Lecture 12:

Data Augmentation & Transfer Learning

Alireza Akhavan Pour

CLASS.VISION
Part 1: Review
Classic networks:

- LeNet-5
- AlexNet
- ZFNet
- VGG

Inception

ResNet
Review Resnet
Residual block

\[
\begin{align*}
  a^{[l]} & \rightarrow a^{[l+1]} & a^{[l+2]} \\
  z^{[l+1]} &= W^{[l+1]} a^{[l]} + b^{[l+1]} & a^{[l+1]} &= g(z^{[l+1]}) \\
  z^{[l+2]} &= W^{[l+2]} a^{[l+1]} + b^{[l+2]} & a^{[l+2]} &= g(z^{[l+2]} + a^{[l]})
\end{align*}
\]

[He et al., 2015. Deep residual networks for image recognition]
Residual Network

\[ \text{X} \rightarrow \text{ResNet} \rightarrow a^{[l]} \]

Plain

ResNet

Training error vs. # layers

He et al., 2015. Deep residual networks for image recognition

SRTTU – A.Akhavan
Case Study: ResNet [He et al., 2015]

MSRA @ ILSVRC & COCO 2015 Competitions

• **1st places in all five main tracks**
  - ImageNet Classification: “Ultra-deep” (quote Yann) **152-layer** nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

*Improvements are relative numbers


Slide from Kaiming He’s recent presentation
https://www.youtube.com/watch?v=1PGLj-uKT1w
Case Study: ResNet

[He et al., 2015]
ILSVRC 2015 winner (3.6% top 5 error)

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

ResNet, 152 layers (ILSVRC 2015)

2-3 weeks of training on 8 GPU machine
at runtime: faster than a VGGNet! (even though it has 8x more layers)

(slide from Kaiming He’s recent presentation)
Case Study: ResNet

[He et al., 2015]

Spatial dimension only 56x56!
Revolution of Depth

(slide from Kaiming He’s recent presentation)
Part 2: Data Augmentation
Data Augmentation

Load image and label

“cat”

CNN

Compute loss
Data Augmentation

Load image and label

“cat”

Transform image

Compute loss

CNN

Load image and label

“cat”

Transform image
Data Augmentation

- Change the pixels without changing the label
- Train on transformed data
- VERY widely used
Common augmentation method

1- Mirroring (Horizontal flips)

2- Random cropping
Data Augmentation

3. Color shifting (Color jitter)

- Color shifting
- $+20, -20, +20$
- $-20, +20, +20$
- $+5, 0, +50$
Data Augmentation
3. Color shifting (Color jitter)

Simple:
Randomly jitter contrast

Complex (PCA color augmentation):

1. Apply PCA to all [R, G, B] pixels in training set
2. Sample a “color offset” along principal component directions

(As seen in [Krizhevsky et al. 2012], ResNet, etc)
Data Augmentation

4. Get creative!

Random mix/combinations of:
- translation
- rotation
- stretching
- shearing, ...
- lens distortions, ... (go crazy)
Bottlenecks
to be aware of
Implementation distortions during training

Load → CPU thread → mini-batch → Training

Distortion

Hard Disk → CPU / GPU
Implementation distortions during training

- CPU thread
- Distortion
- Load
- Mini-batch
- CPU / GPU
- Training
- Hard Disk

SRTTU – A.Akhavan

Lecture 12 - 21
Part 3: Transfer Learning
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]
نگاهی به فرآیند بینایی در مغز!
چرا مغز نیاز به تبدیل تصاویر از محیط پیکسل دارد؟

DiCarlo and Cox, TICS (2007)
Invariant object recognition

Clutter, occlusion (intra-class)
فضای نرونی و خمینه (manifold)
فضای نرونی "خوب"

فضای نرونی "خوب"

Separating hyperplane
در هم تنیده، اطلاعات ضمنی از اشیاء جدایی‌نما، اطلاعات صریح و روشن از اشیاء تبدیل فضای نورونی در مغز

**DiCarlo and Cox, TICS (2007), DiCarlo, Zoccolan and Rust, Neuron (2012)**
Transfer Learning

“You need a lot of data if you want to train/use CNNs”
Transfer Learning

“You need a lot of data if you want to train/use CNNs.”
Transfer Learning
Transfer Learning

廊 • 
Persian Cat
Tiger Cat
Neither

$\textbf{x}$

$\textbf{freeze}$

$\text{softmax}$

$1000$

$\text{softmax}$

$P$

$T$

$N$

$\hat{y}$

\[ \begin{align*}
\text{میتوان از قبل خروجی لایه را برای تصاویر آموزشی حساب کرد.} \\
\checkmark
\end{align*} \]

SRTTU – A.Akhavan

Lecture 12 - 33
Transfer Learning

Persian Cat

Tiger cat

Neither

softmax

P

T

N

\[ T \]

\[ P \]

\[ N \]

Train

freeze

X → ...

X → ...

X → ...

X → ...

X → ...

X → ...

\[ \hat{y} \]
Transfer Learning
Transfer Learning with CNNs

1. Train on Imagenet

2. Small dataset: feature extractor
   - Freeze these
   - Train this

3. Medium dataset: finetuning
   - more data = retrain more of the network (or all of it)
   - Freeze these
   - Train this
Transfer Learning with CNNs

1. Train on Imagenet

2. Small dataset: feature extractor
   - Freeze these
   - Train this

3. Medium dataset: finetuning
   - more data = retrain more of the network (or all of it)
   - Freeze these
   - Train this
منابع

• https://www.slideshare.net/Alirezaakhavanpour/presentation10-68048331
• https://www.coursera.org/specializations/deep-learning
• http://cs231n.stanford.edu/