Visual Object Recognition
Computational Models and Neurophysiological Mechanisms
Neuro 130/230. Harvard College/GSAS 78454
Visual Object Recognition
Computational Models and Neurophysiological Mechanisms
Neurobiology 230. Harvard College/GSAS 78454

Class 1 [09/01/2021]. Introduction to Vision
Note: no class on 09/06/2021
Class 2 [09/13/2021]. Natural image statistics and the retina
Class 3 [09/20/2021]. The Phenomenology of Vision
Class 4 [09/27/2021]. Learning from Lesions
Class 5 [10/04/2021]. Primary Visual Cortex
Note: no class on 10/11/2021
Class 6 [10/18/2021]. Adventures into terra incognita
Class 7 [10/25/2021]. From the Highest Echelons of Visual Processing to Cognition
Class 8 [11/01/2021]. First Steps into in silico vision [Will Xiao]
Class 9 [11/08/2021]. Teaching Computers how to see
Class 10 [11/15/2021]. Computer Vision
Class 11 [11/22/2021]. Connecting Vision to the rest of Cognition

FINAL EXAM, PAPER DUE 12/14/2021. No extensions.
A big happy family of neural networks

https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464
Deep convolutional neural networks: AlexNet

Krizhevsky et al. 2012
Formulation of the visual recognition problem
A more ambitious formulation

- Explain human perception
- Explain neural mechanisms
- Classification

Image (or video)
A brief history of computational models

Hubel and Wiesel, simple and complex cells (1950s’)

Neocognitron (Fukushima 1980)

HMAX (Poggio 1999), Work on MNIST (LeCun 1998)

Deep convolutional neural networks (circa 2012)
Some of the typical computational operations

- Convolution
- Normalization
- ReLU
- Pooling
The convolution operation

\[ f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \]
ReLU

\[ f(u) = \max(0, u) \]
Max pooling

The image illustrates the process of max pooling in a 2D grid. The original grid contains the following values:

<table>
<thead>
<tr>
<th></th>
<th>13</th>
<th>220</th>
<th>117</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>65</td>
<td>54</td>
<td>145</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>41</td>
<td>67</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>89</td>
<td>198</td>
<td>28</td>
<td></td>
</tr>
</tbody>
</table>

After max pooling, the grid is reduced to:

<table>
<thead>
<tr>
<th></th>
<th>220</th>
<th>117</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>198</td>
<td></td>
</tr>
</tbody>
</table>
The HMAX model

Riesenhuber and Poggio 1999
The HMAX model

Serre and Poggio 2007
The model captures the effects of clutter in visual responses.
The model captures selectivity and invariance in V4 responses to curvatures

Cadieu et al 2007
The model approximates decoding of object information from IT cortex.

Serre et al. 2007
The model captures rapid recognition behavior

The model is able to predict the level of performance of human observers (overall 82% for the model vs. 80% for human observers). For both the model and human observers, the level of performance is highest on the close-body condition and drops gradually as the amount of clutter increases in the image from close-body to medium-body and far-body. (Adapted with permission from Serre et al., 2007, Fig. 3A.)
### Traditional approaches to visual recognition

<table>
<thead>
<tr>
<th>Hand-crafted features</th>
<th>Learn readout (e.g. Support Vector Machine)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example hand-crafted features</td>
<td></td>
</tr>
<tr>
<td>- Edges</td>
<td></td>
</tr>
<tr>
<td>- Textures</td>
<td></td>
</tr>
<tr>
<td>- Colors</td>
<td></td>
</tr>
<tr>
<td>- Corners</td>
<td></td>
</tr>
<tr>
<td>- Principal components</td>
<td></td>
</tr>
<tr>
<td>- Spatial frequency decomposition</td>
<td></td>
</tr>
<tr>
<td>- SIFT (Scale-invariant feature transform)</td>
<td></td>
</tr>
</tbody>
</table>

**Image (or video)** | **Hand-crafted features** | **Learn readout** (e.g. Support Vector Machine)
Deep learning

Learn features and readout

Image (or video)
The credit assignment problem

Motor outputs

Synaptic changes
Back-propagation

\[ E_{total} = E_{o1} + E_{o2} = 0.5[(target_{o1} - o_1)^2 + (target_{o2} - o_2)^2] \]

\[ o_1 = \frac{1}{1 + e^{-net_{o1}}} \]

\[ net_{o1} = h_1 \ast w_{ho}(1,1) + h_2 \ast w_{ho}(2,1) + b_{ho} \]

\[ h_1 = \frac{1}{1 + e^{-net_{h1}}} \]

\[ net_{h1} = i_1 \ast w_{ih}(1,1) + i_2 \ast w_{ih}(2,1) + b_{ih} \]
Back-propagation

\[ E_{total} = E_{o_1} + E_{o_2} \]

\[ E_{o_1} = 0.5 (\text{target}_{o_1} - o_1)^2 \]

\[ \frac{\partial E_{total}}{\partial (w_{ho2,1})} = \frac{\partial E_{total}}{\partial o_1} \cdot \frac{\partial o_1}{\partial \text{net}_{o_1}} \cdot \frac{\partial \text{net}_{o_1}}{\partial (w_{ho2,1})} \]

\[ \frac{\partial E_{total}}{\partial o_1} = 2 \cdot 0.5 \cdot (o_1 - \text{target}_{o_1}) \]

\[ \frac{\partial E_{total}}{\partial \text{net}_{o_1}} = 2 \cdot 0.5 \cdot (o_1 - \text{target}_{o_1}) \]

\[ \frac{\partial \text{net}_{o_1}}{\partial (w_{ho2,1})} = h_2 \]

\[ w_{ho}(2,1) \rightarrow w_{ho}(2,1) - \epsilon \cdot \frac{\partial E_{total}}{\partial w_{ho}(2,1)} \]
Back-propagation

\[ E_{total} = E_{o_1} + E_{o_2} \]
\[ E_{o_1} = 0.5(\text{target}_{o_1} - o_1)^2 \]
\[ E_{o_2} = 0.5(\text{target}_{o_2} - o_2)^2 \]
Is back-propagation biologically plausible?

Symmetric feed-forward and feed-back weights

Signed error signals

Large gradients

Feedback alters neuronal activity (and weights only indirectly)

Supervised learning requires many training examples
Deeper and deeper

VGG (2014) – 6.8%
Baidu (2015) – 5.33%

MSRA (2015) – 4.94%
- Spatial pyramid pooling
- Optimized PReLU
- Improved (random) initialization

GoogleNet (2014) – 6.67%
- Inception module
- Multi-scale convolutions (including 1x1 filters)
- Minimal dense layers
- Auxiliary classifiers

AlexNet (2012) – 15.3%
Clarifai (2013) – 11.7%
Putting it all together
To err is human and algorithmic
Putting it all together
CNNs in action: example

Kreiman, 2019
# Computational models versus biology

<table>
<thead>
<tr>
<th></th>
<th>Biology</th>
<th>Computer vision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Receptive field increase through hierarchy</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Convolution-like operations</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Backpropagation</td>
<td>?</td>
<td>✔️</td>
</tr>
<tr>
<td>Supervised learning</td>
<td>~</td>
<td>✔️</td>
</tr>
<tr>
<td>Unsupervised learning</td>
<td>✔️</td>
<td>~</td>
</tr>
<tr>
<td>Interactions between areas</td>
<td>✔️</td>
<td>~</td>
</tr>
</tbody>
</table>
Summary

- Visual recognition ~ extraction of task-dependent adequate features plus read-out
- Computation emerges from combination of simple elementary functions: convolution, normalization, rectification, pooling
- Hierarchical models capture essential neural and behavioral properties of visual processing
- Weights can be learned via back-propagation
- Current models provide only a coarse approximation to the complexities of the visual system