CS 4824/ECE 4424: Convolutional Neural Networks II

Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.
Digit Recognition

![Diagram of a convolutional neural network for digit recognition](image.png)
Benefits of CNN

- Sparse interactions
  - Fewer connections

- Parameter sharing
  - Fewer weights

- Locally equivariant representation
  - Locally invariant to translations
  - Handle inputs of varying length
Parameters

- **# of filters**: integer indicating the number of filters applied to each window

- **Kernel size**: tuple (width, height) indicating the size of the window

- **Stride**: tuple (horizontal, vertical) indicating the horizontal and vertical shift between each window

- **Padding**: “valid” or “same”. Valid indicates no input padding. Same indicates that the input is padded with a border of zeros to ensure that the output has the same size as the input
Examples

1. Kernel (2, 2), stride (1, 1), padding = "valid"
2. Kernel (3, 3), stride (1, 1), padding = "same"
3. Kernel (2, 2), stride (2, 2), padding = "valid"
Training CNN

- Convolutional neural networks are trained in the same way as other neural networks through backpropagation
  - AdaGrad, RMSprop, Adam

- Weight sharing:
  - Combine gradients of shared weights into a single gradient
Architecture design

- What is the preferred filter size?
  - VGG (Visual Geometry Group at Oxford, 2014): stack of small filters is often preferred to single large filter
    - Fewer parameters
    - Deeper network
  - Schematic:
Residual Networks

- **Idea**: Addressing vanishing gradient problem by introducing residual connections (a.k.a. skip connections) to shorten paths (He et al. 2015)

- **Schematic:**
Applications

- Speech Recognition
- Image recognition
- Machine translation
- Control
- ...
- Data with sequential, spatial or tensor patterns
Image Recognition

- Convolutional Neural Network
  - With rectified linear units and dropout
  - Data augmentation for transformation invariance
ImageNet Breakthrough

- Results: ILSVRC-2012
  - Krizhevsky, Sutskever, Hinton

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1 (val)</th>
<th>Top-5 (val)</th>
<th>Top-5 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT + FVs [7]</td>
<td>—</td>
<td>—</td>
<td>26.2%</td>
</tr>
<tr>
<td>1 CNN</td>
<td>40.7%</td>
<td>18.2%</td>
<td>—</td>
</tr>
<tr>
<td>5 CNNs</td>
<td>38.1%</td>
<td>16.4%</td>
<td>16.4%</td>
</tr>
<tr>
<td>1 CNN*</td>
<td>39.0%</td>
<td>16.6%</td>
<td>—</td>
</tr>
<tr>
<td>7 CNNs*</td>
<td>36.7%</td>
<td>15.4%</td>
<td>15.3%</td>
</tr>
</tbody>
</table>

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were “pre-trained” to classify the entire ImageNet 2011 Fall release. See Section 6 for details.
ImageNet Breakthrough

- From Krizhevsky, Sutskever, Hinton