Understanding the Effects of Pre-Training for Object Detectors via Eigenspectrum


ICCV 2019 Neural Architects Workshop

Yosuke Shinya (DENSO CORPORATION)
Edgar Simo-Serra (Waseda University)
Taiji Suzuki (The University of Tokyo / RIKEN)
Background

- Prior works:
  - $I$ can achieve high accuracy [Girshick et al., CVPR2014]
  - $S$ can achieve similar accuracy to $I$ [He et al., ICCV2019]

- This work:
  Do $I$ and $S$ converge to similar models? No!
(Extrinsic) Architecture

ResNet-50 backbone of ImageNet fine-tuned detector $I$

ResNet-50 backbone of scratch-trained detector $S$

- Same (extrinsic) architectures
- Similar accuracy

How to analyze the difference?
Intrinsic dimensionalities

[Suzuki, AISTATS2018]
- Effective dimension of networks is less than actual number of parameters
- Intrinsic dimensionalities can be quantified by eigenspectra of covariance matrices of feature maps

\[
\Sigma = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{W_i H_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} F_{i,x,y} F_{i,x,y}^T
\]

• Intrinsic dimensionalities: numbers of eigenvalues $> 10^{-3}$

Settings in our experiments

- Covariance matrix:

Highly redundant expression

Small number of large eigenvalues

Large number of small eigenvalues

[Suzuki et al., Tokyo Deep Learning Workshop 2018]
Our method: Intrinsic Architecture Search

I

\[
\text{Bottleneck bloc}
\quad \text{3x3 conv.}
\quad \text{1x1 conv.}
\quad 7x7 conv.
\]

S
Our method: Intrinsic Architecture Search

Clearly different intrinsic architectures
Our method: Intrinsic Architecture Search

- **I**
  - Shrinking
  - Adjusting

- **S**
  - Shrinking
  - Adjusting

- Bottleneck block
- 3x3 conv.
- 1x1 conv.
- 7x7 conv.
Our method: Intrinsic Architecture Search

\[ I \rightarrow S \]

- **Shrinking**: Reduces the complexity of the architecture.
- **Adjusting**: Modifies the architecture to maintain performance.
- **Expanding**: Increases the complexity of the architecture.

- **Bottleneck block**
- **3x3 conv.**
- **1x1 conv.**
- **7x7 conv.**
Our method: Intrinsic Architecture Search

- Discovered backbone
  - Similar COCO AP
  - Similar MACs
  - 27% fewer parameters
  - Lower accuracy in ImageNet classification (Compared with ResNet-50)
Eigenspectrum dynamics during fine-tuning (Dropping)

A feature map in ResNet stage 5

Forgetting pre-trained features in the first 10k iterations
Eigenspectrum dynamics during fine-tuning (Rebounding)

Training on small learning rate
→ captures detailed information about training data
→ increases eigenvalues
→ increases variance [Suzuki, AISTATS2018]
→ causes overfitting

Learning rate schedule of “Rethinking ImageNet Pre-Training”

First learning rate decay

Do lengthen

Don’t lengthen

[He et al., ICCV2019]
Conclusions: Effects of pre-training

- ImageNet pre-training increases intrinsic dimensionalities in higher layers
- Increase of parameters caused by them
  - does not improve COCO AP
  - improve classification ability

- Current standard architectures and fine-tuning methods of object detectors are insufficient for utilizing the classification ability due to the forgetting

More appropriate architectures and knowledge-transfer methods are needed

- Hand-crafted architectures for sharing parameters $\rightarrow$ Automatic sharing
- Parameter transfer (fine-tuning) $\rightarrow$ Feature transfer or others
Where are we going?

(1) Understanding multi-task training
   • How to deal with compression of task-irrelevant information?

(2) Task-specific architectures and NAS methods
   • How to design architectures considering task differences?
   • How to overcome long training time for object detection NAS?
     - Reusing ImageNet pre-trained weights are effective but insufficient

(3) Simultaneous optimization of architectures and parameters
   • How to optimize simultaneously?
     - TWEANNs, Differentiable NAS, Neural Rejuvenation [Qiao et al., CVPR2019], … ?
     - Eigenspectrum is related to both architectures and parameters