Summary of Paper:
FitNets: Hints For Thin Deep Nets (ICLR 2015)

Muthu Chidambaram

Department of Computer Science, University of Virginia

https://qdata.github.io/deep2Read/

Greedy Layer-Wise Training of Deep Networks (2006)

- Authors: Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle
- Greedy layer-wise unsupervised training can aid optimization by obtaining a good weight initialization
- Deep architectures require exponentially fewer parameters to express similar capacities as shallow architectures

FitNets: Hints For Thin Deep Nets (ICLR 2015)

- Authors: Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, Yoshua Bengio
- Use outputs of teacher network to train deeper student network
- Wide and deep networks are memory/runtime intensive
- Builds off Knowledge Distillation: compresses ensemble of deep networks into a student network of similar depth

FitNets: Hints For Thin Deep Nets (ICLR 2015)

- Literature supports deep architectures for better representation learning
- Recent optimization work has involved guiding intermediate layers
- Extends Knowledge Distillation using intermediate hints

FitNets: Hints For Thin Deep Nets (ICLR 2015)

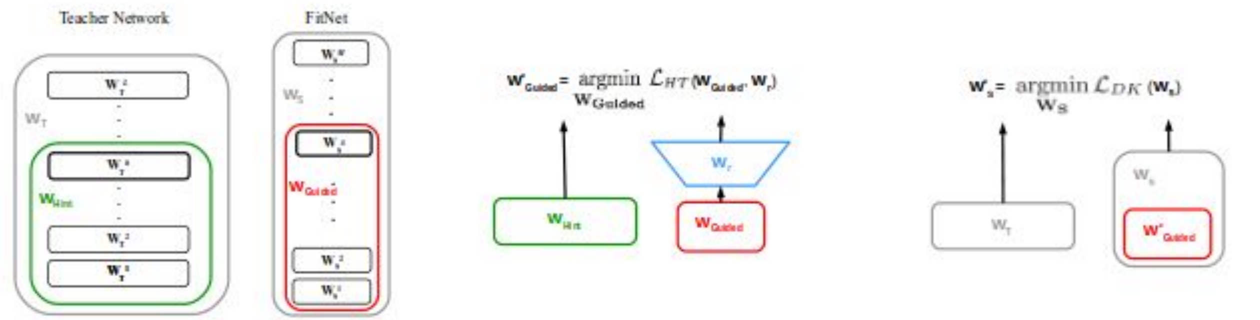
- T is teacher network, S is student network, a_T represents average pre-softmax outputs, Tau is relaxation constant for softening signal
- Hint layer: middle layer of teacher network, guided layer: middle layer of student network
- Train up to guided layer using Lht loss, train after using Lkd loss

$$\mathrm{P}_{\mathrm{T}}^{\tau} = \mathrm{softmax}\left(\frac{\mathbf{a}_T}{\tau}\right), \quad \mathrm{P}_{\mathrm{S}}^{\tau} = \mathrm{softmax}\left(\frac{\mathbf{a}_S}{\tau}\right)$$

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

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

FitNets: Hints For Thin Deep Nets (ICLR 2015)



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

Figure 1: Training a student network using hints.

FitNets: Hints For Thin Deep Nets (ICLR 2015)

Algorithm 1 FitNet Stage-Wise Training.

The algorithm receives as input the trained parameters $\mathbf{W_T}$ of a teacher, the randomly initialized parameters $\mathbf{W_S}$ of a FitNet, and two indices h and g corresponding to hint/guided layers, respectively. Let $\mathbf{W_{Hint}}$ be the teacher's parameters up to the hint layer h. Let $\mathbf{W_{Guided}}$ be the FitNet's parameters up to the guided layer g. Let $\mathbf{W_r}$ be the regressor's parameters. The first stage consists in pre-training the student network up to the guided layer, based on the prediction error of the teacher's hint layer (line 4). The second stage is a KD training of the whole network (line 6).

Input: $\mathbf{W_S}, \mathbf{W_T}, g, h$

Output: $\mathbf{W_S^*}$

1: $\mathbf{W_{Hint}} \leftarrow \{\mathbf{W_T}^1, \ldots, \mathbf{W_T}^h\}$

2: $\mathbf{W_{Guided}} \leftarrow \{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\}$

3: Intialize $\mathbf{W_r}$ to small random values

4: $\mathbf{W^*_{Guided}} \leftarrow \underset{\mathbf{W_{Guided}}}{\operatorname{argmin}} \mathcal{L}_{HT}(\mathbf{W_{Guided}}, \mathbf{W_r})$

5: $\{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\} \leftarrow \{\mathbf{W_{Guided}}^{*1}, \ldots, \mathbf{W_{Guided}}^{*g}\}$

6: $\mathbf{W_S^*} \leftarrow \underset{\mathbf{W_S}}{\operatorname{argmin}} \mathcal{L}_{KD}(\mathbf{W_S})$

FitNets: Hints For Thin Deep Nets (ICLR 2015)

- Hint-based training with knowledge distillation can be seen as curriculum learning
- Student-teacher model is a generic curriculum learning approach
    - Decay lambda in loss to decrease influence of easier examples (ones teacher has high degree of confidence in)
- Tested on CIFAR-10, CIFAR-100, SVHN, MNIST, AFLW

FitNets: Hints For Thin Deep Nets (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

FitNets: Hints For Thin Deep Nets (ICLR 2015)

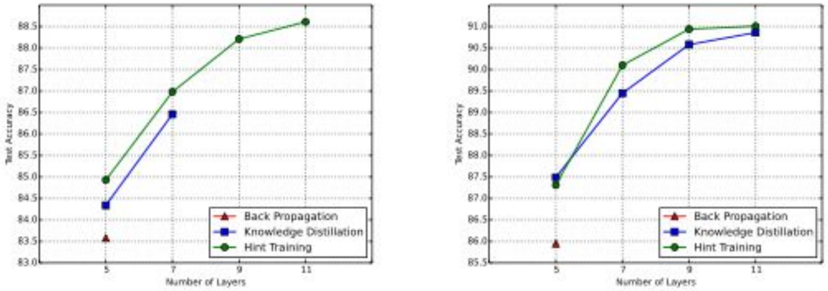
Algorithm	# params	Misclass
Compression		
FitNet	~1.5M	2.42%
Teacher	~4.9M	**2.38%**
State-of-the-art methods		
Maxout		2.47%
Network in Network		2.35%
Deeply-Supervised Networks		**1.92%**

Table 3: SVHN error

Algorithm	# params	Misclass
Compression		
Teacher	~361K	0.55%
Standard backprop	~30K	1.9%
KD	~30K	0.65%
FitNet	~30K	**0.51%**
State-of-the-art methods		
Maxout		0.45%
Network in Network		0.47%
Deeply-Supervised Networks		**0.39%**

Table 4: MNIST error

FitNets: Hints For Thin Deep Nets (ICLR 2015)



(a) 30M Multiplications (b) 107M Multiplications

Figure 2: Comparison of Standard Back-Propagation, Knowledge Distillation and Hint-based Training on CIFAR-10.

Network	# layers	# params	# mult	Acc	Speed-up	Compression rate
Teacher	5	~9M	~725M	90.18%	1	1
FitNet 1	11	~250K	~30M	89.01%	**13.36**	**36**
FitNet 2	11	~862K	~108M	91.06%	4.64	10.44
FitNet 3	13	~1.6M	~392M	91.10%	1.37	5.62
FitNet 4	19	~2.5M	~382M	**91.61%**	1.52	3.60

Table 5: Accuracy/Speed Trade-off on CIFAR-10.

FitNets: Hints For Thin Deep Nets (ICLR 2015)

- Hint-based Training can be used to provide better initialization for optimization
- Difference between KD and HT: HT provides a “starting point” in the parameter space using hints
- Conclusion: HT provides a means of compressing networks by more than 10x while maintaining accuracy

References

- http://arxiv.org/pdf/1412.6550v4.pdf
-