Musings on Continual Learning

Pulkit Agrawal
R-CNN, therefore, can be seen more broadly as a flexible point competition, and at the same time runs at 5 fps. Mask R-CNN surpasses the winner of the 2016 COCO key-point competition, and can be readily applied to detect instance-specific poses. Without tricks, binary mask, with minimal modification Mask R-CNN can be applied to detect instance-specific poses. Without tricks, binary mask, with minimal modification Mask R-CNN can be applied to detect instance-specific poses.

8-GPU machine. We believe the fast train and test speeds, and training on COCO takes one to two days on a single GPU. We will release code to facilitate future research.

Driven by the effectiveness of R-CNN and better accuracy. Faster R-CNN Advanced this…

Figure 2.

Without bells and whistles, Mask R-CNN surpasses all instance-level recognition and can be readily engineered entries from the 2016 competition winner. As a by-product, our method also excels on the COCO object detection task. In ablation experiments, we evaluate multi-stage RoIAlign, which couples segmentation and classification, and based on this, we found it essential to use RoIAlign instead of RoIPooling. In Figure 2, a seemingly minor change, RoIAlign has a large impact: it achieves significantly better performance for instance segmentation. The effect of RoIAlign is strongly amplified when combined with Mask R-CNN.

Instead, our method is based on an “fully convolutional instance segmentation” (FCIS). The framework is predicted a binary mask for each class independently, without tricks. With multi-stage RoIAlign, Mask R-CNN surpasses all state-of-the-art approaches on the COCO instance segmentation benchmark, with an absolute gain of 2.3%. Second, we found it essential to use RoIAlign in the RoIPooling stage. Without bells and whistles, Mask R-CNN surpasses all state-of-the-art approaches on the COCO instance segmentation benchmark, with an absolute gain of 2.3%.
What is a zebra?
What is a zebra?
Success in Reinforcement Learning

ATARI Games

~10-50 million interactions!

21 million games!

AlphaGo
Today’s AI

Task Specific

???

AI we want

Generalists
Learn to perform N tasks $\rightarrow$ Solve the (N+1)th task faster

or,
more complex task
Success on Imagenet
Training on N tasks —> Object classification knowledge

Knowledge for classification

Images from Imagenet

Is Elephant?
Is Sock?
Is Beaker?
Training on N tasks —> Object classification knowledge

Knowledge for classification

Images from Imagenet

Is Elephant?
Is Sock?
-is Beaker?
Reuse knowledge by fine-tuning

Apple?

Orange?
Imagenet: 1000 examples/class

New task: ~100 examples/class
Still need hundreds of “labelled” data points!

Fine-tuning with very few data points, won’t be effective!
Problem Setup

Training Set

Apple

\( x_1 \) \( y_1 \)

Orange

\( x_2 \) \( y_2 \)
Problem Setup

Training Set

Apple

\[ x_1 \quad y_1 \]

Orange

\[ x_2 \quad y_2 \]

Test

Apple or Orange?
Use Nearest Neighbors

Training Set

Apple

$x_1$

$y_1$

Orange

$x_2$

$y_2$

Test

Apple or Orange?

$x$

$z$
Use Nearest Neighbors

**Training Set**

\[ x_1 \quad y_1 \quad \text{Apple} \]

\[ x_2 \quad y_2 \quad \text{Orange} \]

**Test**

\[ x \quad \text{Apple or Orange?} \]

\[ z \]

\[ \mathbf{y}_k \]

\[ k = \arg \min_i \| z_i - z \|_2^2 \]
What does the performance depend on??

Training Set

Apple

$x_1$ $y_1$

Orange

$x_2$ $y_2$

Test

Apple or Orange?

$y_k$

$k = \arg \min_i \|z_i - z\|_2^2$
What does the performance depend on??

Training Set

Apple

Orange

Apple or Orange?

Features might not be optimized for matching!

\[ k = \arg \min_i \| z_i - z \|_2^2 \]
Metric Learning via Siamese Networks*

Instead of one v/s all classification

(*Hadsell et. al. 2006)
Metric Learning via Siamese Networks*

(*Hadsell et. al. 2006)
Metric Learning via Siamese Networks*

Same class: Output = 1

(*Hadsell et. al. 2006)
Metric Learning via Siamese Networks

\[ \phi; \theta_z \rightarrow z_1 \rightarrow f; \theta_f \rightarrow 1 \]

Same class: Output = 1

(*Hadsell et. al. 2006)
Metric Learning via Siamese Networks

Same class: Output = 1
Different class: Output = 0

(*Hadsell et. al. 2006)
Metric Learning via Siamese Networks

\[ \min_{\theta_z, \theta_f} \]

\[ \text{Same class: Output } = 1 \]

\[ \text{Different class: Output } = 0 \]

(*Hadsell et. al. 2006)
Solving using Siamese Network

Training Set

Apple

\[ x_1 \] \hspace{1cm} \[ y_1 \]

Orange

\[ x_2 \] \hspace{1cm} \[ y_2 \]

Test

Apple or Orange?
Solving using Siamese Network

Training Set

Apple

\[ x_1 \quad y_1 \]

Orange

\[ x_2 \quad y_2 \]

Siamese Net

0.1
Solving using Siamese Network

Training Set

\( x_1, y_1 \) Apple

\( x_2, y_2 \) Orange

Siamese Net

0.1

Siamese Net

0.8
Solving using Siamese Network

Training Set

Apple

Orange

$x_1$  $y_1$

$x_2$  $y_2$

Also look at Matching Networks, Vinyals et al. 2017

Siamese Net

0.1

Siamese Net

0.8
Another perspective

$\theta$: parameters after training on say Imagenet
Another perspective

Task 1: Apple v/s Orange

\( \theta \): parameters after training on say Imagenet
Another perspective

Task 1: Apple v/s Orange

\[ \theta : \text{parameters after training on say Imagenet} \]
Another perspective

Task 1: Apple v/s Orange
Task 2: Dog v/s Cat

$\theta$: parameters after training on say Imagenet
Another perspective

Task 1: Apple v/s Orange
Task 2: Dog v/s Cat

$\theta$: parameters after training on say Imagenet
Another perspective

Task 1: Apple v/s Orange
Task 2: Dog v/s Cat

Amount of fine-tuning: \[ \approx (\Delta \theta_1 + \Delta \theta_2) \]
What if?

Task 1: Apple v/s Orange
Task 2: Dog v/s Cat

Fine-tuning would be faster!
Can we optimize $\theta$ to make fine-tuning easier?

Amount of fine-tuning: $\approx (\Delta \theta_1 + \Delta \theta_2)$
How to do it?

Task 1: Apple v/s Orange

$$\min_{\theta} \mathcal{L}_{\tau_1}(f_{\theta})$$
How to do it?

Task 1: Apple v/s Orange

$$\min_{\theta} \mathcal{L}_{\tau_1}(f_\theta)$$

$$\theta'_1 = \theta - \alpha \nabla \mathcal{L}_{\tau_1}(f_\theta)$$

Hariharan et al. 2016, Finn et al. 2017
How to do it?

Task 1: Apple v/s Orange

\[
\min_{\theta} \mathcal{L}_{\tau_1}(f_{\theta})
\]

\[
\min_{\theta} \mathcal{L}_{\tau_1}(f_{\theta'})
\]

(i.e. train for fast fine-tuning!)

\[
\theta' = \theta - \alpha \nabla \mathcal{L}_{\tau_1}(f_{\theta})
\]

Hariharan et al. 2016, Finn et al. 2017
Generalizing to N tasks

Task1: Apple v/s Orange

\[ \min_{\theta} \mathcal{L}_{\tau_1}(f_{\theta}) \]

\[ \min_{\theta} \sum_{i} \mathcal{L}_{\tau_i}(f_{\theta_i}) \]

\[ \theta'_1 = \theta - \alpha \nabla \mathcal{L}_{\tau_1}(f_{\theta}) \]

Hariharan et al. 2016, Finn et al. 2017
More Details

Task 1: Apple v/s Orange

Minimize $\mathcal{L}_{\tau_1}(f_\theta)$

Low Shot Visual Recognition
Hariharan et al. 2016

Model Agnostic Meta-learning
Finn et al. 2017

$\theta'_1 = \theta - \alpha \nabla \mathcal{L}_{\tau_1}(f_\theta)$

Hariharan et al. 2016, Finn et al. 2017
Until Now

Finetuning

Nearest Neighbor Matching

Siamese Network based Metric Learning

Meta-Learning: Training for fine-tuning

Better Features —> Better Transfer!
In practice, how good are these features?

Dog from Imagenet

Accuracy ~80%

Dog

Accuracy ~20%
Consider the task of identifying cars ...
Testing the model
Learning Spurious Correlations

Unbiased look at Dataset bias, Torralba et al. 2011
More parameters in the network

More chances of learning spurious correlations!!

Maybe this problem will be avoided if we first learn simple tasks and then more complex ones??
Sequential/Continual Task Learning

Catastrophic Forgetting!!!

Poor performance on Task-1 !!!!
Catastrophic forgetting in closely related tasks

<table>
<thead>
<tr>
<th>Training on rotating MNIST</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

High Accuracy

Low Accuracy
In machine learning, we generally assume IID* data.

Sample batches of data!

Each batch: uniform distribution of rotations

*IID: Independently and Identically Distributed*
In machine learning, we generally assume IID* data.

Sample batches of data!

Each batch: uniform distribution of rotations

In real world, data is often not batched :)

*IID: Independently and Identically Distributed*
Continual learning is natural ...
In the context of reinforcement learning
Investigating Human Priors for Playing Video Games, Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Alyosha Efros, Tom Griffiths (ICML 2018)
Humans make use of prior knowledge for exploration

(a)

(b)

Humans make use of prior knowledge for exploration
What about Reinforcement Learning Agents?
In a simpler version of the game,..
For RL agents, both games are the same!
Equip Reinforcement Learning Agents with prior knowledge?
Common-Sense/Prior Knowledge

Hand-design
Common-Sense/Prior Knowledge

Hand-design

Learn from Experience

Transfer in Reinforcement Learning —> Very limited success

Good solution to continual learning required!
How to deal with catastrophic forgetting?

Just remember the weights for each task!
(1) Baseline 1

Progressive Networks (Rusu et al. 2016)
Can we do something smarter than storing all the weights?
Overcoming Catastrophic Forgetting (Kirkpatrick et al. 2017)

\[ \mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta_{A,i}^*)^2 \]

EWC: Elastic Weight Consolidation

Don’t change weights that are informative of task A

Fisher Information
Overcoming Catastrophic Forgetting (Kirkpatrick et al. 2017)
Eventually we will run out of capacity!

Is there a better way to make use of the neural network capacity?
Neural Networks are compressible post-training

(Slide adapted from Brian Cheung)
Neural Networks are compressible post-training

(Slide adapted from Brian Cheung)  
(Han et. al. 2015)
Negligible performance change after pruning —> Neural Networks are over-parameterized

Can we make use of over-parameterization?

We will have to make use of “excess” capacity during training
Superposition of many models into one (Cheung et al., 2019)

Superposition:

One Model:

Implementation: $y = W(c(1) \odot x)$

Refer to the paper for details