CNN Architectures

April 22\textsuperscript{nd}, 2021
The ConvNet Building Blocks

Conv Layer

Max Pooling

FC Layer

Batch Norm

Activation Func
Visual Recognition Challenge (ILSVRC)
- 1.2M training images
- 100K testing images
- 1K categories

AlexNet

- Fukushima 1980
- LeCunn et al. 1989
- LeCunn et al. 1998

AlexNet

Details:

- ReLU activation
- Max Pooling
- Dropout
- Batch
- SGD Momentum
ZFNet

VGG

Receptive Field  5
# Parameters  25C²
# FLOPs  9C²+9C² = 18C²

VGG

Receptive Field  
# Parameters  
# FLOPs  
Memory Size (Output)

VGG

Receptive Field
# Parameters
# FLOPs
Memory Size (Output)

5
25C²
25C²HW
HWC

5
9C²+9C² = 18C²
18C²HW
2HWC

GoogLeNet

ResNet

- Lin et al. 2010: 28.2
- Sanchez & Perronnin 2011: 25.8
- AlexNet 2012: 16.4
- ZFNet 2013: 11.7
- VGG 2014: 7.3
- GoogLeNet 2014: 6.7
- ResNet 2015: 3.57

ResNet
ResNet

![Graph showing training and test error over iterations for 56-layer and 20-layer ResNet models.](image-url)
Residual Building Block

\[ x \rightarrow 3\times3 \text{ conv, } C \rightarrow 3\times3 \text{ conv, } C \rightarrow F(x) \]
Residual Building Block

\[ F(x) = F(x) + x \]
Residual Building Block

\[ F(x) + x \]

3X3 conv, C

3X3 conv, C

\[ F(x) \]
Residual Building Block

\[ F(x) + x \]
ResNet Architecture

![ResNet Architecture Diagram]

- AlexNet
- VGG
- ResNet
- Effi. Net
- Tran. lr

**Parameters**
- Memory Size (MB)
- GFLOPS

**VGG Layer**
- Memory Size
- GFLOPS
- Parameters

**Diagram Details**
- Convolutional layers labeled with 'conv, 64' or 'conv, 128'
- Pooling layers labeled with 'pool, /2'
- Fully Connected layers labeled with 'fc'

**Graphs**
- Memory Size vs. GFLOPS for VGG layers
- Parameters for each VGG layer

---

DL4CV Weizmann
ResNet Architecture

![Diagram of ResNet Architecture]

- **Parameters**
- **Memory Size (MB)**
- **GFLOPs**

**Layers:**
- AlexNet
- VGG
- ResNet
- EffiNet
- Tranlr

**Layers:**
- Convolutional layers
- Pooling layers
- Fully Connected (FC) layers
Global Average Pooling

- Average Pooling: $k=7$
- Fully Connected Layer

### Memory Size (MB)

<table>
<thead>
<tr>
<th>VGG Layer</th>
<th>Memory Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>7x7x512</td>
<td>120</td>
</tr>
<tr>
<td>1x1x512</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
</tr>
</tbody>
</table>

### GFLOPs

<table>
<thead>
<tr>
<th>VGG Layer</th>
<th>GFLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>7x7x512</td>
<td>10000</td>
</tr>
<tr>
<td>1x1x512</td>
<td>10000</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
</tr>
</tbody>
</table>

### Parameters

<table>
<thead>
<tr>
<th>VGG Layer</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>7x7x512</td>
<td>16</td>
</tr>
<tr>
<td>1x1x512</td>
<td>17</td>
</tr>
<tr>
<td>1000</td>
<td>18</td>
</tr>
<tr>
<td>1000</td>
<td>19</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
More architectures

**Accuracy**
- DenseNet, Huang et al. 2017
- ResNext, Xie et al. 2017
- SENet, Hu et al. 2018

**Efficiency**
- MobileNet, Howard et al. 2017
- ShuffleNet, Zhang et al. 2018
- Neural Architecture Search, Zoph and Le. 2017
EfficientNet

EfficientNet

#channels

layer_i

resolution HxW

(a) baseline
The scaled model

Conv Layer: $F_i \left(X_{<H_i,W_i,C_i>}\right)$

Conv Stage: $F_{iL} \left(X_{<H_i,W_i,C_i>}\right)$

Conv Network: $\bigcirc_{i=1,...,s} F_{iL} \left(X_{<H_i,W_i,C_i>}\right)$

The scaled model:

$\phi$ is our scaling factor

$d, r, w$ are hyper-parameters

$\bigcirc_{i=1,...,s} F_{i}^{\phi L} \left(X_{<r^\phi H_i, r^\phi W_i, w^\phi C_i>}\right)$

# FLOPs will increase by $2^\phi$
The scaled model:

The scaled model:

\[ \bigcirc_{i=1,...,s} F_i^{d \phi L_i} \left( X_{<r \phi H_i, r \phi W_i, w \phi C_i>} \right) \]

# FLOPs will increase by \(2^\phi\)

Find \(d, r,\) and \(w:\)

1. Fix \(\phi = 2\)
2. Small grid search for \(d, r, w\)
Results
CNN architecture

• Design your network according to your task and resources

• Take into consideration
  • # Parameters
  • # FLOPs
  • Memory Size

• Use identity mapping (Skip connections)

• For most cases, use existing architecture

• For most cases, use a pre-trained model
Transfer Learning

\[ x \in \mathbb{R}^D \]

\[ W \in \mathbb{R}^{D \times 1000} \]

\[ W \in \mathbb{R}^{1000} \]
Transfer Learning

$\mathbf{x} \in \mathbb{R}^D$

$W \in \mathbb{R}^{D \times L}$
Questions?