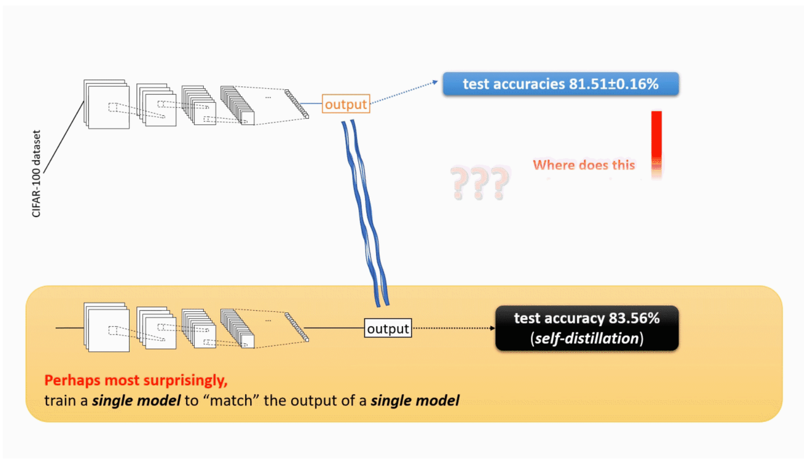
Microsoft Research Blog

Three mysteries in deep learning: Ensemble, knowledge distillation, and self-distillation

Published January 19, 2021

By Zeyuan Allen-Zhu, Senior Researcher; Yuanzhi Li, Assistant Professor, Carnegie Mellon University

Research Area
Artificial intelligence



Under now-standard techniques, such as over-parameterization, batch-normalization, and adding residual links, "modern age" neural network training—at least for image classification tasks and many others—is usually quite stable. Using standard neural network architectures and training algorithms (typically SGD with momentum), the learned models perform consistently well, not only in terms of training accuracy but even in test accuracy, regardless of which random initialization or random data order is used during the training. For instance, if one trains the same WideResNet-28-10 architecture on the CIFAR-100 dataset 10 times with different random seeds, the mean test accuracy is 81.51% while the standard deviation is only 0.16%.

In a new paper, "Towards Understanding Ensemble, Knowledge Distillation, and Self-Distillation in Deep Learning," we focus on studying the discrepancy of neural networks during the training process that has arisen purely from randomizations. We ask the following questions: besides this small deviation in test accuracies, do the neural networks trained from different random initializations actually learn very different functions? If so, where does the discrepancy come from? How do we reduce such discrepancy and make the neural network more stable or even better? These questions turn out to be quite nontrivial, and they relate to the mysteries of three techniques widely used in deep learning.

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

Three of the mysteries in deep learning

Mystery 1: Ensemble. The learned networks $F_1,\ldots F_{10}$ using different random seeds—despite having very similar test performance—are observed to associate with very different functions. Indeed, using a well-known technique called ensemble, which merely takes the unweighted average of the outputs of these independently trained networks, one can obtain a huge boost in test-time performance in many deep learning applications. (See Figure 1 below.) This implies the individual functions $F_1,\ldots F_{10}$ must be different. However, why does ensemble work with a sudden performance boost? Alternatively, if one directly trains $(F_1+\cdots+F_{10})/10$ altogether, why does the performance boost disappear?

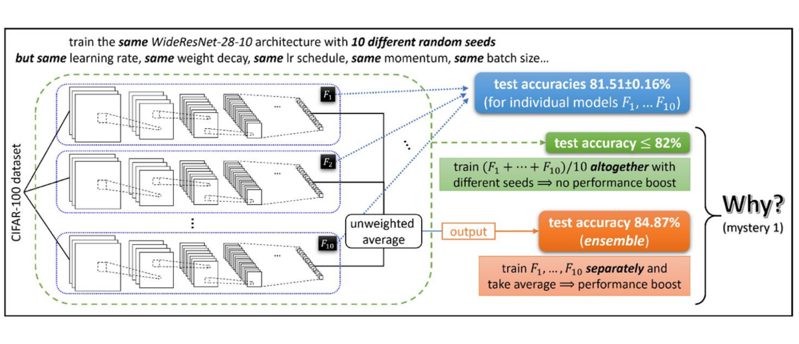


Figure 1: Ensemble gives a performance boost to test accuracies in deep learning applications, but such accuracy gains cannot be matched by training the average of the models directly.

Mystery 2: Knowledge distillation. While ensemble is great for improving test-time performance, it becomes 10 times slower during inference time (that is, test time): we need to compute the outputs of 10 neural networks instead of one. This is an issue when we deploy such models in a low-energy, mobile environment. To fix it, a seminal technique called knowledge distillation was proposed. That is, *knowledge distillation* simply trains another individual model to match the *output* of the ensemble. Here, the output of the ensemble (also called the *dark knowledge*) on a cat image may look like "80% cat + 10% dog + 10% car," while the true training label is "100% cat." (See Figure 2 below.)

It turns out the so-trained individual model can, to great extent, match the test-time performance of a 10-times-bigger ensemble. However, this leads to more questions. Why does matching the outputs of the ensemble give us better test accuracy compared to matching the true labels? Moreover, can we perform ensemble learning over the models after knowledge distillation to further improve test accuracy?

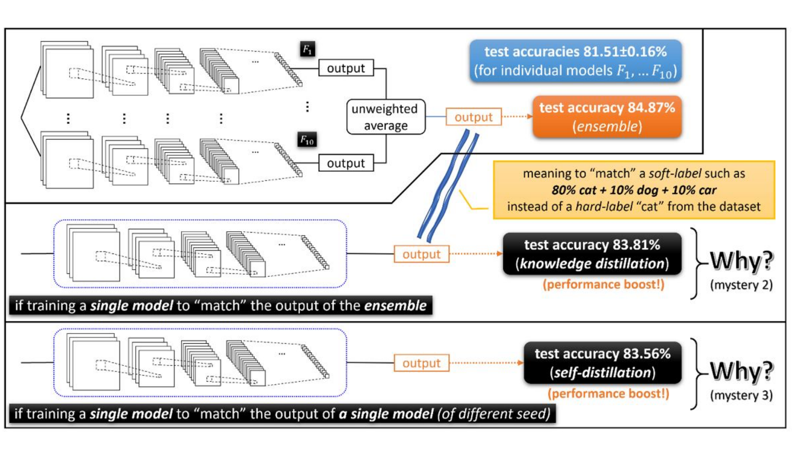


Figure 2: Knowledge distillation and self-distillation also give performance boosts in deep learning.

Mystery 3: Self-distillation. Note that knowledge distillation at least intuitively makes sense: the teacher ensemble model has 84.8% test accuracy, so the student individual model can achieve 83.8%. The following phenomenon, called *self-distillation* (or "Be Your Own Teacher"), is completely astonishing—by performing knowledge distillation against an individual model of the same architecture, test accuracy can also be improved. (See Figure 2 above.) Consider this: if training an individual model only gives 81.5% test accuracy, then how come "training the same model again *using itself as the teacher*" suddenly boosts the test accuracy consistently to 83.5%?

Ensemble of neural networks versus ensemble of feature mappings

Most existing theories on ensemble only apply to the case where individual models are fundamentally different (for example, decision trees supported on different subsets of the variables) or trained over different datasets (such as bootstrapping). They cannot justify the aforementioned phenomenon in the deep learning world, where individually trained neural networks are of the *same* architecture and using the *same* training data—their only difference comes from the randomness during training.

Perhaps the existing theorem closest to matching ensemble in deep learning is the *ensemble of random feature mappings*. On one hand, combining multiple linear models of random (prescribed) features should improve test-time performance because it increases the number of features. On the other hand, in certain parameter regimes, neural network weights can stay very close to their initializations (known as the neural tangent kernel, or NTK, regime), and the resulting network is merely learning a linear function over prescribed feature mappings that are completely decided by the random initialization (see this work). When combining these two, one may conjecture that ensemble in deep learning shares the principle of ensemble in random feature mappings. That leads us to the following question:

Does ensemble/knowledge distillation work in the same way in deep learning compared to that in random feature mappings (namely, the NTK feature mappings)?

Answer: not really, as evidenced by the experiment in Figure 3 below. This figure compares ensemble and knowledge distillation in deep learning versus that in a linear model over random feature mappings. Ensemble works in *both* cases. However, the accuracies in Figure 3 clearly show that they work for *completely different* reasons. Specifically:

- Unlike in the deep learning case, the superior performance of ensemble in the random feature setting cannot be distilled to an individual model. For instance, in Figure 3, the neural tangent kernel (NTK) models' ensemble achieves 70.54% accuracy on the CIFAR-10 dataset, but after knowledge distillation, it goes down to 66.01%, even worse than the test accuracy of 66.68% on the individual model.
- In deep learning, direct training the average of models $(F_1+\cdots+F_{10})/10$ offers no benefit compared to training one individual model F_i; while in the random feature setting, training the average outperforms both individual models and their ensemble. For instance, in Figure 3, the NTK models' ensemble achieves 70.54% accuracy, but this is even worse than directly training the average of 10 models, which gives 72.86% accuracy.

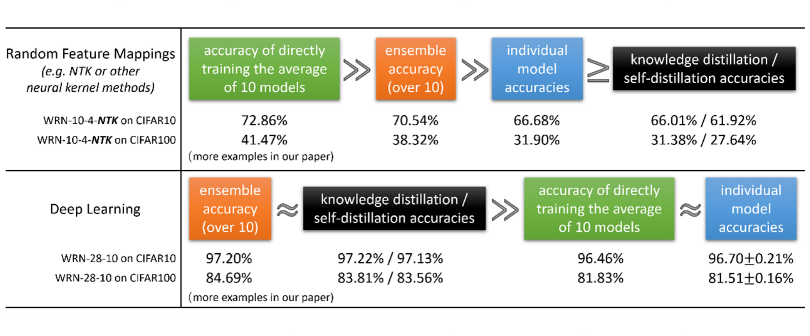


Figure 3: Ensemble works in random feature mappings (but for a completely different reason from that in deep learning), and knowledge distillation does not work in random feature mapping.

The reason for this is that the neural network is performing *hierarchical feature learning*—each individual model F_i, despite having different random initializations, is still capable of learning the same set of features as the others. Thus, their average offers almost no additional capacity compared to an individual network. However, in the linear setting, each F_i uses a different set of prescribed features; so, although combining these features (using either ensemble or direct training average) does offer an advantage, they just cannot by distilled into an individual model due to the scarcity of features.

Ensemble versus reducing variance of individual models

Besides ensemble of random features, one might also conjecture that, due to the high complexity of the neural network, each individual model F_i might learn a function $F_i(x) = y + \xi_i$, where ξ_i is some noise that depends on the randomness used during training. Classical statistics suggests that if all the ξ_i's are roughly independent, then averaging them greatly reduces the amount of noise. Thus,

Can "ensemble reduces the variance" be the reason for ensemble's performance boost?

Answer: our evidence shows that this hypothesis of reducing variance is very questionable in the context of deep learning:

- Ensemble does not increase test accuracy forever: ensemble over 100 individual models typically makes no difference when compared to ensemble over 10 individual models. So, the average of 100 ξ_i's does not reduce the variance anymore comparing to 10 ξ_i's—indicating that these ξ_i's are (1) potentially not independent, and/or (2) can be biased so the mean is not zero. In the event of (1), it is difficult to argue how much error can be reduced by averaging these ξ_i's
- Even if one wishes to accept the idealistic belief that (1) does not occur so all these ξ_i's are just biased, or in symbols, $F_i(x) = y + \xi + \xi_i'$ where ξ is a common error and ξ_i' is an individual, independent error, then, why does knowledge distillation work? After ensemble we expect the network to output close to y+ξ with a common bias ξ. Then, why is this output with error ξ (also known as dark knowledge) better than the original true label for training?
- In Figure 4, we see that ensemble learning of neural networks does not always improve test accuracy—at least when the input is Gaussian-like. In other words, "averaging these " does not lead to any accuracy gain in these networks. Therefore, we need to understand ensemble in deep learning with more care than a generic statement such as "reducing variance."

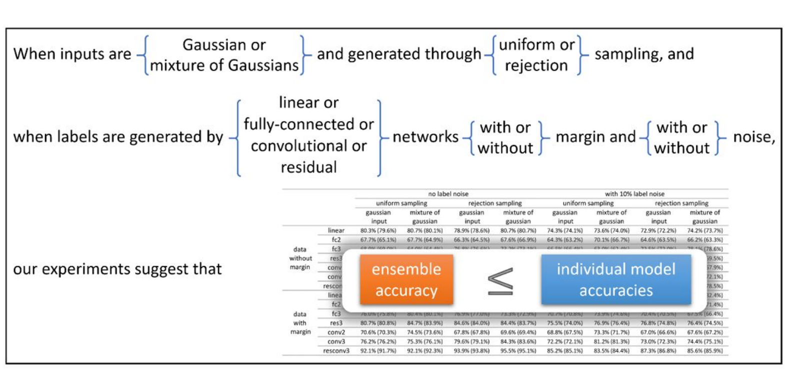


Figure 4: When inputs are Gaussian-like, experiments suggest that ensemble does not improve test accuracy.

Multi-view data: A new approach to justify ensemble in deep learning

Since ensemble is less likely to work under unstructured random inputs (see Figure 4), we have to look at special structure in the data to understand it properly.

In our new work, we propose studying a common structure that can be found in many of the datasets where deep learning excels. In vision datasets in particular, the object can usually be classified using *multiple views*. For example, a car image can be classified as a car by looking at the headlights, the wheels, or the windows. For a typical placement of a car in images, we can observe all these features, and it suffices to use one of the features to classify it as a car. However, there are some car images taken from a particular angle, where one or more of these features are missing. For example, an image of a car facing forward might be missing the wheel feature. We give real-life examples in Figure 5.

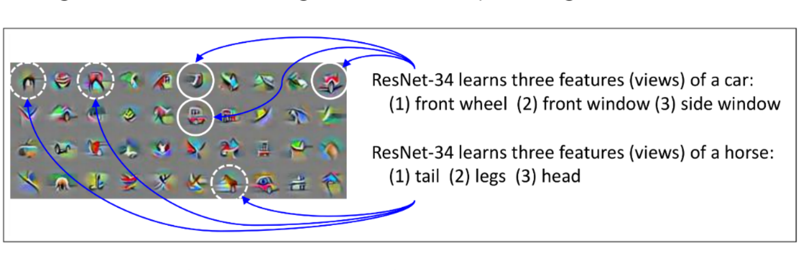


Figure 5: Visualization of some channels in layer 23 of ResNet-34 trained on CIFAR-10

We refer to this structure as "multi-view," where each class of the data has multiple view features. In most of the data, almost all of the view features will show up, but in some data, some view features might be missing. (More broadly speaking, the "multi-view" structure shows up not only in the input pixel space but also in the intermediate layers; we refer interested readers to Figure 6 of our paper for an experimental justification.)

We develop a theorem showing that, during the training process of a neural network under multi-view data, the network will:

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

The first point implies that ensemble of different networks will collect all these learnable view features, hence achieving a higher test accuracy. The second point implies that individual models do not learn all the view features *not* because they do not have enough capacity, but rather because there are not sufficiently many training data left to learn these views. Most of the data has already been classified correctly with existing view features, so they essentially provide no gradient at this stage of training.

Knowledge distillation: Forcing an individual model to learn multiple views

In this new work, we continue to show how knowledge distillation works. In real-life scenarios, some car images might look "more like a cat" compared to others: for example, the headlights on some car images might look like cat eyes. When this happens, the ensemble model can provide meaningful dark knowledge: for instance, "car image X is 10% like a cat."

Now comes the key observation. When training an individual neural network model, if the "headlight" view is not learned, then even if the remaining views can still correctly label image X as a car, they cannot be used to match the dark knowledge "image X is 10% like a cat." In other words, during knowledge distillation, the individual model is forced to learn every possible view feature, matching the performance of ensemble. Note that the crux of knowledge distillation in deep learning is that an individual model, as a neural network, is performing feature learning and therefore capable of learning all the features of ensemble. This is consistent with what we observe in practice. (See Figure 6.)

	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%

(no performance boost!) (no performance boost!)

Figure 6: Knowledge distillation has learned most of the view features from the ensemble, and so ensemble learning on models after knowledge distillation offers no more performance boost.

Self-distillation: Implicitly combining ensemble and knowledge distillation

In this new work, we also give theoretical support to knowledge *self-distillation* (recall Figure 3). Training an individual model to match the output of another identical individual model (but using a different random seed) somehow gives a performance boost.

At a high level, we view self-distillation as combining ensemble and knowledge distillation in a more compact manner. When learning an individual model F_2 from random initialization to match the output of a separately trained individual model F_1, one can expect F_2 to learn a subset of the features depending on its own random initialization. In addition to this, F_2 also has the incentive to learn the subset of features already learned by F_1. In other words, one can view this process as "ensemble learning two individual models F_1, F_2 and distilling it to F_2." The final learned model F_2 may not necessarily cover all the learnable views in the dataset, but it has the potential to at least learn all the views that can be covered through ensemble learning over two individual models. This is where the test-time performance boost comes from! (Recall Figures 2 and 3.)

Conclusion and going forward

In this work, we show, to the best of our knowledge, the first theoretical proof toward understanding how ensemble works in deep learning. We also provide empirical evidence to support our theory and our "multi-view" data hypothesis. We believe our framework can be applied to other settings. For example, data augmentation using random cropping could be potentially regarded as another way to enforce the network to learn "multi-views." We hope that, in practice, our new theoretical insights on how neural networks pick up features during training can also help design new, principled approaches to improve the test accuracy of a neural network, potentially matching that of ensemble.

Latest from

Zeyuan Allen-Zhu
I'm a senior researcher in the Machine Learning and Optimization group here at Microsoft Research AI (Redmond). My full list of publications can...
View profile

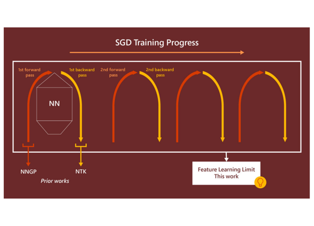
Yuanzhi Li
Assistant Professor, Carnegie Mellon University
View profile

Related to this article

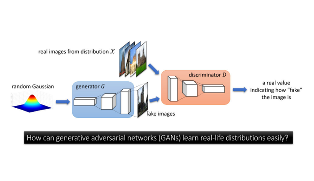
Labs
Microsoft Research Lab - Redmond

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

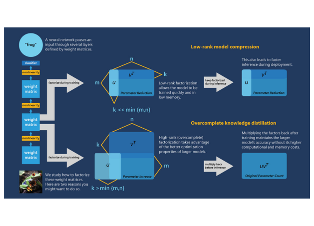
Up next
See all blog posts >



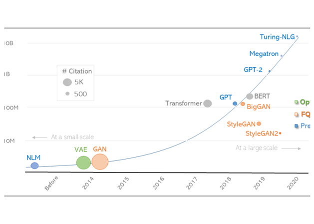
On infinitely wide neural networks that exhibit feature learning >



How can generative adversarial networks learn real-life distributions easily >



Factorized layers revisited: Compressing deep networks without playing the lottery >



A deep generative model trifecta: Three advances that work towards harnessing large-scale power >