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

ICLR 2023

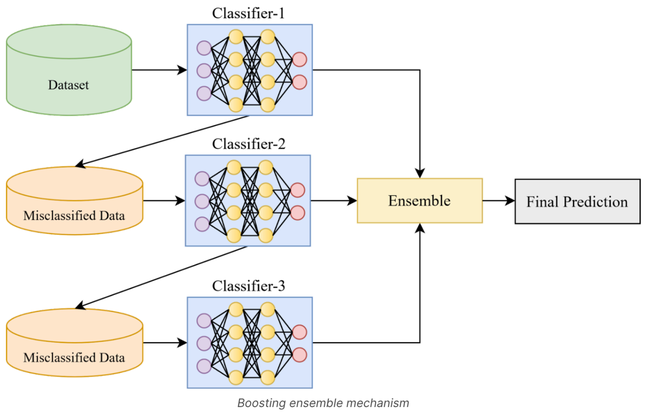
Zeyuan Allen-Zhu
Meta FAIR Labs
zeyuanallenzhu@meta.com

Yuanzhi Li
Mohamed bin Zayed University of AI
Yuanzhi.Li@mbzuai.ac.ae

Jiwoon Lee
jwlee9702@postech.ac.kr
EffL@POSTECH, Korea

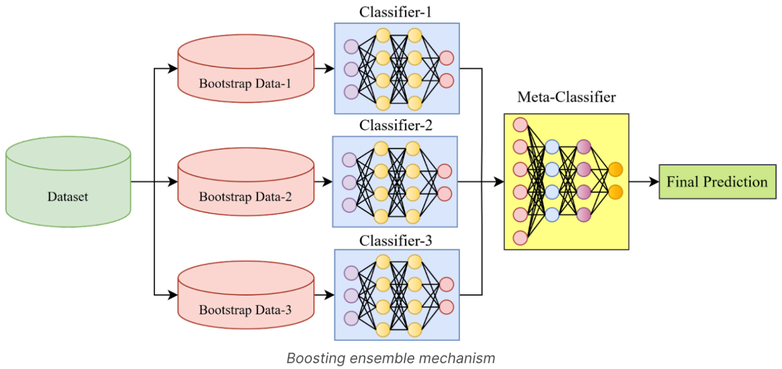
Background : Model Ensemble

- Model ensemble : running two or more models and synthesizing the results into a single score or spread.
  - Ensembled output outperforms single model performance.



Boosting ensemble mechanism

,

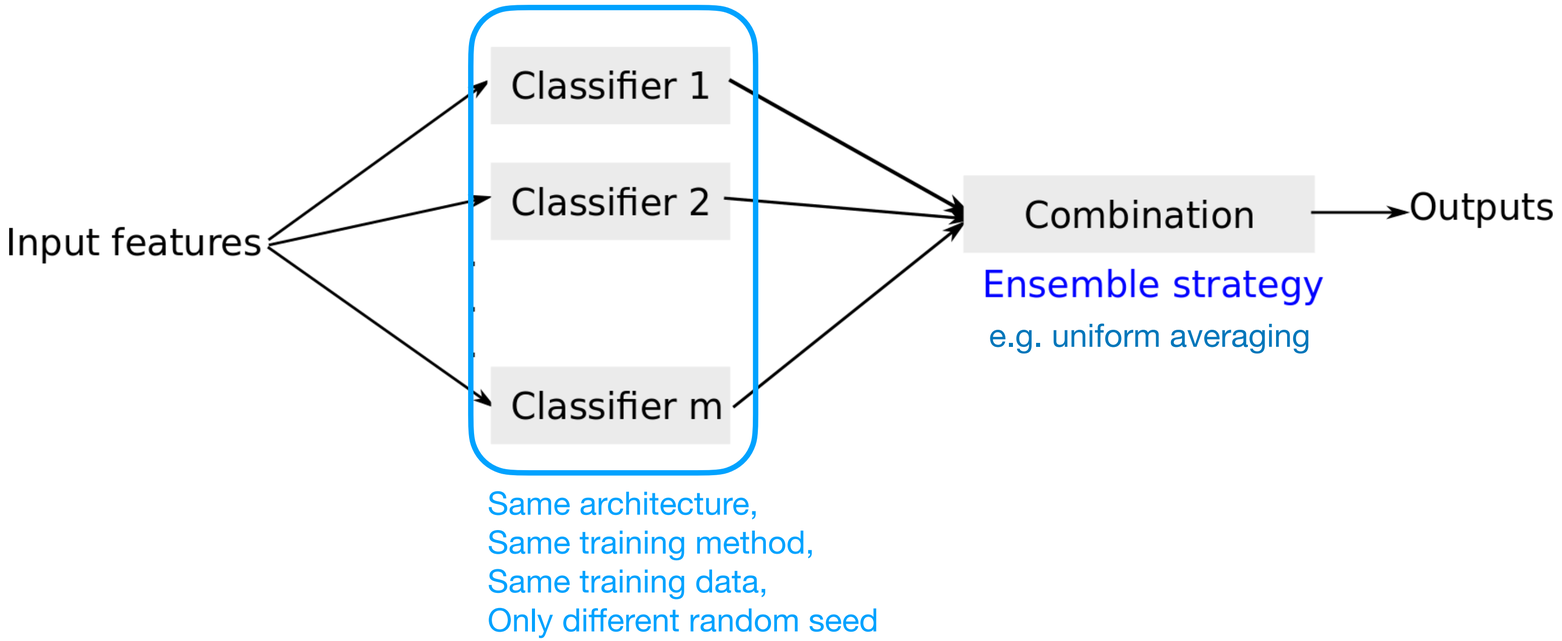


Boosting ensemble mechanism

...

Background : Model Ensemble

- Then, how ensemble of deep learning models can improve test accuracy?
  - Many works exist but *none of those works apply to the particular type of ensemble.*



Background : Model Ensemble

- Then, how ensemble of deep learning models can improve test accuracy?
  - Plus, traditional views cannot theoretically explain some characteristics of model ensemble *in deep learning*.
    - e.g. ensemble in NTK [1] vs Deep learning

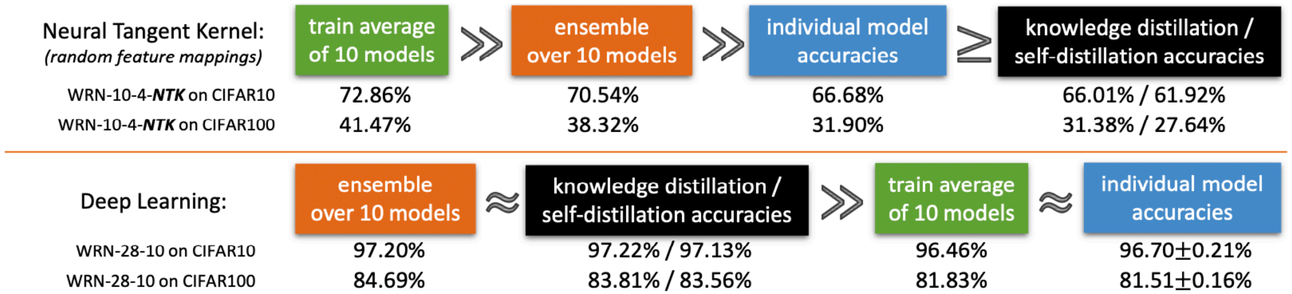


Figure 1: Ensemble in deep learning is very different from ensemble in random feature mappings. Details in Figure 6.

- So, authors' goal is to close gap between theory and practice in multi-class classification.

[1] Jacot et al., "Neural Tangent Kernel: Convergence and Generalization in Neural Network," NIPS 2018.

Neural Tangent Kernel (NTK) and ensemble

- NTK[1] is an approach to approximate neural network, as linear function over random features.
  - With $f : \mathbb{R}^{D+d} \to \mathbb{R},\ x \in \mathbb{R}^d$ and $W \in \mathbb{R}^D, f(W, x) \approx f(W_0, x) + \langle W - W_0, \nabla_W f(W_0, x) \rangle$
- Ensemble works for random features.
  - By enlarging feature space from $\nabla_W f(W_0, x)$ to $\{ \nabla_W f(W_0^i, x) \}_{i \in [L]}$.
- (Contradiction) Training average ($F(x) = \frac{1}{L} \sum_{i=1}^{L} f_i$) works even better in NTK, while it is worse in deep learning.
- **Summary**. It may be more accurate to study ensemble in deep learning as a *feature learning process*, instead of *feature selection process.*

[1] Jacot et al., “Neural Tangent Kernel: Convergence and Generalization in Neural Network,” NIPS 2018.

Feature learning and ensemble

- Some works[1] explain the benefit of ensemble as *reducing variance of individual solutions*.
  - However, reducing variance can reduce only a test loss, not necessarily the test classification error.

		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 7: 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* improve test accuracy. "xx % (yy %)" means $xx\%$ accuracy for single model and $yy\%$ for ensemble. More experiments in Appendix B.4 (Figure 10 and 11).

To know: [2] argues that models can learn features when input is Gaussian or Gaussian like.

[1] Mohabi et al., "Self-Distillation Amplifies Regularization in Hilbert space," NIPS 2020.
[2] Kenji Kawaguchi, "Deep Learning without Poor Local Minima," NIPS 2016.

Main approach : Multi-View Data

- Consider a binary classification problem with four features v_1, v_2, v_3 and v_4.

- When the label is class 1, then:[3]

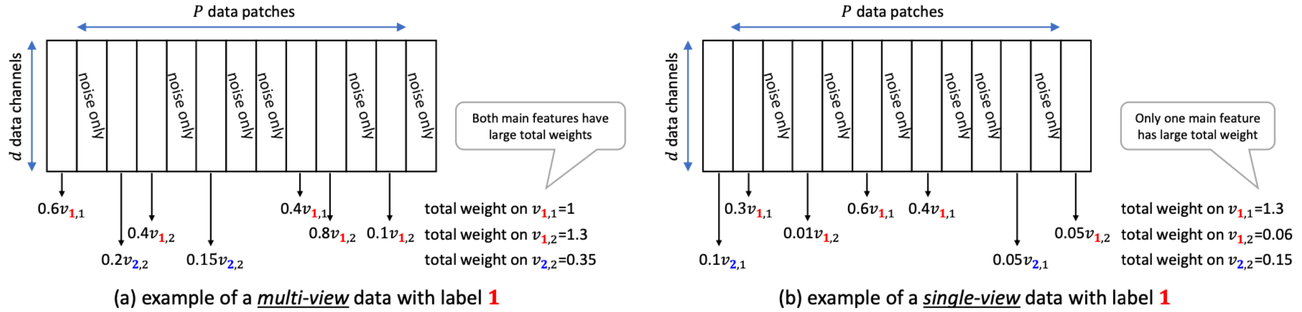
  both v_1, v_2 appears with weight 1, one of v_3, v_4 appears with weight 0.1 w.p. 80%;
  only v_1 appears with weight 1, one of v_3, v_4 appears with weight 0.1 w.p. 10%;
  only v_2 appears with weight 1, one of v_3, v_4 appears with weight 0.1 w.p. 10%.

Multi-view data

- When the label is class 2, then

  both v_3, v_4 appears with weight 1, one of v_1, v_2 appears with weight 0.1 w.p. 80%;
  only v_3 appears with weight 1, one of v_1, v_2 appears with weight 0.1 w.p. 10%;
  only v_4 appears with weight 1, one of v_1, v_2 appears with weight 0.1 w.p. 10%.

Single-view data



(a) example of a *multi-view* data with label **1**

(b) example of a *single-view* data with label **1**

Figure 4: Illustration of a multi-view and a single-view data point; the feature vectors can also be combined with feature noise and random noise, see Def. 3.1.

Main approach : Multi-View Data

k : number of classification label
μ : portion of single-view data

- How individual model learn features?
  - Model can answer correctly for all training data.
    - Model learns one of $\{v_1, v_2\}$ for label 1 and one of $\{v_3, v_4\}$ for label 2.
      - This model can classify 90% of data from each class by learning correct feature.
    - For 10% of data, network memorize it using *noise space* in the data (as known as overfitting).

Theorem 1 (single model, restated). *For sufficiently large $k > 0$, every $m \in [\mathsf{polylog}(k), \frac{1}{s\sigma_0^q \mathsf{polylog}(k)}]$, every $\eta \le \frac{1}{\mathsf{poly}(k)}$, after $T = \frac{\mathsf{poly}(k)}{\eta}$ many iterations, when Parameter D.1 is satisfied, with probability at least $1 - e^{-\Omega(\log^2 k)}$:*

- *(training accuracy is perfect) for every $(X, y) \in \mathcal{Z}$:*

$$\forall i \neq y\colon F_y^{(T)}(X) \ge F_i^{(T)}(X) + \Omega(\log k).$$

- *(multi-view testing is good) for every $i, j \in [k]$ we have $\widetilde{O}(1) \ge \Phi_i^{(T)} \ge 0.4\Phi_j^{(T)} + \Omega(\log k)$, and thus*

$$\mathop{\mathbf{Pr}}_{(X,y)\in\mathcal{D}_m}\left[F_y^{(T)}(X) \ge \max_{j\neq y} F_j^{(T)}(X) + \Omega(\log k)\right] \ge 1 - e^{-\Omega(\log^2 k)}$$

- *(single-view testing is bad) for every $(i, \ell) \in \mathcal{M}$ we have $\Phi_{i,3-\ell}^{(T)} \le \widetilde{O}(\sigma_0 m) \ll \frac{1}{\mathsf{polylog}(k)}$, and since $|\mathcal{M}| \ge k(1 - o(1))$, we have*[22]

$$\mathop{\mathbf{Pr}}_{(X,y)\in\mathcal{D}_s}\left[F_y^{(T)}(X) \ge \max_{j\neq y} F_j^{(T)}(X) - \frac{1}{\mathsf{polylog}(k)}\right] \le \frac{1}{2}\big(1 + o(1)\big)$$

- ***(test accuracy is consistently bad): meaning that:***

$$\mathbf{Pr}_{(X,y)\sim\mathcal{D}}[\exists i \in [k] \setminus \{y\}\colon F_y^{(T)}(X) < F_i^{(T)}(X)] \in [0.49\mu, 0.51\mu] \ .$$

Main approach : Multi-View Data

- How ensemble improves test accuracy?
  - Randomly initialized individual model can capture v_1 or v_2 each with 50%.
  - Hence, ensemble of many independent models will capture both features for both classes.

Ensemble. Suppose $\{F^{[\ell]}\}_{\ell\in[K]}$ are $K = \widetilde{\Omega}(1)$ independently trained models of F with $m = \mathsf{polylog}(k)$ for $T = O\big(\frac{\mathsf{poly}(k)}{\eta}\big)$ iterations (i.e., the same setting as Theorem 1 except we only need a small over-parameterization $m = \mathsf{polylog}(k)$). Let us define their ensemble

$$G(X) = \frac{\widetilde{\Theta}(1)}{K}\sum_{\ell} F^{[\ell]}(X) \tag{4.1}$$

Theorem 2 (ensemble accuracy, restated)**.** *In the same setting as above, suppose $\{F^{[w]}\}_{w\in[K]}$ are K independently randomly trained models with $m \in \big[\log^{\Omega(1)}(k), \log^{O(1)} k\big]$ for $T = \frac{\mathsf{poly}(k)}{\eta}$ iterations each. Let us define $G(X) = \frac{1}{K}\sum_{w} F^{[w]}(X)$.*

- *(training is perfect) same as the single model;*
- *(multi-view testing is good) same as the single model;*
- *(single-view testing is good) when $K \geq \mathsf{polylog}(k)$, ensemble model satisfies*

$$\Pr_{(X,y)\sim\mathcal{D}_s}\Big[G_y(X) \geq \max_{i\in[k]\setminus\{y\}} G_i(X) + \frac{1}{\mathsf{polylog}(k)}\Big] \geq 1 - e^{-\Omega(\log^2 k)}$$

- *(test accuracy is almost perfect): meaning that:*

$$\Pr_{(X,y)\sim\mathcal{D}}[\exists i \in [k]\setminus\{y\} : G_y(X) < G_i(X)] \leq 0.001\mu \ .$$

Main approach : Multi-View Data

- How ensembled knowledge distillation works?
  - Ensemble with all features actually outputs $(2,0.1) = (v_1 + v_2, v_3 + v_4)$.
  - Single model without feature v_4 outputs $(2,0) = (v_1 + v_2, v_3)$.
    - In $(0,0.1) = (\varnothing, ?)$, $?$ whose identity is v_4 is *dark knowledge* in hidden output of ensemble model.

Theorem 3 (ensemble distillation, restated). *For sufficiently large $k > 0$, for every $m \in \left[\log^{\Omega(1)}(k), \log^{O(1)} k\right]$, every $\eta \leq \frac{1}{\mathsf{poly}(k)}$, setting $\eta' = \eta\mathsf{poly}(k)$, after $T = \frac{\mathsf{poly}(k)}{\eta}$ many iterations, when Parameter G.2 is satisfied, with probability at least $1 - e^{-\Omega(\log^2 k)}$, for at least 90% of the iterations $t \leq T$:*

- *(training accuracy is perfect) for every $(X, y) \in \mathcal{Z}$:*

$$\forall i \neq y \colon F_y^{(t)}(X) \geq F_i^{(t)}(X) + \Omega(\log k).$$

- *(multi-view testing is good) for every $i, j \in [k]$ we have $\widetilde{O}(1) \geq \Phi_i^{(t)} \geq 0.4\Phi_j^{(t)} + \Omega(\log k)$, and thus*

$$\Pr_{(X,y)\in\mathcal{D}_m}\left[F_y^{(t)}(X) \geq \max_{j\neq y} F_j^{(t)}(X) + \Omega(\log k)\right] \geq 1 - e^{-\Omega(\log^2 k)}$$

- *(single-view testing is good) for every $i \in [k]$ and $\ell \in [2]$ we have $\Phi_{i,\ell}^{(t)} \geq \Omega(\log k)$ and thus*

$$\Pr_{(X,y)\in\mathcal{D}_s}\left[F_y^{(T)}(X) \geq \max_{j\neq y} F_j^{(T)}(X) + \Omega(\log k)\right] \leq 1 - e^{-\Omega(\log^2 k)}$$

→

- *(test accuracy is almost perfect): meaning that:*

$$\Pr_{(X,y)\sim\mathcal{D}}[\exists i \in [k] \setminus \{y\} \colon F_y^{(t)}(X) < F_i^{(t)}(X)] \leq 0.001\mu \ .$$

Main approach : Multi-View Data

- How self knowledge distillation works?
  - Self-distillation is performing implict ensemble + knowledge distillation.

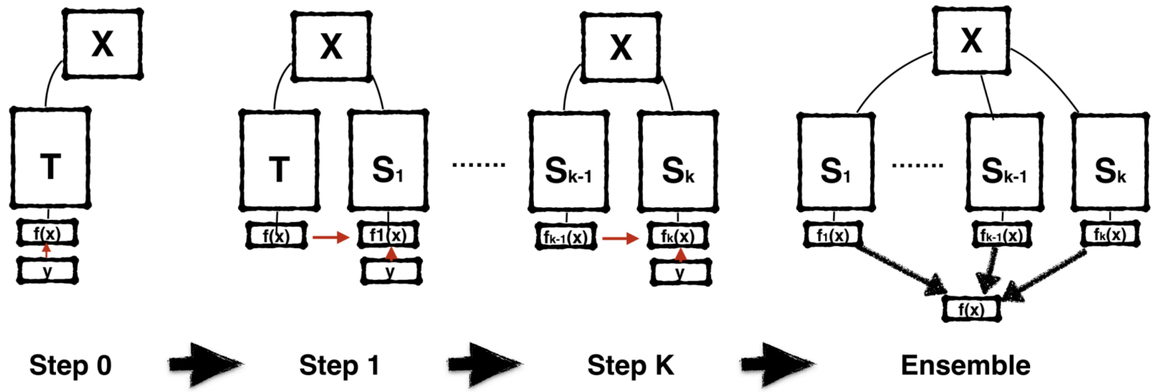


Figure 1. **Graphical representation of the BAN training procedure**: during the first step the teacher model T is trained from the labels Y. Then, at each consecutive step, a new identical model is initialized from a different random seed and trained from the supervision of the earlier generation. At the end of the procedure, additional gains can be achieved with an ensemble of multiple students generations.

[1] Furlanello et al., “Born-Again Neural Networks,” ICML 2018.

Main approach : Multi-View Data

- How self knowledge distillation works?
  - Self-distillation is performing implict ensemble + knowledge distillation.

Assumption 4.1 (balanced $\mathcal{D}_m$)**.** *In Def. 3.1, for multi-view data (X, y), we additionally assume that the marginal distributions of $\sum_{p\in\mathcal{P}_v(X)} z_p^q \in [1, 1 + o(1)]$ for $v \in \{v_{y,1}, v_{y,2}\}$.*

Theorem 4 (self distillation, restated)**.** *Suppose the data satisfies Assumption 4.1. For sufficiently large $k > 0$, for every $m \in \left[\log^{\Omega(1)}(k), k\right]$, every $\eta \le \frac{1}{\mathsf{poly}(k)}$, setting $T = \frac{\mathsf{poly}(k)}{\eta}$ and $T' = \frac{\mathsf{poly}(k)}{\eta}$, when Parameter D.1 is satisfied, with probability at least $1 - e^{-\Omega(\log^2 k)}$:*

- *(training accuracy is perfect) for every $(X, y) \in \mathcal{Z}$:*

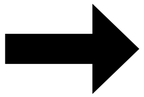
$$\forall i \neq y\colon F_y^{(T+T')}(X) \ge F_i^{(T+T')}(X) + \Omega(\log k).$$

- *(multi-view testing is good) for every $i, j \in [k]$ we have $\widetilde{O}(1) \ge \Phi_i^{(T+T')} \ge 0.4\Phi_j^{(T+T')} + \Omega(\log k)$, and thus*

$$\mathop{\mathbf{Pr}}_{(X,y)\in\mathcal{D}_m}\left[F_y^{(T+T')}(X) \ge \max_{j\neq y} F_j^{(T+T')}(X) + \Omega(\log k)\right] \ge 1 - e^{-\Omega(\log^2 k)}$$

- *(single-view testing is better) for every $(i, \ell) \in \mathcal{M}_F \cup \mathcal{M}_G$ we have $\Phi_{i,\ell}^{(T+T')} \ge \Omega\big(\frac{1}{\log k}\big)$, and since $|\mathcal{M}_F \cup \mathcal{M}_G| \ge 1.5k(1 - o(1))$, we have*

$$\mathop{\mathbf{Pr}}_{(X,y)\in\mathcal{D}_s}\left[F_y^{(T+T')}(X) \ge \max_{j\neq y} F_j^{(T+T')}(X) + \Omega(\log k)\right] \ge \frac{3}{4}\big(1 - o(1)\big)$$



- *(test accuracy is better): meaning that:*

$$\mathop{\mathbf{Pr}}_{(X,y)\sim\mathcal{D}}[\exists i \in [k] \setminus \{y\}\colon F_y^{(T+T')}(X) < F_i^{(T+T')}(X)] \le 0.26\mu$$

Justify the multi-view in practice

- Each independently trained model focus on different views.

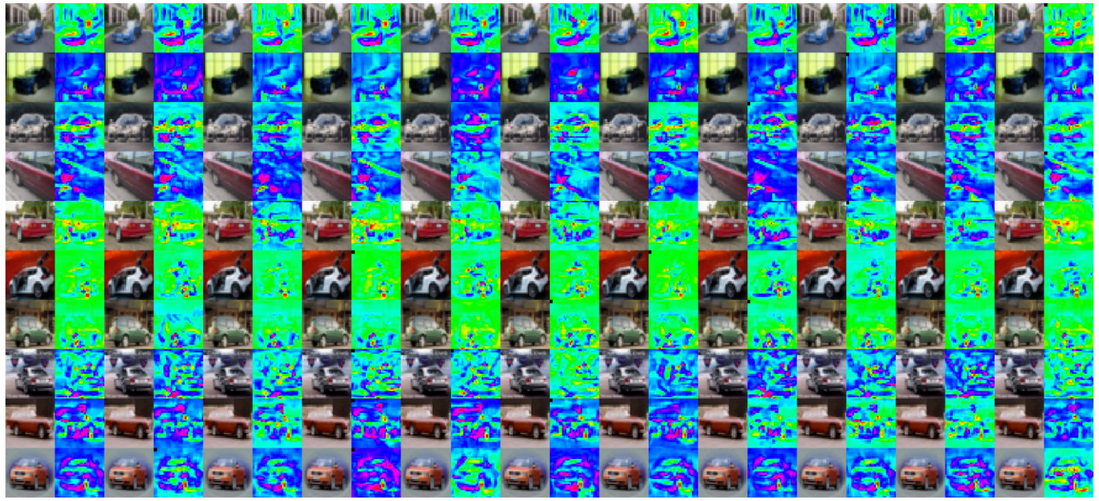


Figure 4: Ten independently trained ResNet-34 models (and their ensemble) detect car images through different reasonings, suggesting that the data has multi views, and independently trained neural networks do utilize this structure. The numerical experiments in Figure 9 also suggest the existence of multi views.

Justify the multi-view in practice

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 9: 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 B.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 3 for an illustration. This indicates that many data can be classified correctly using different 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.**

Empirical result

- Ensemble in deep learning works differently from ensemble of feature selection

finite-width neural kernel models	CIFAR10 test accuracy: single model (best of 10)	ensemble (over 10)	train $\sum_\ell f_\ell$ (over 10)	knowledge distillation	self-distill	CIFAR100 test accuracy: 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.

Empirical result

- Knowledge distillation works for model ensemble in aspect of multi-view

	CIFAR10 test accuracy				CIFAR100 test accuracy			
	single model (over 10)	ensemble (over 10)	10 runs of knowledge distill	ensemble over knowledge distill	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%	76.38±*0.23%*	81.13%	78.94±0.21%	80.35%
ResNet-34	93.65±*0.19%*	94.97%	94.37±0.13%	94.88%	71.66±*0.43%*	76.85%	73.57±0.34%	75.60%
ResNet-34-2	95.45±*0.14%*	96.55%	96.00±0.12%	96.42%	77.01±*0.35%*	81.48%	79.43±0.23%	81.56%
ResNet-16-10	96.08±*0.16%*	96.80%	96.73±0.07%	96.76%	80.03±*0.17%*	83.18%	82.51±0.14%	83.36%
ResNet-22-10	96.44±*0.09%*	97.12%	97.01±0.09%	97.09%	81.17±*0.23%*	84.33%	83.54±0.19%	84.27%
ResNet-28-10	96.70±*0.21%*	97.20%	97.06±0.08%	97.24%	81.51±*0.16%*	84.69%	83.75±0.16%	84.87%

Message ①: an ensemble over *single models* (independently trained) can be distilled into a single model with moderate accuracy loss.
Message ②: an ensemble over models *after knowledge distillation* *does not improve accuracy by much* – in fact, not exceeding the ensemble accuracy of the original single models ③ – despite the training objective is still non-convex and different random seeds are used.
This means, knowledge distillation models (i.e. simply matching the soft labels) *have learned most of the features* from the ensemble, and have less variety comparing to the original single models. This also means that "(huge) non-convexity" in neural networks and SGD with "different random seeds" *even together do not guarantee ensemble advantage unconditionally*; the structure of the data (and hard labels) is extremely important for ensemble to work as we mainly focus on in this paper.

Figure 8: Single models (+ their ensemble) vs. Knowledge distillations (+ their ensemble). Details in Appendix B.2.

Conclusion

- It is a first theoretical work towards understanding how ensemble work *in deep learning*.

- Authors **propose** a generic structure of the data refered to as **multi-view.**
  - With this view, they prove that
    - **Ensemble can improve test accuracy** for two-layer neural networks in the setting.
    - **Ensembled model can be distilled** into a single model.

Q&A