Born-Again Neural Networks

Tommaso Furlanello [1] **Zachary C. Lipton** [2 3] **Michael Tschannen** [4] **Laurent Itti** [1] **Anima Anandkumar** [5 3]

Abstract

Knowledge Distillation (KD) consists of transferring "knowledge" from one machine learning model (the *teacher*) to another (the *student*). Commonly, the teacher is a high-capacity model with formidable performance, while the student is more compact. By transferring knowledge, one hopes to benefit from the student's compactness, without sacrificing too much performance. We study KD from a new perspective: rather than compressing models, we train students parameterized identically to their teachers. Surprisingly, these *Born-Again Networks* (BANs), outperform their teachers significantly, both on computer vision and language modeling tasks. Our experiments with BANs based on DenseNets demonstrate state-of-the-art performance on the CIFAR-10 (3.5%) and CIFAR-100 (15.5%) datasets, by validation error. Additional experiments explore two distillation objectives: (i) *Confidence-Weighted by Teacher Max* (CWTM) and (ii) *Dark Knowledge with Permuted Predictions* (DKPP). Both methods elucidate the essential components of KD, demonstrating the effect of the teacher outputs on both predicted and non-predicted classes.

1. Introduction

In a 2001 paper on statistical modeling (Breiman et al., 2001), Leo Breiman noted that different stochastic algorithmic procedures (Hansen & Salamon, 1990; Liaw et al., 2002; Chen & Guestrin, 2016) can lead to diverse models with similar validation performances. Moreover, he noted that we can often compose these models into an ensemble that achieves predictive power superior to each of the constituent models. Interestingly, given such a powerful ensemble, one can often find a simpler model — no more complex than one of the ensemble's constituents — that mimics the ensemble and achieves its performance. Previously, in *Born-Again Trees* Breiman & Shang (1996) pioneered this idea, learning single trees that match the performance of multiple-tree predictors. These born-again trees approximate the ensemble decision but offer some desired properties of individual decision trees, such as their purported amenability to interpretation. A number of subsequent papers have proposed variations the idea of *born-again* models. In the neural network community, similar ideas emerged in papers on *model compression* by Bucilua et al. (2006) and related work on *knowledge distillation* (KD) by Hinton et al. (2015). In both cases, the idea is typically to transfer the knowledge of a high-capacity teacher with desired high performance to a more compact student (Ba & Caruana, 2014; Urban et al., 2016; Rusu et al., 2015). Although the student cannot match the teacher when trained directly on the data, the *distillation* process brings the student closer to matching the predictive power of the teacher.

We propose to revisit KD with the objective of disentangling the benefits of this training technique from its use in model compression. In experiments transferring knowledge from teachers to students of identical capacity, we make the surprising discovery that the students become the masters, outperforming their teachers by significant margins. In a manner reminiscent to Minsky's *Sequence of Teaching Selves* (Minsky, 1991), we develop a simple re-training procedure: after the teacher model converges, we initialize a new student and train it with the dual goals of predicting the correct labels and matching the output distribution of the teacher. We call these students *Born-Again Networks* (BANs) and show that applied to DenseNets, ResNets and LSTM-based sequence models, BANs consistently have lower validation errors than their teachers. For DenseNets, we show that this procedure can be applied for multiple steps, albeit with diminishing returns.

We observe that the gradient induced by KD can be decomposed into two terms: a *dark knowledge* term, containing the information on the wrong outputs, and a ground-truth component which corresponds to a simple rescaling of the original gradient that would be obtained using the real labels. We interpret the second term as training from the real

[1]University of Southern California, Los Angeles, CA, USA [2]Carnegie Mellon University, Pittsburgh, PA, USA [3]Amazon AI, Palo Alto, CA, USA [4]ETH Zürich, Zürich, Switzerland [5]Caltech, Pasadena, CA, USA. Correspondence to: Tommaso Furlanello <furlanel@usc.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

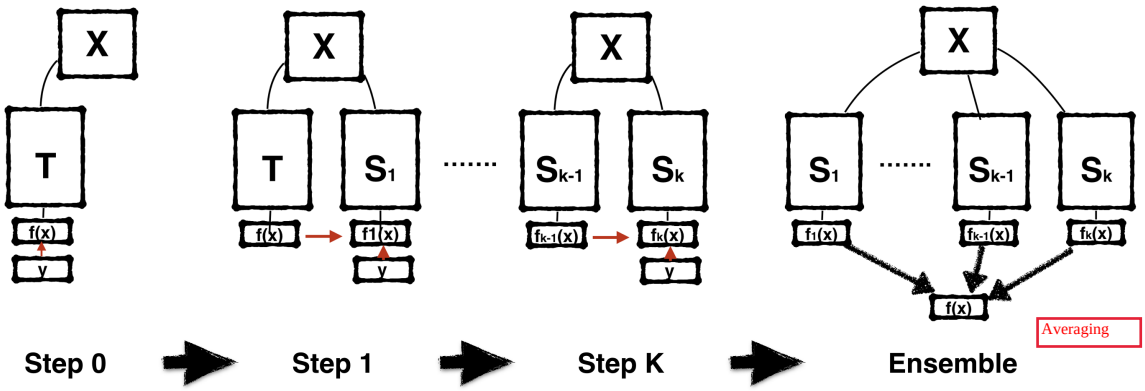

Figure 1. **Graphical representation of the BAN training procedure**: during the first step the teacher model T is trained from the labels Y. Then, at each consecutive step, a new identical model is initialized from a different random seed and trained from the supervision of the earlier generation. At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

labels using importance weights for each sample based on the teacher's confidence in its maximum value. Experiments investigating the importance of each term are aimed at quantifying the contribution of dark knowledge to the success of KD.

Furthermore, we explore whether the objective function induced by the DenseNet teacher can be used to improve a simpler architecture like ResNet bringing it close to state-of-the-art accuracy. We construct *Wide-ResNets* (Zagoruyko & Komodakis, 2016b) and *Bottleneck-ResNets* (He et al., 2016b) of comparable complexity to their teacher and show that these BAN-as-ResNets surpass their DenseNet teachers. Analogously we train DenseNet students from Wide-ResNet teachers, which drastically outperform standard ResNets. Thus, we demonstrate that weak masters can still improve performance of students, and KD need not be used with strong masters.

2. Related Literature

We briefly review the related literature on knowledge distillation and the models used in our experiments.

2.1. Knowledge Distillation

A long line of papers have sought to transfer knowledge between one model and another for various purposes. Sometimes the goal is compression: to produce a compact model that retains the accuracy of a larger model that takes up more space and/or requires more computation to make predictions (Bucilua et al., 2006; Hinton et al., 2015). Breiman & Shang (1996) proposed compressing neural networks and multiple-tree predictors by approximating them with a single tree. More recently, others have proposed to transfer knowledge from neural networks by approximating them with simpler models like decision trees (Chandra et al., 2007) and generalized additive models (Tan et al., 2018) for the purpose of increasing *transparency* or *interpretability*. Further, Frosst & Hinton (2017) proposed distilling deep networks into decision trees for the purpose of explaining decisions. We note that in each of these cases, what precisely is meant by *interpretability* or *transparency* is often undeclared and the topic remains fraught with ambiguity (Lipton, 2016).

Among papers seeking to compress models, the goal of knowledge transfer is simple: produce a student model that achieves better accuracy by virtue of knowledge transfer from the teacher model than it would if trained directly. This research is often motivated by the resource constraints of underpowered devices like cellphones and internet-of-things devices. In a pioneering work, Bucilua et al. (2006) compress the information in an ensemble of neural networks into a single neural network. Subsequently, with modern deep learning tools, Ba & Caruana (2014) demonstrated a method to increase the accuracy of shallow neural networks, by training them to mimic deep neural networks, using an penalizing the L2 norm of the difference between the student's and teacher's logits. In another recent work, Romero et al. (2014) aim to compress models by approximating the mappings between teacher and student hidden layers, using linear projection layers to train the relatively narrower students.

Interest in KD increased following Hinton et al. (2015), who demonstrated a method called *dark knowledge*, in which a student model trains with the objective of matching the full softmax distribution of the teacher model. One paper applying ML to Higgs Boson and supersymmetry detection, made the (perhaps inevitable) leap to applying dark knowledge to the search for dark matter (Sadowski et al.,

2015). Urban et al. (2016) train a *super teacher* consisting of an ensemble of 16 convolutional neural networks and compresses the learned function into shallow multilayer perceptrons containing 1, 2, 3, 4, and 5 layers. In a different approach, Zagoruyko & Komodakis (2016a) force the student to match the attention map of the teacher (norm across the channel dimension in each spatial location) at the end of each residual stage. Czarnecki et al. (2017) try to minimize the difference between teacher and student derivatives of the loss with respect to the input in addition to minimizing the divergence from teacher predictions.

Interest in KD has also spread beyond supervised learning. In the deep reinforcement learning community, for example, Rusu et al. (2015) distill multiple DQN models into a single one. A number of recent papers (Furlanello et al., 2016; Li & Hoiem, 2016; Shin et al., 2017) employ KD for the purpose of minimizing forgetting in continual learning. (Papernot et al., 2016) incorporate KD into an adversarial training scheme. Recently, Lopez-Paz et al. (2015) pointed out some connections between KD and a theory of on learning with privileged information (Pechyony & Vapnik, 2010).

In a superficially similar work to our own, Yim et al. (2017) propose applying KD from a DNN to another DNN of identical architecture, and report that the student model trains faster and achieves greater accuracy than the teacher. They employ a loss which is calculated as follows: for a number of pairs of layers $\{(i, j)\}$ of same dimensionality, they (i) calculate a number of inner products $G_{i,j}(\boldsymbol{x})$ between the activation tensors at the layers i and j, and (ii) they construct a loss that requires the student to match the statistics of these inner products to the corresponding statistics calculated on the teacher (for the same example), by minimizing $||G^T_{i,j}(\boldsymbol{x}) - G^S_{i,j}(\boldsymbol{x})||^2_2$. The authors exploit a statistic used in Gatys et al. (2015) to capture style similarity between images (given the same network).

Key differences Our work differs from (Yim et al., 2017) in several key ways. First, their novel loss function, while technically imaginative, is not demonstrated to outperform more standard KD techniques. Our work is the first, to our knowledge, to demonstrate that dark knowledge, applied for self-distillation, even without softening the logits results in significant boosts in performance. Indeed, when distilling to a model of identical architecture we achieve the current second-best performance on the CIFAR100 dataset. Moreover, this paper offers empirical rigor, providing several experiments aimed at understanding the efficacy of self-distillation, and demonstrating that the technique is successful in domains other than images.

2.2. Residual and Densely Connected Neural Networks

First described in (He et al., 2016a), deep residual networks employ design principles that are rapidly becoming ubiquitous among modern computer vision models. The Resnet passes representations through a sequence of consecutive *residual-blocks*, each of which applies several sub-modules, denoted *residual units*), each of which consists of convolutions and skip-connections, interspersed with spatial downsampling. Multiple extensions (He et al., 2016b; Zagoruyko & Komodakis, 2016b; Xie et al., 2016; Han et al., 2016) have been proposed, progressively increasing their accuracy on CIFAR100 (Krizhevsky & Hinton, 2009) and ImageNet (Russakovsky et al., 2015). Densely connected networks (DenseNets) (Huang et al., 2016) are a recently proposed variation where the summation operation at the end of each unit is substituted by a concatenation of the input and output of the unit.

3. Born-Again Networks

Consider the classical image classification setting where we have a training dataset consisting of tuples of images and labels $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and we are interested in fitting a function $f(x) : \mathcal{X} \mapsto \mathcal{Y}$, able to generalize to unseen data. Commonly, the mapping $f(x)$ is parametrized by a neural network $f(x, \theta_1)$, θ_1 with parameters in some space Θ_1. We learn the parameters via Empirical Risk Minimization (ERM), producing a resulting model θ^*_1 that minimizes some loss function:

$$\theta^*_1 = \arg\min_{\theta_1} \mathcal{L}(y, f(x, \theta_1)), \tag{1}$$

typically optimized by some variant of Stochastic Gradient Descent (SGD).

Born-Again Networks (BANs) are based on the empirical finding demonstrated in knowledge distillation / model compression papers that generalization error, can be reduced by modifying the loss function. This should not be surprising: the most common such modifications are the classical regularization penalties which limit the complexity of the learned model. BANs instead exploit the idea demonstrated in KD, that the information contained in a teacher model's output distribution $f(x, \theta^*_1)$ can provide a rich source of training signal, leading to a second solution $f(x, \theta^*_2)$, $\theta_2 \in \Theta_2$, with better generalization ability. We explore techniques to modify, substitute, or regularize the original loss function with a KD term based on the cross-entropy between the new model's outputs and the outputs of the original model:

$$\mathcal{L}(f(x, \arg\min_{\theta_1} \mathcal{L}(y, f(x, \theta_1))), f(x, \theta_2)). \tag{2}$$

Unlike the original works on KD, we address the case when

the teacher and student networks have identical architectures. Additionally, we present experiments addressing the case when the teacher and student networks have similar capacity but different architectures. For example we perform knowledge transfer from a DenseNet teacher to a ResNet student with similar number of parameters.

3.1. Sequence of Teaching Selves Born-Again Networks Ensemble

Inspired by the impressive recent results of SGDR Wide-Resnet (Loshchilov & Hutter, 2016) and Coupled-DenseNet (Dutt et al., 2017) ensembles on CIFAR100, we apply BANs sequentially with multiple generations of knowledge transfer. In each case, the k-th model is trained, with knowledge transferred from the $k-1$-th student:

$$\mathcal{L}(f(x, \arg\min_{\theta_{k-1}} \mathcal{L}(f(x, \theta_{k-1}))), f(x, \theta_k)). \tag{3}$$

Finally, similarly to ensembling multiple snapshots (Huang et al., 2017) of SGD with restart (Loshchilov & Hutter, 2016), we produce Born-Again Network Ensembles (BANE) by averaging the prediction of multiple generations of BANs.

$$\hat{f}^k(x) = \sum_{i=1}^{k} f(x, \theta_i)/k. \tag{4}$$

We find the improvements of the sequence to saturate, but we are able to produce significant gains through ensembling.

3.2. Dark Knowledge Under the Light

The authors in (Hinton et al., 2015) suggest that the success of KD depends on the dark knowledge hidden in the distribution of logits of the *wrong* responses, that carry information on the similarity between output categories. Another plausible explanations might be found by comparing the gradients flowing through output node corresponding to the correct class during distillation vs. normal supervised training. Note that restricting attention to this gradient, the knowledge distillation might resemble importance-weighting where the weight corresponds to the teacher's confidence in the correct prediction.

The single-sample gradient of the cross-entropy between student logits z_j and teacher logits t_j with respect to the ith output is given by:

$$\frac{\partial \mathcal{L}_i}{\partial z_i} = q_i - p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} - \frac{e^{t_i}}{\sum_{j=1}^{n} e^{t_j}}. \tag{5}$$

When the target probability distribution function corresponds to the ground truth $*$ one-hot label $p_* = y_* = 1$ this reduces to:

$$\frac{\partial \mathcal{L}_*}{\partial z_*} = q_* - y_* = \frac{e^{z_*}}{\sum_{j=1}^{n} e^{z_j}} - 1 \tag{6}$$

When the loss is computed with respect to the complete teacher output, the student back-propagates the mean of the gradients with respect to correct and incorrect outputs across all the b samples s of the mini-batch (assuming without loss of generality the nth label is the ground truth label $*$):

$$\sum_{s=1}^{b}\sum_{i=1}^{n} \frac{\partial \mathcal{L}_{i,s}}{\partial z_{i,s}} = \sum_{s=1}^{b}(q_{*,s} - p_{*,s}) + \sum_{s=1}^{b}\sum_{i=1}^{n-1}(q_{i,s} - p_{i,s}), \tag{7}$$

up to a rescaling factor $1/b$. The second term corresponds to the information incoming from all the wrong outputs, via dark knowledge. The first term corresponds to the gradient from the correct choice and can be rewritten as

$$\frac{1}{b}\sum_{s=1}^{b}(q_{*,s} - p_{*,s}y_{*,s}) \tag{8}$$

which allows the interpretation of the output of the teacher p_* as a weighting factor of the original ground truth label y_*.

When the teacher is correct and confident in its output, i.e. $p_{*,s} \approx 1$, Eq. (8) reduces to the ground truth gradient in Eq. (6), while samples with lower confidence have their gradients rescaled by a factor $p_{*,s}$ and have reduced contribution to the overall training signal.

We notice that this form has a relationship with importance weighting of samples where the gradient of each sample in a mini-batch is balanced based on its importance weight w_s. When the importance weights correspond to the output of a teacher for the correct dimension we have:

$$\sum_{s=1}^{b} \frac{w_s}{\sum_{u=1}^{b} w_u}(q_{*,s} - y_{*,s}) = \sum_{s=1}^{b} \frac{p_{*,s}}{\sum_{u=1}^{b} p_{*,u}}(q_{*,s} - y_{*,s}). \tag{9}$$

So we ask the following question: does the success of dark knowledge owe to the information contained in the non-argmax outputs of the teacher? Or is dark knowledge simply performing a kind of importance weighting? To explore these questions, we develop two treatments. In the first treatment, Confidence Weighted by Teacher Max (CWTM), we weight each example in the student's loss function (standard cross-entropy with ground truth labels) by the confidence of the teacher model on that example (even if the teacher wrong). We train BAN models using an approximation of Eq. (9), where we substitute the correct answer $p_{*,s}$ with

the max output of the teacher $\max p_{.,s}$:

$$\sum_{s=1}^{b} \frac{\max p_{.,s}}{\sum_{u=1}^{b} \max p_{.,u}} (q_{*,s} - y_{*,s}). \quad (10)$$

In the second treatment, dark knowledge with Permuted Predictions (DKPP), **we permute the non-argmax outputs of the teacher's predicted distribution**. We use the original formulation of Eq. (7), substituting the $*$ operator with $\max$ and permuting the teacher dimensions of the dark knowledge term, leading to:

$$\sum_{s=1}^{b}\sum_{i=1}^{n} \frac{\partial \mathcal{L}_{i,s}}{\partial z_{i,s}} = \sum_{s=1}^{b} (q_{*,s} - \max p_{.,s}) + \sum_{s=1}^{b}\sum_{i=1}^{n-1} q_{i,s} - \phi(p_{j,s}), \quad (11)$$

where $\phi(p_{j,s})$ are the permuted outputs of the teacher. In DKPP we scramble the correct attribution of dark knowledge to each non-argmax output dimension, destroying the pairwise similarities of the original output covariance matrix.

3.3. BANs Stability to Depth and Width Variations

DenseNet architectures are parametrized by depth, growth, and compression factors. Depth corresponds to the number of dense blocks. The growth factor defines how many new features are concatenated at each new dense block, while the compression factor controls by how much features are reduced at the end of each stage.

Variations in these hyper-parameters induce a tradeoff between number of parameters, memory use and the number of sequential operations for each pass. We test the possibility of expressing the same function of the DenseNet teacher with different architectural hyperparameters. In order to construct a fair comparison, we construct DenseNets whose output dimensionality at each spatial transition matches that of the DenseNet-90-60 teacher. Keeping the size of the hidden states constant, we modulate the growth factor indirectly via the choice the number of blocks. Additionally, we can drastically reduce the growth factor by reducing the compression factor before or after each spatial transition.

3.4. DenseNets Born-Again as ResNets

Since BAN-DenseNets perform at the same level as plain DenseNets with multiples of their parameters, we test whether the BAN procedure can be used to improve ResNets as well. Instead of the weaker ResNet teacher, we employ a DenseNet-90-60 as teacher and construct comparable ResNet students by switching *Dense Blocks* with *Wide Residual Blocks* and *Bottleneck Residual Blocks*.

4. Experiments

All experiments performed on CIFAR-100 use the same preprocessing and training setting as for Wide-ResNet (Zagoruyko & Komodakis, 2016b) except for Mean-Std normalization. The only form of regularization used other than the KD loss are weight decay and, in the case of Wide-ResNet drop-out.

4.1. CIFAR-10/100

Baselines To get a strong teacher baseline without the prohibitive memory usage of the original architectures, we explore multiple heights and growth factors for DenseNets. We find a good configuration in relatively shallower architectures with increased growth factor and comparable number of parameters to the largest configuration of the original paper. Classical ResNet baselines are trained following (Zagoruyko & Komodakis, 2016b). Finally, we construct Wide-ResNet and bottleneck-ResNet networks that match the output shape of DenseNet-90-60 at each block, as baselines for our BAN-ResNet with DenseNet teacher experiment.

BAN-DenseNet and ResNet We perform BAN retraining after convergence, using the same training schedule originally used to train the teacher networks. We employ DenseNet-(116-33, 90-60, 80-80, 80-120) and train a sequence of BANs for each configuration. We test the ensemble performance for sequences of 2 and 3 BANs. We explored other forms of knowledge transfer for training BANs. Specifically, we tried progressively constraining the BANs to be more similar to their teachers, sharing the first and last layers between student and teacher, or adding losses that penalize the L2 distance between student and teacher activations. However, we found these variations to systematically perform slightly worse than the simple KD via cross entropy. For BAN-ResNet experiments with a ResNet teacher we use Wide-ResNet-(28-1, 28-2, 28-5, 28-10).

BAN without Dark Knowledge In the first treatment, CWTM, we fully exclude the effect of all the teacher's output except for the argmax dimension. To do so, we train the students with the normal label loss where samples are weighted by their importance. We interpret the max of the teacher's output for each sample as the importance weight and use it to rescale each sample of the student's loss.

In the second treatment, DKPP, we maintain the overall high order moments of the teachers output, but randomly permute each output dimension except the argmax one. We maintain the rest of the training scheme and the architecture unchanged.

Both methods alter the covariance between outputs, such

that any improvement cannot be fully attributed to the classical dark knowledge interpretation.

Variations in Depth, Width and Compression Rate We also train variations of DenseNet-90-60, with increased or decreased number of units in each block and different number of channels determined through a ratio of the original activation sizes.

BAN-Resnet with DenseNet teacher In all the BAN-ResNet with DenseNet teacher experiments, the student shares the first and last layers of the teacher. We modulate the complexity of the ResNet by changing the number of units, starting from the depth of the successful Wide-ResNet-28 (Zagoruyko & Komodakis, 2016b) and reducing until only a single residual unit per block remains. Since the number of channels in each block is the same for every residual unit, we match it with a proportion of the corresponding dense block output after the 1×1 convolution, before the spatial down-sampling. We explore mostly architectures with a ratio of 1, but we also show the effect of halving the width of the network.

BAN-DenseNet with ResNet teacher With this experiment we test whether a weaker ResNet teacher is able to successfully train DenseNet-90-60 students. We use multiple configurations of Wide-ResNet teacher and train the Ban-DenseNet student with the same hyper parameters of the other DenseNet experiments.

4.2. Penn Tree Bank

To validate our method beyond computer vision applications, we also apply the BAN framework to language models and evaluate it on the Penn Tree Bank (PTB) dataset (Marcus et al., 1993) using the standard train/test/validation split by (Mikolov et al., 2010). We consider two BAN language models: a single layer LSTM (Hochreiter & Schmidhuber, 1997) with 1500 units (Zaremba et al., 2014) and a smaller model from (Kim et al., 2016) combining a convolutional layers, highway layers, and a 2-layer LSTM (referred to as CNN-LSTM).

For the LSTM model we use weight tying (Press & Wolf, 2016), 65% dropout and train for 40 epochs using SGD with a mini-batch size of 32. An adaptive learning rate schedule is used with an initial learning rate 1 that is multiplied by a factor of 0.25 if the validation perplexity does not decrease after an epoch.

The CNN-LSTM is trained with SGD for the same number of epochs with a mini-batch size of 20. The initial learning rate is set to 2 and is multiplied by a factor of 0.5 if the validation perplexity does not decrease by at least 0.5 after an epoch (this schedule slightly differs from (Kim et al., 2016), but worked better for the teacher model in our experiments).

Both models are unrolled for 35 steps and the KD loss is simply applied between the softmax outputs of the unrolled teacher an student.

5. Results

We report the surprising finding that by performing KD across models of similar architecture, BAN student models tend to improve over their teachers across all configurations.

5.1. CIFAR-10

As can be observed in Table 1 the CIFAR-10 test error is systematically lower or equal for both Wide-ResNet and DenseNet student trained from an identical teacher. It is worth to note how for BAN-DenseNet the gap between architectures of different complexity is quickly reduced leading to implicit gains in the parameters to error rate ratio.

Table 1. **Test error on CIFAR-10** for Wide-ResNet with different depth and width and DenseNet of different depth and growth factor.

Network	Parameters	Teacher	BAN
Wide-ResNet-28-1	0.38 M	6.69	**6.64**
Wide-ResNet-28-2	1.48 M	5.06	**4.86**
Wide-ResNet-28-5	9.16 M	4.13	**4.03**
Wide-ResNet-28-10	36 M	**3.77**	3.86
DenseNet-112-33	6.3 M	3.84	**3.61**
DenseNet-90-60	16.1 M	3.81	**3.5**
DenseNet-80-80	22.4 M	**3.48**	3.49
DenseNet-80-120	50.4 M	**3.37**	3.54

5.2. CIFAR-100

For CIFAR-100 we find stronger improvements for all BAN-DenseNet models. We focus therefore most of our experiments to explore and understand the born-again phenomena on this dataset.

BAN-DenseNet and BAN-ResNet In Table 2 we report test error rates using both labels and teacher outputs (BAN+L) or only the latter (BAN). The improvement of fully removing the label supervision is systematic across modality, it is worth noting that the smallest student BAN-DenseNet-112-33 reaches an error of 16.95% with only 6.5 M parameters, comparable to the 16.87% error of the DenseNet-80-120 teacher with almost eight times more parameters.

In Table 3 all but one Wide-ResNnet student improve over their identical teacher.

Sequence of Teaching Selves Training BANs for multiple generations leads to inconsistent but positive improvements, that saturate after a few generations. The third gener-

Table 2. **Test error on CIFAR-100** *Left Side:* DenseNet of different depth and growth factor and respective BAN student. BAN models are trained only with the teacher loss, BAN+L with both label and teacher loss. CWTM are trained with sample importance weighted label, the importance of the sample is determined by the max of the teacher's output. DKPP are trained only from teacher outputs with all the dimensions but the argmax permuted. *Right Side:* test error on CIFAR-100 sequence of BAN-DenseNet, and the BAN-ensembles resulting from the sequence. Each BAN in the sequence is trained from cross-entropy with respect to the model at its left. BAN and BAN-1 models are trained from Teacher but have different random seeds. We include the teacher as a member of the ensemble for Ens*3 for 80-120 since we did not train a BAN-3 for this configuration.

Network	Teacher	BAN	BAN+L	CWTM	DKPP	BAN-1	BAN-2	BAN-3	Ens*2	Ens*3
DenseNet-112-33	18.25	**16.95**	17.68	17.84	17.84	17.61	17.22	**16.59**	15.77	15.68
DenseNet-90-60	17.69	**16.69**	16.93	17.42	17.43	16.62	**16.44**	16.72	15.39	15.74
DenseNet-80-80	17.16	**16.36**	16.5	17.16	16.84	16.26	16.30	**15.5**	15.46	15.14
DenseNet-80-120	16.87	**16.00**	16.41	17.12	16.34	**16.13**	16.13	/	**15.13**	**14.9**

Table 3. **Test error on CIFAR-100** for Wide-ResNet students trained from identical Wide-ResNet teachers and for DenseNet-90-60 students trained from Wide-ResNet teachers

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

ation of BAN-3-DenseNet-80-80 produces our single best model with 22M parameters, achieving 15.5% error on CIFAR0100 (Table 2). To our knowledge, this is currently the SOTA non-ensemble model without shake-shake regularization. It is only beaten by Yamada et al. (2018) who use a *pyramidal ResNet* trained for 1800 epochs with a combination of shake-shake (Gastaldi, 2017), pyramid-drop (Yamada et al., 2016) and cut-out regularization (DeVries & Taylor, 2017).

BAN-Ensemble Similarly, our largest ensemble BAN-3-DenseNet-BC-80-120 with 150M parameters and an error of 14.9% is the lowest reported ensemble result in the same setting. BAN-3-DenseNet-112-33 is based on the building block of the best coupled-ensemble of (Dutt et al., 2017) and reaches a single-error model of 16.59% with only 6.3M parameters, furthermore the ensembles of two or three consecutive generations reach a comparable error of 15.77% and 15.68% with the baseline error of 15.68% reported in (Dutt et al., 2017) where four models were used.

Effect of non-argmax Logits As can be observed in the two rightmost columns if the left side of Table 2 we find that removing part of the dark knowledge still generally brings improvements to the training procedure with respect to the baseline. Importance weights CWTM lead to weak improvements over the teacher in all models but the largest DenseNet. Instead, in DKPP we find a comparable but systematic improvement effect of permuting all but the argmax dimensions.

These results demonstrate that KD does not simply contribute information on each specific non-correct output. DKPP demonstrates that the higher order moments of the output distribution that are invariant to the permutation procedure still systematically contribute to improved generalization. Furthermore, the complete removal of wrong logit information in the CWTM treatment still brings improvements for three models out of four, suggesting that the information contained in pre-trained models can be used to rebalance the training set, by giving less weight to training samples for which the teacher's output distribution is not concentrated on the max.

DenseNet to modified DenseNet students It can be seen in Table 4 that DenseNet students are particularly robust to the variations in the number of layers. The most shallow model with only half the number of its teacher layers DenseNet-7-1-2 still improves over the DenseNet-90-60 teacher with an error rate of 16.95%. Deeper variations are competitive or even better than the original student. The best modified student result is 16.43% error with twice the number of layers (half the growth factor) of its DenseNet-90-60 teacher.

The biggest instabilities as well as parameter saving is obtained by modifying the compression rate of the network, indirectly reducing the dimensionality of each hidden layer. Halving the number of filters after each spatial dimension reduction in DenseNet-14-0.5-1 gives an error of 19.83%, the worst across all trained DenseNets. Smaller reductions lead to larger parameter savings with lower accuracy losses, but directly choosing a smaller network retrained with BAN procedure like DenseNet-106-33 seems to lead to higher parameter efficiency.

DenseNet Teacher to ResNet Student Surprisingly, we find (Ttable 5) that our Wide-ResNet and Pre-ResNet students that match the output shapes at each stage of their DenseNet teachers tend to outperform classical ResNets, their teachers, and their baseline.

Table 4. **Test error on CIFAR-100-Modified Densenet:** a Densenet-90-60 is used as teacher with students that share the same size of hidden states after each spatial transition but differs in depth and compression rate

Densenet-90-60	Teacher	0.5*Depth	2*Depth	3*Depth	4*Depth	0.5*Compr	0.75*Compr	1.5*compr
Error	17.69	16.95	16.43	16.64	16.64	19.83	17.3	18.89
Parameters	22.4 M	21.2 M	13.7 M	12.9 M	1 2.6 M	5.1 M	10.1 M	80.5 M

Table 5. **DenseNet to ResNet:** CIFAR-100 test error for BAN-ResNets trained from a DenseNet-90-60 teacher with different numbers of blocks and compression factors. In all the BAN architectures, the number of units per block is indicated first, followed by the ratio of input and output channels with respect to a DenseNet-90-60 block. All BAN architectures share the first (conv1) and last(fc-output) layer with the teacher which are frozen. Every dense block is effectively substituted by residual blocks

DenseNet 90-60	Parameters	Baseline	BAN
Pre-activation ResNet-1001	10.2 M	22.71	/
BAN-Pre-ResNet-14-0.5	7.3 M	20.28	18.8
BAN-Pre-ResNet-14-1	17.7 M	18.84	**17.39**
BAN-Wide-ResNet-1-1	20.9 M	20.4	19.12
BAN-Match-Wide-ResNet-2-1	43.1 M	18.83	17.42
BAN-Wide-ResNet-4-0.5	24.3 M	19.63	**17.13**
BAN-Wide-ResNet-4-1	87.3 M	18.77	17.18

Table 6. **Validation/Test perplexity on PTB** (lower is better) for BAN-LSTM language model of different complexity

Network	Parameters	Teacher Val	BAN+L Val	Teacher Test	BAN+L Test
ConvLSTM	19M	83.69	80.27	80.05	**76.97**
LSTM	52M	75.11	71.19	71.87	**68.56**

Both BAN-Pre-ResNet with 14 blocks per stage and BAN-Wide-ResNet with 4 blocks per stage and 50% compression factor reach respectively a test error of 17.39% and 17.13% using a parameter budget that is comparable with their teachers. We find that for BAN-Wide-ResNets, only limiting the number of blocks to 1 per stage leads to inferior performance compared to the teacher.

Similar to how adapting the depth of the models offers a nice tradeoff between memory consumption and number of sequential operations, exchanging dense and residual blocks allows to choose between concatenation and additions. By using additions, ResNets overwrite old memory banks, saving RAM, at the cost of heavier models that do not share layers offering another technical tradeoff to choose from.

ResNet Teacher to DenseNet Students The converse experiment, training a DenseNet-90-60 student from ResNet student confirms the trend of students surpassing their teachers. The improvement from ResNet to DenseNet (Table 3, right-most column) over simple label supervision is significant as indicated by 16.79% error of the DenseNet-90-60 student trained from the Wide-ResNet-28-10.

5.3. Penn Tree Bank

Although we did not use the state-of-the-art bag of tricks (Merity et al., 2017) for training LSTMs, nor the recently proposed improvements on KD for sequence models (Kim & Rush, 2016), we found significant decreases in perplexity on both validation and testing set for our benchmark language models. The smaller BAN-LSTM-CNN model decreases test perplexity from 80.05 to 76.97, while the bigger BAN-LSTM model improves from 71.87 to 68.56. Unlike the CNNs trained for CIFAR classification, we find that LSTM models work only when trained with a combination of teacher outputs and label loss (BAN+L). One potential explanation for this finding might be that teachers generally reach 100% accuracy on the CIFAR training sets while the PTB training perplexity is far from being minimized.

6. Discussion

In Marvin Minsky's Society of Mind (Minsky, 1991), the analysis of human development led to the idea of a *sequence of teaching selves*. Minsky suggested that sudden spurts in intelligence during childhood may be due to longer and hidden training of new "student" models under the guidance of the older self. Minsky concluded that our perception of a long-term self is constructed by an ensemble of multiple generations of internal models, which we can use for guidance when the most current model falls short. Our results show several instances where such transfer was successful in artificial neural networks.

Acknowledgements

This work was supported by the National Science Foundation (grant numbers CCF-1317433 and CNS-1545089), C-BRIC (one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA), and the Intel Corporation. The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

References

Ba, J. and Caruana, R. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pp. 2654–2662, 2014.

Breiman, L. and Shang, N. Born again trees. *Available online at: ftp://ftp.stat.berkeley.edu/pub/users/breiman/BAtrees.ps*, 1996.

Breiman, L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, 2001.

Bucilua, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 535–541. ACM, 2006.

Chandra, R., Chaudhary, K., and Kumar, A. The combination and comparison of neural networks with decision trees for wine classification. *School of Sciences and Technology, University of Fiji*, 2007.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM, 2016.

Czarnecki, W. M., Osindero, S., Jaderberg, M., Świrszcz, G., and Pascanu, R. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems*, pp. 4281–4290, 2017.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 2017.

Dutt, A., Pellerin, D., and Quenot, G. Coupled Ensembles of Neural Networks. *arXiv:1709.06053*, 2017.

Frosst, N. and Hinton, G. Distilling a neural network into a soft decision tree. *arXiv:1711.09784*, 2017.

Furlanello, T., Zhao, J., Saxe, A. M., Itti, L., and Tjan, B. S. Active long term memory networks. *arXiv:1606.02355*, 2016.

Gastaldi, X.. Shake-shake regularization. *arXiv:1705.07485*, 2017.

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *arXiv:1508.06576*, 2015.

Han, D., Kim, J., and Kim, J. Deep pyramidal residual networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6307–6315, 2017.

Hansen, L. K. and Salamon, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645. Springer, 2016b.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. Snapshot ensembles: Train 1, get M for free. In *International Conference on Learning Representations*, 2017.

Kim, Y. and Rush, A. M. Sequence-level knowledge distillation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 2741–2749. AAAI Press, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. 2009.

Li, Z. and Hoiem, D. Learning without forgetting. In *European Conference on Computer Vision*, pp. 614–629. Springer, 2016.

Liaw, A., Wiener, M. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

Lipton, Z. C. The mythos of model interpretability. *ICML Workshop on Human Interpretability in Machine Learning, arXiv:1606.03490*, 2016.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with restarts. In *International Conference on Learning Representations*, 2017.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of English: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Merity, S., Keskar, N. S., and Socher, R. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*, 2017.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

Minsky, M. Society of mind: A response to four reviews. *Artificial Intelligence*, 48(3):371–396, 1991.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, pp. 582–597. IEEE, 2016.

Pechyony, D. and Vapnik, V. On the theory of learning with privileged information. In *Advances in Neural Information Processing Systems*, pp. 1894–1902, 2010.

Press, O. and Wolf, L. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 157–163, 2016.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv:1412.6550*, 2014.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., and Hadsell, R. Policy distillation. In *International Conference on Learning Representations*, 2016.

Sadowski, P., Collado, J., Whiteson, D., and Baldi, P. Deep learning, dark knowledge, and dark matter. In *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pp. 81–87, 2015.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pp. 2994–3003, 2017.

Tan, S., Caruana, R., Hooker, G., and Gordo, A. Transparent model distillation. *arXiv:1801.08640*, 2018.

Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., and Richardson, M. Do deep convolutional nets really need to be deep and convolutional? In *International Conference on Learning Representations*, 2017.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995, 2017.

Yamada, Y., Iwamura, M., and Kise, K. ShakeDrop regularization. *arXiv:1802.02375*, 2018.

Yamada, Y., Iwamura, M., and Kise, K.. Deep pyramidal residual networks with separated stochastic depth. *arXiv:1612.01230*, 2016.

Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7130–7138, 2017.

Zagoruyko, S. and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2016a.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1-87.12, 2016b.

Zaremba, W., Sutskever, I., and Vinyals, O. Recurrent neural network regularization. *arXiv:1409.2329*, 2014.