Chapter 2 - Section 5

Image Classification

Dr. Li Hongyang

Thursday, March 24, 2022

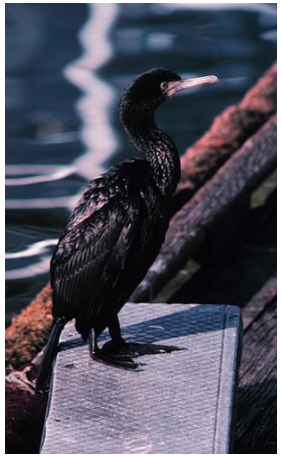
Acknowledge : Wang Cheng

商汤 sensetime

清華大學 Tsinghua University

Outline

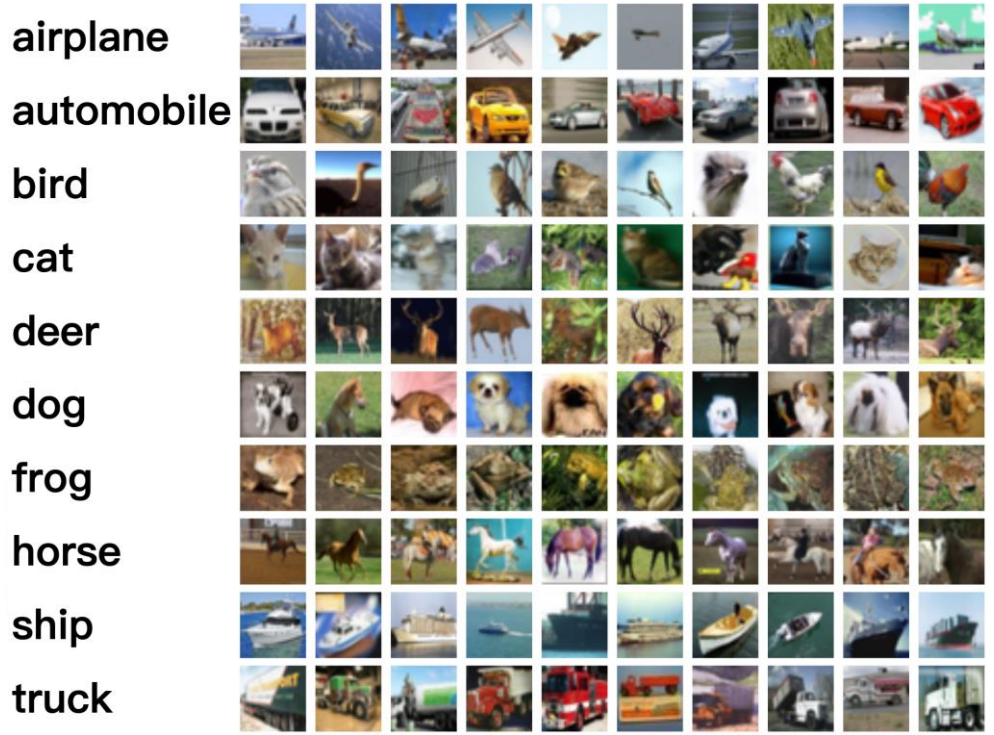
- Definition
  - Attempts to comprehend an entire image as a whole.
  - Classify the image by assigning it to a specific label.



Bird
Car
Dog
Cat
Human
Flower

- Different types of image classification tasks

The MNIST database of handwritten digits



Ten classes in CIFAR-10

LeCun Y. The MNIST database of handwritten digits. http://yann. lecun. com/exdb/mnist/. 1998.
Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images.

- Different types of image classification tasks

Traffic sign recognition

Flower recognition

- Different types of image classification tasks

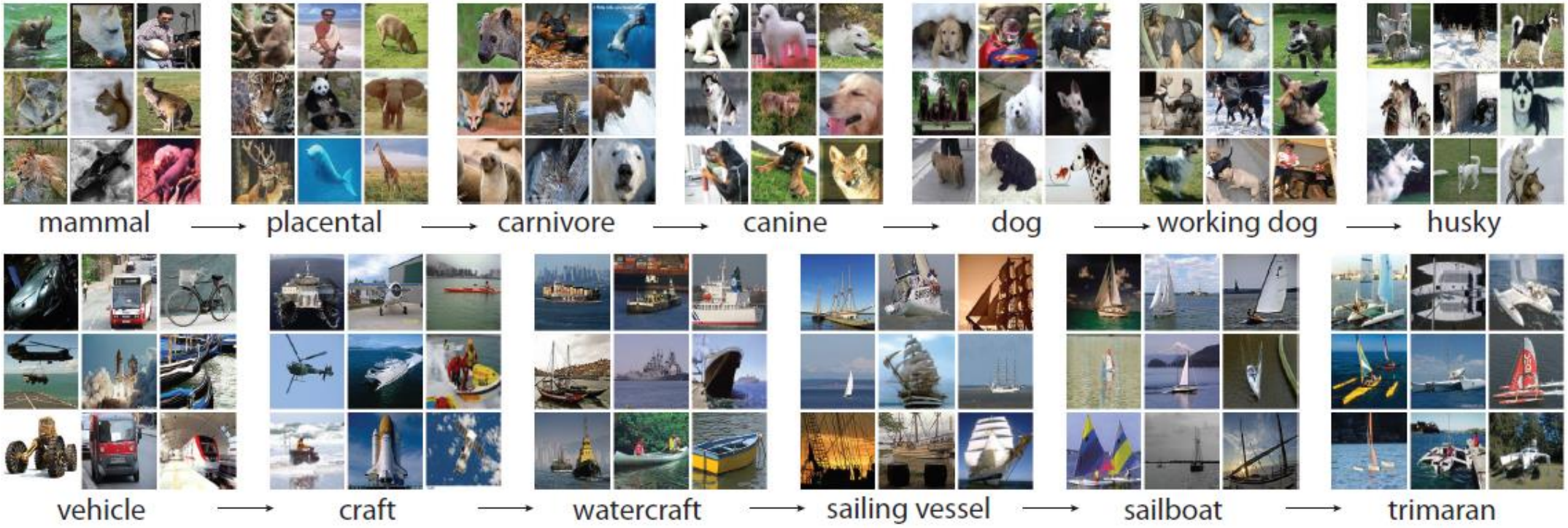


Figure 1: A snapshot of two root-to-leaf branches of ImageNet: the **top** row is from the mammal subtree; the **bottom** row is from the vehicle subtree. For each synset, 9 randomly sampled images are presented.

ImageNet

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, **ImageNet: A Large-Scale Hierarchical Image Database.** *IEEE Computer Vision and Pattern Recognition (CVPR), 2009*

Background and Research Progress

- ImageNet Datasets
  - The ImageNet dataset contains 14,197,122 annotated images according to the WordNet hierarchy
    - Total number of non-empty WordNet synsets: 21841
    - Total number of images: 14,197,122
    - Number of images with bounding box annotations: 1,034,908
    - Number of synsets with SIFT features: 1000
    - Number of images with SIFT features: 1.2 million

Background and Research Progress

- Challenges
  - Viewpoint variation
  - Illumination
  - Deformation
  - Occlusion
  - Background Clutter
  - Intraclass variation

Background and Research Progress

- Viewpoint variation
  - Images are selected from CUB-100-2011

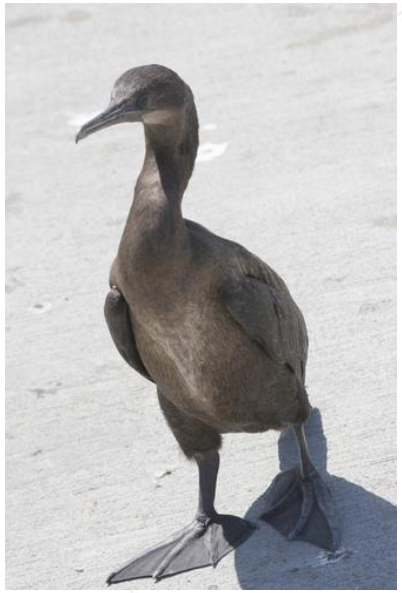
Brandt Cormorant (Left)

Brandt Cormorant (Right)

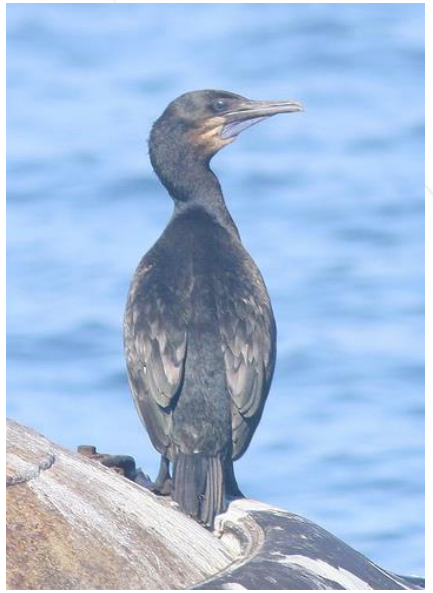
Welinder P, Branson S, Mita T, Wah C, Schroff F, Belongie S, Perona P. Caltech-UCSD birds 200.

- Illumination

Brandt Cormorant



Brandt Cormorant (Light)



Brandt Cormorant (Lighter)

- Deformation

Brandt Cormorant

Brandt Cormorant (Open)

- Occlusion

Brandt Cormorant

Brandt Cormorant (Occlusion)

- Background Clutter

Brandt Cormorant

Brandt Cormorant (Background Clutter)

- Intraclass variation

Brandt Cormorant

Yellow Throat

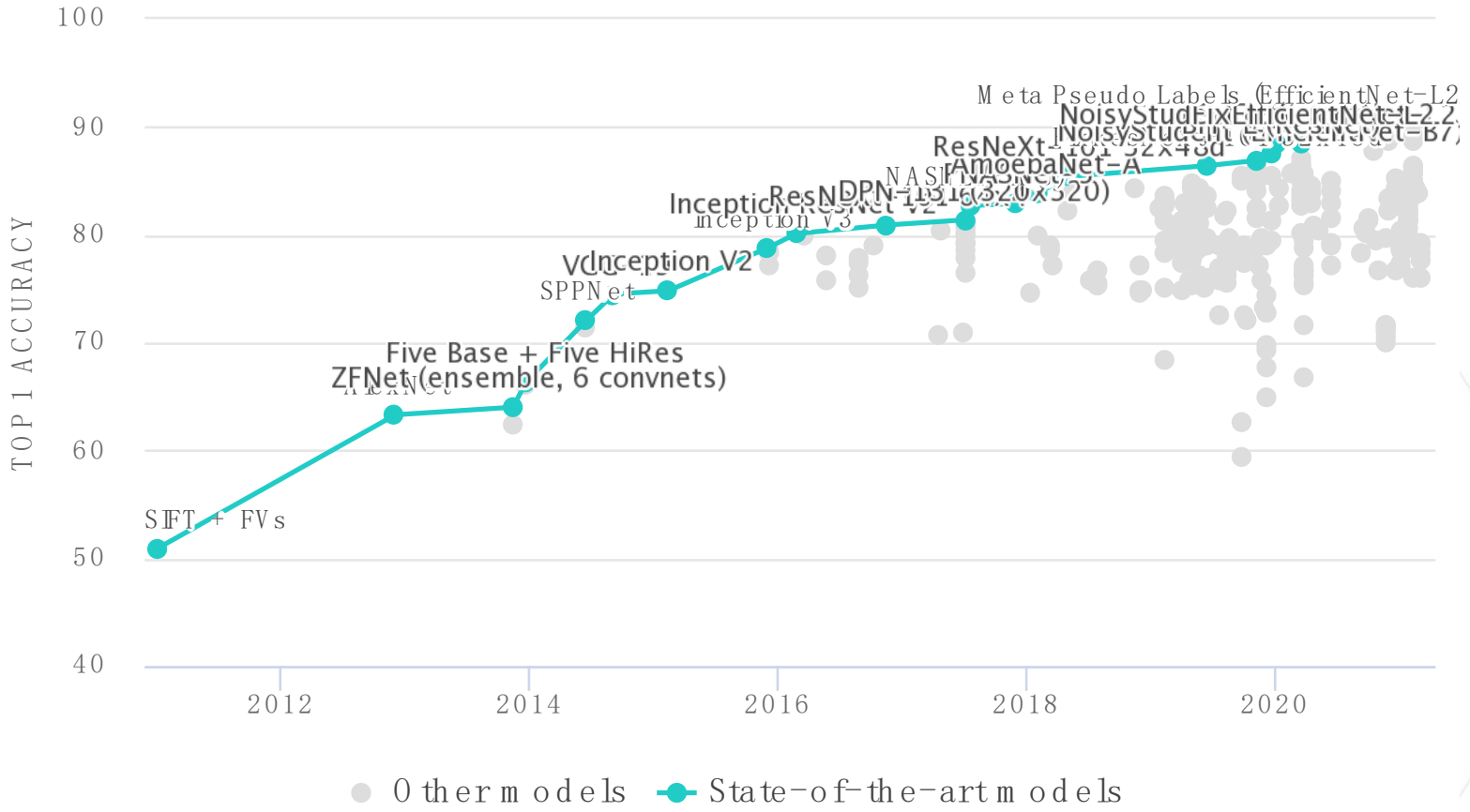
Black and White Warbler

Background and Research Progress

- Progress
  - K-Nearest Neighbor
  - Naive Bayes
  - Support Vector Machines
  - Neural network

Datasets and Evaluation Metrics

- Overview of Models in ImageNet Challenge



J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. CVPR2009.

Datasets and Evaluation Metrics

- Several datasets and evaluation metrics

Topic	Dataset Name	Task	Scale
Objects	Cifar-10	Multi-class Classification	60,000 images, 10 classes
	Cifar-100	Fine-grained Classification	60,000 images, 100 classes
	ImageNet	Multi-class Classification	14,000,000 images, 20,000 classes
	PASCAL VOC	Multi-class Classification	17,125 images, 20 classes
Human	SMILES	Binary Classification	13,165 images, 2 classes
	Facial Expression Recognition	Multi-class Classification	35,888, 7 classes
Flowers	Flowers-17	Multi-class Classification	1,360, 17 classes
	Flowers-102	Fine-grained Classification	8,189, 102 classes
Text	MNIST	Multi-class Classification	70,000 images, 10 classes
	USPS	Multi-class Classification	9,900 images, 10 classes

Datasets and Evaluation Metrics

- Several datasets and evaluation metrics
  - Accuracy
  - F1 score
  - Precision-Recall (PR) curves
  - Receiver Operator Characteristic (ROC) curves, AUC

- F1 score
  - Both consider $precision$ and $recall$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2 \times precsion \times recall}{precsion + recall}$$

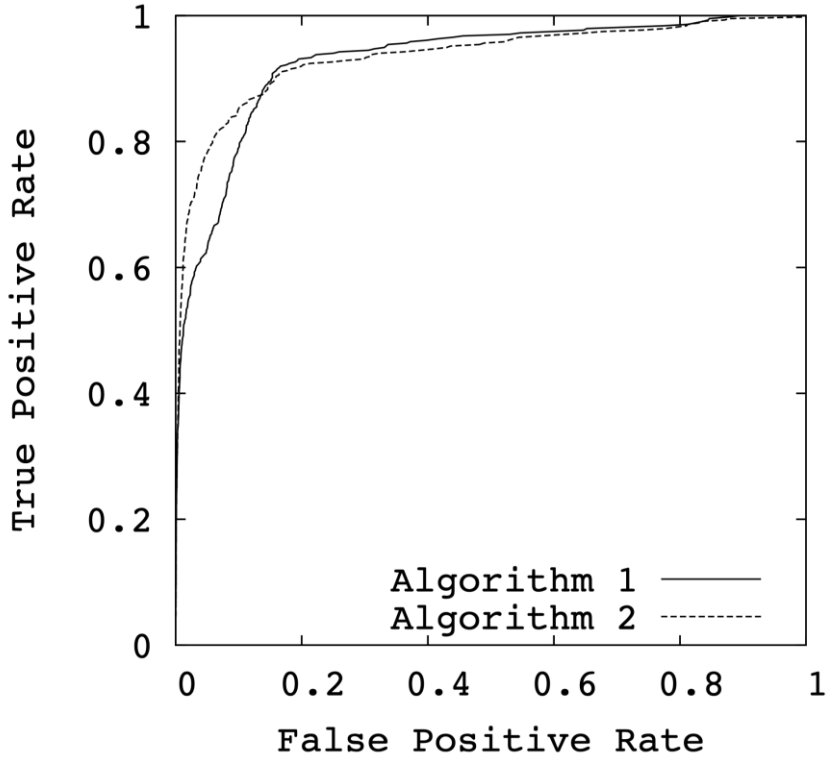
$$F_\beta = (1 + \beta^2)\frac{precsion \times recall}{(\beta^2 \times precsion) + recall}$$

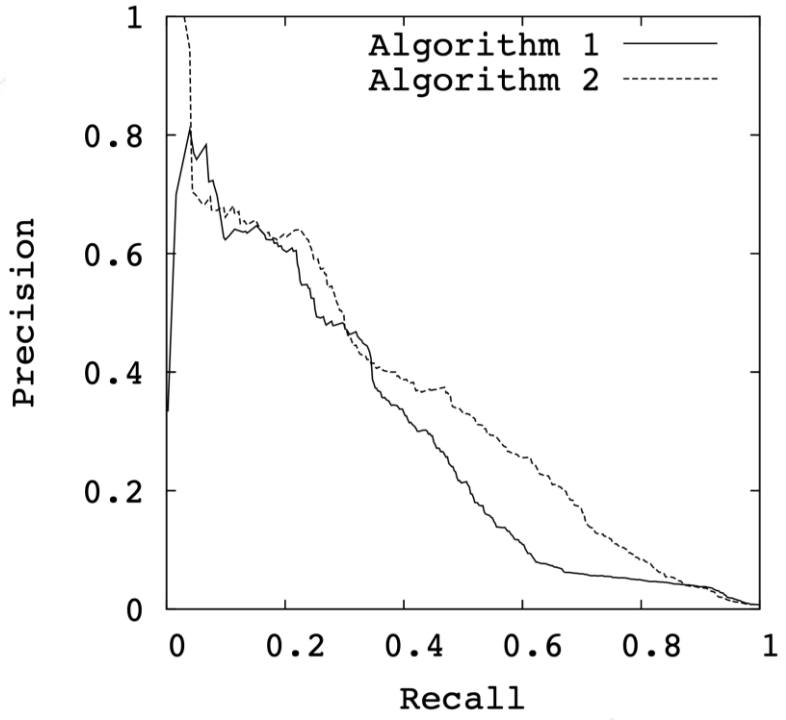
$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$specificity = \frac{TN}{TN + FP}$$

Datasets and Evaluation Metrics

- PR vs ROC
  - Which algorithm is better?

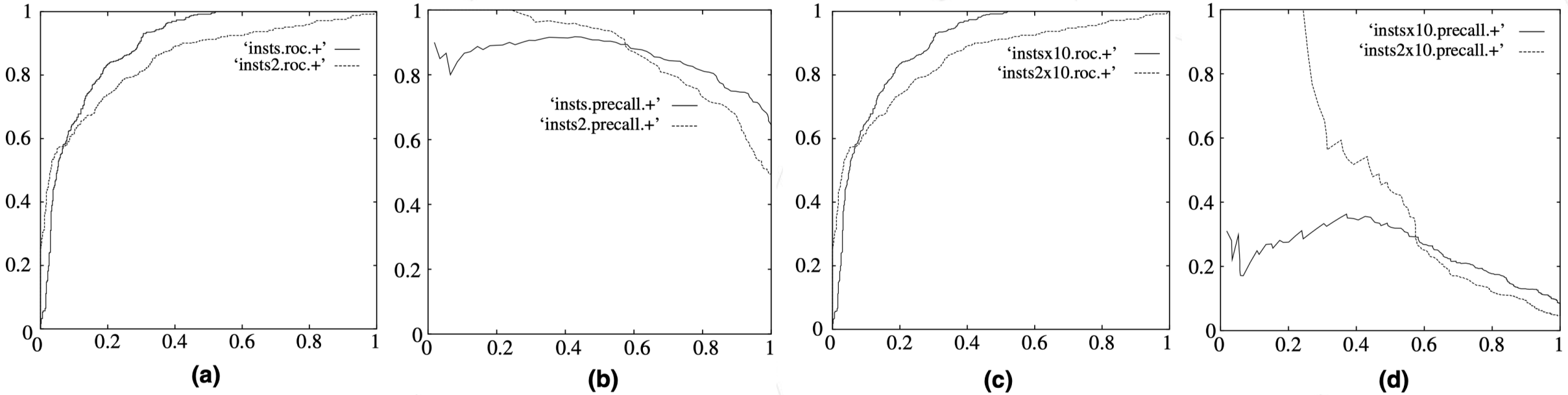




Davis, J. and Goadrich, M., The relationship between Precision-Recall and ROC curves. ICML, 2006.

Datasets and Evaluation Metrics

- PR vs ROC
  - Which algorithm is better?



Fawcett T. An introduction to ROC analysis. Pattern recognition letters. 2006 Jun 1;27(8):861-74.

Datasets and Evaluation Metrics

- PR vs ROC
  - A curve dominates in ROC space if and only if it dominates in PR space
  - The ROC curve and PR curve for a given algorithm contain the same points
  - Optimizes the area under the ROC curve is not guaranteed to optimize the area under the PR curve

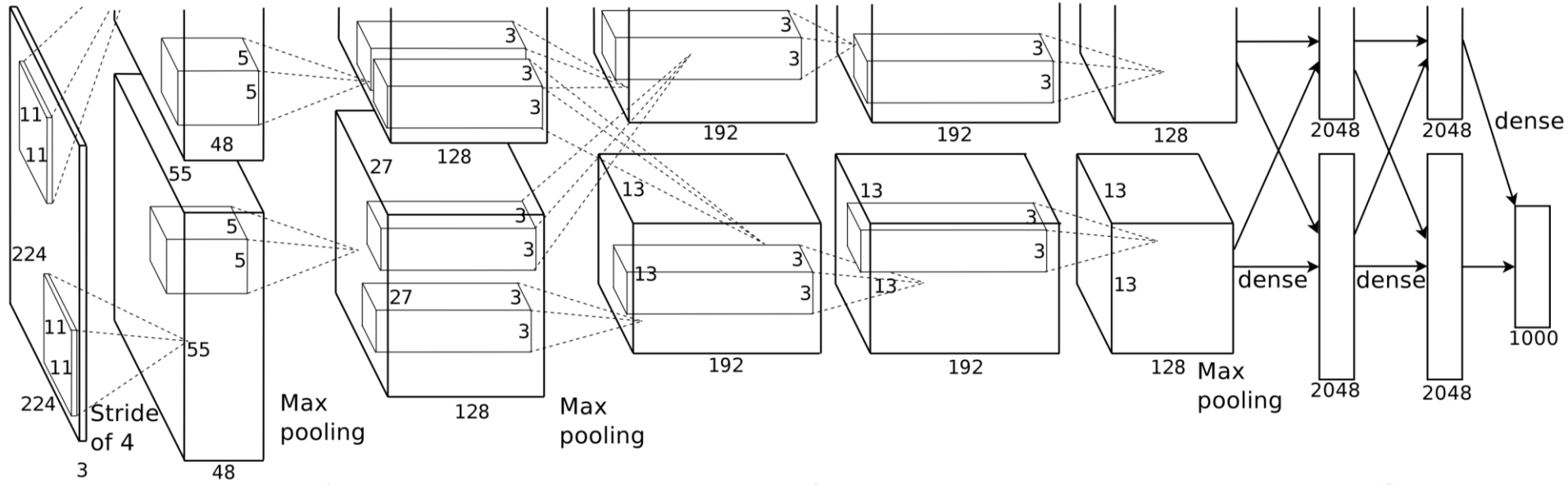
商汤 sensetime

清華大學 Tsinghua University

Outline

Popular Neural Networks

- 2012: AlexNet
  - Top-1 accuracy rate: 62.5%
  - Parameters: 60M
  - Takes between five and six days to train on two GTX 580 3GB GPUs



Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. NIPS. 2012;25:1097-105.

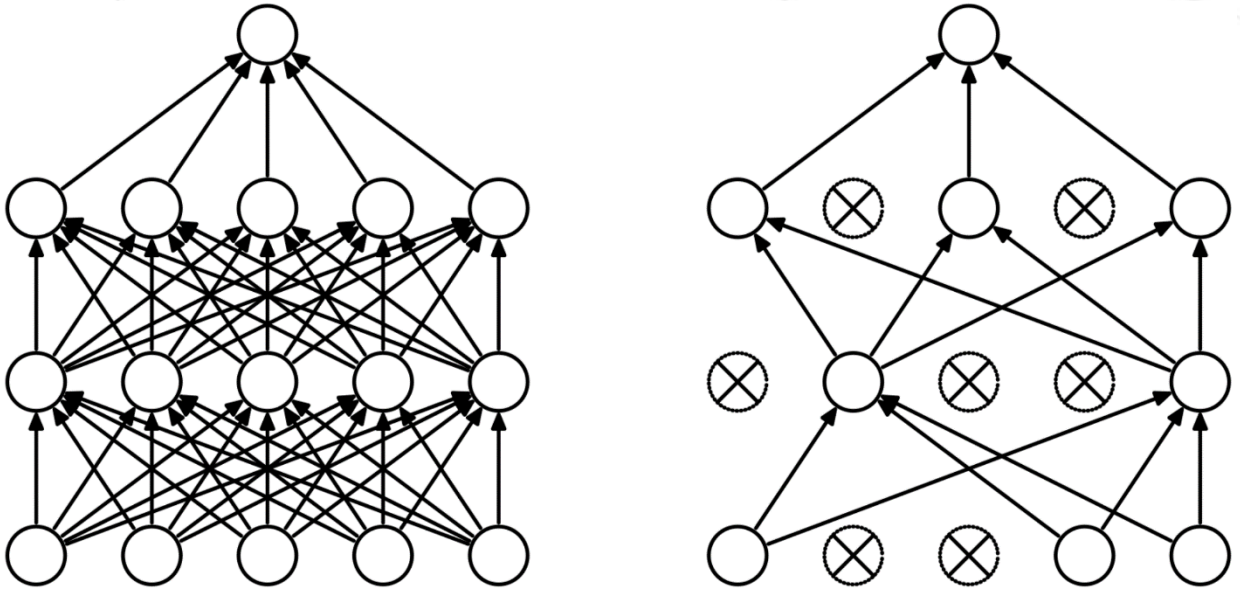
Popular Neural Networks

- Rectified linear unit (ReLU)

$$f(x) = x^+ = \max(0, x)$$

  - Biological plausibility: One-sided
  - Sparse activation: In a randomly initialized network, only about 50% of hidden units are activated
  - Better gradient propagation: Fewer vanishing gradient problems
  - Efficient computation: Only comparison, addition and multiplication.
  - Scale-invariant: $\max(0, ax) = a \max(0, x)$ for $a \geq 0$

- Dropout in Neural Networks
  - Training neural network with dropout can be seen as training a collection of 2^n thinned networks with extensive weight sharing

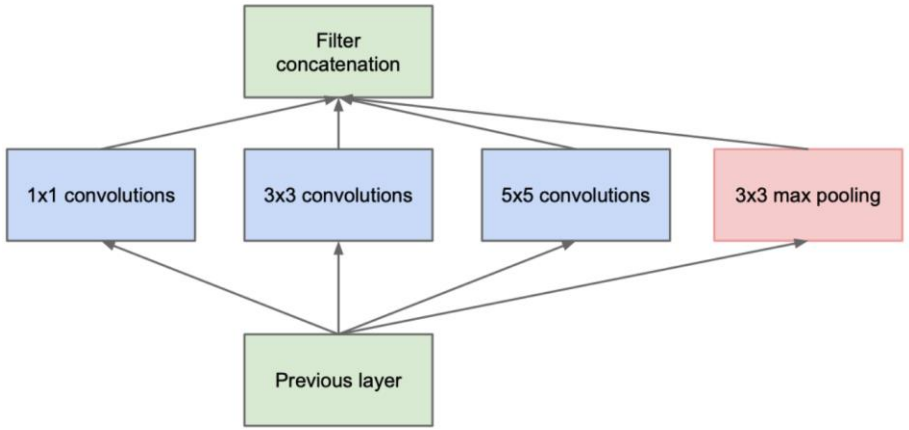


(a) Standard Neural Net (b) After applying dropout.

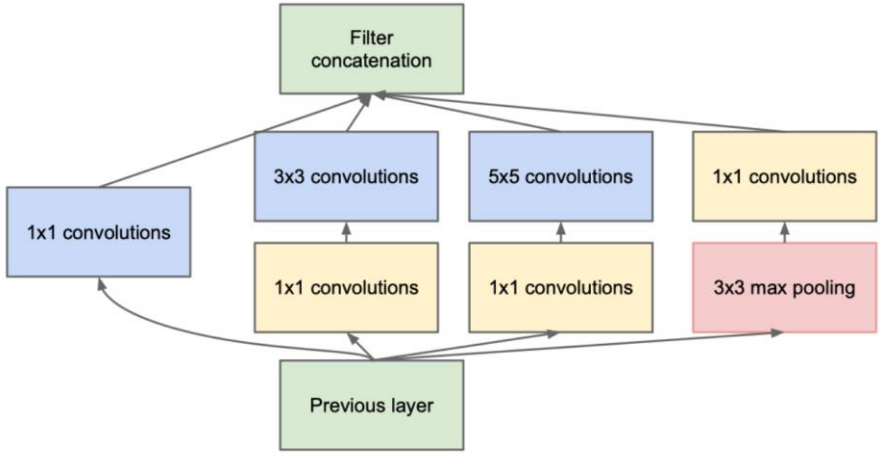
Srivastava N, Hinton G, et al. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014 Jan 1;15(1):1929-58.

Popular Neural Networks

- 2014: Inception V1
  - Top-1 accuracy rate: 69.8%
  - Parameters: 5M
  - Introduce a new level of organization in the form of the "Inception module"



(a) Inception module, naïve version



(b) Inception module with dimension reductions

Szegedy C, Liu W, et al. Going deeper with convolutions. CVPR2015 (pp. 1-9).

Popular Neural Networks

- Inception module and GoogLeNet
  - An optimal local sparse structure can be approximated by dense components
  - 1×1 convolutions are used to compute reductions
  - $2 - 3 \times$ faster than similarly performing networks with non-Inception architecture
  - Auxiliary classifiers
    - Encourage discrimination in the lower stages in the classifier
    - Increase the gradient signal that gets propagated back
    - Provide additional regularization.

Popular Neural Networks

- 2015: Inception V3
  - Top-1 accuracy rate: 78.8%
  - Parameters: 23.8M
  - Factorizing Convolutions with Large Filter Size
    - 5*5 -> 3*3 + 3*3, 3*3 -> 1*3 + 3*1
  - Model Regularization via Label Smoothing

$$y_i = \begin{cases} 1-\varepsilon & if \quad i = true, \\ \dfrac{\varepsilon}{K-1} & otherwise \end{cases}$$

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. CVPR2016.

Popular Neural Networks

- 2015: ResNet 152
  - Top-1 accuracy rate: 78.6%
  - Parameters: 117.4M

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. CVPR2016 (pp. 770-778).

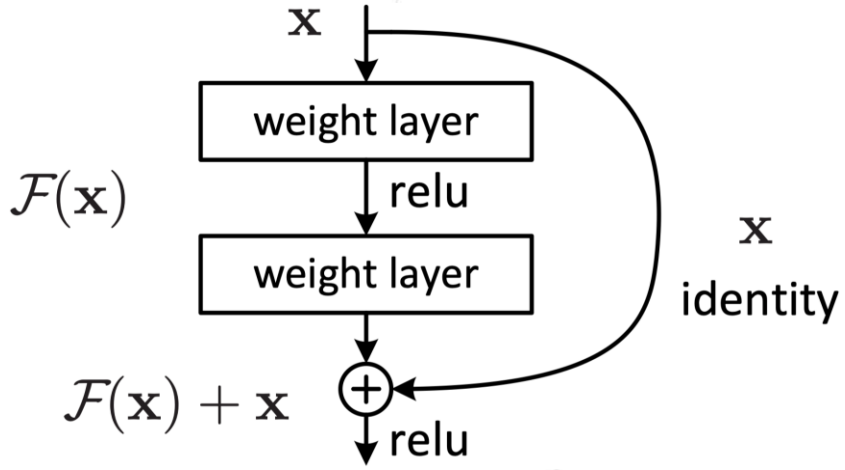
Popular Neural Networks

- Is learning better networks as easy as stacking more layers?
  - Vanishing/exploding gradients
    - Batch Normalization; ReLU
  - Overfitting
    - More data; Dropout
  - High resource consumption
    - More GPUs
  - Higher training error, degradation problem
    - Why, and how to solve it?

- Deep residual learning framework
  - Data Processing Inequality

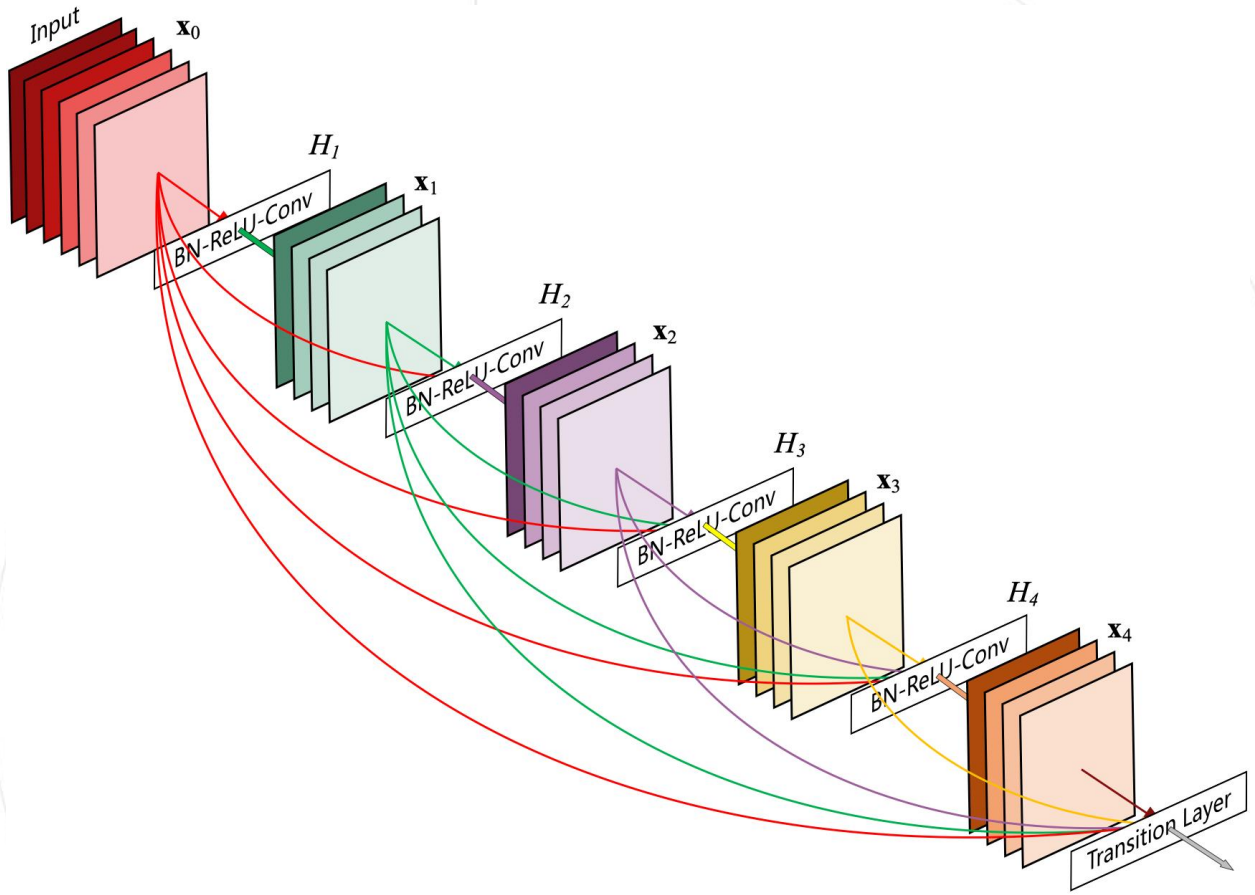
$$I(X;Y) \geqslant I(X;Z)$$

  - Shortcut connections
    - a deeper model should have training error no greater than its shallower counterpart



Popular Neural Networks

- 2016: DenseNet-264
  - Top-1 accuracy rate: 79.2%
  - Create short paths from early layers to later layers
    - Each layer has direct access to the gradients from the loss function and the original input signal



Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. CVPR2017 (pp. 4700-4708).

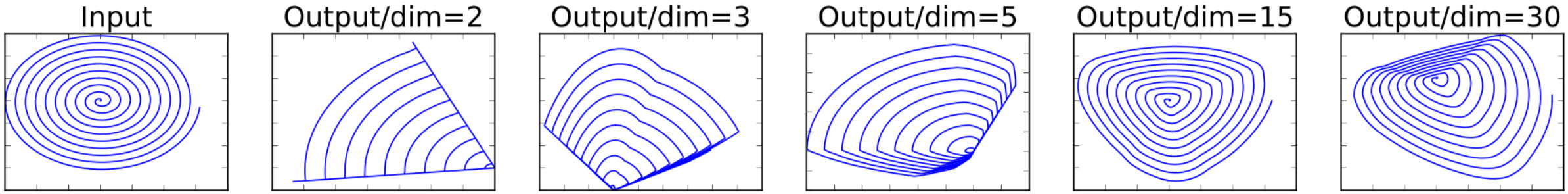
Lightweight Networks

- 2018: MobileNetV2
  - Top-1 accuracy rate: 74.7%
  - Parameters: 6.9M
  - Depthwise separable convolution
  - Linear bottleneck
  - Inverted residuals

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. CVPR2018 (pp. 4510-4520).

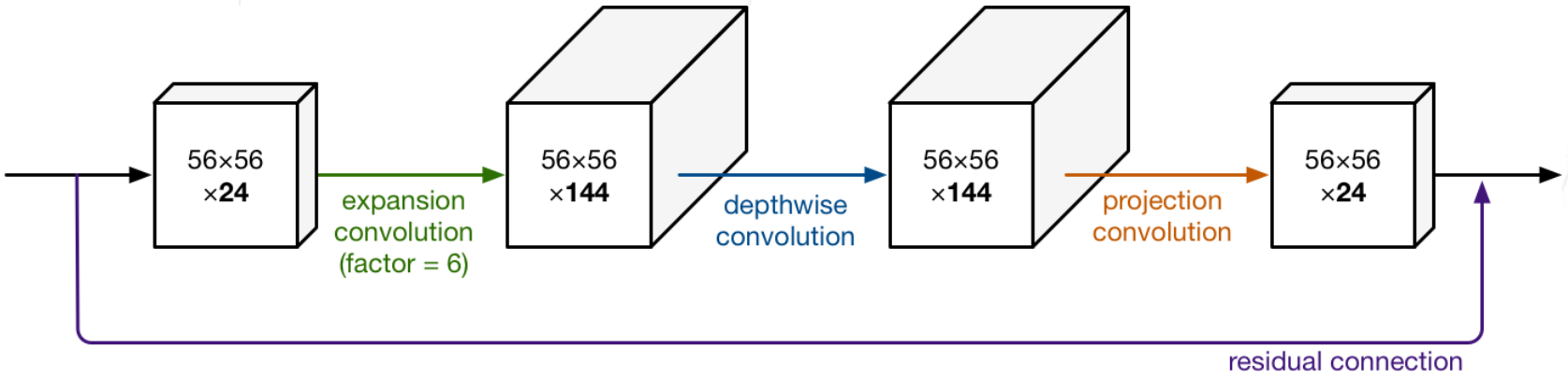
- Depthwise Separable Convolution
  - Depthwise Convolution (left) + Pointwise Convolution (right)

- Linear bottleneck
  - ReLU collapses the channel

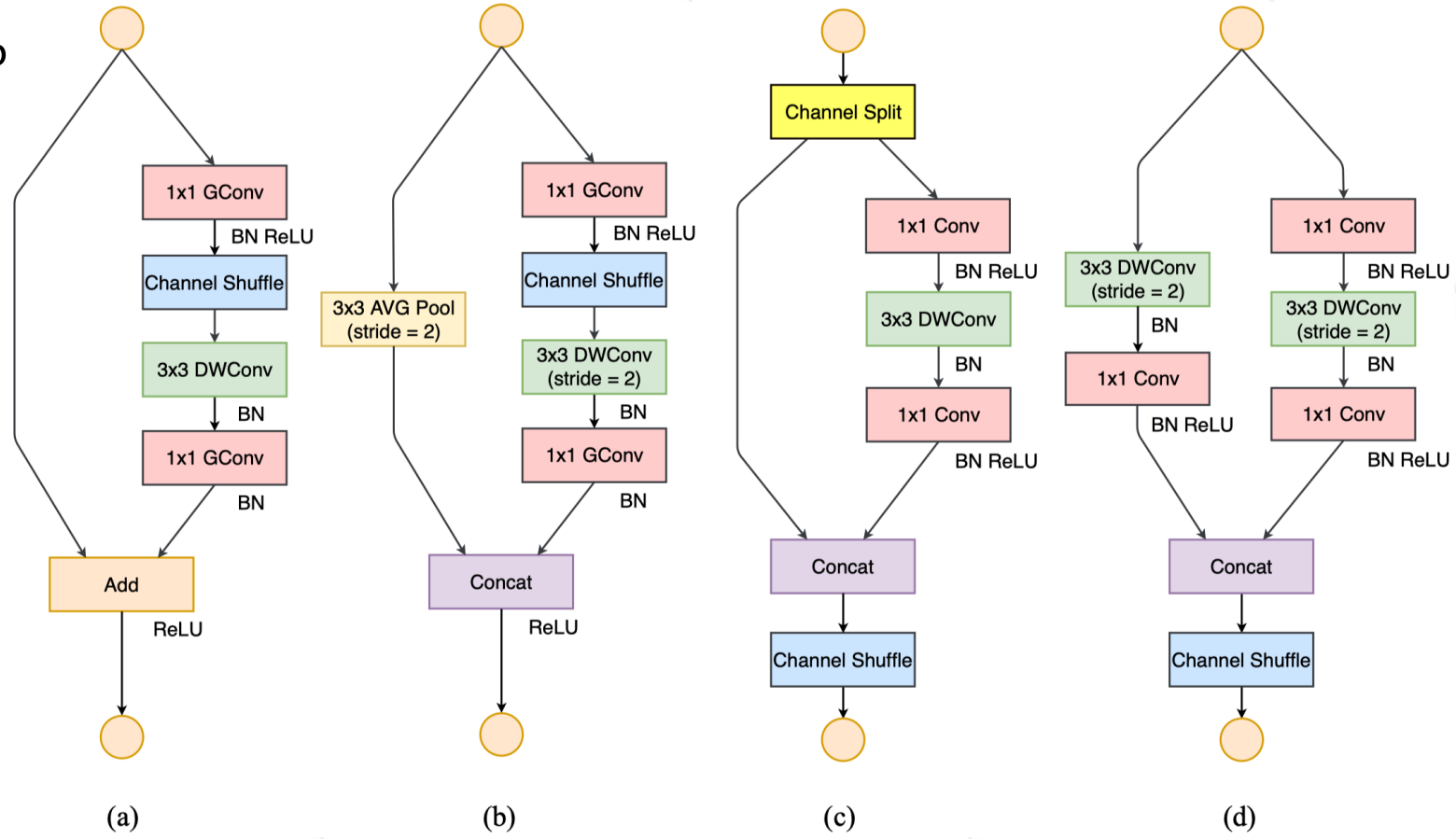


  - Using linear layers prevents nonlinearities from destroying too much information
  - Expansion ratio
    - The ratio between the size of the input bottleneck and the inner size

- Inverted residuals
  - Expansion Convolution
    - low-dim to high-dim
  - Depthwise Convolution with ReLU
  - Projection layer
    - high-dim to low-dim



- 2018: ShuffleNet V2 2×
  - Top-1 accuracy rate:75.4%
  - Parameters:7.4M
  - Channel Split

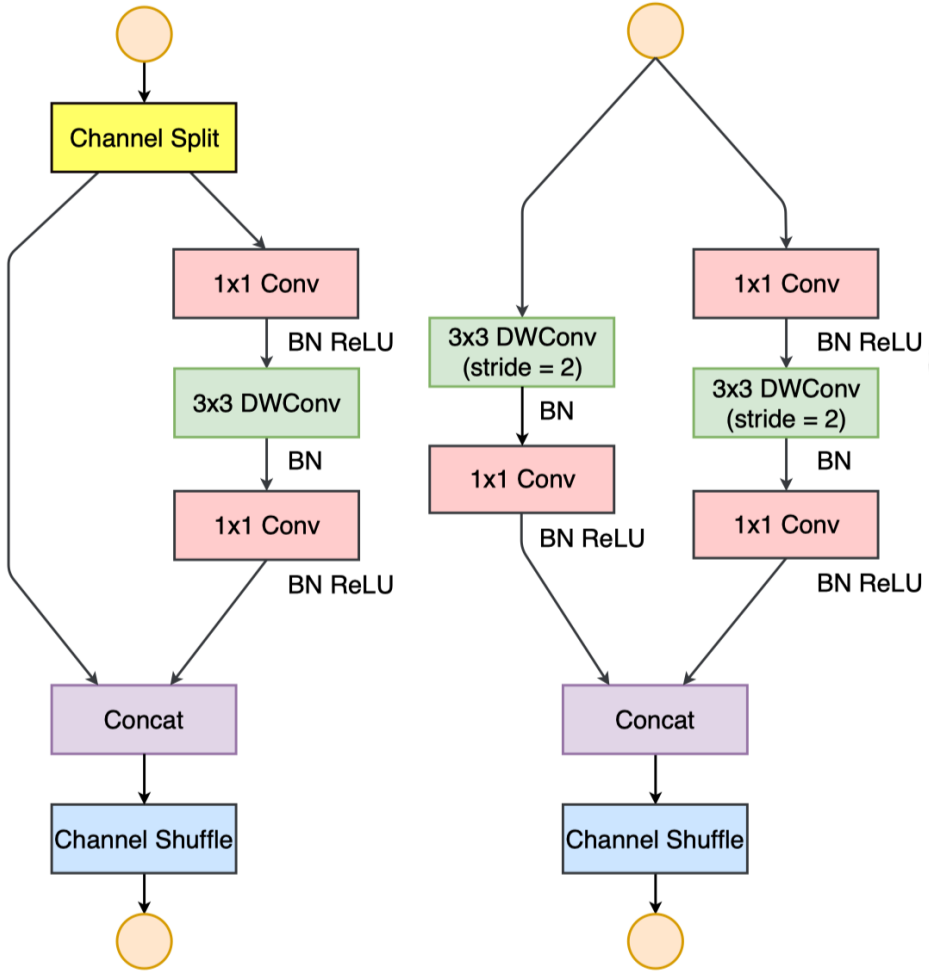


Ma N, Zhang X, Zheng HT, Sun J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. ECCV2018.

Lightweight Networks

- Four practical guidelines for efficient network architecture design
  - G1) Equal channel width minimizes memory access cost (MAC)
  - G2) Excessive group convolution increases MAC
  - G3) Network fragmentation reduces degree of parallelism
  - G4) Element-wise operations are non-negligible

- ShuffleNet V2
  - One branch remains as identity (G3)
  - No group convolution (G2)
  - The number of channels keeps the same (G1)
  - “Concat”， “Channel Shufle” and “Channel Split”, are merged into a single element-wise operation (G4)



(c) (d)

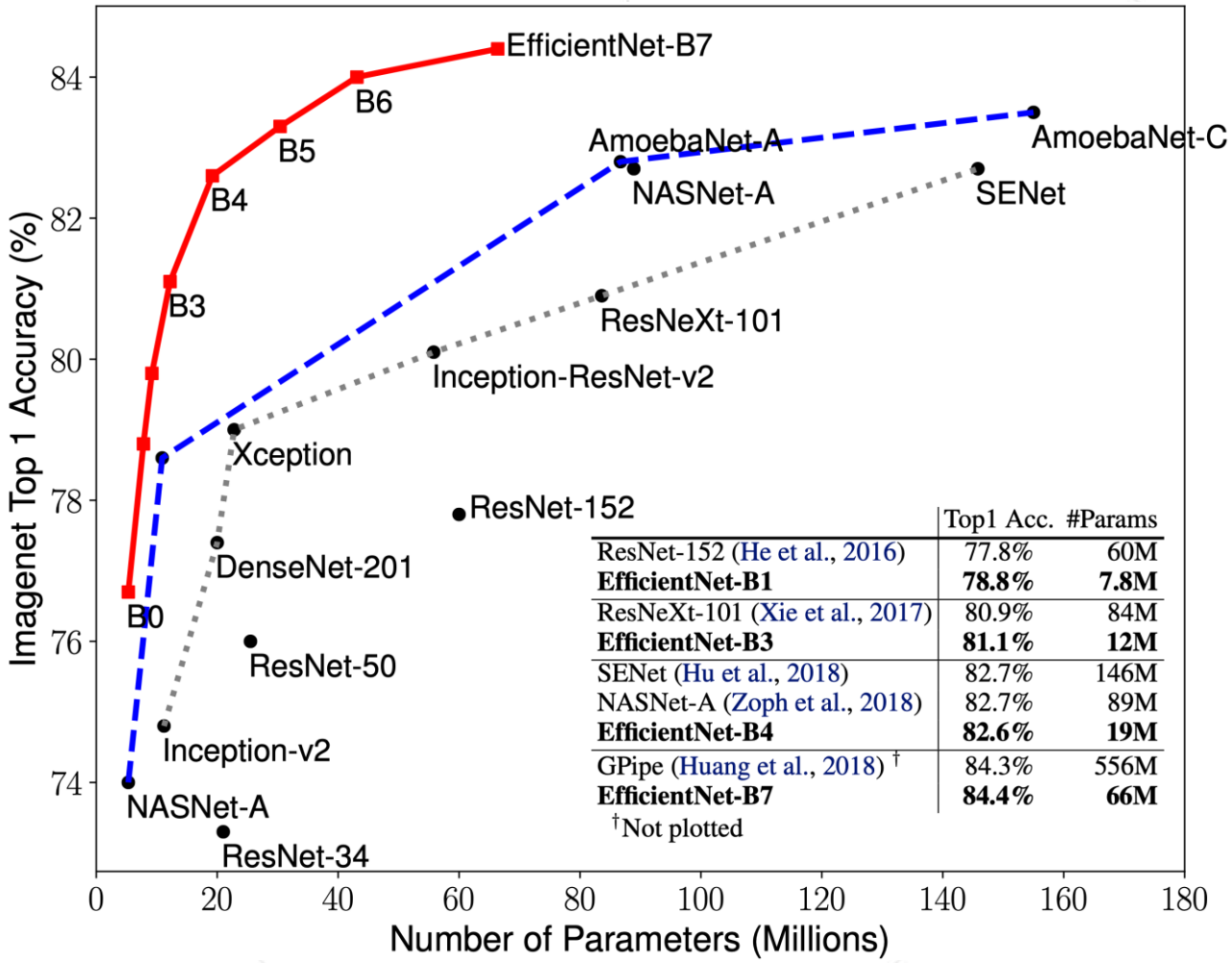
Neural Architecture Search

- Some NAS models

ConvNet	ImageNet top1 acc	ImageNet top5 acc	Published In
NASNet	82.7	96.2	CVPR2018
PNASNet	82.9	96.2	ECCV2018
MNASNet	76.13	92.85	CVPR2018
ProxylessNAS	75.1	92.5	ICLR2019
EfficientNet	84.3	97.0	ICML2019

Neural Architecture Search

- 2019: EfficientNet
  - Top-1 accuracy rate: 84.4%
  - Parameters: 66M
  - Compound scaling method



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. ICML2019

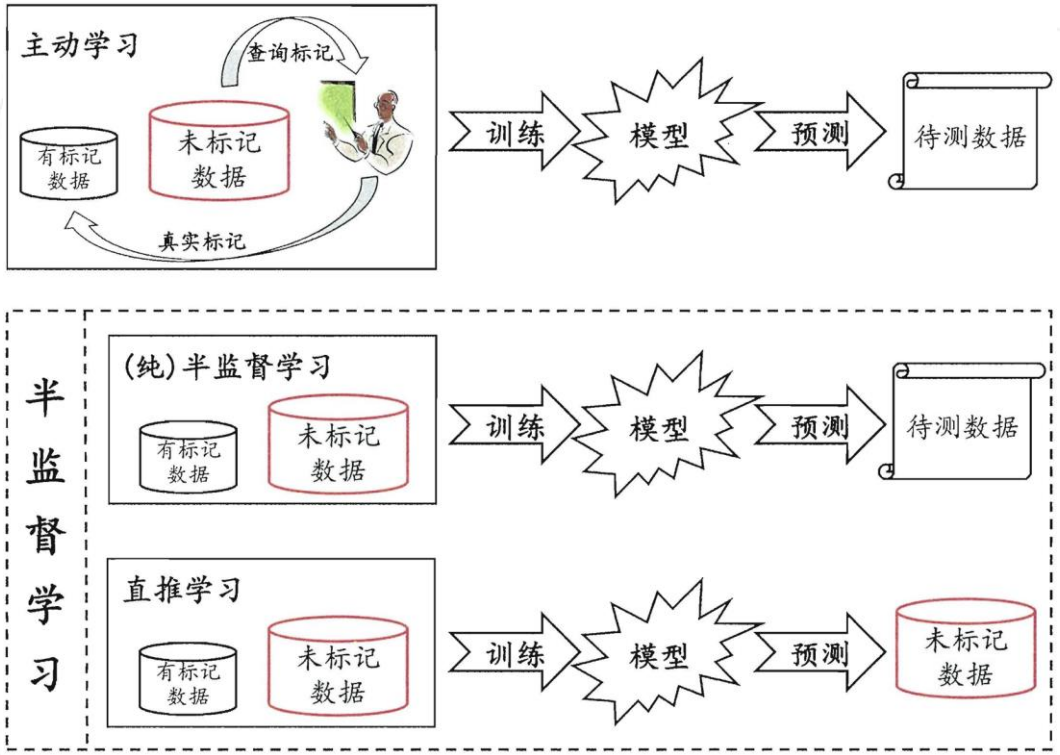
商汤 sensetime

清華大學 Tsinghua University

Outline

Semi-Unsupervised Classification Model

- Definition
  - Semi-supervised image classification leverages unlabeled data as well as labelled data to increase classification performance



Pseudo-Labelling

- Background
  - Consistency Regularization

$$\|\mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta) - \mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta)\|_2^2$$

  - Entropy Minimization
    - Minimizes the entropy of $p_{model}(y|x;\theta)$
  - Traditional Regularization

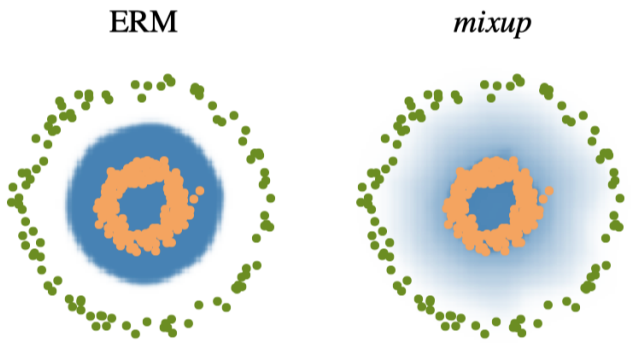
Pseudo-Labelling

- mixup: Beyond Empirical Risk Minimization
  - A simple and data-agnostic data augmentation method

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \qquad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \qquad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

(a) One epoch of *mixup* training in PyTorch.



(b) Effect of *mixup* ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1|x)$.

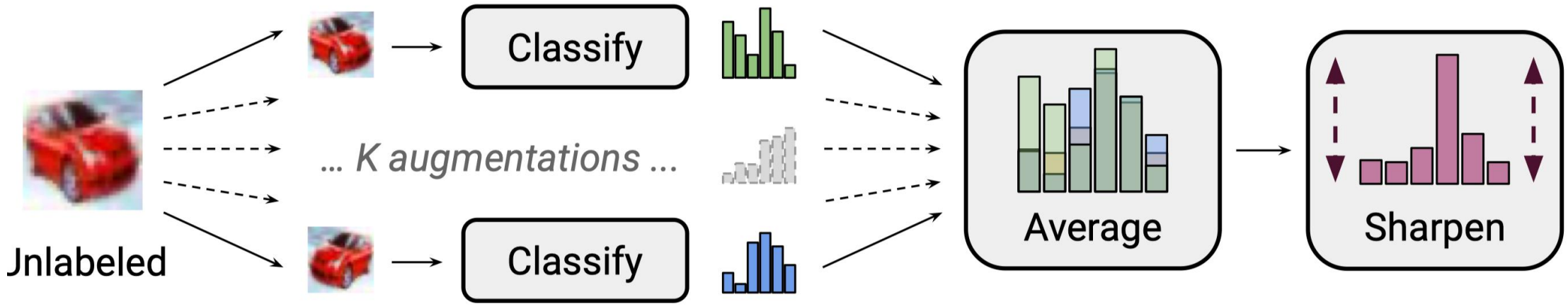
Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412. 2017 Oct 25.

Pseudo-Labelling

- mixup: Beyond Empirical Risk Minimization
  - Why use beta distribution?
  - Similar to label smooth?
  - Why mixup works? (Generalization gap between training data and real data)

Pseudo-Labelling

- MixMatch: A Holistic Approach to Semi-Supervised Learning
  - Stochastic data augmentation is applied to an unlabeled image K times
  - The average of these K predictions is “sharpened” by adjusting the distribution’s temperature



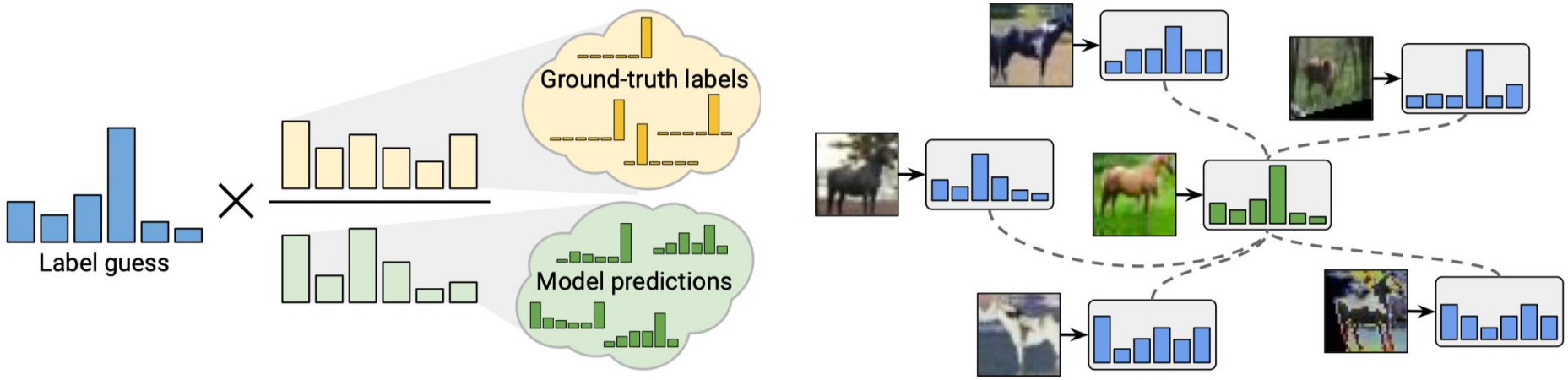
Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel C. Mixmatch: A holistic approach to semi-supervised learning. arXiv preprint arXiv:1905.02249. 2019 May 6.

- MixMatch: A Holistic Approach to Semi-Supervised Learning

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \dots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.
for $b = 1$ **to** B **do**
  $\hat{x}_b = \text{Augment}(x_b)$ *// Apply data augmentation to* x_b
  for $k = 1$ **to** K **do**
    $\hat{u}_{b,k} = \text{Augment}(u_b)$ *// Apply* k^{th} *round of data augmentation to* u_b
  end for
  $\bar{q}_b = \frac{1}{K} \sum_k \text{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of* u_b
  $q_b = \text{Sharpen}(\bar{q}_b, T)$ *// Apply temperature sharpening to the average prediction (see eq. (7))*
end for
$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ *// Augmented labeled examples and their labels*
$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ *// Augmented unlabeled examples, guessed labels*
$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ *// Combine and shuffle labeled and unlabeled data*
$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ *// Apply* MixUp *to labeled data and entries from* $\mathcal{W}$
$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ *// Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$
return $\mathcal{X}', \mathcal{U}'$

Pseudo-Labelling

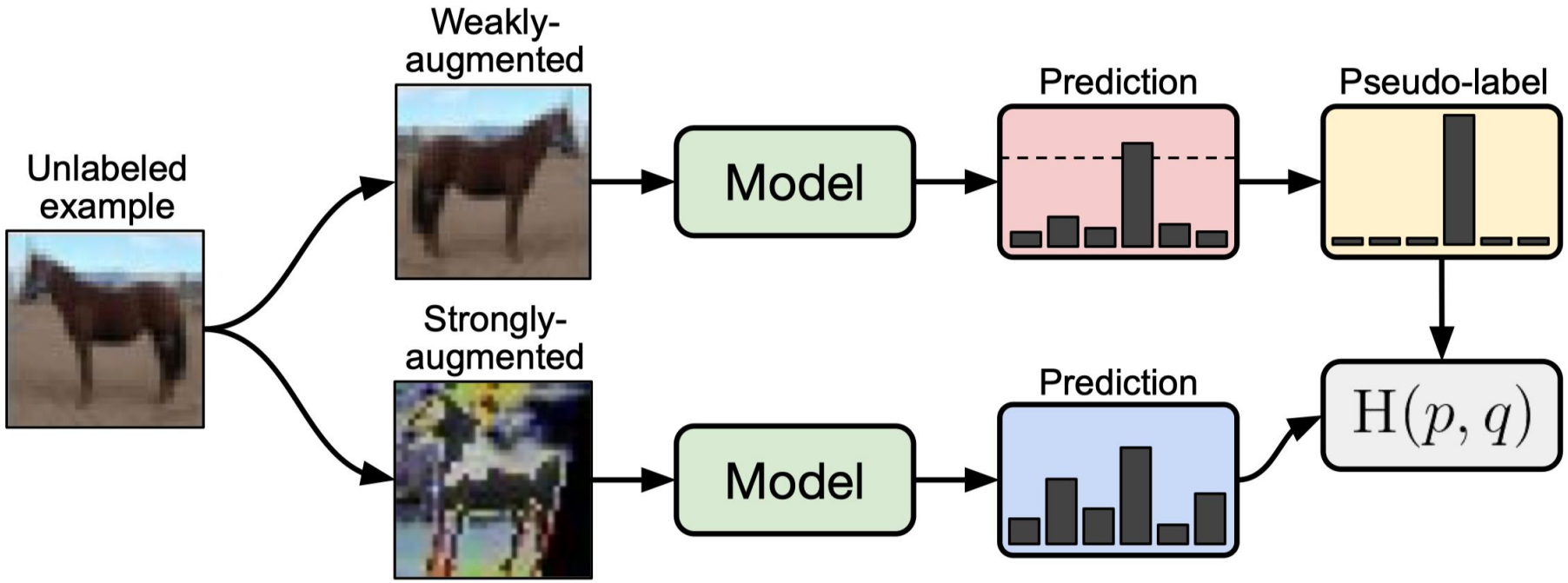
- ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring
  - Improved version of MixMatch
  - Distribution Alignment (left) and Augmentation Anchor (right)



Berthelot D, Carlini N, et al. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv preprint arXiv:1911.09785. 2019 Nov 21.

Pseudo-Labelling

- FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence
  - Combination of Consistency regularization and pseudo-labeling.



Sohn K, Berthelot D, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685. 2020 Jan 21.

Pseudo-Labelling

- FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence
  - FixMatch consists of two loss terms: a supervised loss ℓ_s and an unsupervised loss ℓ_u
  - ℓ_s is the standard cross-entropy loss

$$\ell_s = \frac{1}{B}\sum_{b=1}^{B} \mathrm{H}(p_b, p_\mathrm{m}(y \mid \alpha(x_b)))$$

  - Convert the prediction on the weakly-augmented image to a one-hot pseudo-label
  - ℓ_u is the cross-entropy loss against the model' s output for the strongly-augmented image

$$\ell_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau)\, \mathrm{H}(\hat{q}_b, p_\mathrm{m}(y \mid \mathcal{A}(u_b)))$$

Generative Methods

- 生成式方法 (generative methods)
  - 假设所有数据(无论是否有标记)都是由同一个潜在的模型“生成”的
  - 通过潜在模型的参数将未标记数据与学习目标联系起来，而未标记数据的标记则可看作模型的缺失参数
  - 通常可基于 EM 算法进行极大似然估计求解

- 生成式方法 (generative methods)

- E 步：根据当前模型参数计算未标记样本 $\boldsymbol{x}_j$ 属于各高斯混合成分的概率

$$\gamma_{ji} = \frac{\alpha_i \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{i=1}^{N} \alpha_i \cdot p(\boldsymbol{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} ; \tag{13.5}$$

- M 步：基于 γ_{ji} 更新模型参数，其中 l_i 表示第 i 类的有标记样本数目

$$\boldsymbol{\mu}_i = \frac{1}{\sum_{\boldsymbol{x}_j \in D_u} \gamma_{ji} + l_i} \left(\sum_{\boldsymbol{x}_j \in D_u} \gamma_{ji} \boldsymbol{x}_j + \sum_{(\boldsymbol{x}_j, y_j) \in D_l \wedge y_j = i} \boldsymbol{x}_j \right) , \tag{13.6}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{\sum_{\boldsymbol{x}_j \in D_u} \gamma_{ji} + l_i} \left(\sum_{\boldsymbol{x}_j \in D_u} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^{\mathrm{T}} + \sum_{(\boldsymbol{x}_j, y_j) \in D_l \wedge y_j = i} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^{\mathrm{T}} \right) , \tag{13.7}$$

$$\alpha_i = \frac{1}{m} \left(\sum_{\boldsymbol{x}_j \in D_u} \gamma_{ji} + l_i \right) . \tag{13.8}$$

周志华 著. 机器学习, 北京: 清华大学出版社, 2016年1月. (ISBN 978-7-302-206853-6)

商汤 sensetime

清華大學 Tsinghua University

Outline

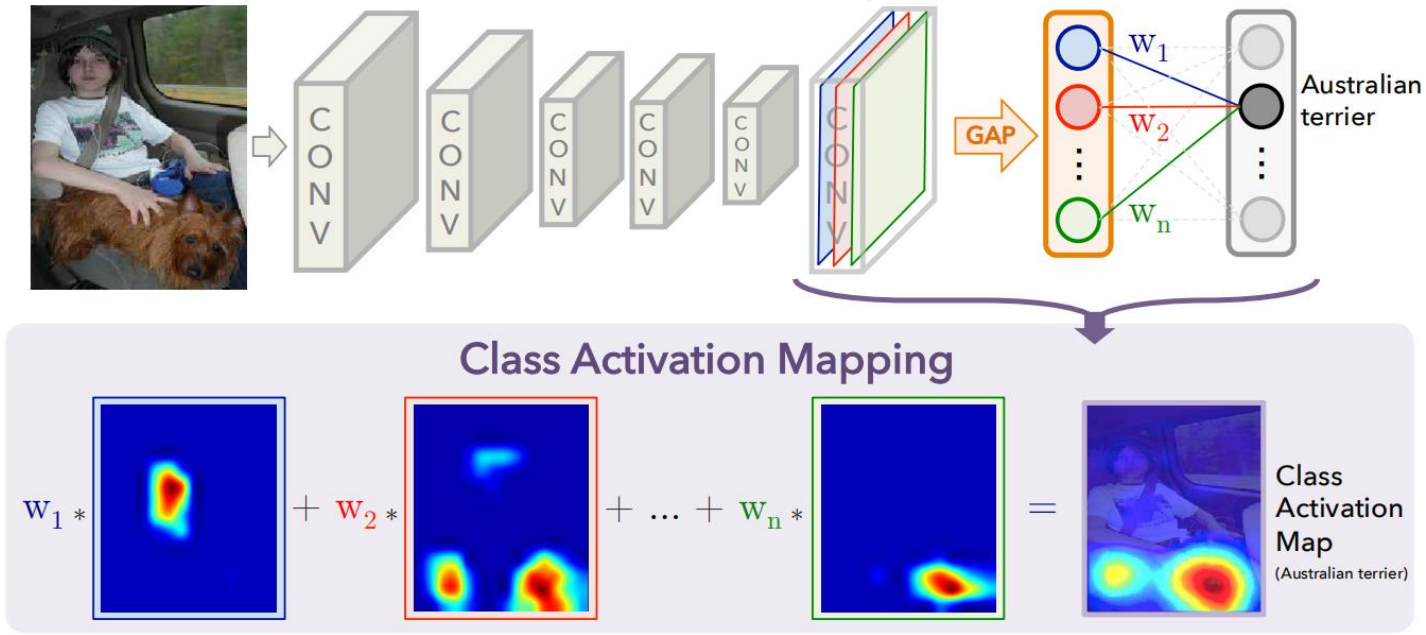
Adversarial Training

- Adversarial Training
  - Adversarial machine learning is a machine learning technique that attempts to fool models by supplying deceptive input
  - keeping track via this link

https://en.wikipedia.org/wiki/Adversarial_machine_learning

Visualization of Classification Model

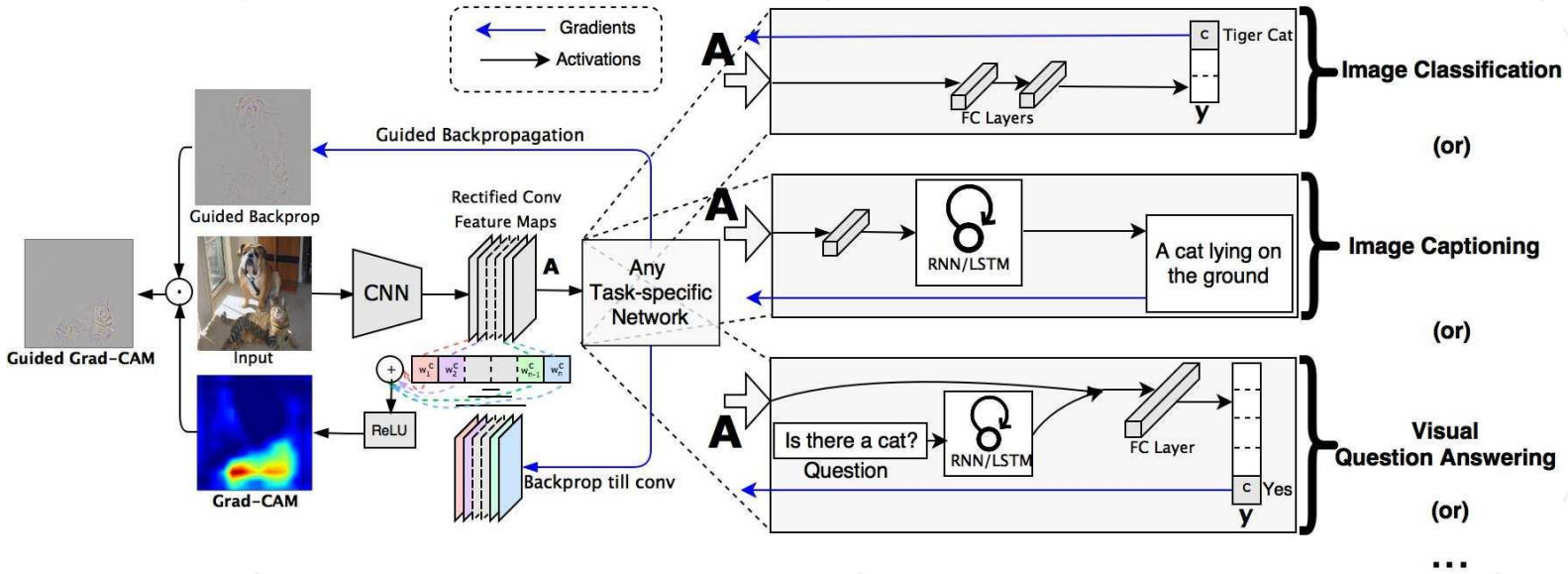
- Learning deep features for discriminative localization
  - Replace FC layer with Convs + GAP + Softmax
  - The class activation map is the weighted sum of weights and Convs outputs



Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. CVPR2016 (pp. 2921-2929).

Visualization of Classification Model

- Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization
  - Need no modification on network
  - Grad-CAM is a generalization to CAM



Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. CVPR2017 (pp. 618-626).

Visualization of Classification Model

- Obtain the class discriminative localization map
  - Compute the gradient of the score for class c with respect to feature maps of a convolutional layer, i.e. $\partial y^c / A_{ij}^k$
  - Use GAP to obtain the neuron importance weights α_k^c

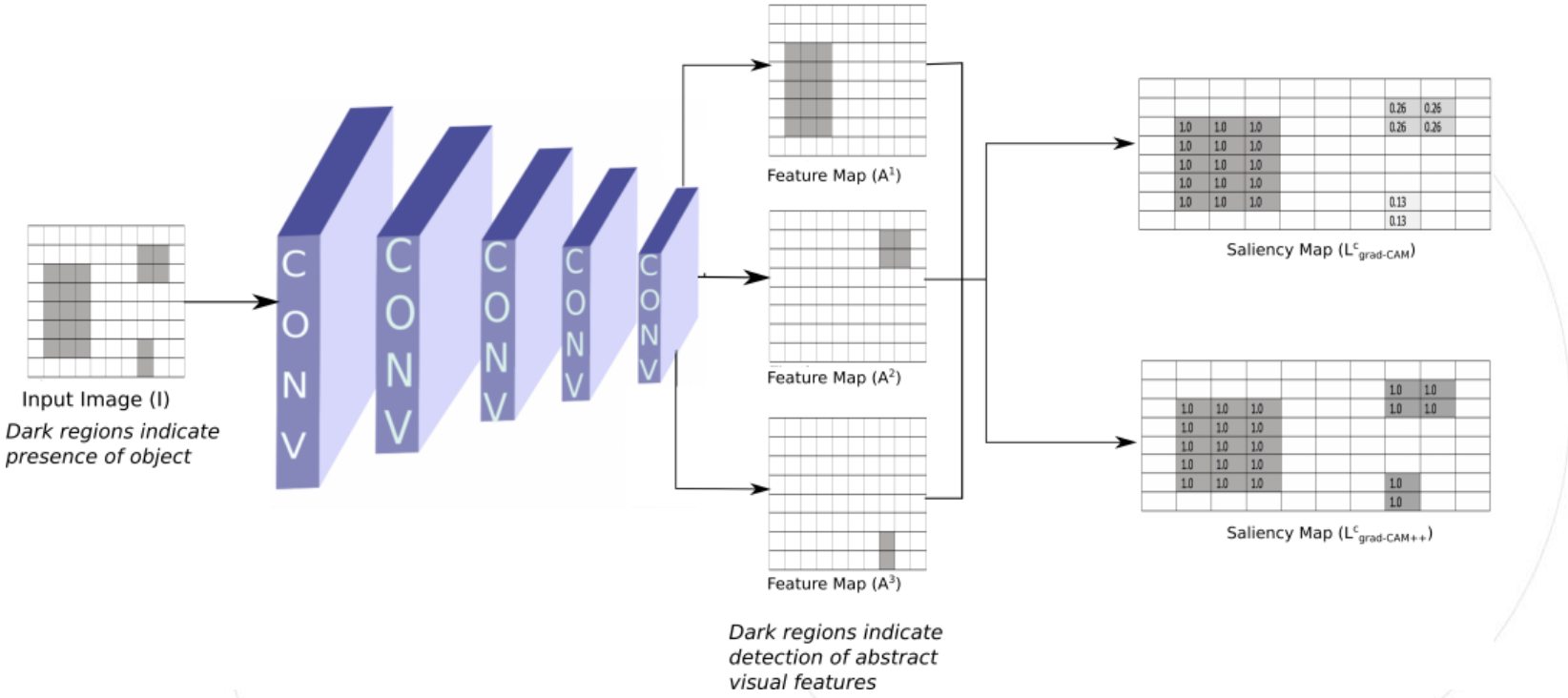
$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

  - Weighted combination of forward activation maps followed by a ReLU

$$L_{\text{Grad-CAM}}^c = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Visualization of Classification Model

- Grad-CAM++
  - Grad-CAM fails when localizing multiple occurrences of the same class
  - Grad-CAM heatmaps often do not capture the entire object in completeness



Chattopadhay A, Sarkar A, et al. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. WACV2018 Mar 12 (pp. 839-847). IEEE.

Visualization of Classification Model

- Grad-CAM++
  - Taking a weighted average of the pixel-wise gradients

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

$$\alpha_k^c = \overbrace{\frac{1}{Z} \sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

  - Combining above two equations

$$Y^c = \sum_k \{ \sum_a \sum_b \alpha_{ab}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ab}^k}) \} [\sum_i \sum_j A_{ij}^k]$$

Visualization of Classification Model

- Grad-CAM++
  - Taking partial derivative w.r.t. A_{ij}^k on both sides:

$$\frac{\partial Y^c}{\partial A_{ij}^k} = \sum_a \sum_b \alpha_{ab}^{kc} \cdot \frac{\partial Y^c}{\partial A_{ab}^k} + \sum_a \sum_b A_{ab}^k \{\alpha_{ij}^{kc} \cdot \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}\}$$

  - Taking a further partial derivative w.r.t. A_{ij}^k:

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} = 2.\alpha_{ij}^{kc} \cdot \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\alpha_{ij}^{kc} \cdot \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}$$

  - Rearranging terms, we get:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}$$

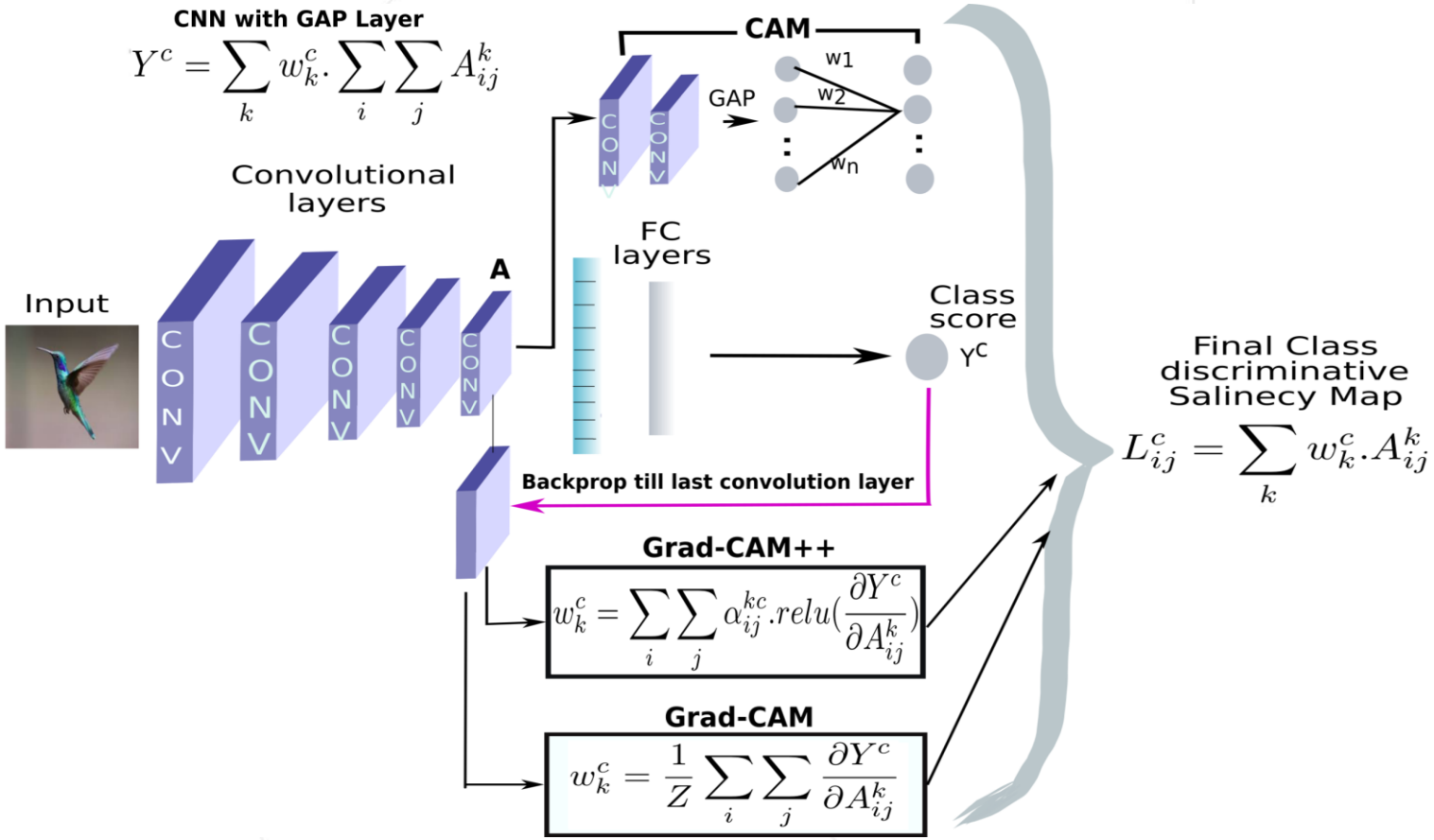
Visualization of Classification Model

- Grad-CAM++
  - Substituting Eqn 10 in Eqn 5, we get the following GradCAM++ weights:

$$w_k^c = \sum_i \sum_j [\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}].relu(\frac{\partial Y^c}{\partial A_{ij}^k}) \quad (11)$$

Visualization of Classification Model

- Overview of CAM, Grad CAM and Grad-CAM++



商汤 sensetime

清華大學 Tsinghua University

Outline

Competitions

- FGVC8 , The Eight Workshop on Fine-Grained Visual Categorization
  - CVPR2021 workshop
  - https://sites.google.com/view/fgvc8

Prior FGVC Workshops

- 7th FGVC Workshop @ CVPR 2020, Virtual
- 6th FGVC Workshop @ CVPR 2019, Long Beach, CA
- 5th FGVC Workshop @ CVPR 2018, Salt Lake City, UT
- 4th FGVC Workshop @ CVPR 2017, Honolulu, HI
- 3rd FGVC Workshop @ CVPR 2015, Boston, MA
- 2nd FGVC Workshop @ CVPR 2013, Columbus, OH
- 1st FGVC Workshop @ CVPR 2011, Colorado Springs, CO

Most Starred Repos

- Related Repository
  - Deep Residual Learning for Image Recognition
  - EfficientNet
  - Mixup
  - MixMatch
  - ReMixMatch
  - Learning to Navigate for Fine-grained Classification

清華大學 Tsinghua University

商汤 sensetime

END