Ensemble and Distillation

Lantian Li

2021.2.22

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

December 18, 2020

(version 1.5)

Abstract

We formally study how *ensemble* of deep learning models can improve test accuracy, and how the superior performance of ensemble can be distilled into a single model using *knowledge distillation*. We consider the challenging case where the ensemble is simply an average of the outputs of a few independently trained neural networks with the *same* architecture, trained using the *same* algorithm on the *same* data set, and they only differ by the random seeds used in the initialization.

We empirically show that ensemble/knowledge distillation in deep learning works very differently from traditional learning theory, especially differently from ensemble of random feature mappings or the neural-tangent-kernel feature mappings, and is potentially out of the scope of existing theorems. Thus, to properly understand ensemble and knowledge distillation in deep learning, we develop a theory showing that when data has a structure we refer to as "multi-view", then ensemble of independently trained neural networks can provably improve test accuracy, and such superior test accuracy can also be provably distilled into a single model by training a single model to match the output of the ensemble instead of the true label. Our result sheds light on how ensemble works in deep learning in a way that is completely different from traditional theorems, and how the "dark knowledge" is hidden in the outputs of the ensemble— that can be used in knowledge distillation— comparing to the true data labels. In the end, we prove that self-distillation can also be viewed as implicitly combining ensemble and knowledge distillation to improve test accuracy.

Ensemble

- Model average
  - Get a set of classifiers $f_1(x)$, $f_2(x)$, $f_3(x)$, ...
  - Average all the f_i to F

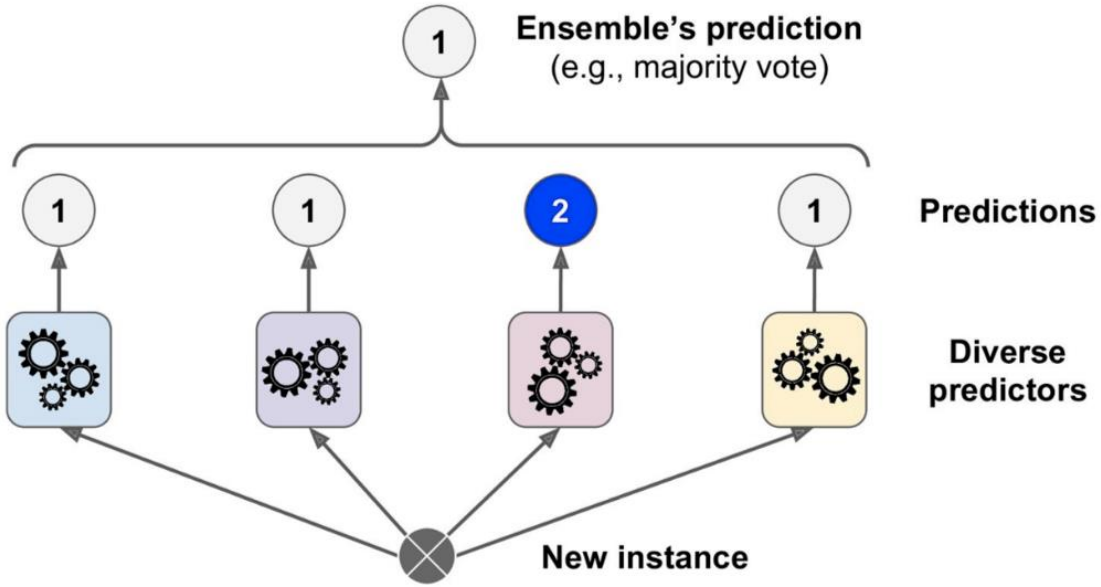
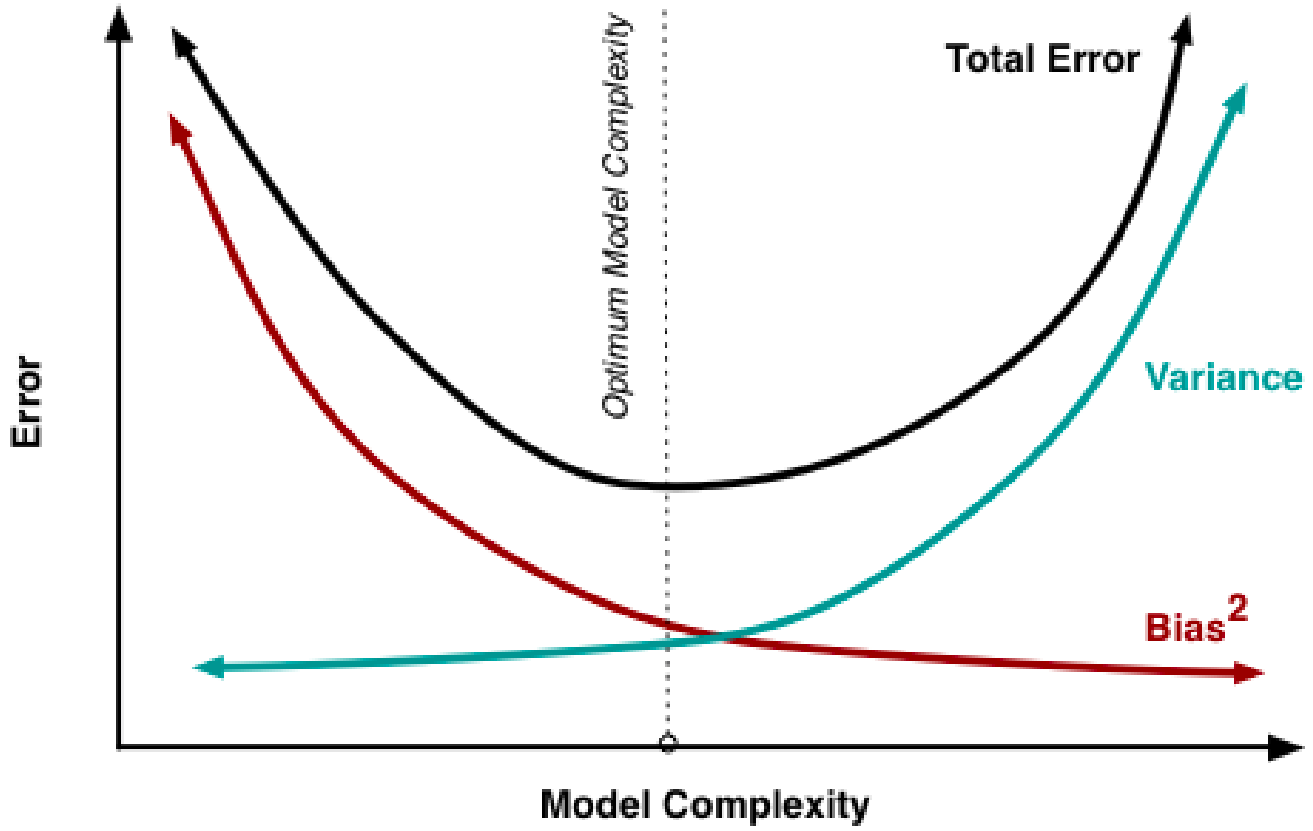


Figure 7-2. Hard voting classifier predictions

No Free Lunch

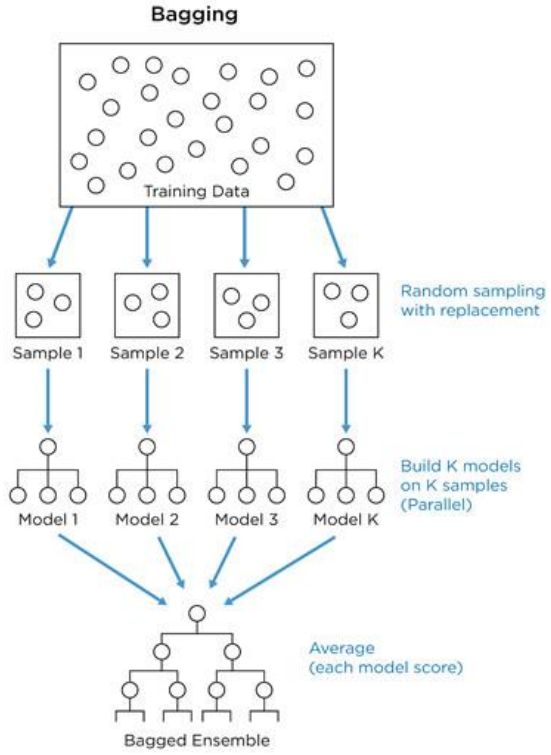


Generalizability

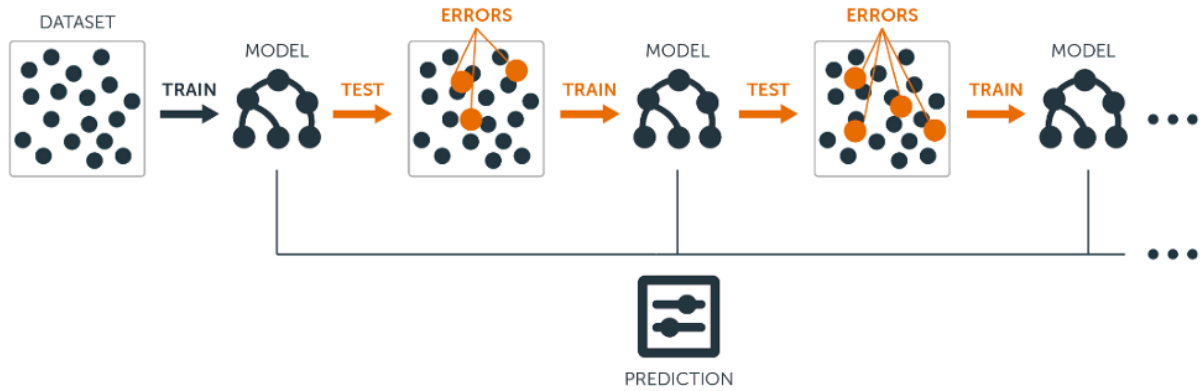
Representability

Two major approaches

- Bagging
  - Decision Tree -> Random Forest
  - Reducing Variance

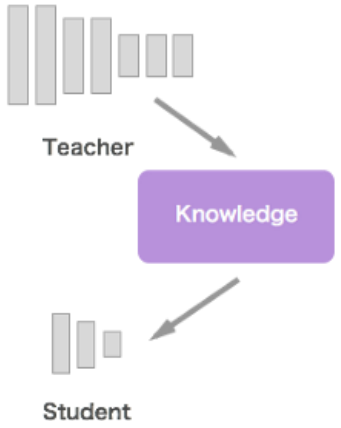


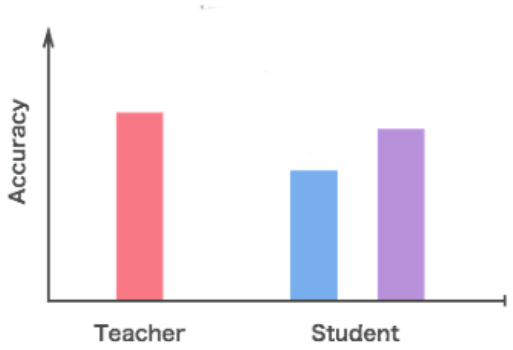
- Boosting
  - AdaBoost
  - Sequential learning process
  - Combining weak learners into a stronger one.

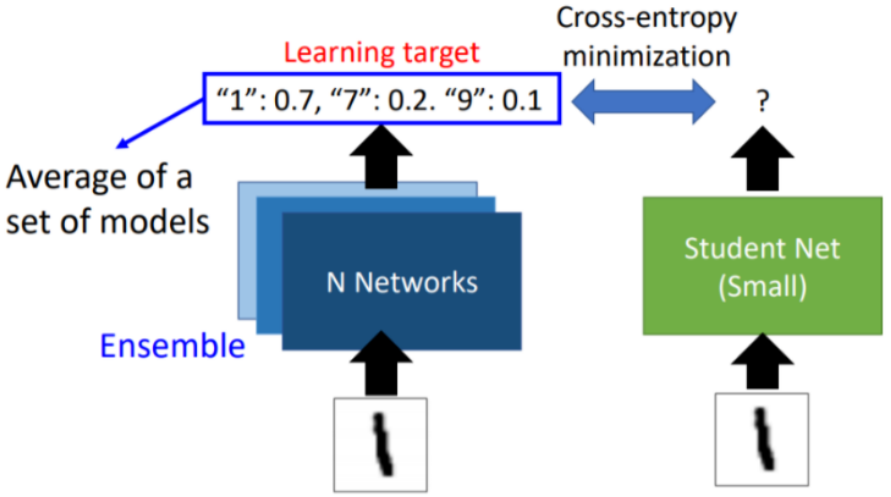


Distillation

- *Distilling the Knowledge in a Neural Network*. Geoffrey Hinton, etc. 2015.
- Logits, Feature, Attention, Relation ……







Hidden information in relationship between categories

Ensemble in deep learning

- Guarantees
  - A few **independently** NNs are trained.
  - All the NNs have the **same** architecture and are trained using the **same** training algorithm over the **same** training data set.
  - The only difference is the **randomness** used to initialize these NNs and/or the randomness during training.
  - The ensemble model is obtained by merely taking an **unweighted average** the output of these independently trained NNs.

Empirical results

neural networks	single model (over 10) ⑦	ensemble (over 10)	train $\sum_\ell f_\ell$ (over 10)	knowledge distillation	self-distill	single model (over 10) ⑦	ensemble (over 10)	train $\sum_\ell f_\ell$ (over 10)	knowledge distillation	self-distill
ResNet-28-2	95.22±*0.14%*	96.33%	95.02%	96.16%	95.78%	76.38±*0.23%*	81.13%	73.18%	79.03%	78.12%
ResNet-34	93.65±*0.19%*	94.97%	93.12%	94.59%	94.21%	71.66±*0.43%*	76.85%	68.88%	73.74%	73.14%
ResNet-34-2	95.45±*0.14%*	96.55%	95.00%	96.08%	95.86%	77.01±*0.35%*	81.48%	72.99%	79.23%	79.07%
ResNet-16-10	96.08±*0.16%*	96.80%	95.88% (over 6)◊	96.81%	96.62%	80.03±*0.17%*	83.18%	80.53% (over 6)◊	82.67%	82.25%
ResNet-22-10	96.44±*0.09%*	97.12%	96.41% (over 5)◊	97.09%	97.05%	81.17±*0.23%*	84.33%	81.59% (over 5)◊	83.71%	83.26%
ResNet-28-10	96.70±*0.21%*	97.20%	96.46% (over 4)◊	97.22%	97.13%	81.51±*0.16%*	84.69%	81.83% (over 4)◊	83.81%	83.56%

Message ④: for neural nets, ensemble helps on improving test accuracies, **and** this accuracy gain *cannot* be matched by training the sum of the individuals directly. In other words, the benefit of using ensemble comes from somewhere other than enlarging the model.

Message ⑤: for neural nets, the superior test performance of ensemble *can be distilled* into single model by a large extent.

Message ⑥: for neural nets, self-distillation *clearly improves* the test performance of single models.

Message ⑦: for neural nets, the superior performance of ensemble *does not* come from the variance of test accuracies in single models.

Two theoretical questions

Our theoretical questions:

How does ensemble improve the test-time performance in deep learning when we simply average over a few independently trained neural networks? – Especially when all the neural networks have the same architecture, are trained over the same data set using the same standard training algorithm (i.e. stochastic gradient descent with the same learning rate and sample regularization), and even when all single models already have 100% *training accuracy*? How can such superior test-time performance of ensemble be later "distilled" into a single neural network of the same architecture, simply by training the single model to match the output of the ensemble over the same training data set?

Ensemble in Deep Learning v.s. Ensemble of Feature Mappings

- Model averaging (i.e., ensemble) in deep learning works very differently from model averaging in random feature mappings.

- Random feature mappings

41, 45, 55, 59, 86, 92]. In particular, the theory shows that when $f : \mathbb{R}^{D+d} \to \mathbb{R}$ is a neural network with inputs $x \in \mathbb{R}^d$ and weights $W \in \mathbb{R}^D$, in some cases, $f(W, x)$ can be approximated by:

$$f(W, x) \approx f(W_0, x) + \langle W - W_0, \nabla_W f(W_0, x) \rangle$$

Where W_0 is the random initialization of the neural network, and $\Phi_{W_0}(x) := \nabla_W f(W_0, x)$ is the neural tangent kernel (NTK) feature mapping. This is known as the neural tangent kernel (NTK) approach. If this approximation holds, then training a neural network can be approximated by learning a linear function over $\Phi_{W_0}(x)$, which is very theory-friendly.

	CIFAR10 test accuracy					CIFAR100 test accuracy				
finite-width neural kernel models	single model (best of 10)	ensemble (over 10)	train $\sum_\ell f_\ell$ (over 10)	knowledge distillation	self-distill	single model (best of 10)	ensemble (over 10)	train $\sum_\ell f_\ell$ (over 10)	knowledge distillation	self-distill
SimpleCNN-10-3-NTK	64.36%	67.38%	69.37%	64.63%	65.24%	out of memory				
ResNet10-2-NTK	69.15%	73.29%	74.71%	68.82%	66.09%					
ResNet16-2-NTK	68.32%	73.79%	74.62% (over 7)◊	66.12%	70.61%					
ResNet16-5-NTK	74.21%	78.46%	out of memory	70.23%	75.66%					
ResNet10-10-NTK	76.66%	80.39%	out of memory	77.25%	74.46%					
SimpleCNN10-6-NTK'	59.92%	63.43%	65.69%	59.12%	57.81%	18.99%	26.54%	28.28%	18.27%	18.40%
ResNet10-4-NTK'	66.68%	70.54%	72.86%	66.01%	62.91%	31.90%	38.32%	41.47%	31.38%	27.64%
SimpleCNN-10-6-GP	30.48%	35.33%	40.08%	29.43%	29.10%	9.82%	11.82%	12.22%	8.95%	9.33%
ResNet-10-4-GP	42.17%	48.60%	53.17%	39.45%	41.63%	18.89%	22.92%	25.88%	16.91%	16.59%

◊due to memory restriction, trained $\sum_\ell f_\ell$ over fewer than 10 models.

Message ①: for neural kernel methods, ensemble helps on improving test accuracies, but ensemble is *not better than* training the sum of the individuals directly. In other words, the benefit of using ensemble here merely comes from the richer set of prescribed features.

Message ②: for neural kernel methods, the superior test performance of ensemble *cannot be distilled* into a single model.

Message ③: for neural kernel methods, self-distillation is generally *no better than* a single model's test performance.

neural networks	single model (over 10)⑦	ensemble (over 10)	train $\sum_\ell f_\ell$ (over 10)	knowledge distillation	self-distill	single model (over 10)⑦	ensemble (over 10)	train $\sum_\ell f_\ell$ (over 10)	knowledge distillation	self-distill
ResNet-28-2	95.22±*0.14*%	96.33%	95.02%	96.16%	95.78%	76.38±*0.23*%	81.13%	73.18%	79.03%	78.12%
ResNet-34	93.65±*0.19*%	94.97%	93.12%	94.59%	94.21%	71.66±*0.43*%	76.85%	68.88%	73.74%	73.14%
ResNet-34-2	95.45±*0.14*%	96.55%	95.00%	96.08%	95.86%	77.01±*0.35*%	81.48%	72.99%	79.23%	79.07%
ResNet-16-10	96.08±*0.16*%	96.80%	95.88% (over 6)◊	96.81%	96.62%	80.03±*0.17*%	83.18%	80.53% (over 6)◊	82.67%	82.25%
ResNet-22-10	96.44±*0.09*%	97.12%	96.41% (over 5)◊	97.09%	97.05%	81.17±*0.23*%	84.33%	81.59% (over 5)◊	83.71%	83.26%
ResNet-28-10	96.70±*0.21*%	97.20%	96.46% (over 4)◊	97.22%	97.13%	81.51±*0.16*%	84.69%	81.83% (over 4)◊	83.81%	83.56%

Message ④: for neural nets, ensemble helps on improving test accuracies, **and** this accuracy gain *cannot* be matched by training the sum of the individuals directly. In other words, the benefit of using ensemble comes from somewhere other than enlarging the model.

Message ⑤: for neural nets, the superior test performance of ensemble *can be distilled* into single model by a large extent.

Message ⑥: for neural nets, self-distillation *clearly improves* the test performance of single models.

Message ⑦: for neural nets, the superior performance of ensemble *does not* come from the variance of test accuracies in single models.

Evidence 1

- In actual deep learning, ensemble **does not enlarge feature space**: an individual model f(x) is still capable of learning the features of the ensemble model.
- What is the dark knowledge hidden in the output of ensemble comparing to the original label?

- To understand the benefit of ensemble and knowledge distillation in deep learning, it is perhaps inevitable to study deep learning as a **feature learning process**, instead of **feature selection process**.

Ensemble in Deep Learning: a Feature Learning Process

		no label noise				with 10% label noise			
		uniform sampling		rejection sampling		uniform sampling		rejection sampling	
		gaussian input	mixture of gaussian	gaussian input	mixture of gaussian	gaussian input	mixture of gaussian	gaussian input	mixture of gaussian
data without margin	linear	80.3% (79.6%)	80.7% (80.1%)	78.9% (78.6%)	80.7% (80.7%)	74.3% (74.1%)	73.6% (74.0%)	72.9% (72.2%)	74.2% (73.7%)
	fc2	67.7% (65.1%)	67.7% (64.9%)	66.3% (64.5%)	67.6% (66.9%)	64.3% (63.2%)	70.1% (66.7%)	64.6% (63.5%)	66.2% (63.3%)
	fc3	68.9% (69.0%)	64.0% (64.4%)	76.8% (76.6%)	73.2% (73.1%)	66.5% (66.4%)	63.0% (62.4%)	72.5% (72.0%)	78.1% (78.6%)
	res3	69.1% (68.0%)	70.7% (71.2%)	69.3% (69.0%)	69.9% (69.4%)	68.7% (65.9%)	66.9% (63.8%)	68.1% (68.1%)	68.8% (69.5%)
	conv2	65.4% (65.7%)	67.0% (66.8%)	68.3% (68.2%)	68.3% (68.2%)	67.1% (66.2%)	65.1% (65.5%)	65.8% (66.0%)	67.5% (67.9%)
	conv3	68.7% (68.5%)	70.7% (71.2%)	77.8% (77.1%)	80.3% (80.3%)	67.5% (68.2%)	67.7% (67.5%)	73.6% (73.4%)	71.8% (72.1%)
	resconv3	78.3% (78.3%)	79.6% (79.3%)	83.8% (82.6%)	82.1% (81.8%)	74.1% (73.9%)	73.4% (73.1%)	78.7% (78.5%)	79.2% (78.5%)
data with margin	linear	79.0% (78.0%)	79.0% (77.2%)	78.4% (77.3%)	80.0% (80.0%)	82.1% (81.7%)	80.7% (80.0%)	81.6% (80.2%)	84.1% (82.4%)
	fc2	80.6% (79.0%)	80.4% (78.4%)	78.4% (76.6%)	78.4% (77.0%)	77.4% (75.3%)	73.7% (73.9%)	74.7% (72.2%)	75.7% (71.4%)
	fc3	76.0% (75.8%)	80.4% (80.1%)	76.9% (77.0%)	73.3% (72.9%)	70.7% (70.8%)	73.9% (74.6%)	70.4% (70.5%)	67.5% (66.4%)
	res3	80.7% (80.8%)	84.7% (83.9%)	84.6% (84.0%)	84.4% (83.7%)	75.5% (74.0%)	76.9% (76.4%)	76.8% (74.8%)	76.4% (74.5%)
	conv2	70.6% (70.3%)	74.5% (73.6%)	67.8% (67.8%)	69.6% (69.4%)	68.8% (67.5%)	73.3% (71.7%)	67.0% (66.6%)	67.6% (67.2%)
	conv3	76.2% (76.2%)	75.3% (76.1%)	79.6% (79.1%)	84.3% (83.6%)	72.2% (72.1%)	81.2% (81.3%)	73.0% (72.3%)	74.4% (75.1%)
	resconv3	92.1% (91.7%)	92.1% (92.3%)	93.9% (93.8%)	95.5% (95.1%)	85.2% (85.1%)	83.5% (84.4%)	87.3% (86.8%)	85.6% (85.9%)

Figure 3: When data is Gaussian-like, and when the target label is generated by some fully-connected(fc) / residual(res) / convolutional(conv) network, *ensemble does not* seem to improve test accuracy. " xx % (yy %)" means $xx\%$ accuracy for single model and $yy\%$ for ensemble. More experiments in Appendix A.4.

Evidence 2

- Ensemble in DL might not improve test accuracy when inputs are Gaussian-like.

- Data structure is very important.

Learning Multi-View Data

- Consider a binary classification problem
  - four features: v1, v2, v3, v4.
  - v1, v2 correspond to the first class label.
  - v3, v4 correspond to the second class label.

Multi-view data

- When the label is class 1, then:

$$\begin{cases} \text{both } v_1, v_2 \text{ appears with weight 1, one of } v_3, v_4 \text{ appears with weight 0.1} & \text{w.p. } 80\%; \\ \text{only } v_1 \text{ appears with weight 1, one of } v_3, v_4 \text{ appears with weight 0.1} & \text{w.p. } 10\%; \\ \text{only } v_2 \text{ appears with weight 1, one of } v_3, v_4 \text{ appears with weight 0.1} & \text{w.p. } 10\%. \end{cases}$$

- When the label is class 2, then

$$\begin{cases} \text{both } v_3, v_4 \text{ appears with weight 1, one of } v_1, v_2 \text{ appears with weight 0.1} & \text{w.p. } 80\%; \\ \text{only } v_3 \text{ appears with weight 1, one of } v_1, v_2 \text{ appears with weight 0.1} & \text{w.p. } 10\%; \\ \text{only } v_4 \text{ appears with weight 1, one of } v_1, v_2 \text{ appears with weight 0.1} & \text{w.p. } 10\%. \end{cases}$$

How individual neural networks learn. In this data set, if we train the neural network using the cross-entropy loss via gradient descent (GD) from random initialization, during the training process of the individual networks, we show that:

- The network will quickly pick up one of the feature $v \in \{v_1, v_2\}$ for the first label, and one of the features $v' \in \{v_3, v_4\}$ for the second label. So, 90% of the training examples, consisting of all multi-view data and half of the single-view data (those with feature v or v'), are classified correctly. These data begin to contribute negligible gradient after wards.
- The network will memorize (using for example the noise in the data) the remaining 10% of the training examples without learning any new features, due to insufficient amount of left-over samples after the first phase, thus achieving training accuracy 100% but test accuracy 90%.

How ensemble improves test accuracy. We show that depending on the randomness of the initialization, each individual network will pick up v_1 or v_2 each with probability 0.5. Hence, we can prove that as long as we ensemble $\widetilde{O}(1)$ many independently trained models, w.h.p. the ensemble model will be able to pick up both features $\{v_1, v_2\}$, and both features $\{v_3, v_4\}$. Thus, all the data will be classified correctly.

How knowledge distillation works. Since ensemble learns all the features v_1, v_2, v_3, v_4, on the multi-view data with label 1, the ensemble model will actually output $\propto (2, 0.1)$, where the 2 comes from features v_1, v_2 and 0.1 comes from one of v_3, v_4. On the other hand, an individual model that only learns one of v_3, v_4 will actually output $\propto (2, 0)$ when the feature v_3 or v_4 in the data does not match the one learned by the model. Hence, by training the individual model to match the output of the ensemble, the individual model is forced to learn both features v_3, v_4, even though it has already perfectly classified the training data. **This is the "dark knowledge" hidden in the ensemble model.**

Generality of our multi-view hypothesis

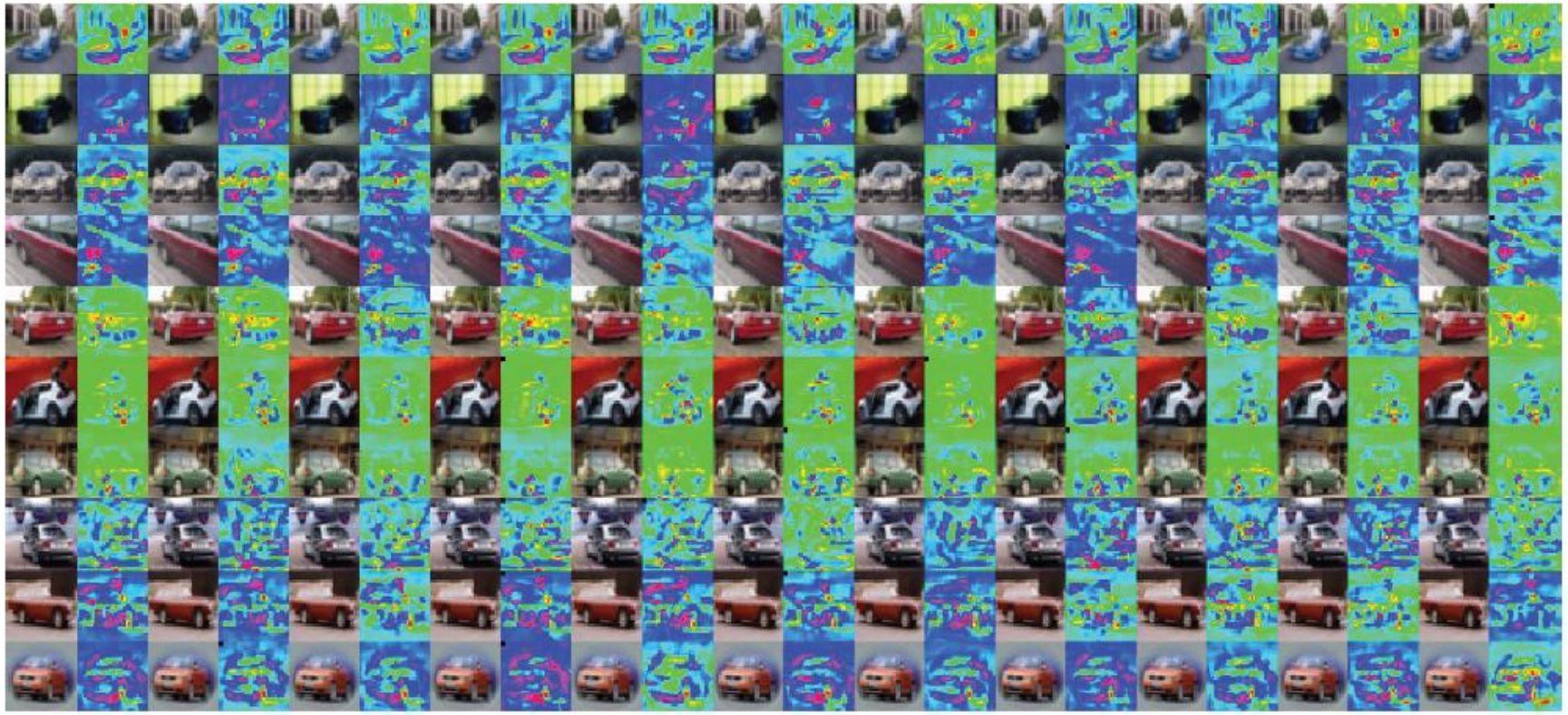


Figure 4: Ten independently trained ResNet-34 models (and their ensemble) detect car images through different reasonings, suggesting that the data has multi views, and independently trained neural networks do utilize this structure. The numerical experiments in Figure 6 also suggest the existence of multi views.

CIFAR100	# input channels		original	split to 2	split to 4	split to 8	avg over 2	avg over 4	avg over 8
ResNet-28 (a)	16		70.44±0.29%	68.77±0.25%	66.70±0.66%	-	69.00±0.43%	66.45±0.15%	-
ResNet-28 (b)	32		70.49±0.29%	67.62±0.89%	63.28±0.50%	-	67.99±0.15%	63.89±0.31%	-
ResNet-28-2 (a)	32	single	76.09±0.23%	74.50±0.68%	72.47±1.78%	70.84±1.32%	75.31±0.23%	73.69±0.34%	71.60±0.34%
ResNet-28-2 (b)	64	model	76.12±0.23%	74.88±0.22%	72.81±0.29%	69.21±0.49%	74.58±0.33%	72.71±0.29%	68.85±0.28%
ResNet-28-4 (a)	64	test	79.10±0.18%	78.57±0.29%	77.94±0.43%	76.88±0.35%	78.42±0.35%	78.14±0.16%	77.52±0.20%
ResNet-28-4 (b)	128	accuracy	78.53±0.16%	77.72±0.20%	76.62±0.29%	74.93±0.40%	77.95±0.22%	76.88±0.33%	75.17±0.25%
ResNet-28-10 (a)	160		81.23±0.23%	81.03±0.17%	80.53±0.09%	80.12±0.26%	80.58±0.28%	81.06±0.22%	80.63±0.22%
ResNet-28-10 (b)	320		80.76±0.27%	80.41±0.24%	80.09±0.16%	79.02±0.22%	80.54±0.23%	80.03±0.16%	79.38±0.27%
ResNet-28 (a)	16		75.52%	74.07%	73.63%	-	74.05%	70.98%	-
ResNet-28 (b)	32		74.47%	73.58%	72.17%	-	71.97%	68.03%	-
ResNet-28-2 (a)	32	ensemble	80.33%	79.73%	79.58%	78.75%	79.24%	78.19%	76.31%
ResNet-28-2 (b)	64	model	79.63%	80.18%	79.17%	78.20%	78.42%	76.81%	72.90%
ResNet-28-4 (a)	64	test	82.64%	82.81%	82.56%	82.24%	82.26%	82.12%	81.71%
ResNet-28-4 (b)	128	accuracy	81.84%	82.06%	81.89%	81.74%	81.28%	80.63%	79.14%
ResNet-28-10 (a)	160		84.05%	84.08%	83.65%	83.51%	83.79%	84.12%	83.69%
ResNet-28-10 (b)	320		83.10%	83.40%	83.81%	83.53%	83.21%	83.00%	82.19%

Figure 6: Justify the multi-view hypothesis in practice. We regard some intermediate layer of a pre-trained ResNet as "input" with multiple channels (this pre-trained network stays fixed and shared for all individual models). Then, we train a new model either starting from this input (i.e. the "*original*" column), or from a fraction of the input (i.e., "*split into 4*" means using only 1/4 of the input channels), or from an average of the input (i.e., "*average over 4*" means averaging every four channels). Details in Appendix A.3.

Observation 1. Even when we significantly collapse the input channels (through averaging or throwing away most of them), most of the single model test accuracies do not drop by much. Moreover, it's known [65] that in ResNet, most channels are indeed learning different features (views) of the input, also see Figure 5 for an illustration. This indicates that many data can be classified correctly using completely views.

Observation 2. Even when single model accuracy drops noticeably, ensemble accuracy does not change by much. We believe this is a strong evidence that there are multiple views in the data (even at intermediate layers), and **ensemble can collect all of them even when some models have missing views.**

On the theory side

with a special structure that we shall refer to as **multi-view**, with a training set $\mathcal{Z}$ consisting of N i.i.d. samples from some unknown distribution $\mathcal{D}$, for certain two-layer convolution network f with (smoothed-)ReLU activation:

- (Single model has bad test accuracy): there is a value $\mu > 0$ such that when a single model f is trained over $\mathcal{Z}$ using the cross-entropy loss, via gradient descent (GD) starting from random Gaussian initialization, the model can reach zero training error *efficiently*. However, w.h.p. the prediction (classification) error of f over $\mathcal{D}$ is between 0.49μ and 0.51μ.
- (Ensemble provably improves test accuracy): let $f_1, f_2, \cdots, f_L$ be L independently trained single models as above with $L = \widetilde{\Omega}(1)$, then w.h.p. $F = \sum_\ell f_\ell$ will have prediction error at most 0.01μ over $\mathcal{D}$.
- (Ensemble can be distilled into a single model): if we further train (using GD from random initialization) another network f_0 to match the output of F merely over the same training data set $\mathcal{Z}$, then f_0 can be trained *efficiently* and w.h.p. f_0 will have prediction error at most 0.01μ over $\mathcal{D}$ as well.
- (*Self-distillation* [36, 63] also improves test accuracy, see Figure 1): if we further train (using GD from random initialization) another network f' to match the output of *a single model* f_1 merely over the same training data set $\mathcal{Z}$, then f' can be trained *efficiently* and w.h.p. f' will have prediction error at most $\leq 0.26\mu$ over $\mathcal{D}$. The main idea is that self-distillation is performing "*implicit ensemble + knowledge distillation*", as we shall argue in Section 3.2.

Conclusions

- Ensemble in deep learning is a feature learning process.
- This feature learning depends on the structure of data.
- This structure refers to as ‘multi-view’.

Discussions

- For knowledge distillation, an individual model is forced to learn multi-view features (‘dark knowledge’ hidden in the ensemble model) to the logits/features by a soft label refinery.
- For data augmentation, it is another way to enforce the NNs to learn ‘multi-views’. However, it focuses on the raw input data by random cropping.

- Can we use ensemble and distillation to improve the performance on multi-genre or other difficult SRE tasks ?