ConvLens: Visualizing Internal Components of Convolutional Neural Networks

Hooman Shariati
Mahdi Ghodsi
INTRODUCTION
What’s a ConvNet?

- a Supervised Machine Learning Algorithm

Used in:
- Image and Video Recognition
- Recommender Systems
- Natural Language Processing
Recognition + Localization

Source: Kaiming He, ICCV 2015
Image Captioning

Source: Kuarrpaeth yi anq'd F-e i-8Fei, “Deep Visual-Semantic Alignments for Generating Imag6e D eDsceriptio n2s”, 0 CViP6R 2015
Problem

“Neural networks have long been known as “black boxes” because it is difficult to understand exactly how any particular, trained neural network functions due to the large number of interacting, non-linear parts.”

Yajin Zhou
Alexnet

- [227x227x3] INPUT
- [55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
- [27x27x96] MAX POOL1: 3x3 filters at stride 2
- [27x27x96] NORM1: Normalization layer
- [27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
- [13x13x256] MAX POOL2: 3x3 filters at stride 2
- [13x13x256] NORM2: Normalization layer
- [13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
- [13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
- [13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
- [6x6x256] MAX POOL3: 3x3 filters at stride 2
VggNet

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>A-LRN</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 weight layers</td>
<td>11 weight layers</td>
<td>13 weight layers</td>
<td>16 weight layers</td>
<td>16 weight layers</td>
<td>19 weight layers</td>
<td></td>
</tr>
<tr>
<td><strong>conv3-64</strong></td>
<td><strong>conv3-64</strong></td>
<td><strong>LRN</strong></td>
<td><strong>conv3-64</strong></td>
<td><strong>conv3-64</strong></td>
<td><strong>conv3-64</strong></td>
<td><strong>conv3-64</strong></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-128</strong></td>
<td><strong>conv3-128</strong></td>
<td><strong>conv3-128</strong></td>
<td><strong>conv3-128</strong></td>
<td><strong>conv3-128</strong></td>
<td><strong>conv3-128</strong></td>
<td><strong>conv3-128</strong></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-256</strong></td>
<td><strong>conv3-256</strong></td>
<td><strong>conv3-256</strong></td>
<td><strong>conv3-256</strong></td>
<td><strong>conv1-256</strong></td>
<td><strong>conv3-256</strong></td>
<td><strong>conv3-256</strong></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv1-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
</tr>
<tr>
<td><strong>maxpool</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
<td><strong>conv3-512</strong></td>
</tr>
<tr>
<td><strong>FC-4096</strong></td>
<td><strong>FC-4096</strong></td>
<td><strong>FC-4096</strong></td>
<td><strong>FC-1000</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The 6 different architectures of VGG Net. Configuration D produced the best results.

Source: Simonyan and Zisserman, ArXiv 2014
To Optimize a ConvNet:

1. Need to understand its structure
2. Need to know what it has learned given its training
Existing Visualizations
Structure: Mental Model

Source: Krizhevsky, Alex, 2013
Structure: Too Much

Source: Hoo-Chang Shin, IEEE 2013
Structure: Too Little

Source: Holger R. Roth, ArXiv 2015
Structure: Seems Right

Source: Sven Behnke, 2013
Learning: Forward Activation
Learning: Guided Back Propagation

Input  BackPropagation  Guided BackProp
Existing Tools: TensorBoard
Existing Tools: VisToolBox
Existing Tools: Harley's
Existing Tools: CNNVis
Existing Tools: ReVACNN
Our Design Process
What: Data

Hierarchichal Network

Layers:
Type: Categorical
Input: Q,S
Output: Q,S
Rank: Q,S

Conv Layer: 9 Q,S Attributes
Norm Layer: 4 Q,S Attributes
Conv Layer: 9 Q,S Attributes
Pool Layer: 3 Q,S Attributes
ReLU Layer: 0 Attributes
FC Layer: Network of nodes and edges with Q,D weights
Drop Layer: 1 Q,S Attribute
Prob Layer: 1 Q,S Attribute/Class

Filter: Network of nodes and edges with Q,D weights
What: Derived

- For each Filter:
  - Activation Images: Diverging
  - Guided Back Propagation: Sequential
Why: Tasks

- **Explore → Summarize**
  - Over all Architecture of ConvNet
  - Parameters in each layer

- **Locate → Identify**
  - Filters that have learned useful features
  - Filters that are useless
Demo
How: Encode

- Layers: Nodes in a chain
- Rank: Position in the chain, text label on the node
- Output size: Text Label on chain links
- All other attributes: Extra information on click
- Filter number (inside a conv layer): Text label and position in the stack of images.
How: Facet

- Juxtaposed and Coordinate Side-by-Side Views
- Shared data (Magnified, and related content)
How: Manipulate

- Select
  - a layer to show more details for
  - a specific filter to show more details for
  - A filter visualization method to be used for the overview of filters

- Annotate useful/ useless filters
How: Aggregate

- Aggregation: Grouped all the nodes with the same depth into a single layer

- Aggregation: Grouped the information contained in all nodes and edges of each filter into a single image

- Filtering: Eliminated low probability outcomes from Prob Layers.
How: Reduce

- Our Guided BackPropagation reduces the data by eliminating negative weights
How: Scale

- Layers: Hundreds
  - No limit on the complexity of layers.
- Filters: Hundreds
  - No limit on the complexity of the NN inside filters.
Future Works & Limitations

- More dynamic front-end for screens with different aspect ratios
- Display the filter visualization images in a sorted order according to the importance of the filter.
  - Importance is implied by the sizes of weights in each filter.
- Find a meaningful visualization for the Fully Connected layers.
  - Deep Dream is one possible solution, but it only shows what the network has learned during training.
- Integrate the data generation tools (scripts) into ConvLens.
  - This requires heavy computation on the server side.