x-x_0)^2/a^2} e^{-ik_0(x-x_0)} \]
Group Convolution

\[ Y \otimes Z(g) = \sum_{h \in G} Y(h) Z(h^{-1} g) \]

\[ x \ast \phi(u) = \sum_{v} x(v) \phi(u - v) \]
Group Invariant Scattering

Main Result: Conv. Wavelet Networks are translation invariant functions in $L_2(\mathbb{R}^2)$

Furthermore, CWN can be made invariant to action under any compact Lie group.

$$\|\Phi(f) - \Phi(L_\tau f)\|_{\mathcal{H}} \leq C \|f\| \left( \sup_{x \in \mathbb{R}^d} |\nabla \tau(x)| + \sup_{x \in \mathbb{R}^d} |H_\tau(x)| \right)$$
Textures

Sifre and Mallat

<table>
<thead>
<tr>
<th></th>
<th>State-of-the-art [1, 2]</th>
<th>Translation Scattering</th>
<th>Roto-trans Scattering</th>
<th>+ log</th>
<th>+ scale processing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.6</td>
<td>50</td>
<td>23</td>
<td>16</td>
<td>7.7</td>
</tr>
<tr>
<td>3.0</td>
<td>35</td>
<td>10</td>
<td>6</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>20</td>
<td>3.3</td>
<td>1.8</td>
<td>0.6</td>
<td></td>
</tr>
</tbody>
</table>
Basis for images

Learns similar representation as Imagenet CNN

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Layers</th>
<th>Calc.-101</th>
<th>Calc.-256</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCATTERING</td>
<td>1</td>
<td>51.2±0.8</td>
<td>19.3±0.2</td>
</tr>
<tr>
<td>IMAGE NET CCN</td>
<td>1</td>
<td>44.8 ± 0.7</td>
<td>24.6±0.4</td>
</tr>
<tr>
<td>SCATTERING</td>
<td>2</td>
<td>68.8±0.5</td>
<td>34.6±0.2</td>
</tr>
<tr>
<td>IMAGE NET CCN</td>
<td>2</td>
<td>66.2±0.5</td>
<td>39.6±0.3</td>
</tr>
<tr>
<td>IMAGE NET CCN</td>
<td>3</td>
<td>72.3±0.4</td>
<td>46.0±0.3</td>
</tr>
<tr>
<td>IMAGE NET CCN</td>
<td>7</td>
<td>85.5±0.4</td>
<td>72.6±0.2</td>
</tr>
</tbody>
</table>
Learnt Invariances
Optimal network

1) Encoded symmetry
2) Regularize Lipshitz coefficients
3) Compete Neurons in final layers
Final words