Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning

Konstanty Subbotko

Koło Naukowe Uczenia Maszynowego UW

3 listopada 2021

Paper

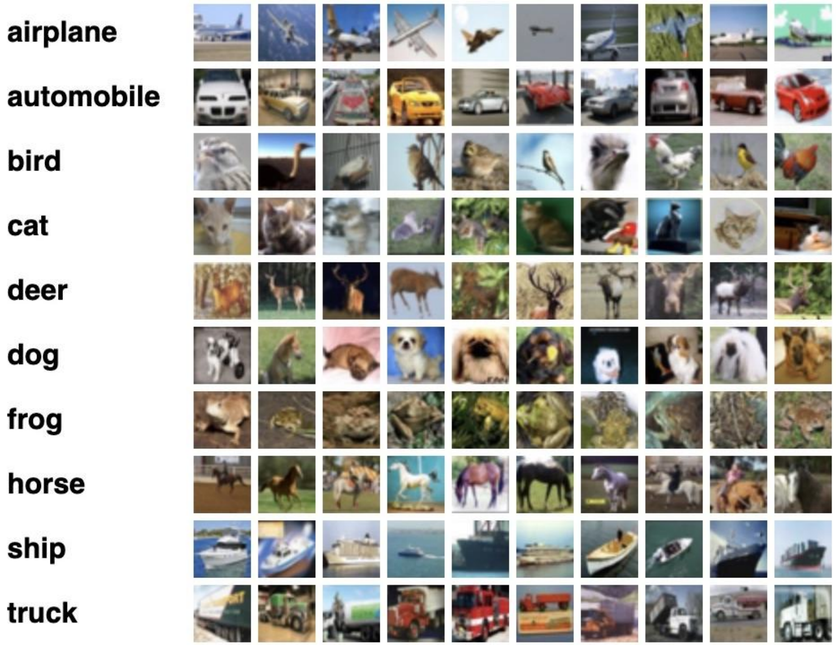
Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Microsoft Research Redmond

Yuanzhi Li
yuanzhil@andrew.cmu.edu
Carnegie Mellon University

Setup

- Experiments on CIFAR-10/CIFAR-100
- ResNet



Mystery 1: Ensemble

F_1, F_2, ..., F_{10} - identical networks with **different initializations** trained **independently**

Model	Test accuracy
F_1 , F_2 , ... , F_{10}	81.51 +/- 0.16%
$(F_1 + F_2 + ... + F_{10})$ / 10 (trained as average)	81.83%
$(F_1 + F_2 + ... + F_{10})$ / 10 (ensemble)	84.87%

Mystery 1: Ensemble

F_1, F_2, ..., F_{10} - identical networks with **different initializations** trained **independently**

Model	Test accuracy
F_1 , F_2 , ... , F_{10}	81.51 +/- 0.16%
$(F_1 + F_2 + ... + F_{10}) / 10$ (trained as average)	81.83%
$(F_1 + F_2 + ... + F_{10}) / 10$ (ensemble)	84.87%

Same models, same dataset but they give better accuracy combined

Mystery 2: Knowledge distillation

Using 10 models during inference is quite expensive!

Solution: train a single network S to emulate ensemble T = $(F_1 + F_2 + \ldots + F_{10}) / 10$

Mystery 2: Knowledge distillation

Using 10 models during inference is quite expensive!

Solution: train a single network S to emulate ensemble $T = (F_1 + F_2 + \ldots + F_{10}) / 10$

$L(x, y) = t * L_{class}(S(x), y) + (1 - t) * H(S(x), T(x))$

where H - cross-entropy loss function

Mystery 2: Knowledge distillation

Using 10 models during inference is quite expensive!

Solution: train a single network S to emulate ensemble T = $(F_1 + F_2 + \ldots + F_{10}) / 10$

$L(x, y) = t * L_{class}(S(x), y) + (1 - t) * H(S(x), T(x))$
where H - cross-entropy loss function

This way S learns distribution of classes from T (known as knowledge distillation)

Mystery 2: Knowledge distillation

Model	Test accuracy
$F_1, F_2, \dots, F_{10}$	81.51 +/- 0.16%
$(F_1 + F_2 + \dots + F_{10}) / 10$ (ensemble)	84.87%
S (distilled from ensemble)	83.81%

Mystery 2: Knowledge distillation

Model	Test accuracy
$F_1, F_2, \dots, F_{10}$	81.51 +/- 0.16%
$(F_1 + F_2 + \dots + F_{10}) / 10$ (ensemble)	84.87%
S (distilled from ensemble)	83.81%

S has the same architecture as F_1, F_2, ..., F_{10} but gives better test accuracy

Mystery 3: Self-distillation

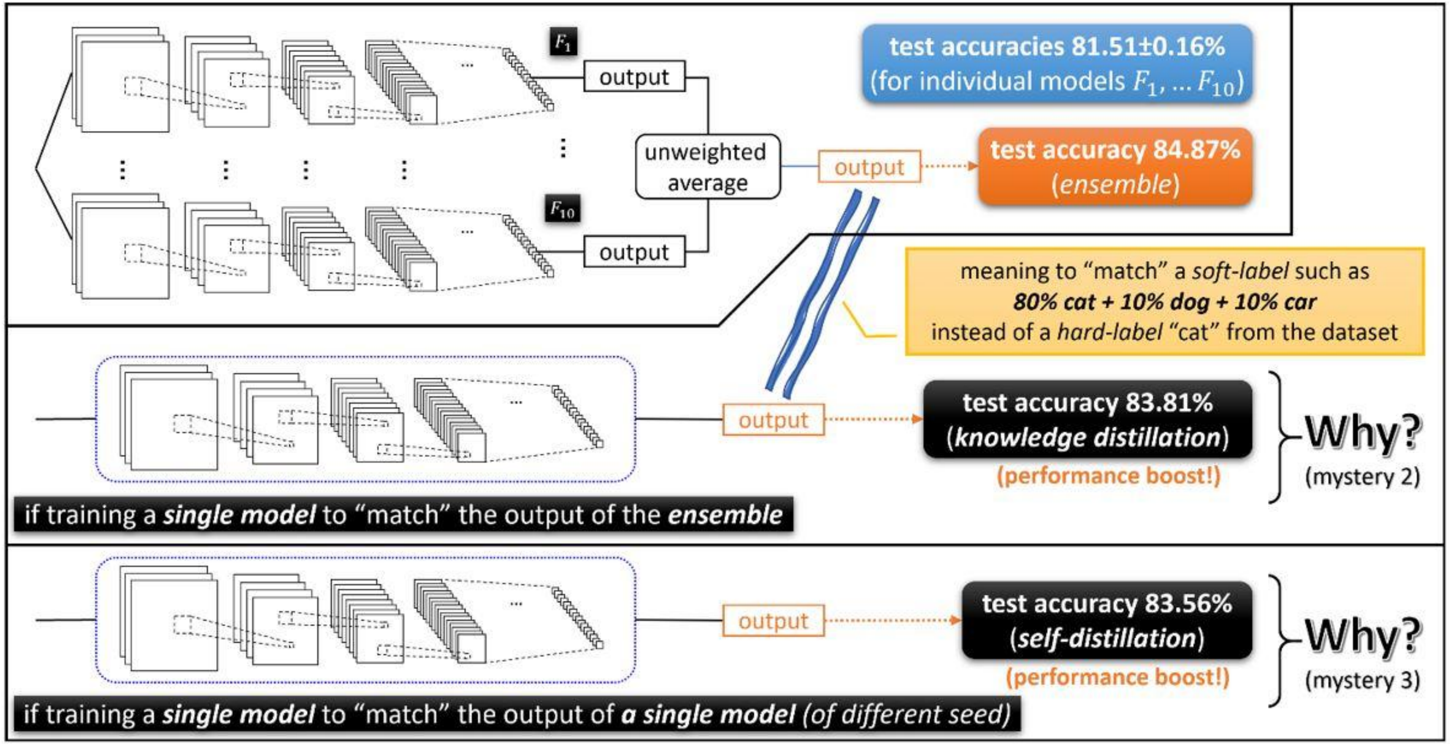
We can also distill from model of the same architecture

Model	Test accuracy
$F_1, F_2, \dots, F_{10}$	81.51 +/- 0.16%
$(F_1 + F_2 + \dots + F_{10}) / 10$ (ensemble)	84.87%
S (distilled from ensemble)	83.81%
S_1 (distilled from e.g. F_1)	83.56%

Mystery 3: Self-distillation

We can also distill from model of the same architecture

Model	Test accuracy
$F_1, F_2, \ldots, F_{10}$	81.51 +/- 0.16%
$(F_1 + F_2 + \ldots + F_{10}) / 10$ (ensemble)	84.87%
S (distilled from ensemble)	83.81%
S_1 (distilled from e.g. F_1)	83.56%

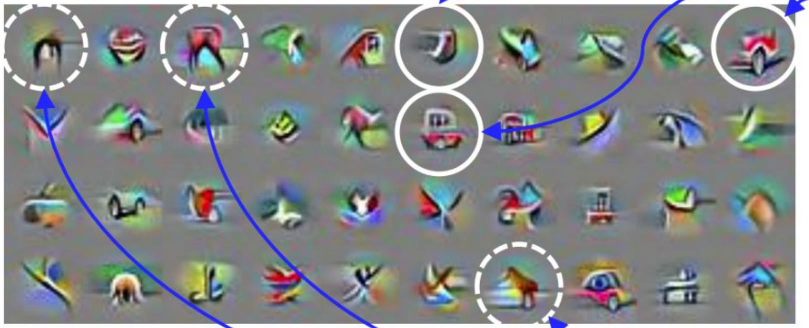


F_1
output
F_{10}
output
unweighted average
output
test accuracies 81.51±0.16% (for individual models $F_1, \ldots F_{10}$)
test accuracy 84.87% (*ensemble*)
meaning to "match" a *soft-label* such as
80% cat + 10% dog + 10% car
instead of a *hard-label* "cat" from the dataset
output
test accuracy 83.81% (***knowledge distillation***)
(performance boost!)
Why?
(mystery 2)
if training a ***single model*** to "match" the output of the ***ensemble***
output
test accuracy 83.56% (***self-distillation***)
(performance boost!)
Why?
(mystery 3)
if training a ***single model*** to "match" the output of ***a single model*** *(of different seed)*

Multi-view data

Claim: ensemble works when data has a multi-view structure
(data that can be classified using multiple different views)

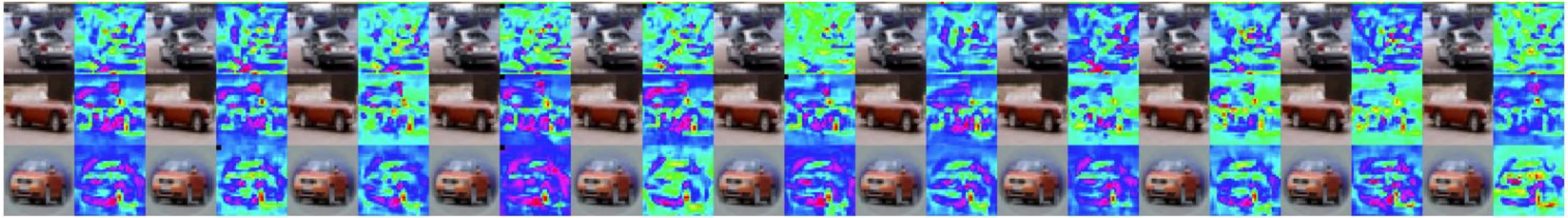
Example from CIFAR-10



ResNet-34 learns three features (views) of a car:
(1) front wheel (2) front window (3) side window

ResNet-34 learns three features (views) of a horse:
(1) tail (2) legs (3) head

Heatmaps of $F_1, \ldots, F_{10}$ and their ensemble



Example

Binary classification: dog vs cat

Four “features”: v1, v2, v3, v4 (tail, ear etc.)

Dog features: v1, v2

Cat features: v3, v4

When the label is “dog”:

- both v1 and v2 appear in 80%
- only v1 appears in 10%
- only v2 appears in 10%

of the data

Example

Binary classification: dog vs cat

Four “features”: v1, v2, v3, v4 (tail, ear etc.)

Dog features: v1, v2

Cat features: v3, v4

When the label is “dog”:

- both v1 and v2 appear in 80%
- only v1 appears in 10%
- only v2 appears in 10%

of the data

Then 80% of the data is multi-view data.

Example

Binary classification: dog vs cat

Four “features”: v1, v2, v3, v4 (tail, ear etc.)

Dog features: v1, v2

Cat features: v3, v4

When the label is “dog”:

- both v1 and v2 appear in 80%
- only v1 appears in 10%
- only v2 appears in 10%

of the data

Then 80% of the data is multi-view data.
The remaining 20% is single-view data.

Ensemble explanation

Intuition: each model learns subset of the features sufficient to classify the input and ensemble collects all of them

Ensemble explanation

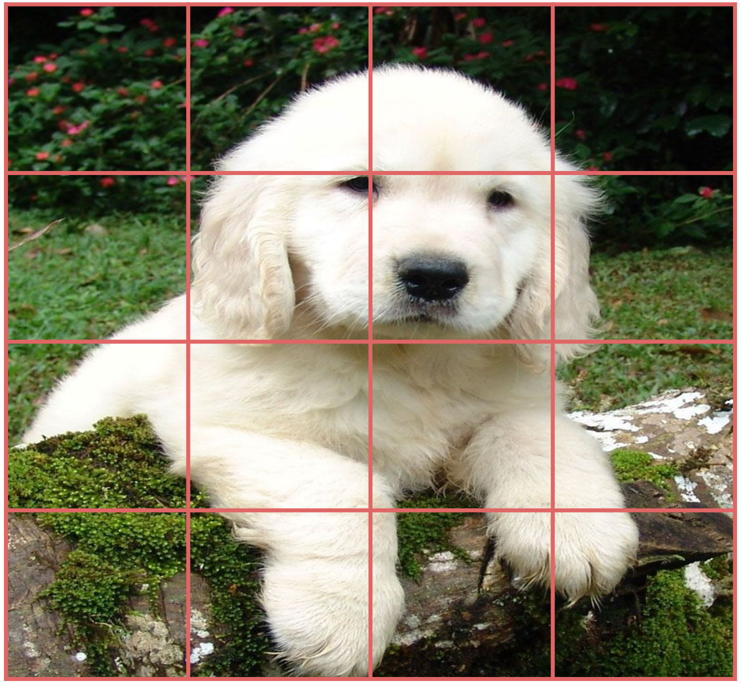
Intuition: each model learns subset of the features sufficient to classify the input and ensemble collects all of them

Training of a single model goes as follows:

1. Quickly learn a *subset* of these features depending on the randomness
2. Memorize the small number of remaining data that cannot be classified correctly using these features

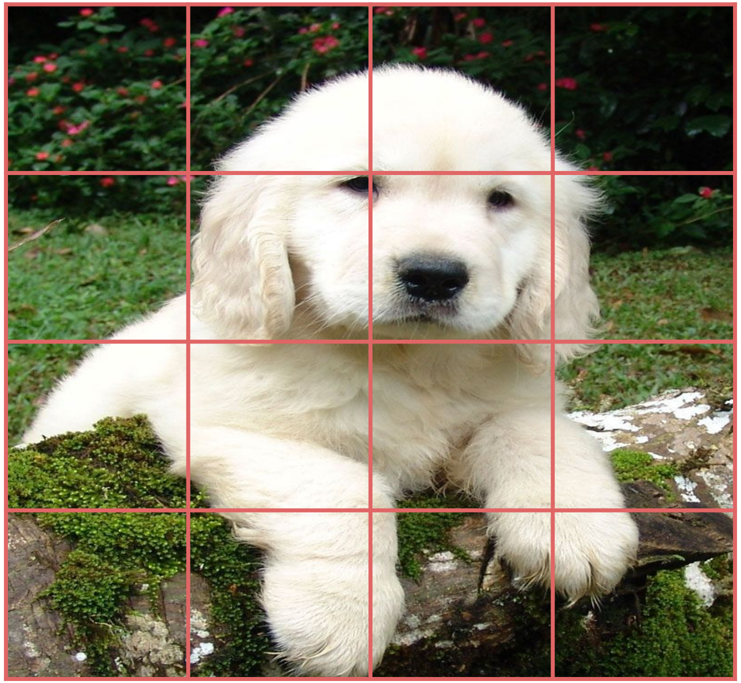
(It’s be proved under simplifying assumptions)

Multi-view data distribution (sketch)



Data: (X, y) where X = (x_1, x_2, ..., x_P) (here P = 16)

Multi-view data distribution (sketch)

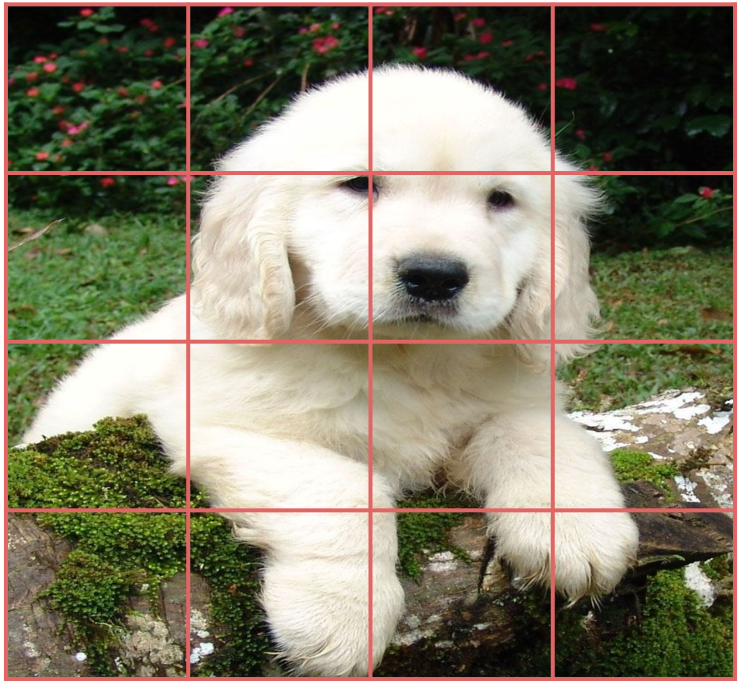


Data: (X, y) where $X = (x_1, x_2, \ldots, x_P)$ (here P = 16)

E.g. 4 features: ear, paw, wood, grass

We sample C_p disjoint patches per feature

Multi-view data distribution (sketch)



Data: (X, y) where X = (x_1, x_2, ..., x_P) (here P = 16)

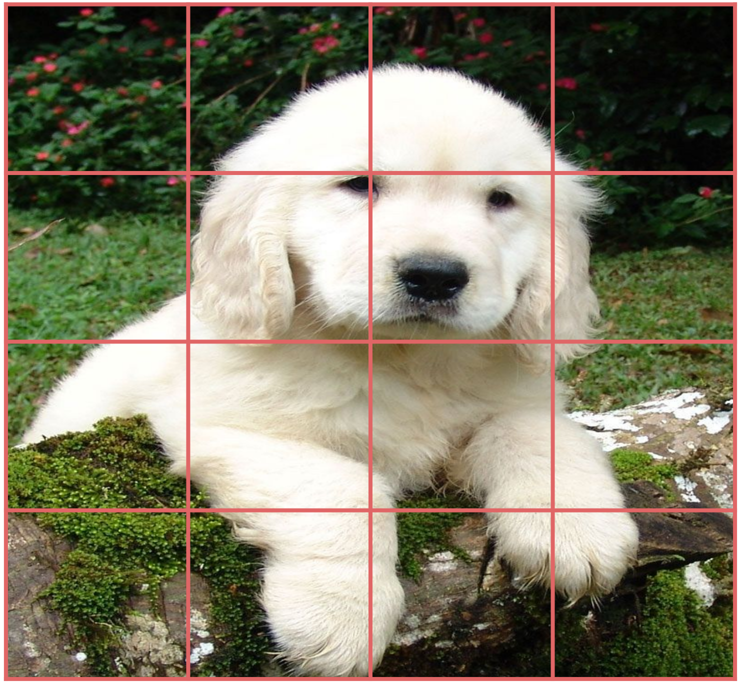
E.g. 4 features: ear, paw, wood, grass

We sample C_p disjoint patches per feature

For feature *v* and (sampled) patch *p*:

$$x_p = z_p v + \sum_{v' \in \mathcal{V}} \alpha_{p,v'} v' + \xi_p$$

Multi-view data distribution (sketch)



Data: (X, y) where X = (x_1, x_2, ..., x_P) (here P = 16)

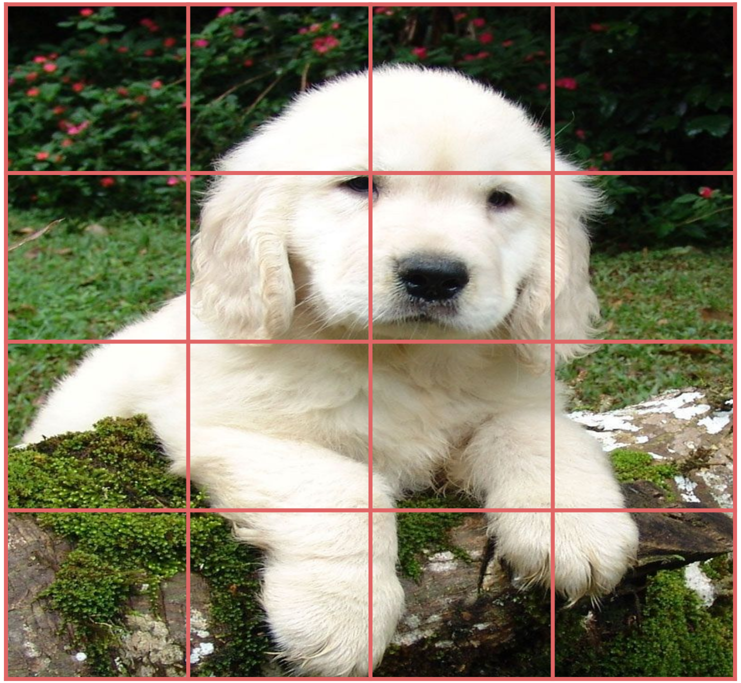
E.g. 4 features: ear, paw, wood, grass

We sample C_p disjoint patches per feature

For feature *v* and (sampled) patch *p*:

$$x_p = \underbrace{z_p}_{\text{feature coefficient}} v + \underbrace{\sum_{v' \in \mathcal{V}} \alpha_{p,v'} v'}_{\text{feature noise}} + \underbrace{\xi_p}_{\text{Gaussian noise}}$$

Multi-view data distribution (sketch)



Data: (X, y) where X = (x_1, x_2, ..., x_P) (here P = 16)

E.g. 4 features: ear, paw, wood, grass

We sample C_p disjoint patches per feature

For feature *v* and (sampled) patch *p*:

$$x_p = \underbrace{z_p}_{\text{feature coefficient}} v + \underbrace{\sum_{v' \in \mathcal{V}} \alpha_{p,v'} v'}_{\text{feature noise}} + \underbrace{\xi_p}_{\text{Gaussian noise}}$$

multi-view data: Σz_p is large when v == ear/paw

single-view data: Σz_p is large for only one of ear/paw

Proofs overview

The proofs in the paper are +/- 40 pages long

Proofs overview

The proofs in the paper are +/- 40 pages long

TLDR:

Assumptions:

- multi-view data distribution
- two-layer CNN with smoothed ReLU
- cross entropy loss function
- random gaussian initialization
- two fixed features per class
- ...

Proofs overview

Under these assumptions:

- Single model has bad test accuracy ($0.49c < E < 0.51c$)
- Ensemble provably improves test accuracy ($E < 0.01c$)
- Ensemble can be efficiently distilled into a single model ($E < 0.01c$)
- Self-distillation also improves test accuracy ($E < 0.26c$)

where **E** denotes the classification error and **c** is a fixed constant

Experiment: ensemble over KD

	CIFAR10 test accuracy			
	single model (over 10)	ensemble (over 10)	10 runs of knowledge distill	ensemble over knowledge distill
ResNet-28-2	95.22±*0.14%*	96.33%	95.89±0.07%	96.21%
ResNet-34	93.65±*0.19%*	94.97%	94.37±0.13%	94.88%
ResNet-34-2	95.45±*0.14%*	96.55%	96.00±0.12%	96.42%
ResNet-16-10	96.08±*0.16%*	96.80%	96.73±0.07%	96.76%
ResNet-22-10	96.44±*0.09%*	97.12%	97.01±0.09%	97.09%
ResNet-28-10	96.70±*0.21%*	97.20%	97.06±0.08%	97.24%

Experiment: ensemble over KD

	CIFAR10 test accuracy			
	single model (over 10)	ensemble (over 10)	10 runs of knowledge distill	ensemble over knowledge distill
ResNet-28-2	95.22±*0.14%*	96.33%	95.89±0.07%	96.21%
ResNet-34	93.65±*0.19%*	94.97%	94.37±0.13%	94.88%
ResNet-34-2	95.45±*0.14%*	96.55%	96.00±0.12%	96.42%
ResNet-16-10	96.08±*0.16%*	96.80%	96.73±0.07%	96.76%
ResNet-22-10	96.44±*0.09%*	97.12%	97.01±0.09%	97.09%
ResNet-28-10	96.70±*0.21%*	97.20%	97.06±0.08%	97.24%

KD models have learned most of the of features from ensemble so they have less variety

Experiment: dropping channels

CIFAR100	# input channels		original	split to 2	split to 4	split to 8
ResNet-28 (a)	16	single model test accuracy	70.44±0.29%	68.77±0.25%	66.70±0.66%	-
ResNet-28 (b)	32		70.49±0.29%	67.62±0.89%	63.28±0.50%	-
ResNet-28-2 (a)	32		76.09±0.23%	74.50±0.68%	72.47±1.78%	70.84±1.32%
ResNet-28-2 (b)	64		76.12±0.23%	74.88±0.22%	72.81±0.29%	69.21±0.49%
ResNet-28-4 (a)	64		79.10±0.18%	78.57±0.29%	77.94±0.43%	76.88±0.35%
ResNet-28-4 (b)	128		78.53±0.16%	77.72±0.20%	76.62±0.29%	74.93±0.40%
ResNet-28-10 (a)	160		81.23±0.23%	81.03±0.17%	80.53±0.09%	80.12±0.26%
ResNet-28-10 (b)	320		80.76±0.27%	80.41±0.24%	80.09±0.16%	79.02±0.22%
ResNet-28 (a)	16	ensemble model test accuracy	75.52%	74.07%	73.63%	-
ResNet-28 (b)	32		74.47%	73.58%	72.17%	-
ResNet-28-2 (a)	32		80.33%	79.73%	79.58%	78.75%
ResNet-28-2 (b)	64		79.63%	80.18%	79.17%	78.20%
ResNet-28-4 (a)	64		82.64%	82.81%	82.56%	82.24%
ResNet-28-4 (b)	128		81.84%	82.06%	81.89%	81.74%
ResNet-28-10 (a)	160		84.05%	84.08%	83.65%	83.51%
ResNet-28-10 (b)	320		83.10%	83.40%	83.81%	83.53%

- Take some intermediate layer of ResNet
- Remove all but 1/n channels
- Train a new network starting from this layer

Experiment: dropping channels

CIFAR100	# input channels		original	split to 2	split to 4	split to 8
ResNet-28 (a)	16	single model test accuracy	70.44±0.29%	68.77±0.25%	66.70±0.66%	-
ResNet-28 (b)	32		70.49±0.29%	67.62±0.89%	63.28±0.50%	-
ResNet-28-2 (a)	32		76.09±0.23%	74.50±0.68%	72.47±1.78%	70.84±1.32%
ResNet-28-2 (b)	64		76.12±0.23%	74.88±0.22%	72.81±0.29%	69.21±0.49%
ResNet-28-4 (a)	64		79.10±0.18%	78.57±0.29%	77.94±0.43%	76.88±0.35%
ResNet-28-4 (b)	128		78.53±0.16%	77.72±0.20%	76.62±0.29%	74.93±0.40%
ResNet-28-10 (a)	160		81.23±0.23%	81.03±0.17%	80.53±0.09%	80.12±0.26%
ResNet-28-10 (b)	320		80.76±0.27%	80.41±0.24%	80.09±0.16%	79.02±0.22%
ResNet-28 (a)	16	ensemble model test accuracy	75.52%	74.07%	73.63%	-
ResNet-28 (b)	32		74.47%	73.58%	72.17%	-
ResNet-28-2 (a)	32		80.33%	79.73%	79.58%	78.75%
ResNet-28-2 (b)	64		79.63%	80.18%	79.17%	78.20%
ResNet-28-4 (a)	64		82.64%	82.81%	82.56%	82.24%
ResNet-28-4 (b)	128		81.84%	82.06%	81.89%	81.74%
ResNet-28-10 (a)	160		84.05%	84.08%	83.65%	83.51%
ResNet-28-10 (b)	320		83.10%	83.40%	83.81%	83.53%

- Take some intermediate layer of ResNet
- Remove all but 1/n channels
- Train a new network starting from this layer

1. Different channels are learning different features that can be used to classify the input

Experiment: dropping channels

CIFAR100	# input channels		original	split to 2	split to 4	split to 8
ResNet-28 (a)	16	single model test accuracy	70.44±0.29%	68.77±0.25%	66.70±0.66%	-
ResNet-28 (b)	32		70.49±0.29%	67.62±0.89%	63.28±0.50%	-
ResNet-28-2 (a)	32		76.09±0.23%	74.50±0.68%	72.47±1.78%	70.84±1.32%
ResNet-28-2 (b)	64		76.12±0.23%	74.88±0.22%	72.81±0.29%	69.21±0.49%
ResNet-28-4 (a)	64		79.10±0.18%	78.57±0.29%	77.94±0.43%	76.88±0.35%
ResNet-28-4 (b)	128		78.53±0.16%	77.72±0.20%	76.62±0.29%	74.93±0.40%
ResNet-28-10 (a)	160		81.23±0.23%	81.03±0.17%	80.53±0.09%	80.12±0.26%
ResNet-28-10 (b)	320		80.76±0.27%	80.41±0.24%	80.09±0.16%	79.02±0.22%
ResNet-28 (a)	16	ensemble model test accuracy	75.52%	74.07%	73.63%	-
ResNet-28 (b)	32		74.47%	73.58%	72.17%	-
ResNet-28-2 (a)	32		80.33%	79.73%	79.58%	78.75%
ResNet-28-2 (b)	64		79.63%	80.18%	79.17%	78.20%
ResNet-28-4 (a)	64		82.64%	82.81%	82.56%	82.24%
ResNet-28-4 (b)	128		81.84%	82.06%	81.89%	81.74%
ResNet-28-10 (a)	160		84.05%	84.08%	83.65%	83.51%
ResNet-28-10 (b)	320		83.10%	83.40%	83.81%	83.53%

- Take some intermediate layer of ResNet
- Remove all but 1/n channels
- Train a new network starting from this layer

1. Different channels are learning different features that can be used to classify the input
2. Ensemble can collect all of multiple views even when some models have missing views

Knowledge distillation & self-distillation

Using multi-view approach we can explain remaining “mysteries”:

Knowledge distillation = forcing individual model to learn every feature

Knowledge distillation & self-distillation

Using multi-view approach we can explain remaining “mysteries”:

Knowledge distillation = forcing individual model to learn every feature

Self-distillation = combining knowledge distillation and ensemble

Bibliography

- [2012.09816] Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning (arxiv.org)
- Three mysteries in deep learning: Ensemble, knowledge distillation, and self-distillation - Microsoft Research