Knowledge Distillation

Seoul AI Meetup

Martin Kersner, 2017/12/23

Introduction

The best results are ussually achieved with

- Ensemble Models (Meta Learning presentation)
- Large Networks

Assumption
Time and cost of running inference in machine learning model is more important than the time and memory of training a model.

Example
Speech recognition (Google)
Face recognition (Apple)

Born Again Trees

Leo Breiman and Nong Shang, 1996

Explainibility of prediction is an important factor for a prediction algorithm.

1. Generate synthetic data by **smeering** method
2. Construct* tree using **repeatedly generated** synthetic data labeled by boosted/bagged predictor
3. Prune tree* and select** subtree

* CART
** Select subtree giving minimum error on the training set

Born Again Trees

Leo Breiman and Nong Shang, 1996

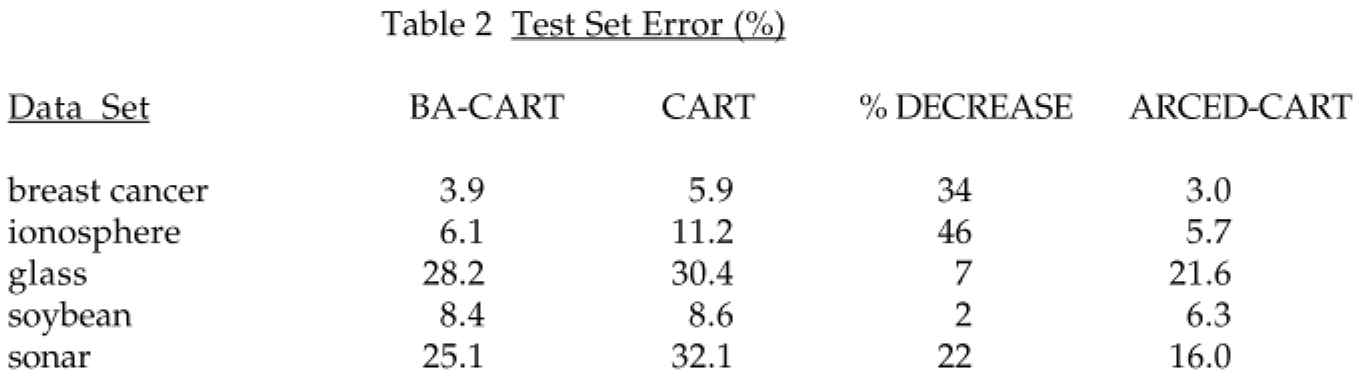


Table 2 Test Set Error (%)

Data Set	BA-CART	CART	% DECREASE	ARCED-CART
breast cancer	3.9	5.9	34	3.0
ionosphere	6.1	11.2	46	5.7
glass	28.2	30.4	7	21.6
soybean	8.4	8.6	2	6.3
sonar	25.1	32.1	22	16.0

Born Again Neural Networks

Tommaso Furlanello et al., NIPS, 2017

Train students that are **parametrized identically*** to their parents. Born Again Networks (BANs) tend to outperform their teacher models.

Training

1. Train parent network from scratch
2. Train student with dual goal

- predicting the correct labels
- matching the output distribution of the teacher.

3. Converged student network is used as a teacher in the next step

* teacher and student networks have identical architectures

Born Again Neural Networks

Tommaso Furlanello et al., NIPS, 2017

The information contained in the original model's output distribution can provide a **rich source of training signal**, leading to a second solution with **better generalization ability**.

$$\min_{\theta_k} \mathcal{L}(y, f(x, \theta_k)) + \mathcal{L}(f(x, \arg\min_{\theta_{k-1}} \mathcal{L}(y, f(x, \theta_{k\_1}))), f(x, \theta_k))$$

Born Again Neural Networks

Tommaso Furlanello et al., NIPS, 2017

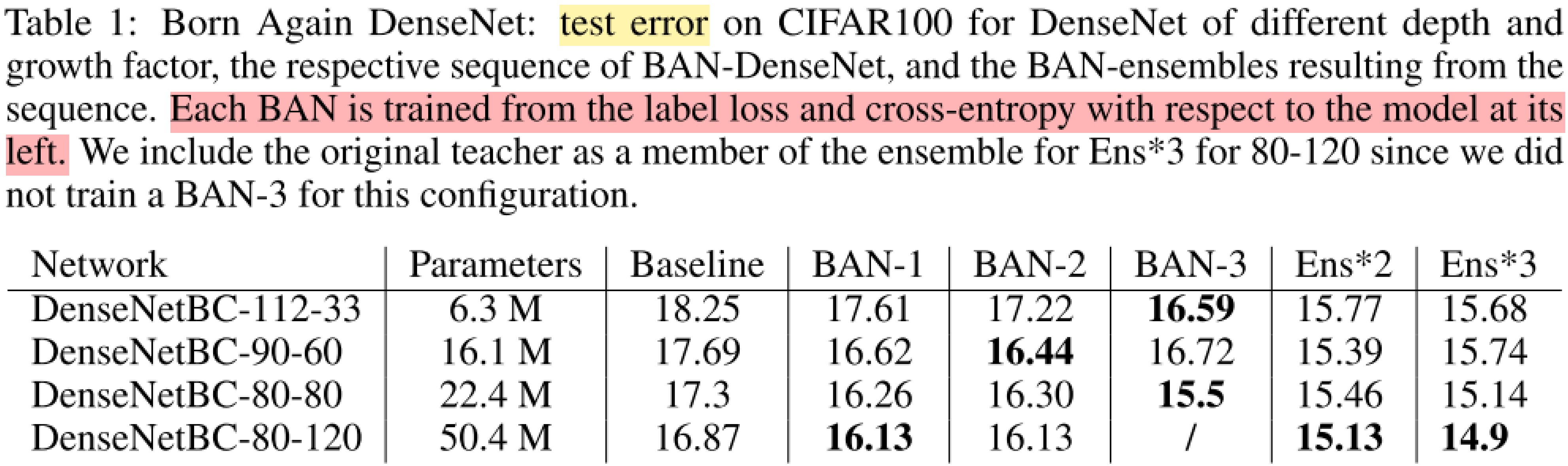


Table 1: Born Again DenseNet: test error on CIFAR100 for DenseNet of different depth and growth factor, the respective sequence of BAN-DenseNet, and the BAN-ensembles resulting from the sequence. Each BAN is trained from the label loss and cross-entropy with respect to the model at its left. We include the original 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	Parameters	Baseline	BAN-1	BAN-2	BAN-3	Ens*2	Ens*3
DenseNetBC-112-33	6.3 M	18.25	17.61	17.22	**16.59**	15.77	15.68
DenseNetBC-90-60	16.1 M	17.69	16.62	**16.44**	16.72	15.39	15.74
DenseNetBC-80-80	22.4 M	17.3	16.26	16.30	**15.5**	15.46	15.14
DenseNetBC-80-120	50.4 M	16.87	**16.13**	16.13	/	**15.13**	**14.9**

State-of-the-art performance on the CIFAR-100 dataset reaching a validation error of 15.5 %.

Model Compression

Cristian Bucila, Rich Caruana et al., KDD, 2006

Method for "compressing" large, complex ensembles into smaller, faster models, usually without significant loss in performance.

Training

1. Generate a synthetic data*
2. Label them by ensemble**
3. Train*** a neural network (single hidden layer with 128 hidden units)

* small training set (4K)
** SVMs, bagged trees, boosted trees, boosted stumps, simple decision trees, ran- dom forests, neural nets, logistic regression, k-nearest neigh- bor, and naive Bayes
*** Training with no special loss function.

Model Compression

Cristian Bucila, Rich Caruana et al., KDD, 2006

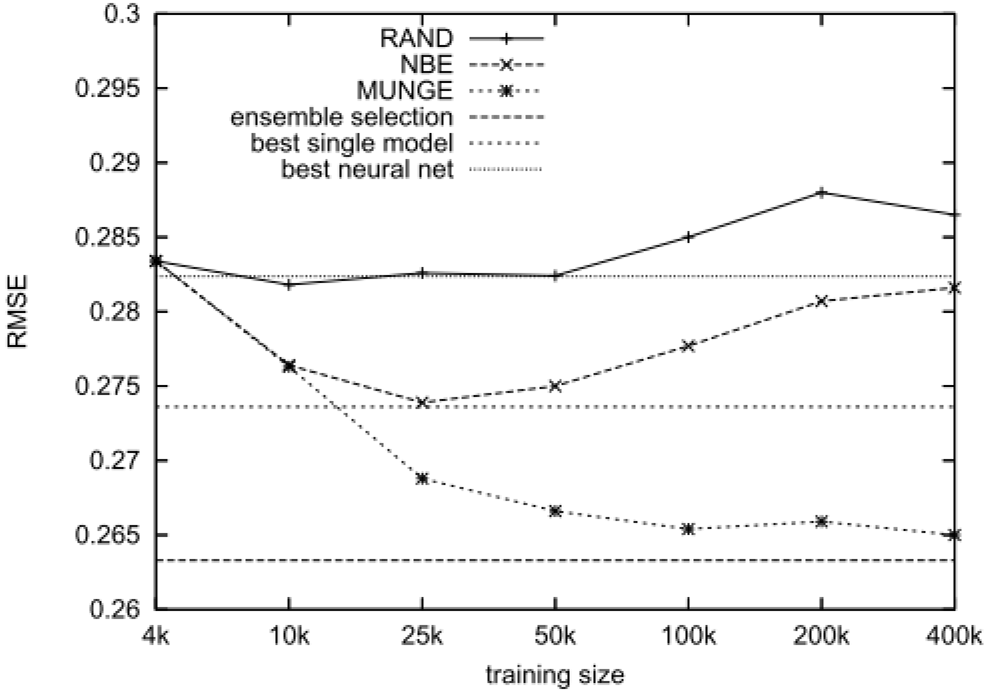


Figure 2: Average perf. over the eight problems.

Do Deep Nets Really Need to be Deep?

Lei Jimmy Ba and Rich Caruana, NIPS, 2014

Shallow feed-forward nets can learn the complex functions previously learned by deep nets and achieve accuracies previously only achievable with deep models.

Training

1. Training a state-of-the-art **deep** model
2. Training a **shallow** model to mimic the deep model

Do Deep Nets Really Need to be Deep?

Lei Jimmy Ba and Rich Caruana, NIPS, 2014

Shallow network architecture*

- convolution layer 128
- pooling layer
- fully connected 1.2K linear units (**Linear layer** to speed up training)
- fully connected 30k non-linear units

Deep network architecture*

128c-p-128c-p-128c-p-1000fc

* Network architectures that were used on CIFAR-10.

Do Deep Nets Really Need to be Deep?

Lei Jimmy Ba and Rich Caruana, NIPS, 2014

Loss

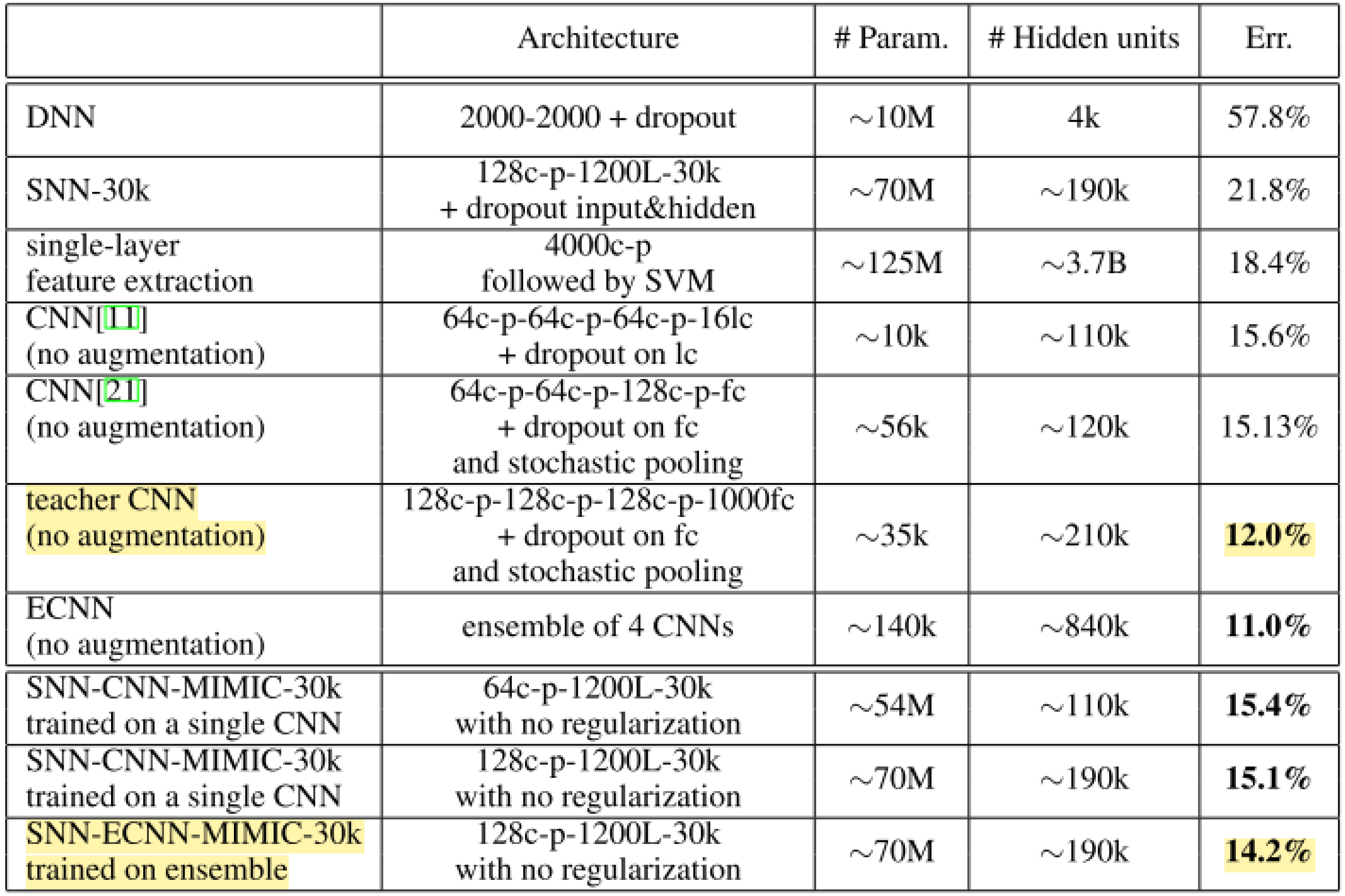
$$\mathcal{L}(W, \beta) = \frac{1}{2T} \sum_t ||g(x^{(t)}; W, \beta) - z^{(t)}||_2^2$$

A lot of information resides in ratios of **logits**.

```
>>> def softmax(Z):
        return np.exp(Z)/np.sum(np.exp(Z))
>>>
>>> softmax([10, 20, 30])
>>> array([  2.06106005e-09,   4.53978686e-05,   9.99954600e-01])
>>>
>>> softmax([-10, 0, 10])
>>> array([  2.06106005e-09,   4.53978686e-05,   9.99954600e-01])
```

Do Deep Nets Really Need to be Deep?

Lei Jimmy Ba and Rich Caruana, NIPS, 2014



	Architecture	# Param.	# Hidden units	Err.
DNN	2000-2000 + dropout	∼10M	4k	57.8%
SNN-30k	128c-p-1200L-30k + dropout input&hidden	∼70M	∼190k	21.8%
single-layer feature extraction	4000c-p followed by SVM	∼125M	∼3.7B	18.4%
CNN[11] (no augmentation)	64c-p-64c-p-64c-p-16lc + dropout on lc	∼10k	∼110k	15.6%
CNN[21] (no augmentation)	64c-p-64c-p-128c-p-fc + dropout on fc and stochastic pooling	∼56k	∼120k	15.13%
teacher CNN (no augmentation)	128c-p-128c-p-128c-p-1000fc + dropout on fc and stochastic pooling	∼35k	∼210k	**12.0%**
ECNN (no augmentation)	ensemble of 4 CNNs	∼140k	∼840k	**11.0%**
SNN-CNN-MIMIC-30k trained on a single CNN	64c-p-1200L-30k with no regularization	∼54M	∼110k	**15.4%**
SNN-CNN-MIMIC-30k trained on a single CNN	128c-p-1200L-30k with no regularization	∼70M	∼190k	**15.1%**
SNN-ECNN-MIMIC-30k trained on ensemble	128c-p-1200L-30k with no regularization	∼70M	∼190k	**14.2%**

Table 2: Comparison of shallow and deep models: classification error rate on CIFAR-10. Key: c, convolution layer; p, pooling layer; lc, locally connected layer; fc, fully connected layer

Distilling the Knowledge in a Neural Network

Geoffrey Hinton et al., 2015

“Distillation” as a way to transfer the knowledge from the cumbersome model to a small model that is **more suitable for deployment**.

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

Using a higher value for T produces a softer probability distribution over classes.

Distilling the Knowledge in a Neural Network

Geoffrey Hinton et al., 2015

Loss

weighted average of two objective functions.

- cross entropy with soft targets
- cross entropy with correct labels (temperature 1) lower weight on this term

Training

1. Train cumbersome model.
2. Transfer knowledge from cumbersome model to small model. **Both model have set up the same high temperature**.

Inference

Small model is using $T = 1$.

Distilling the Knowledge in a Neural Network

Geoffrey Hinton et al., 2015

Cumbersome model

- two hidden layers of 1200 rectified linear hidden units
- dropout

Small model

- two hidden layers of 800 rectified linear hidden units
- no regularization

model type	# error on MNIST
cumbersome	67
small	146
small with KD	74

FitNets: Hints for thin deep nets

Adriana Romero et al., ICLR, 2015

Depth is a fundamental aspect of representation learning

- encourages the reuse of features,
- leads to more abstract and invariant representations at higher layers.

Knowledge transfer layers

- *Hint* layer
- *Guided* layer

The deeper we set the guided layer, the less flexibility we give to the network and, therefore, FitNets are more likely to suffer from over-regularization.

FitNets: Hints for thin deep nets

Adriana Romero et al., ICLR, 2015

Convolutional regressor

- teacher network will usually be wider than the FitNet
- added to guided layer
- less parameters than fully-conected regressor
- approximately the same spatial region of the input image as the teacher hint

Losses

$$\mathcal{L}_{HT}(\mathbf{W}_{\mathbf{Guided}}, \mathbf{W}_{\mathbf{r}}) = \frac{1}{2}||u_h(\mathbf{x}; \mathbf{W}_{\mathbf{Hint}}) - r(v_g(\mathbf{x}; \mathbf{W}_{\mathbf{Guided}}); \mathbf{W}_{\mathbf{r}})||^2$$

$$\mathcal{L}_{KD}(\mathbf{W}_{\mathbf{S}}) = \mathcal{H}(\mathbf{y}_{\mathbf{true}}, \mathrm{P}_{\mathrm{S}}) + \lambda \mathcal{H}(\mathrm{P}_{\mathrm{T}}^{\tau}, \mathrm{P}_{\mathrm{S}}^{\tau})$$

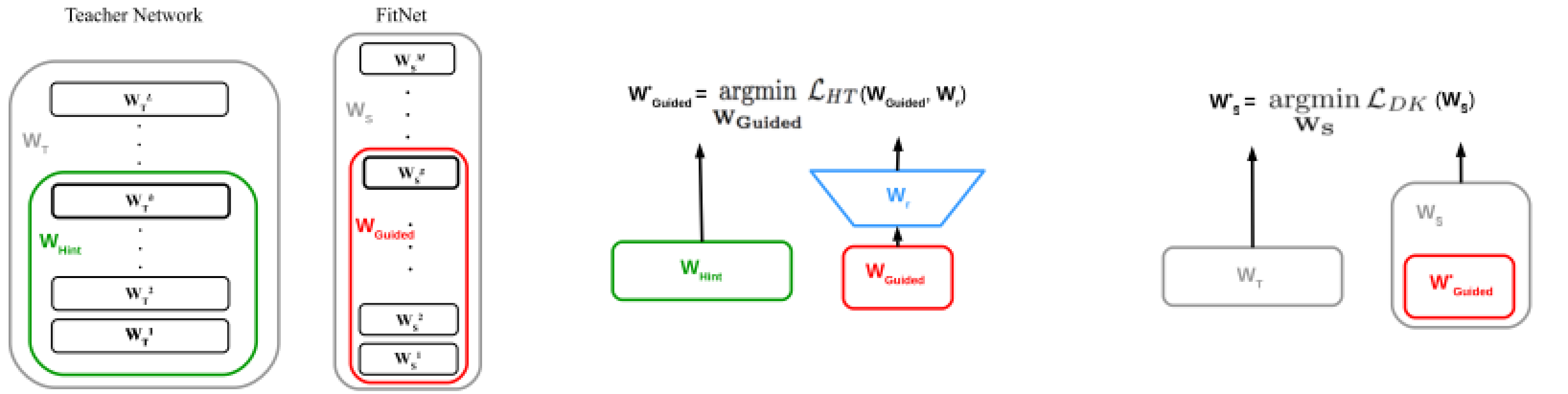
In order to promote the learning of more complex examples (examples with lower teacher confidence), we gradually anneal lambda during the training with a linear decay (resemblance to curriculum learning, Bengio, 2009).

FitNets: Hints for thin deep nets

Adriana Romero et al., ICLR, 2015

Training

1. Train teacher network
2. Train student network through hint layer up to guided layer
3. Knowledge Distillation (as in Distilling the Knowledge in a Neural Network)

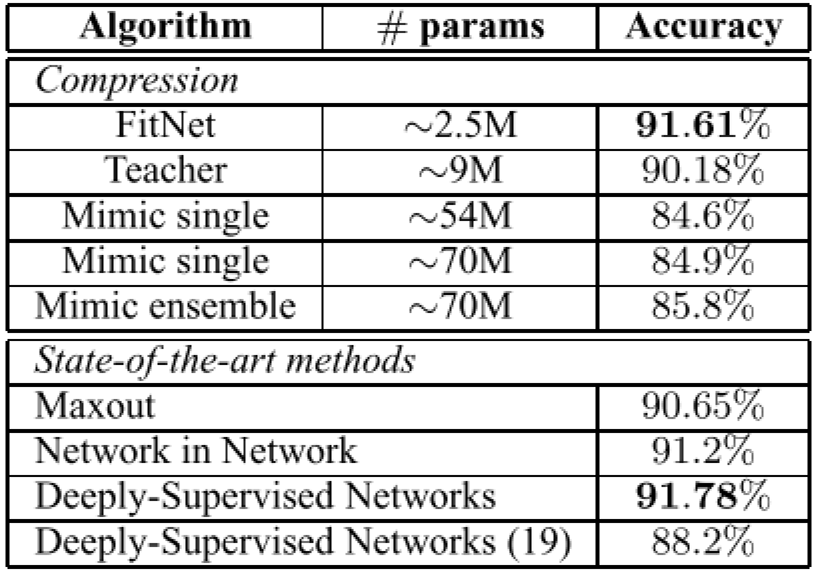


Teacher Network
FitNet
W_T^L
W_T
W_T^h
W_{Hint}
W_T^2
W_T^1
W_S^M
W_S
W_S^g
W_{Guided}
W_S^2
W_S^1
$W^*_{Guided} = \underset{W_{Guided}}{\operatorname{argmin}} \mathcal{L}_{HT}(W_{Guided}, W_r)$
W_r
W_{Hint}
W_{Guided}
$W^*_S = \underset{W_S}{\operatorname{argmin}} \mathcal{L}_{DK}(W_S)$
W_T
W_S
W^*_{Guided}

(a) Teacher and Student Networks (b) Hints Training (c) Knowledge Distillation

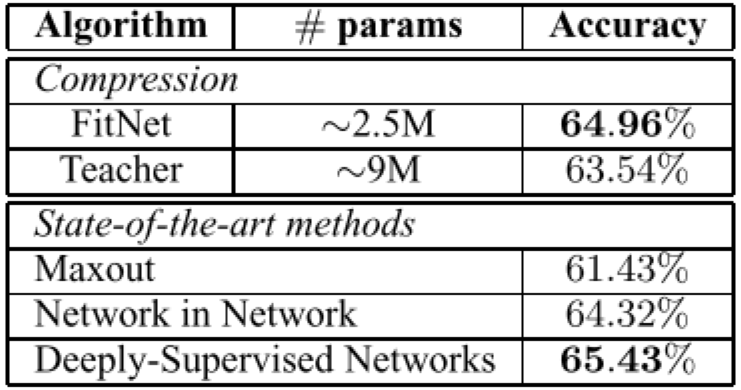
FitNets: Hints for thin deep nets

Adriana Romero et al., ICLR, 2015



Algorithm	# params	Accuracy
Compression		
FitNet	~2.5M	**91.61%**
Teacher	~9M	90.18%
Mimic single	~54M	84.6%
Mimic single	~70M	84.9%
Mimic ensemble	~70M	85.8%
State-of-the-art methods		
Maxout		90.65%
Network in Network		91.2%
Deeply-Supervised Networks		**91.78%**
Deeply-Supervised Networks (19)		88.2%

Table 1: Accuracy on CIFAR-10



Algorithm	# params	Accuracy
Compression		
FitNet	~2.5M	**64.96%**
Teacher	~9M	63.54%
State-of-the-art methods		
Maxout		61.43%
Network in Network		64.32%
Deeply-Supervised Networks		**65.43%**

Table 2: Accuracy on CIFAR-100

Student model outperforms the teacher model, while requiring notably fewer parameters, suggesting that **depth is crucial to achieve better representations**.

A Gift from Knowledge Distillation

Junho Yim et al., CVPR, 2017

Distilled knowledge is transferred in terms of **flow between layers**, which is calculated by computing the inner product between features from two layers.

$$G_{i,j}(x;W) = \sum_{s=1}^{h} \sum_{t=1}^{w} \frac{F^1_{s,t,i}(x;W) \times F^2_{s,t,j}(x;W)}{h \times w}$$

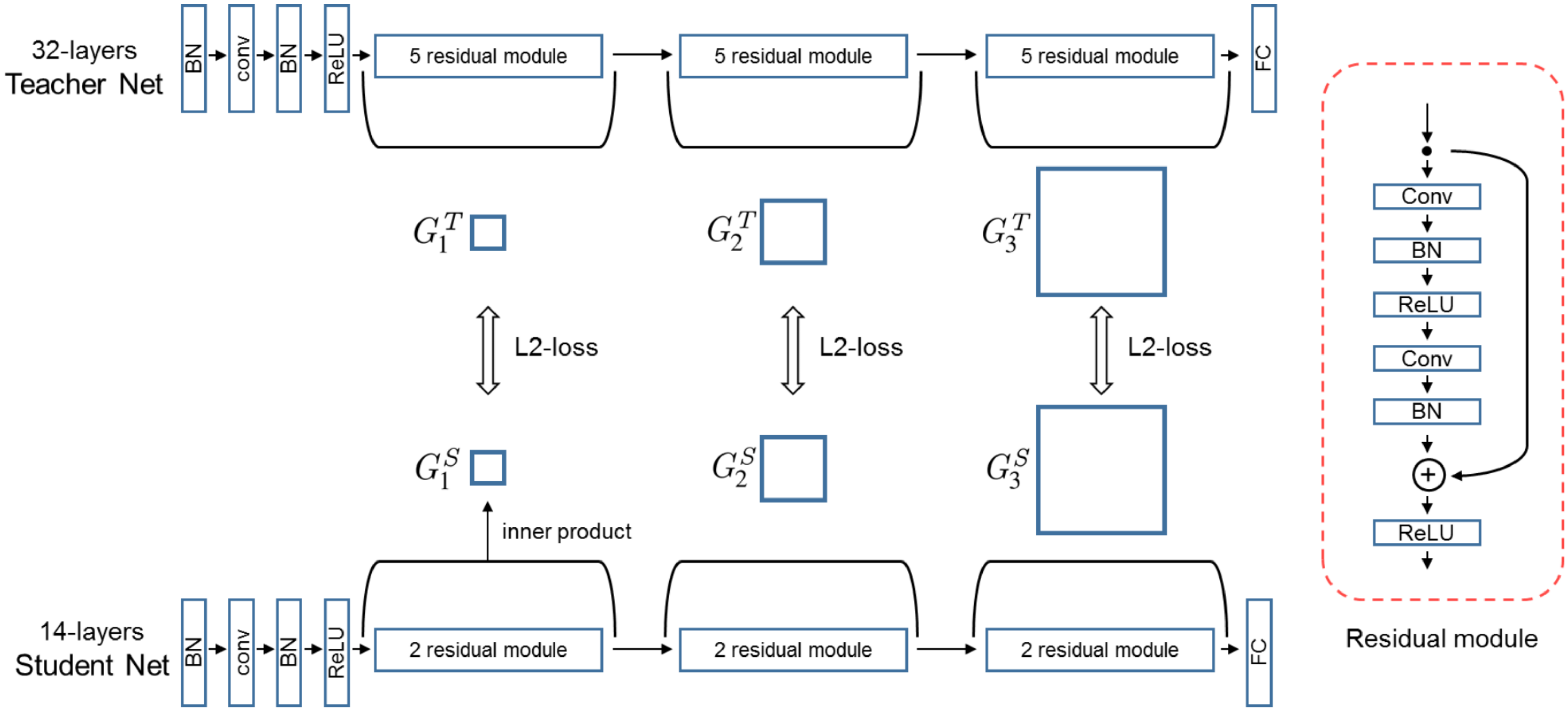


Figure 2. Complete architecture of our proposed method. The numbers of layers of the teacher and student networks can be changed. The FSP matrices are extracted at the three sections that maintain the same spatial size. There are two stages of our proposed method. In stage 1, the student network is trained to minimize the distance between the FSP matrices of the student and teacher networks. Then, the pretrained weights of the student DNN are used for the initial weight in stage 2. Stage 2 represents the normal training procedure.

A Gift from Knowledge Distillation

Junho Yim et al., CVPR, 2017

Training

1. Train teacher network
2. Minimize the loss function LFSP to make the FSP matrix of the student network similar to that of the teacher network
3. Train student network by the main task loss (softmax cross entropy loss)

$$L_{FSP}(W_t, W_s) \\ = \frac{1}{N} \sum_x \sum_{i=1}^{n} \lambda_i \times \|(G_i^T(x; W_t) - G_i^S(x; W_s)\|_2^2$$

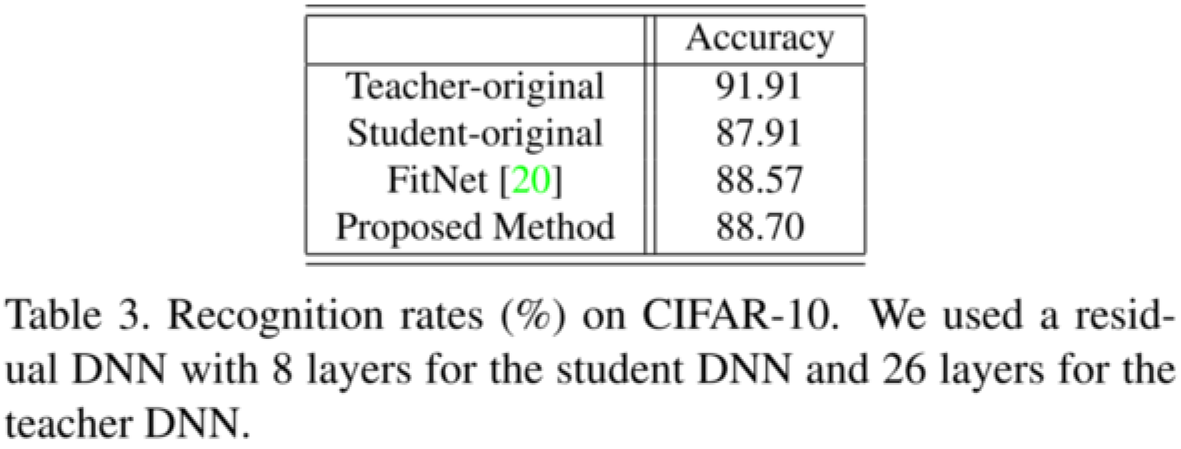
* lambda_i same for all loss terms

A Gift from Knowledge Distillation

Junho Yim et al., CVPR, 2017

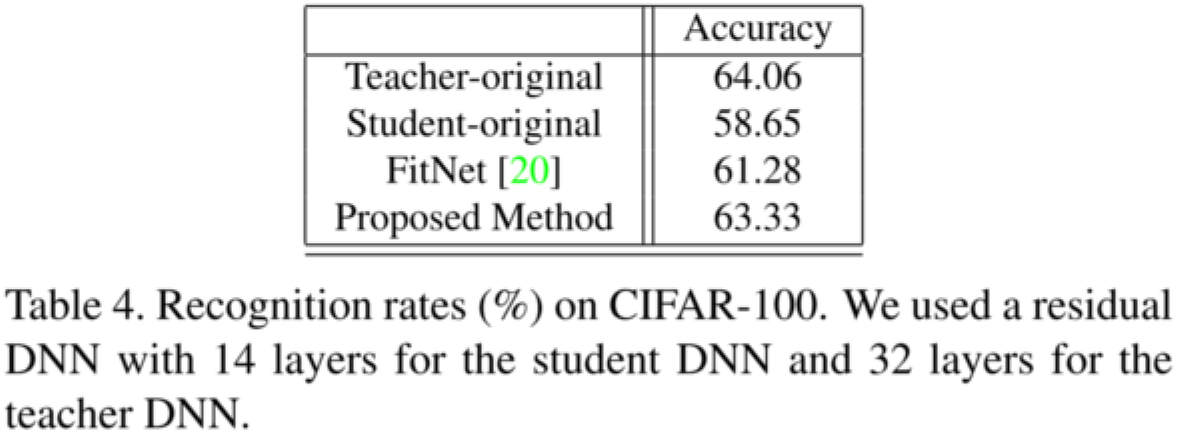
Evaluation

- fast optimization (only third of training steps than for teacher network)
- claim to be better than FitNets, **but** ...



	Accuracy
Teacher-original	91.91
Student-original	87.91
FitNet [20]	88.57
Proposed Method	88.70

Table 3. Recognition rates (%) on CIFAR-10. We used a residual DNN with 8 layers for the student DNN and 26 layers for the teacher DNN.



	Accuracy
Teacher-original	64.06
Student-original	58.65
FitNet [20]	61.28
Proposed Method	63.33

Table 4. Recognition rates (%) on CIFAR-100. We used a residual DNN with 14 layers for the student DNN and 32 layers for the teacher DNN.

References

Deep Learning, 12.1.4 Model Compression, Ian Godfellow et al.

Born Again Trees

Born Again Neural Networks

Model Compression

Do Deep Nets Really Need to be Deep?

Distilling the Knowledge in a Neural Network

FitNets: Hints for thin deep nets

A Gift from Knowledge Distillation