CS6501: Deep Learning for Visual Recognition

CNN Architectures
ILSVRC:
Imagenet Large Scale Visual Recognition Challenge [Russakovsky et al 2014]
The Problem: Classification

Classify an image into 1000 possible classes:
e.g. Abyssinian cat, Bulldog, French Terrier, Cormorant, Chickadee,
red fox, banjo, barbell, hourglass, knot, maze, viaduct, etc.

cat, tabby cat (0.71)
Egyptian cat (0.22)
red fox (0.11)
.....
The Data: ILSVRC

Imagenet Large Scale Visual Recognition Challenge (ILSVRC): Annual Competition

1000 Categories

~1000 training images per Category

~1 million images in total for training

~50k images for validation

Only images released for the test set but no annotations, evaluation is performed centrally by the organizers (max 2 per week)
The Evaluation Metric: Top K-error

True label: Abyssinian cat

<table>
<thead>
<tr>
<th>Top-k error</th>
<th>Top-k accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

cat, tabby cat (0.61)  
Egyptian cat (0.22)  
red fox (0.11)  
Abyssinian cat (0.10)  
French terrier (0.03)  
.....
Top-5 error on this competition (2012)
Alexnet (Krizhevsky et al NIPS 2012)

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky  
University of Toronto  
kriz@cs.utoronto.ca

Ilya Sutskever  
University of Toronto  
ilya@cs.utoronto.ca

Geoffrey E. Hinton  
University of Toronto  
hinton@cs.utoronto.ca
Alexnet

https://www.saagie.com/fr/blog/object-detection-part1
Pytorch Code for Alexnet

• In-class analysis

https://github.com/pytorch/vision/blob/master/torchvision/models/alexnet.py
Dropout Layer

Happens for every batch for a different set of connections only during training

(a) Standard Neural Net  (b) After applying dropout.

Srivastava et al 2014

Important

model.train()
model.eval()
Preprocessing and Data Augmentation
Preprocessing and Data Augmentation
Preprocessing and Data Augmentation

224x224
Preprocessing and Data Augmentation

224x224
True label: Abyssinian cat
Some Important Aspects

• Using ReLUs instead of Sigmoid or Tanh
• Momentum + Weight Decay
• Dropout (Randomly sets Unit outputs to zero during training)
• GPU Computation!

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse coding [2]</td>
<td>47.1%</td>
<td>28.2%</td>
</tr>
<tr>
<td>SIFT + FVs [24]</td>
<td>45.7%</td>
<td>25.7%</td>
</tr>
<tr>
<td>CNN</td>
<td>37.5%</td>
<td>17.0%</td>
</tr>
</tbody>
</table>
What is happening?

https://www.saagie.com/fr/blog/object-detection-part1
Feature extraction (SIFT) → Feature encoding (Fisher vectors) → Classification (SVM or softmax)

Deep Learning

Convolutional Network (includes both feature extraction and classifier)
VGG Network

Simonyan and Zisserman, 2014.


https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py
Batch Normalization Layer

**Input:** Values of $x$ over a mini-batch: $\mathcal{B} = \{x_1...m\}$;  
Parameters to be learned: $\gamma$, $\beta$

**Output:** $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

\[
\mu_\mathcal{B} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \quad \text{ // mini-batch mean}
\]

\[
\sigma^2_\mathcal{B} \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_\mathcal{B})^2 \quad \text{ // mini-batch variance}
\]

\[
\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma^2_\mathcal{B} + \epsilon}} \quad \text{ // normalize}
\]

\[
y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i) \quad \text{ // scale and shift}
\]

GoogLeNet

https://github.com/kuangliu/pytorch-cifar/blob/master/models/googlenet.py

Szegedy et al. 2014

Further Refinements – Inception v3, e.g.

GoogLeNet (Inceptionv1)

Inception v3
ResNet (He et al CVPR 2016)

Sorry, does not fit in slide.

http://felixlaumon.github.io/assets/kaggle-right-whale/resnet.png

https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py
Revolution of Depth

- AlexNet, 8 layers (ILSVRC 2012)
- VGG, 19 layers (ILSVRC 2014)
- ResNet, 152 layers (ILSVRC 2015)

152 layers
Densenet

Input → Convolution → Dense Block 1 → Convolution → Pooling → Dense Block 2 → Convolution → Pooling → Dense Block 3 → Pooling → Linear → Prediction

“horse”
Densenet

Densenet

Object Detection
Object Detection as Classification

deer?
cat?
background?
Object Detection as Classification

CNN

deer?
cat?
background?
Object Detection as Classification

CNN

deer?
cat?
background?
Object Detection as Classification with Sliding Window

deer?
cat?
background?
Object Detection as Classification with Box Proposals
Box Proposal Method – SS: Selective Search

Segmentation As Selective Search for Object Recognition. van de Sande et al. ICCV 2011
RCNN

R-CNN: Regions with CNN features

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions


Questions?