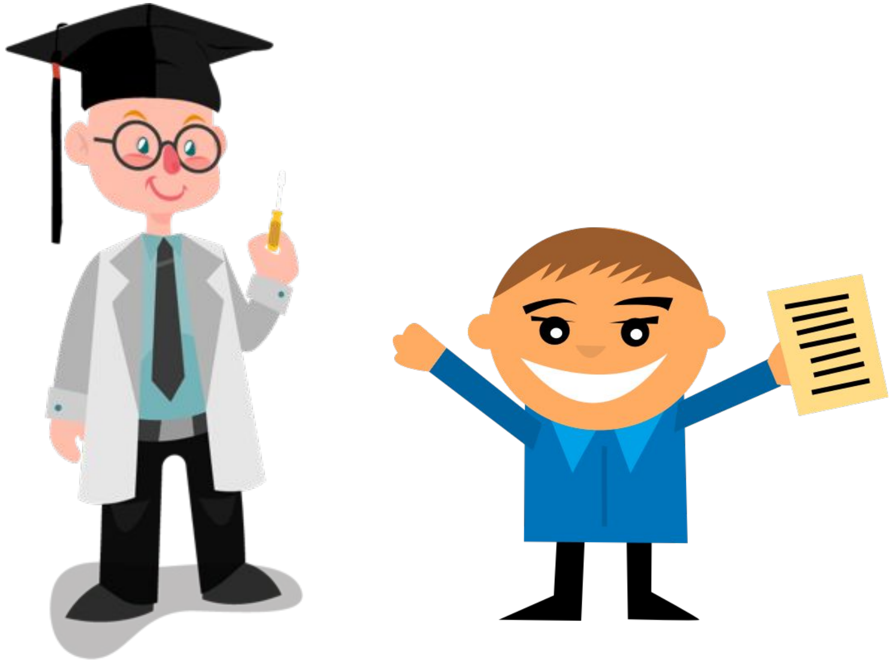
Lecture
Knowledge Distillation

Song Han
songhan@mit.edu

Lecture Plan

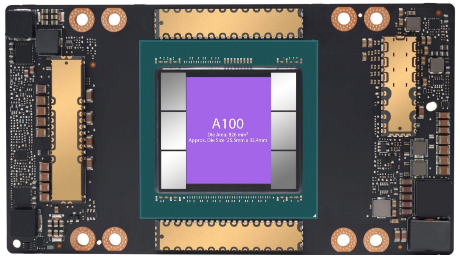
Today we will:

1. What is knowledge distillation;
2. What to match;
3. Self and online distillation;
4. Distillation for different tasks;
5. Network Augmentation, a training technique for tiny machine learning models.



What is knowledge distillation?

Challenge: limited hardware resources

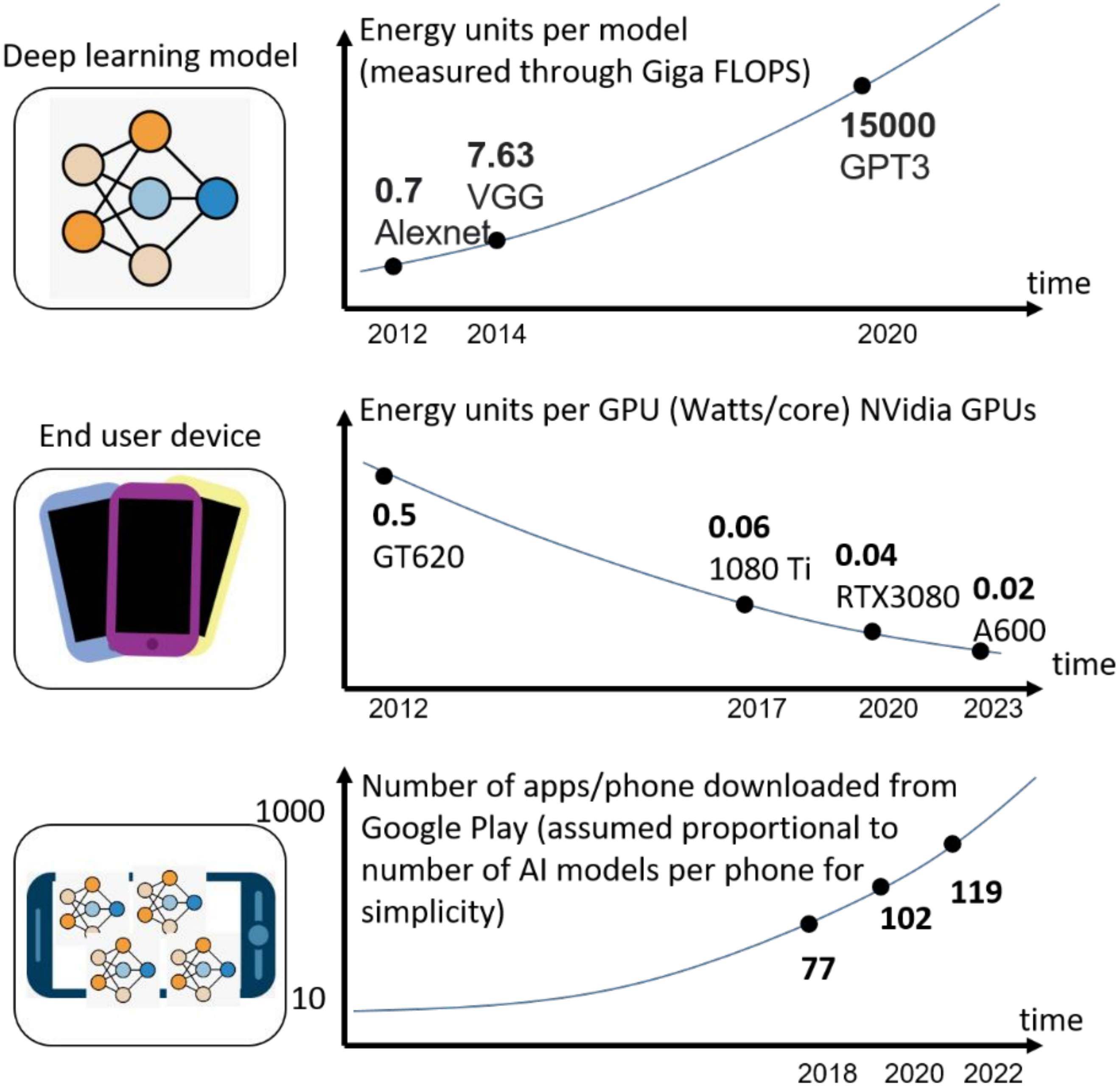


Jetson Nano
4GB GPU

	Cloud AI	Tiny AI
Computation (fp32)	19.5 TFLOPS	MFLOPs
Memory	80GB	256kB
Neural Network Resnet/ViT		MCUNet MobileN etV2-Tin y …

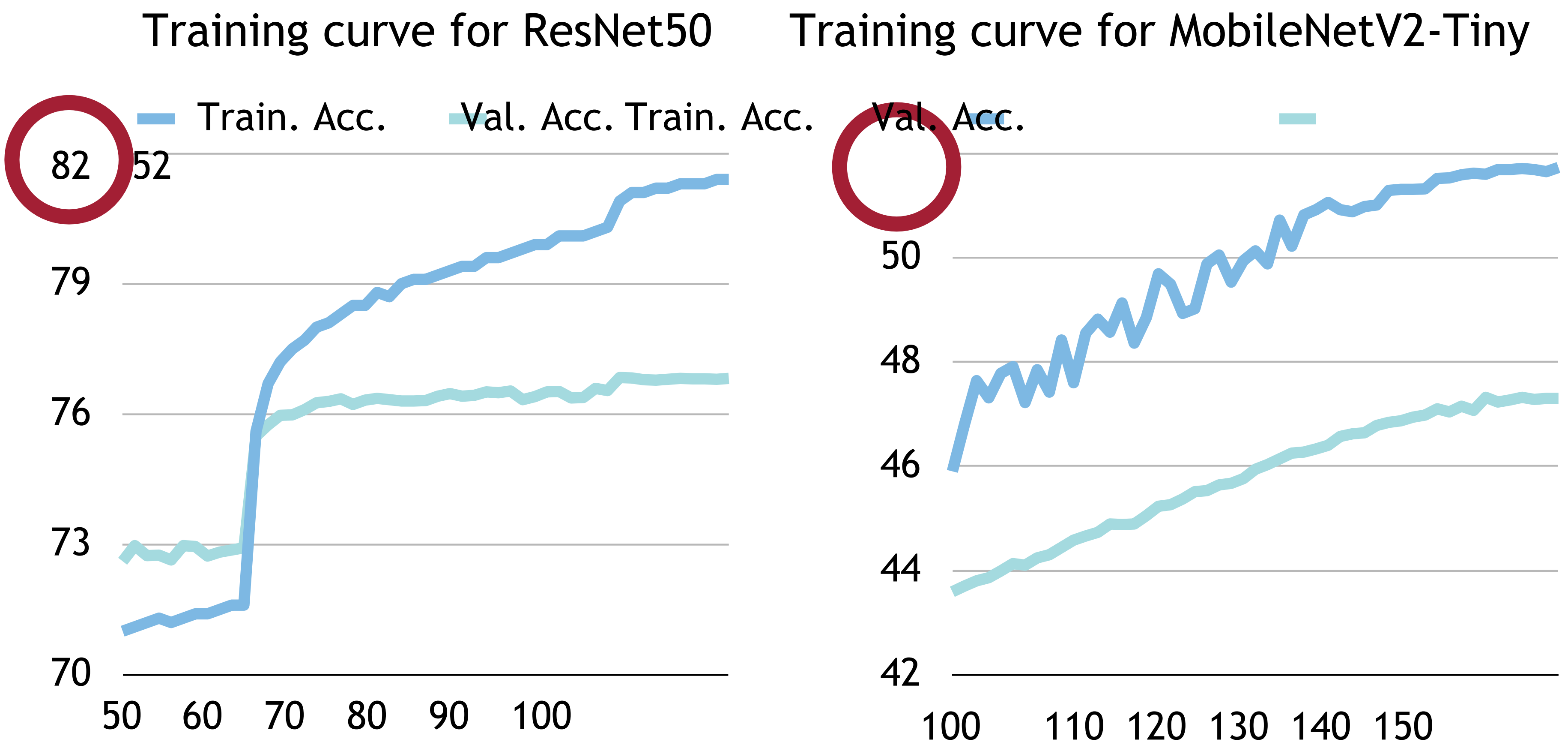
- Must be **tiny** to run efficiently on tiny edge devices.

Network Augmentation for Tiny Deep Learning [Cai *et al.*, ICLR 2022]


Deep learning model
Energy units per model
(measured through Giga FLOPS)
0.7
Alexnet
7.63
VGG
15000
GPT3
time
2012
2014
2020
End user device
Energy units per GPU (Watts/core) NVidia GPUs
0.5
GT620
0.06
1080 Ti
0.04
RTX3080
0.02
A600
time
2012
2017
2020
2023
1000
10
Number of apps/phone downloaded from Google Play (assumed proportional to number of AI models per phone for simplicity)
77
102
119
time
2018
2020
2022

Tiny models are hard to train

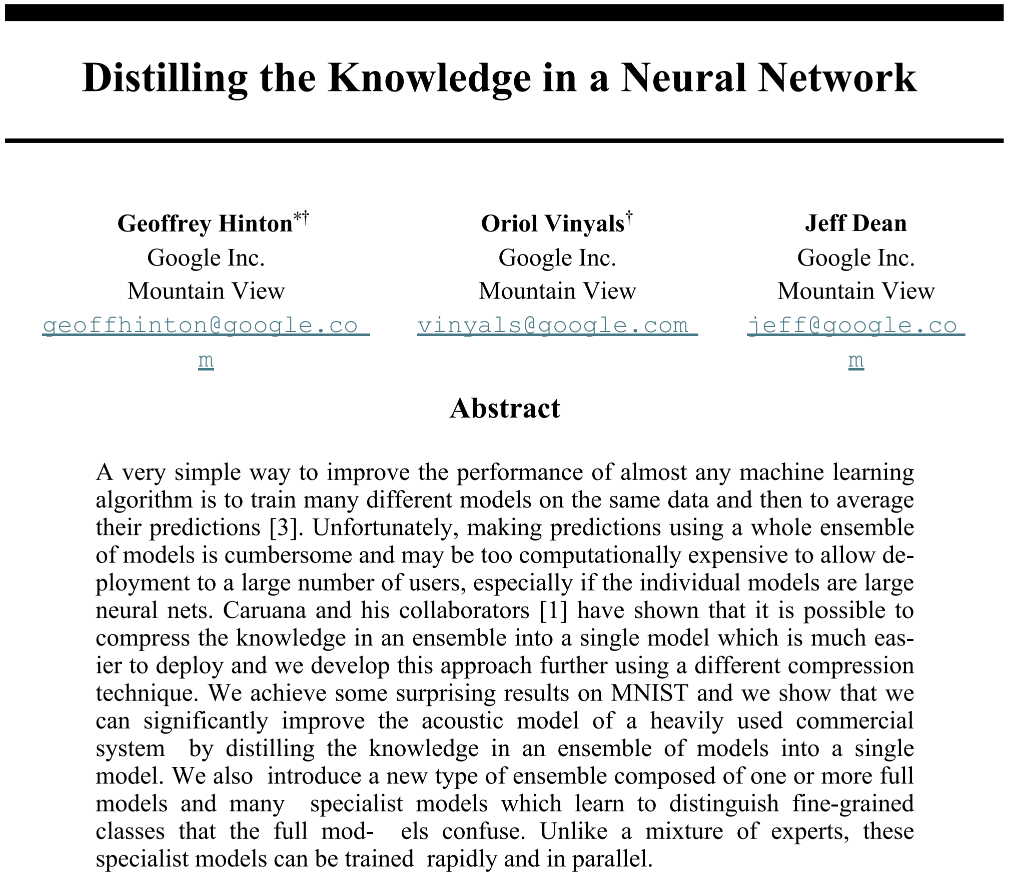
Tiny models underfit large datasets



Question: Can we help the training of tiny models with large models?

Network Augmentation for Tiny Deep Learning [Cai *et al.*, ICLR 2022]

Knowledge Distillation



Distilling the Knowledge in a Neural Network

Geoffrey Hinton[*†]
Google Inc.
Mountain View
geoffhinton@google.co
m

Oriol Vinyals[†]
Google Inc.
Mountain View
vinyals@google.com

Jeff Dean
Google Inc.
Mountain View
jeff@google.co
m

Abstract

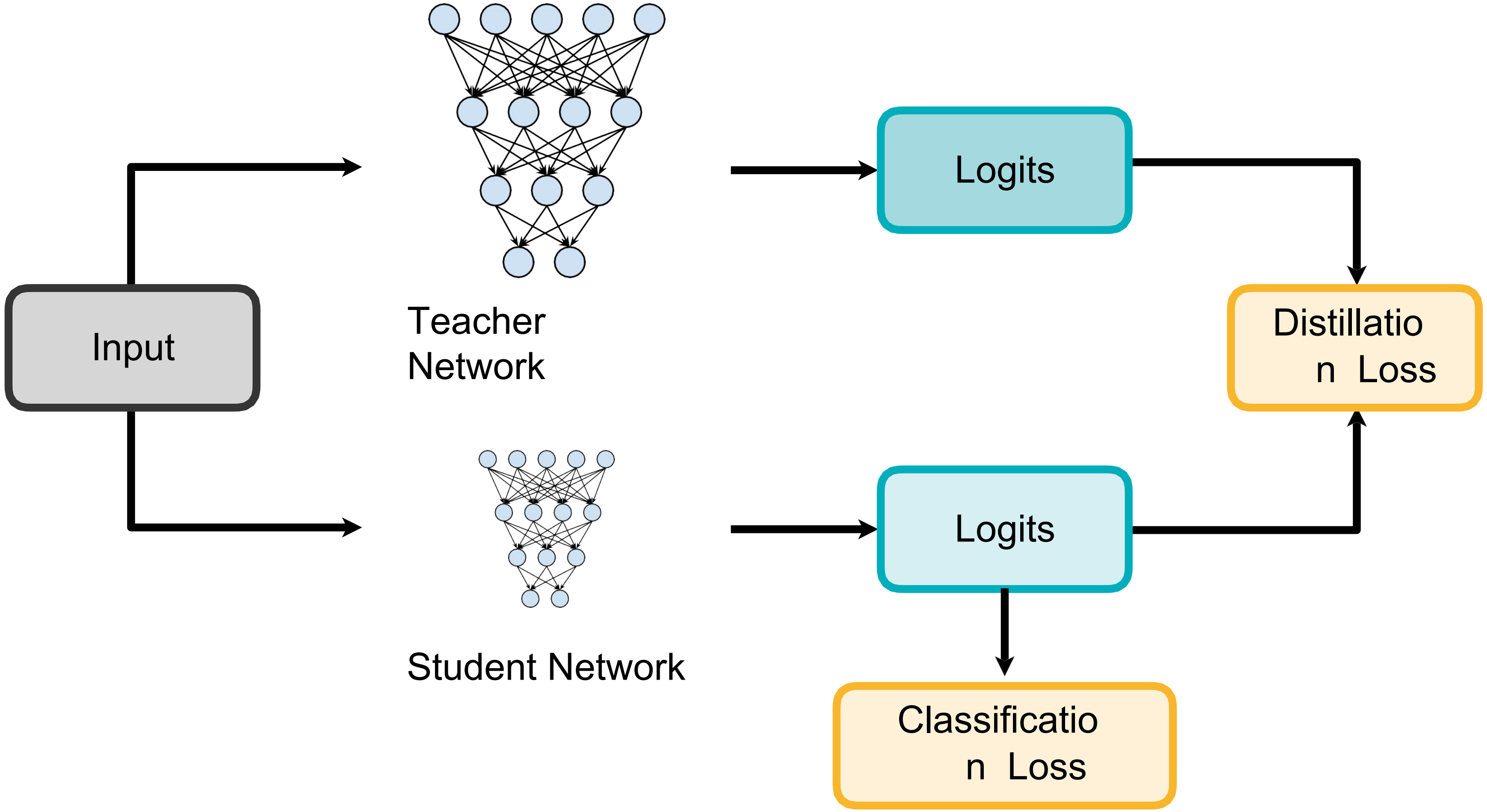
A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions [3]. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators [1] have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full mod- els confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

Distilling the knowledge in a neural network
G Hinton, O Vinyals, J Dean
arXiv preprint arXiv:1503.02531 2 (7)

10910 2015

Distilling the Knowledge in a Neural Network [Hinton *et al.*, NeurIPS Workshops 2014]

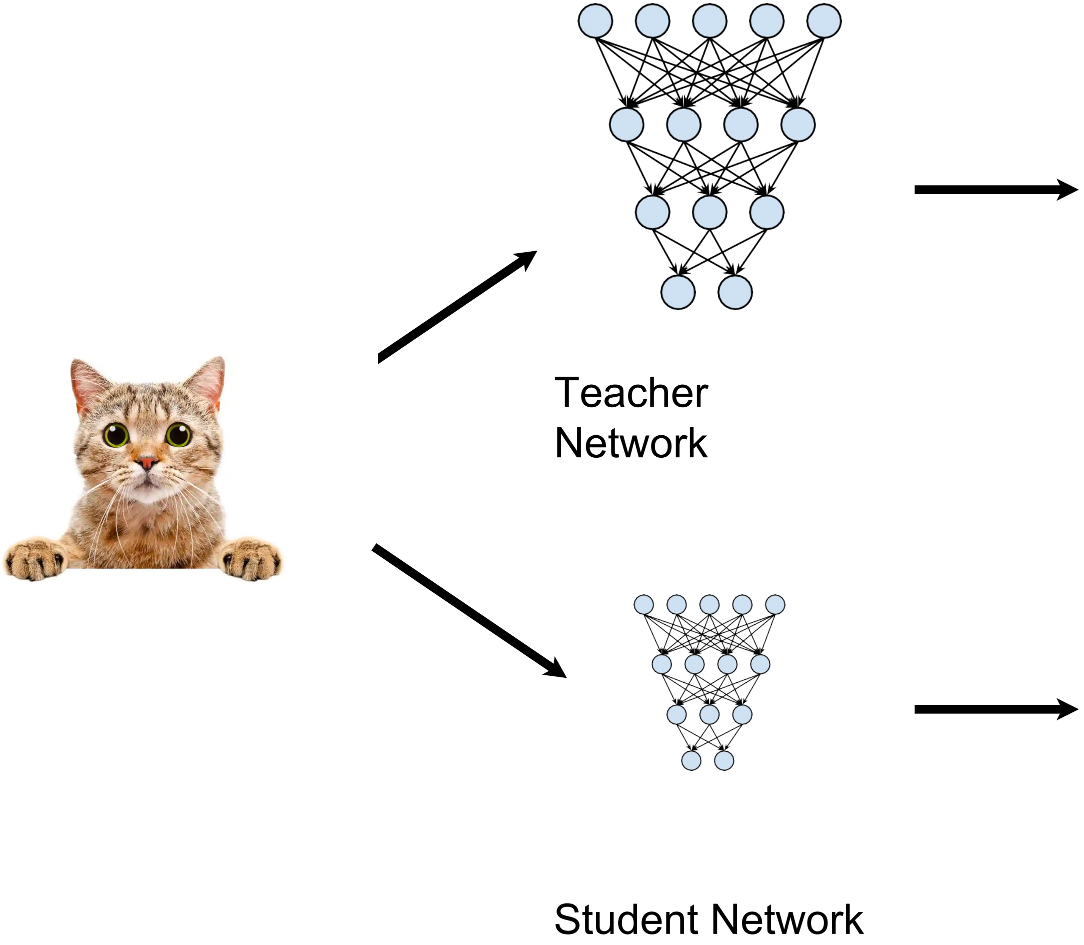
Illustration of knowledge distillation



Knowledge Distillation: A Survey [Gou *et al.*, IJCV 2020]

Intuition of knowledge distillation

Matching prediction probabilities between teacher and student



	Logits	Probabilities
Cat	5	0.982
Dog	1	0.017

$$\frac{\exp(5)}{\exp(5) + \exp(1)}$$

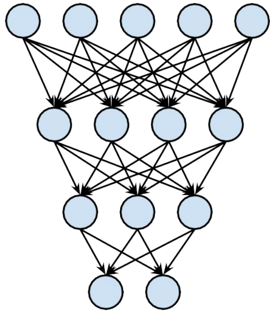
$$\frac{\exp(1)}{\exp(5) + \exp(1)}$$

	Logits	Probabilities
Cat	3	0.731
Dog	2	0.269

The student model is less confident

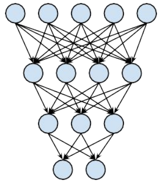
Intuition of knowledge distillation

Matching prediction probabilities between teacher and student



Teacher Network

	Logits	Probabilities
Cat	5	0.982
Dog	1	0.017

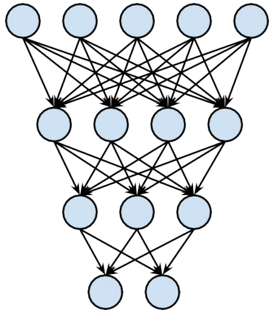


Student Network

	Logits	Probabilities
Cat	3	0.731 ↑
Dog	2	0.269 ↓

Intuition of knowledge distillation

Concept of temperature



$$\frac{\exp(5/1)}{\exp(5/1) + \exp(1/1)}$$

	Logits	Probabilities (T=1)	Probabilities (T=10)
Cat	5	0.982	0.599
Dog	1	0.017	0.401

$$\frac{\exp(5/10)}{\exp(5/10) + \exp(1/10)}$$

A larger temperature smooths the output probability distribution.

Formal Definition of KD

- Neural networks typically use a softmax function to generate the **logits** z_i to class **probabilities** $p(z_i, T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$. Here, $i, j = 0,1,2,...,C -$, where C is the number of classes. T is the temperature, which is normally set to 1.
- The goal of knowledge distillation is to **align the class probability distributions from teacher and student networks**.

Distilling the Knowledge in a Neural Network [Hinton *et al.*, NeurIPS Workshops 2014]

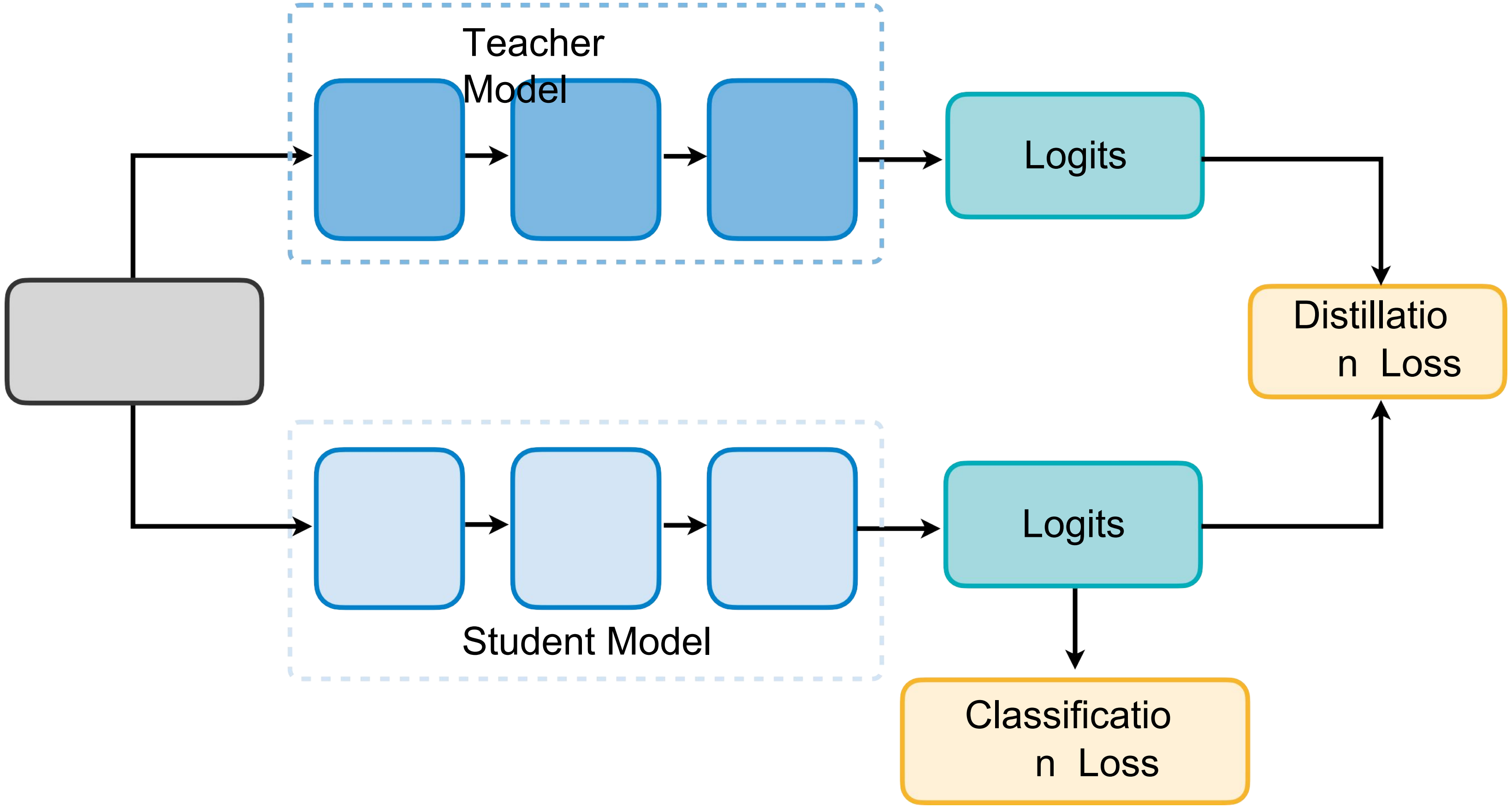
Results

- Resnet50 Teacher
- Resnet18 Student
- CIFAR-10
- T – 95%
- S – 89%
- Train S – 93%

What to match?

1. **Output logits**
2. **Intermediate weights**
3. **Intermediate features**
4. **Gradients**
5. **Sparsity patterns**
6. **Relational information**

Matching output logits



Cross entropy loss:

$E(-p_t \log p_s)$;

L2 loss:

$E(||p_t - p_s||_2^2)$

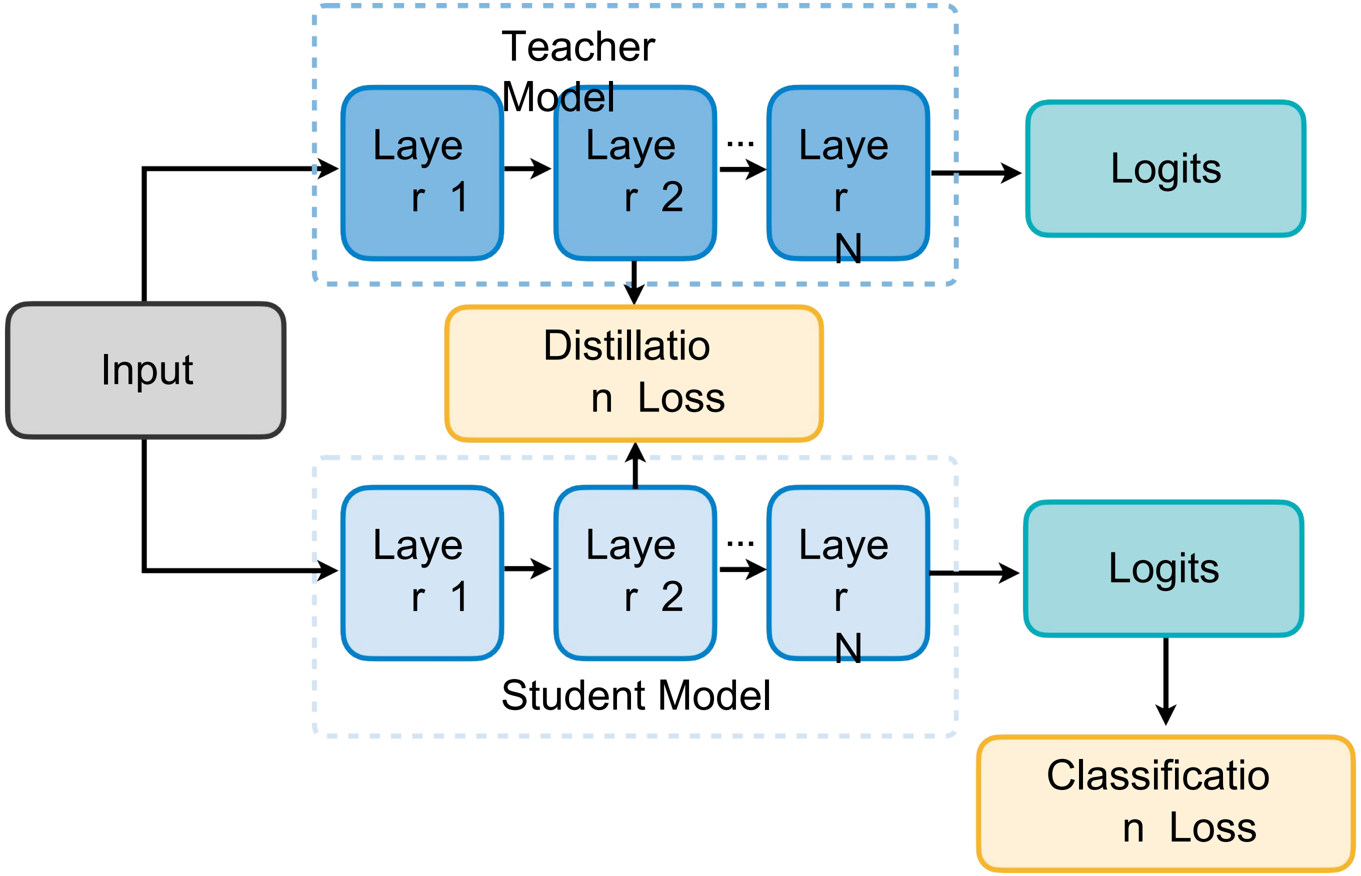
Distilling the Knowledge in a Neural Network [Hinton *et al.*, NeurIPS Workshops 2014]
Do Deep Nets Really Need to be Deep? [Ba and Caruana, NeurIPS 2014]

What to match?

1. Output logits
2. **Intermediate weights**
3. Intermediate features
4. Gradients
5. Sparsity patterns
6. Relational information

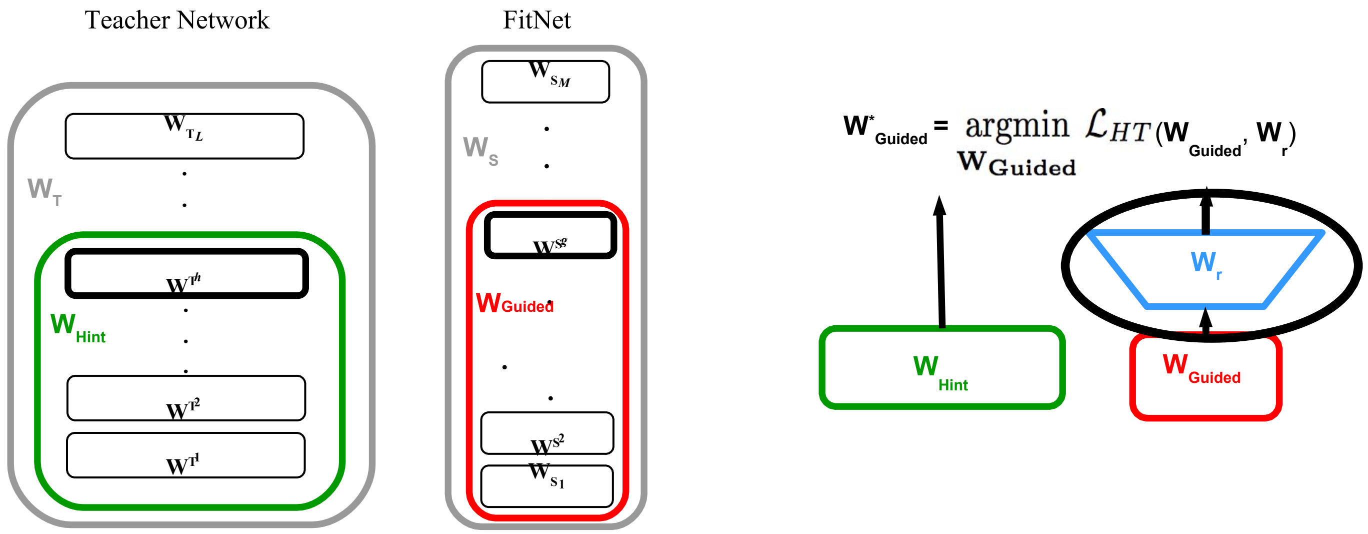
What else to match other than output logits?

Matching intermediate weights



Knowledge Distillation: A Survey [Gou *et al.*, IJCV 2020]

Matching intermediate weights



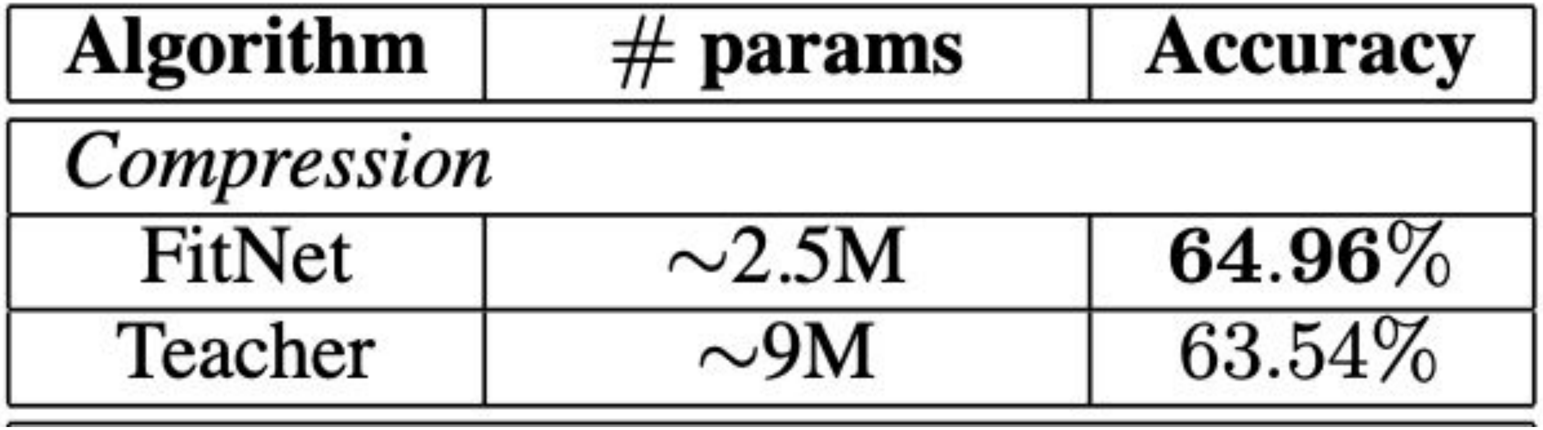
(a) Teacher and Student Networks (b) Hints Training

An FC layer used to align the shapes of teacher and student weights

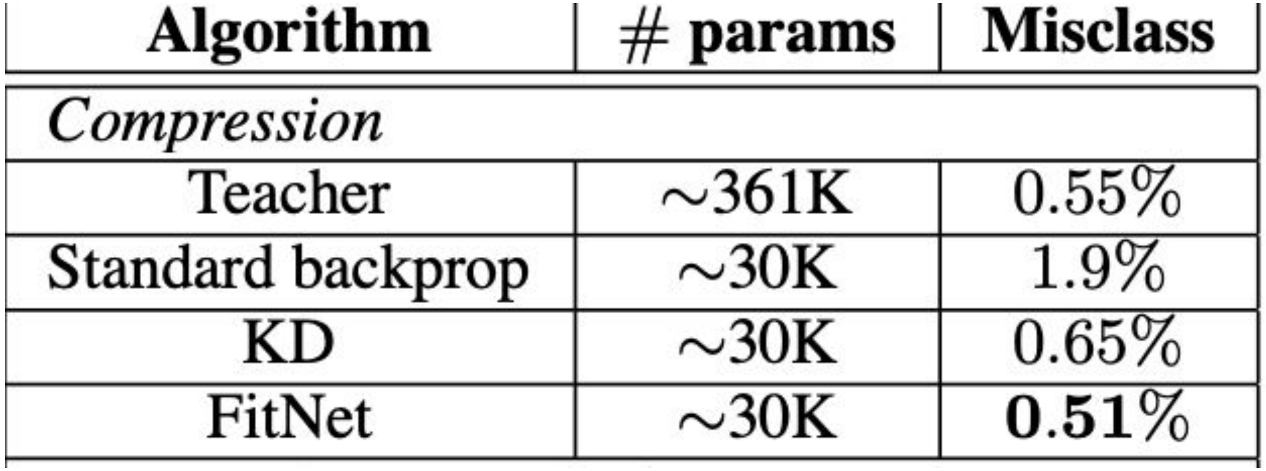
- Other than the cross-entropy distillation loss, also add a L2 loss between teacher weights and student weights (linear transformation is applied to match the dimensionalities).

FitNets: Hints for Thin Deep Nets [Romero *et al.*, ICLR 2015]

CIFAR-100/ MNIST



Algorithm	**# params**	**Accuracy**
Compression		
FitNet	∼2.5M	**64.96%**
Teacher	∼9M	63.54%



Algorithm	**# params**	**Misclass**
Compression		
Teacher	∼361K	0.55%
Standard backprop	∼30K	1.9%
KD	∼30K	0.65%
FitNet	∼30K	**0.51%**

What to match?

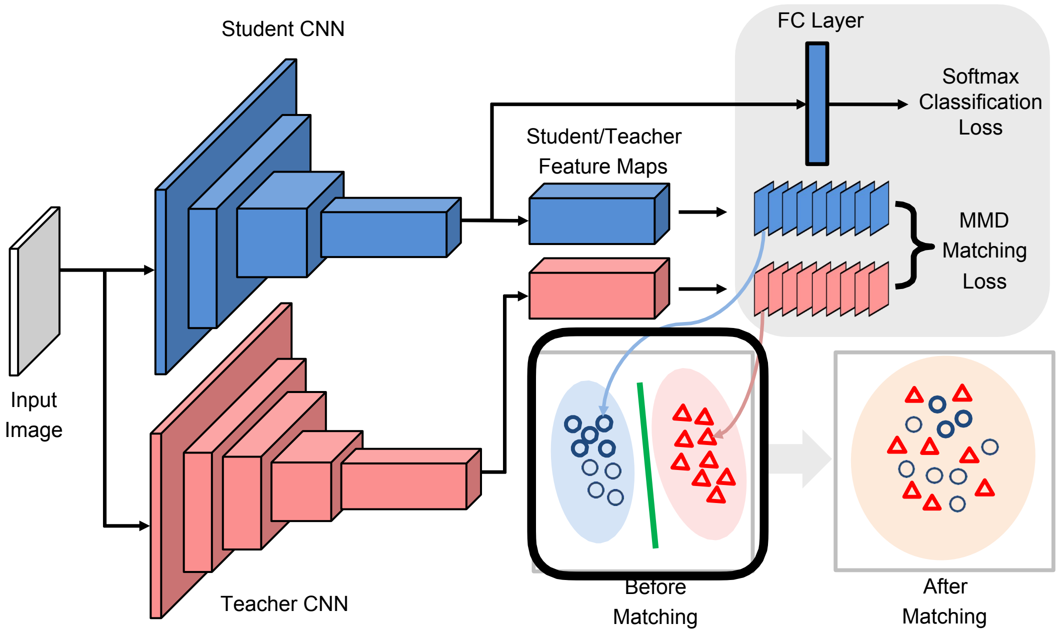
1. Output logits
2. Intermediate weights
3. **Intermediate features**
4. Gradients
5. Sparsity patterns
6. Relational information

Matching intermediate features

Minimizing maximum mean discrepancy between feature maps

- Intuition: teacher and student networks should have similar **feature** distributions, not just output probability distributions.

$$\mathcal{L}_{\mathrm{MMD}^2}(\mathcal{X},\mathcal{Y}) = \|\frac{1}{N}\sum_{i=1}^{N}\phi(\boldsymbol{x}^i) - \frac{1}{M}\sum_{j=1}^{M}\phi(\boldsymbol{y}^j)\|_2^2,$$



$$\begin{aligned}\mathcal{L}_{\mathrm{MMD}^2}(\mathcal{X},\mathcal{Y}) &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N}k(\boldsymbol{x}^i,\boldsymbol{x}^{i'})\\ &+ \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M}k(\boldsymbol{y}^i,\boldsymbol{y}^{i'})\\ &- \frac{2}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}k(\boldsymbol{x}^i,\boldsymbol{y}^j),\end{aligned}$$

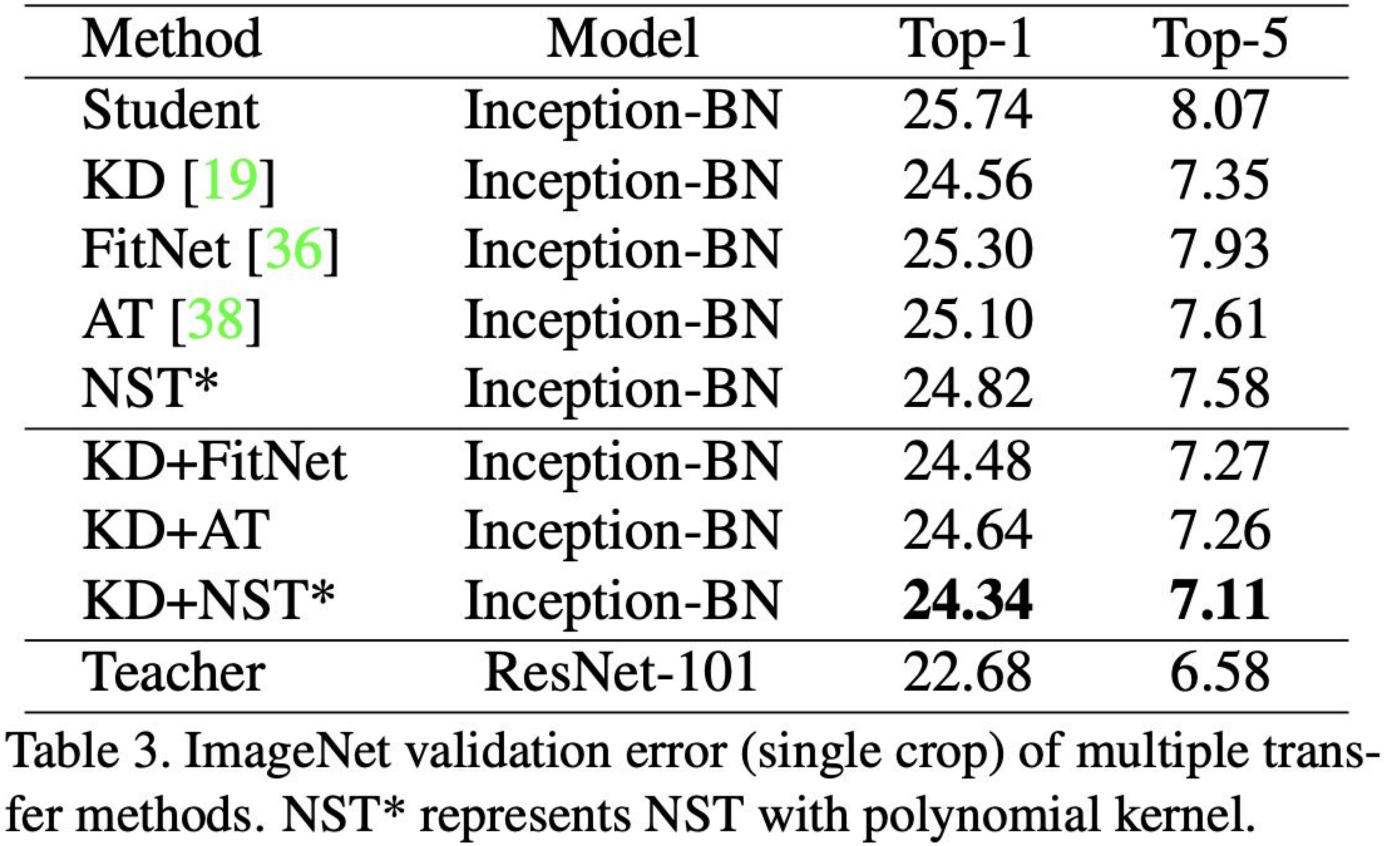
Teacher and student have very different feature distributions without distillation

The feature maps can be interpolated if their dimensions do not match.

Like What You Like: Knowledge Distill via Neuron Selectivity Transfer [Huang and Wang, arXiv 2017]

Kernels and Results

- Linear Kernel: $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$
- Polynomial Kernel: $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y} + c)^d$
- Gaussian Kernel: $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{2\sigma^2})$



Method	Model	Top-1	Top-5
Student	Inception-BN	25.74	8.07
KD [19]	Inception-BN	24.56	7.35
FitNet [36]	Inception-BN	25.30	7.93
AT [38]	Inception-BN	25.10	7.61
NST*	Inception-BN	24.82	7.58
KD+FitNet	Inception-BN	24.48	7.27
KD+AT	Inception-BN	24.64	7.26
KD+NST*	Inception-BN	**24.34**	**7.11**
Teacher	ResNet-101	22.68	6.58

Table 3. ImageNet validation error (single crop) of multiple transfer methods. NST* represents NST with polynomial kernel.

What to match?

1. Output logits
2. Intermediate weights
3. Intermediate features
4. **Gradients**
5. Sparsity patterns
6. Relational information

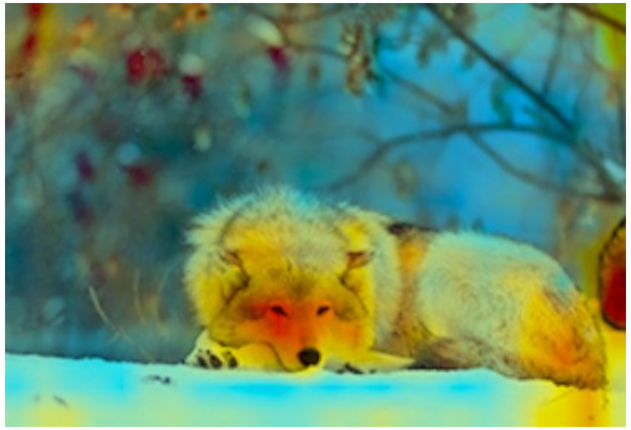
Intermediate attention maps

Gradients of feature maps are used to characterize “attention” of DNNs

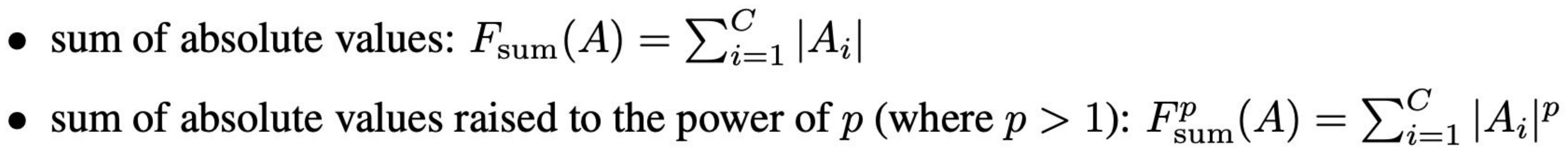
- The attention of a CNN feature map is defined as $\frac{dL}{dx}$, where L is the learning objective.
- Intuition: If $\frac{dL}{dx_{i,j}}$ is large, a small perturbation at i, j will significantly impact the final output. As a result, the network is putting more attention on position i, j.

input image

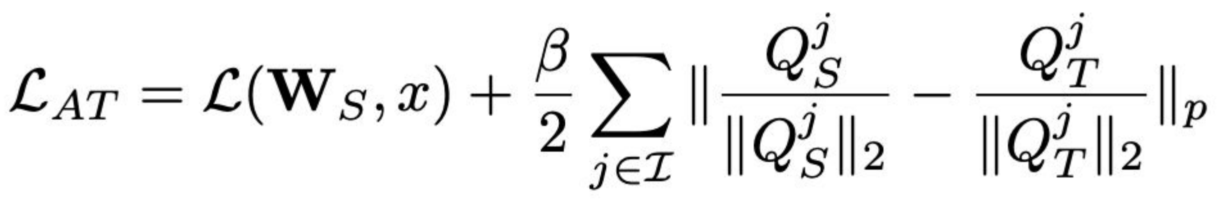
attention map



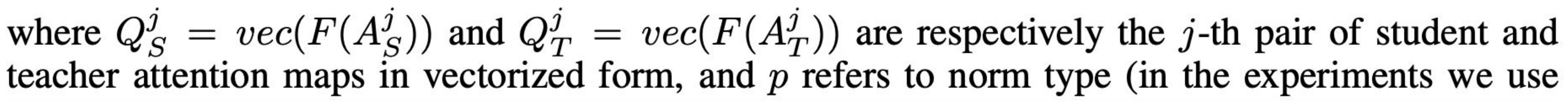
Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer [Zagoruyko and Komodakis, ICLR 2017]



- sum of absolute values: $F_{\text{sum}}(A) = \sum_{i=1}^{C} |A_i|$
- sum of absolute values raised to the power of p (where $p > 1$): $F^p_{\text{sum}}(A) = \sum_{i=1}^{C} |A_i|^p$



$$\mathcal{L}_{AT} = \mathcal{L}(\mathbf{W}_S, x) + \frac{\beta}{2} \sum_{j \in \mathcal{I}} \| \frac{Q_S^j}{\|Q_S^j\|_2} - \frac{Q_T^j}{\|Q_T^j\|_2} \|_p$$

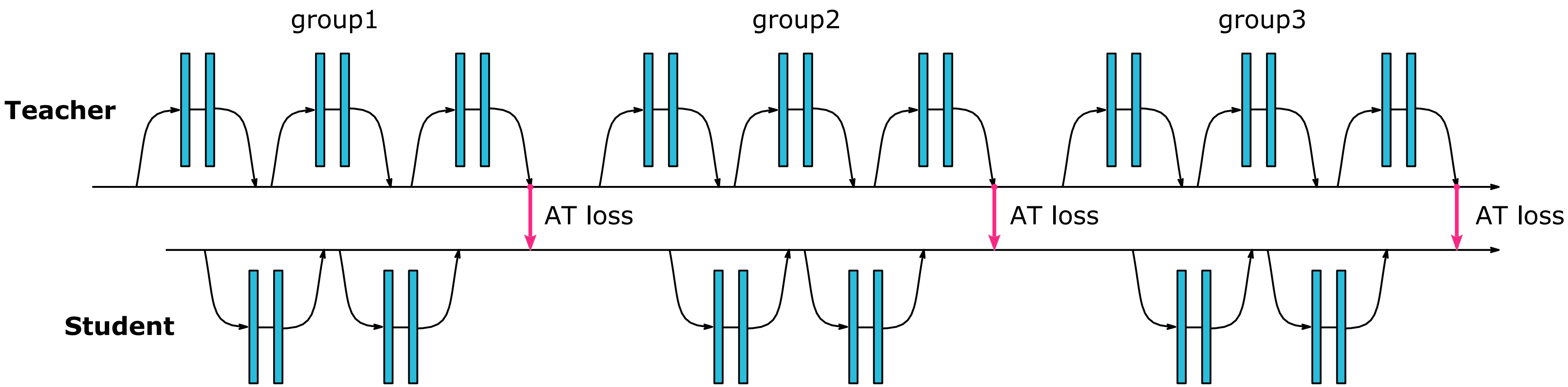


where $Q_S^j = vec(F(A_S^j))$ and $Q_T^j = vec(F(A_T^j))$ are respectively the j-th pair of student and teacher attention maps in vectorized form, and p refers to norm type (in the experiments we use

Without loss of generality, we assume that transfer losses are placed between student and teacher attention maps of same spatial resolution, but, if needed, attention maps can be interpolated.

Matching intermediate attention maps

- The attention transfer objective is defined as: $\frac{\beta}{2}||J_S - J_T||_2^2$, here J_S is the student attention map (gradient of student feature map) and J_T is the teacher attention map. β is a constant.



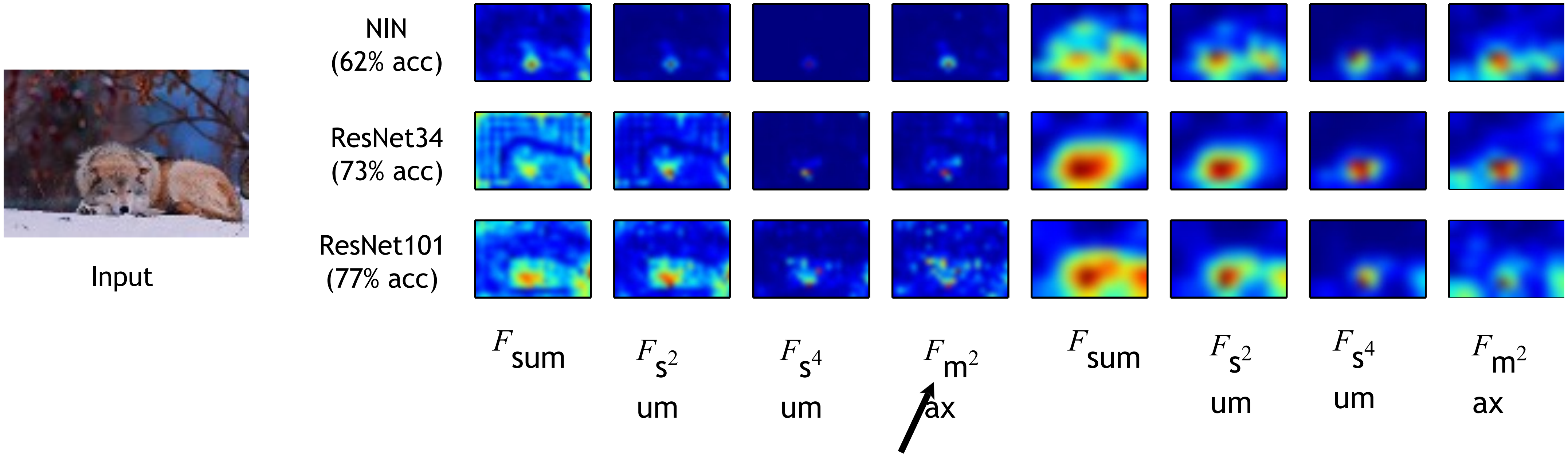
$$J_S = \frac{\partial}{\partial x}\mathcal{L}(\mathbf{W_S}, x), J_T = \frac{\partial}{\partial x}\mathcal{L}(\mathbf{W_T}, x)$$

Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer [Zagoruyko and Komodakis, ICLR 2017]

Intermediate attention maps

Performant models have similar attention maps

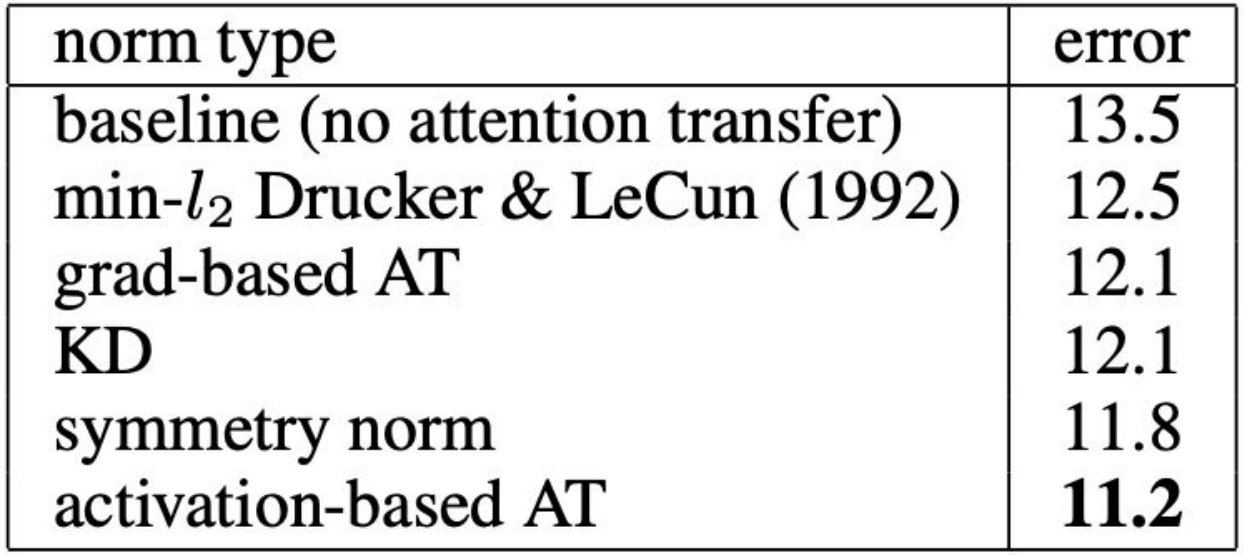
- Attention maps of performant ImageNet models (ResNets) are indeed similar to each other, but the less performant model (NIN) has quite different attention maps.



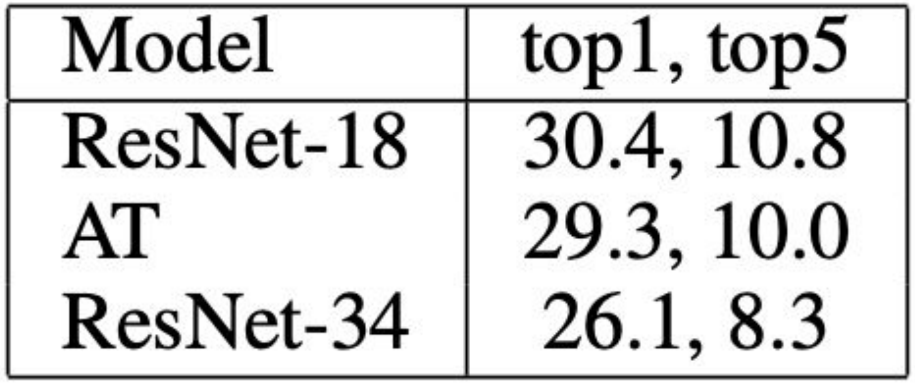
Different reduction methods across the channel dimensions

Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer [Zagoruyko and Komodakis, ICLR 2017]

CIFAR-10 and ImageNet



norm type	error
baseline (no attention transfer)	13.5
min-l_2 Drucker & LeCun (1992)	12.5
grad-based AT	12.1
KD	12.1
symmetry norm	11.8
activation-based AT	**11.2**



Model	top1, top5
ResNet-18	30.4, 10.8
AT	29.3, 10.0
ResNet-34	26.1, 8.3

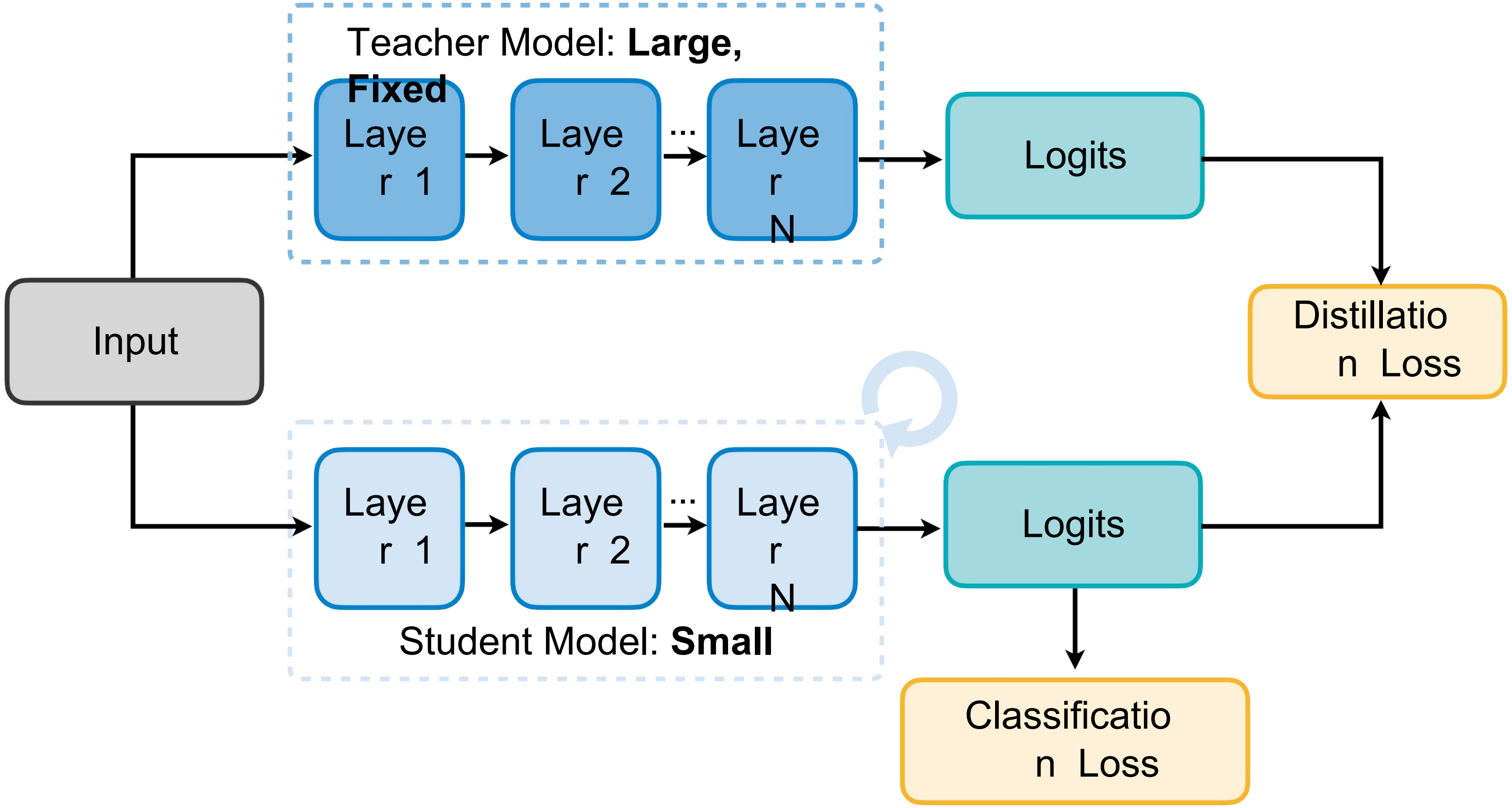
Baseline is a thin NIN network with 0.2M param
Teacher NIN-wide, 1M

Self and Online Distillation

1. **Self Distillation**
2. **Online Distillation**
3. **Combined**

Overview of knowledge distillation

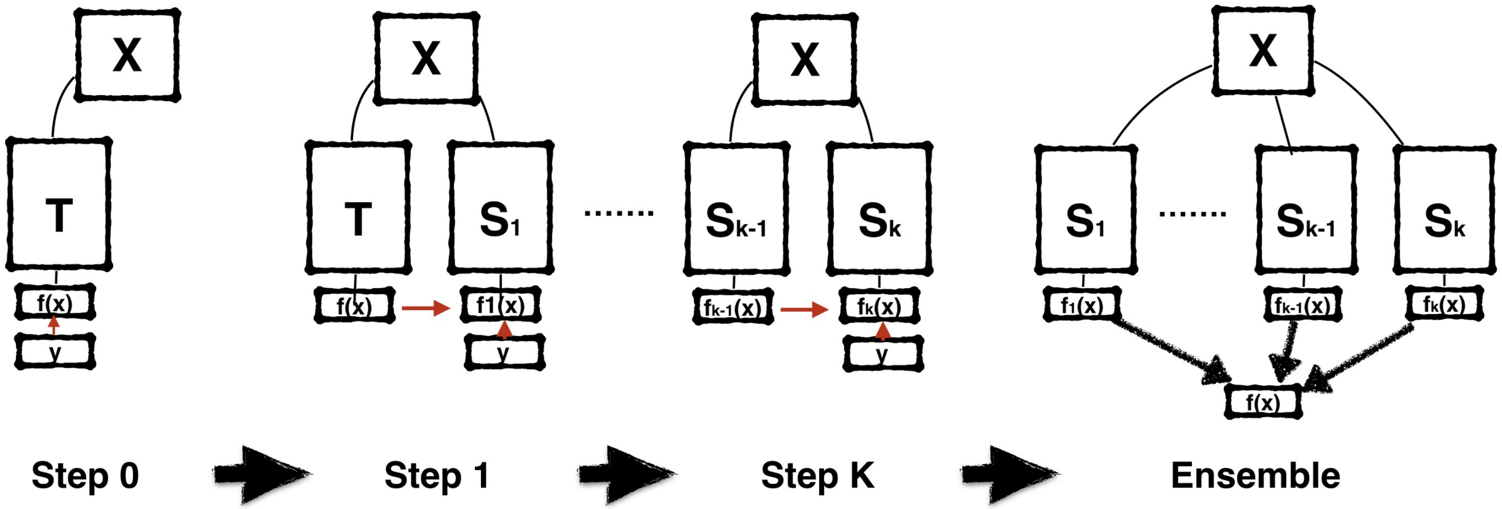
Teacher model is usually larger than the student model and is fixed



Discussion: What is the disadvantage of fixed large teachers? Does it have to be the case that we need a fixed large teacher in KD?

Knowledge Distillation: A Survey [Gou *et al.*, IJCV 2020]

Self-Distillation with Born-Again NNs



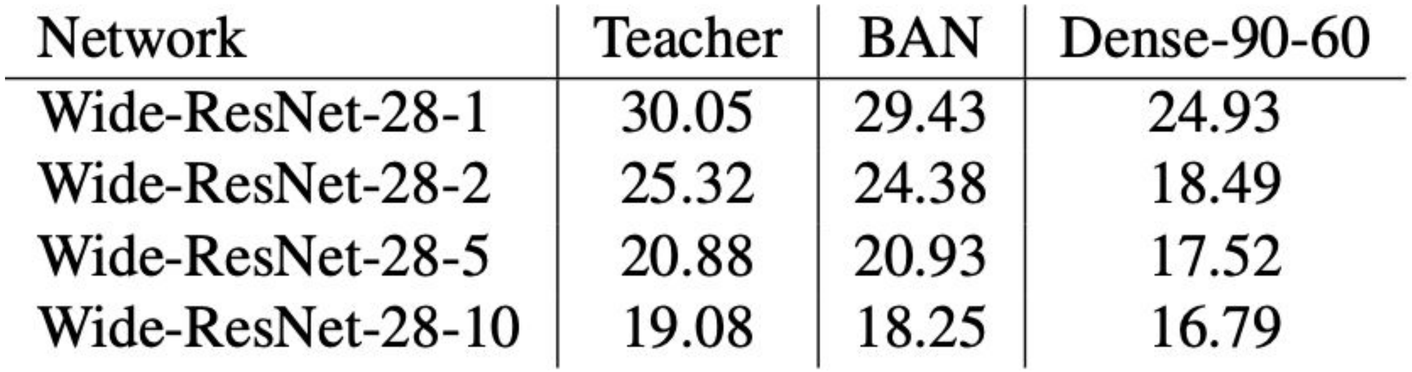
- Born-Again Networks generalizes defensive distillation by **adding iterative training stages** and **using both classification objective and distillation objective** in subsequent stages.
- Network architecture $T = S_1 \quad = S_2 \quad = \ldots$
- $= S_k$. Network accuracy $T < S_1 < S_2 \quad < \ldots$
- $< S_k$.

Can alternatively ensemble $T, S_1, S_2, \ldots, S_k$ to get even better performance.

Teacher is k-1 and student is k, the learnt teacher transfer the knowledge to student

$$\mathcal{L}(f(x, \arg\min_{\theta_1} \mathcal{L}(y, f(x, \theta_1))), f(x, \theta_2)).$$

$$\mathcal{L}(f(x, \arg\min_{\theta_{k-1}} \mathcal{L}(f(x, \theta_{k-1}))), f(x, \theta_k)).$$



Network	Teacher	BAN	Dense-90-60
Wide-ResNet-28-1	30.05	29.43	24.93
Wide-ResNet-28-2	25.32	24.38	18.49
Wide-ResNet-28-5	20.88	20.93	17.52
Wide-ResNet-28-10	19.08	18.25	16.79

Test error on CIFAR-100

BAN-3

Wide-ResNet students trained from identical Wide-ResNet teachers and for DenseNet-90-60 students trained from Wide-ResNet teachers

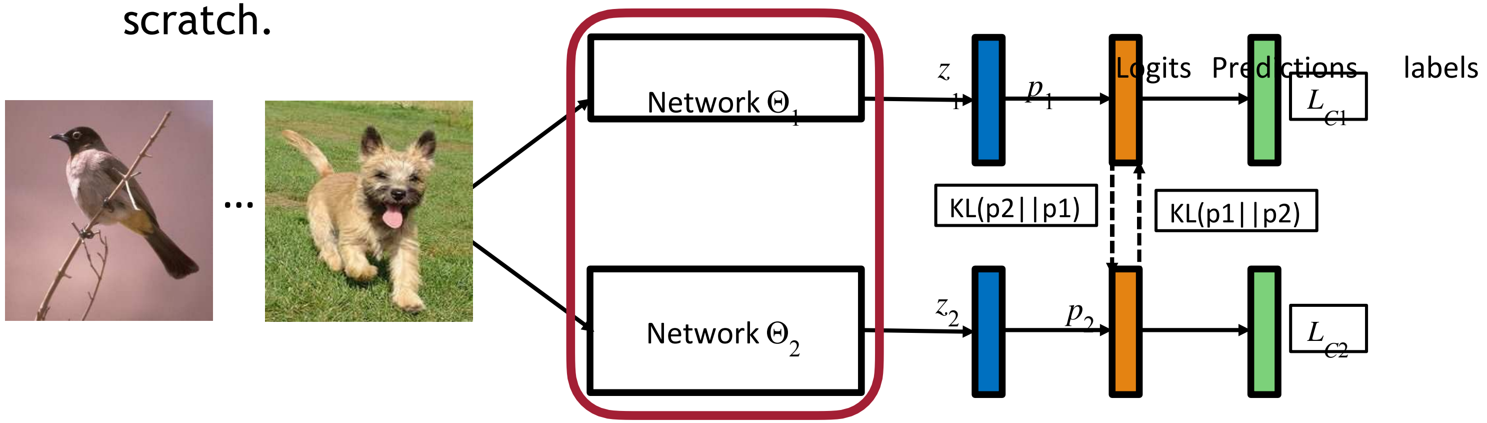
Self and Online Distillation

1. Self Distillation
2. **Online Distillation**
3. Combined

Online Distillation

Deep Mutual Learning

Θ_1, Θ_2 can be the same or different, and they are trained from scratch.



- Idea of deep mutual learning: for both teacher and student networks, we want to add a distillation objective that minimizes the output distribution of the other party.
- $\mathrm{B}(S) = \text{CrossEntropy}(S(I), y) + \text{KL}(S(I), T(I))$;
- $\mathrm{B}(T) = \text{CrossEntropy}(T(I), y) + \text{KL}(T(I), S(I))$.
- Note: it is not necessary to pretrain *T* $S{=}T$ is allowed.
  and

Deep Mutual Learning [Zhang *et al.*, CVPR 2018]

Online Distillation

Deep Mutual Learning

Network Types		CIFAR-10						CIFAR-100					
		Independent		DML		DML-Ind		Independent		DML		DML-Ind	
Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2
Resnet-32	Resnet-32	92.47	92.47	92.68	92.80	0.21	0.33	68.99	68.99	71.19	70.75	2.20	1.76
WRN-28-10	Resnet-32	95.01	92.47	95.75	93.18	0.74	0.71	78.69	68.99	78.96	70.73	0.27	1.74
MobileNet	Resnet-32	93.59	92.47	94.24	93.32	0.65	0.85	73.65	68.99	76.13	71.10	2.48	2.11
MobileNet	MobileNet	93.59	93.59	94.10	94.30	0.51	0.71	73.65	73.65	76.21	76.10	2.56	2.45
WRN-28-10	MobileNet	95.01	93.59	95.73	94.37	0.72	0.78	78.69	73.65	80.28	77.39	1.59	3.74
WRN-28-10	WRN-28-10	95.01	95.01	95.66	95.63	0.65	0.62	78.69	78.69	80.28	80.08	1.59	1.39

Deep mutual learning can improve both student (net 2) and teacher (net 1) models.

Deep Mutual Learning [Zhang *et al.*, CVPR 2018]