FITNETS:
HINTS FOR THIN DEEP NETS

2022/03/15

守山　慧

論文情報

- 会議
  - ICLR 2015
- 著者
  - Adriana Romero（University of Barcelona）
  - Nicolas Ballas（University of Montreal）
  - Samira Ebrahimi Kahou（Ecole Polytechnique of Montreal）
  - Antoine Chassang（University of Montreal）
  - Carlo Gatta（Centre de Visi´o per Computador）
  - Yoshua Bengio（University of Montreal，CIFAR Senior Fellow）

背景

- モデルの隠れ層の深さはパフォーマンスの向上において重要
  a. 画像分類や物体検知でSOTAを達成したモデルの隠れ層は深い

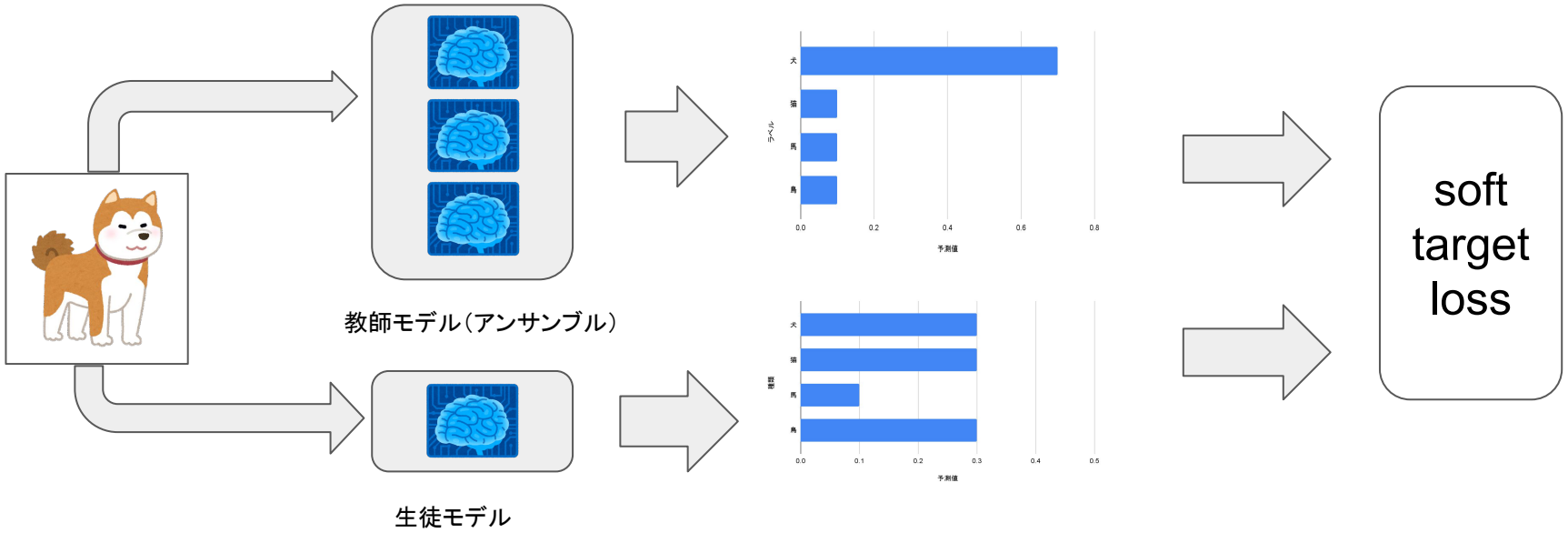
- 層を深くすると実用的に以下の課題が生じる

  課題1：学習時にさまざまな工夫が必要

  課題2：推論時に計算時間と計算リソースを必要とする

背景

- 課題2を解決するために大きな教師モデルから小さな生徒モデルに知識を移す手法が提案された
  - 教師モデル：ネットワークが広く深いモデルまたはアンサンブルモデル
  - 生徒モデル：教師モデルより狭く浅いモデル



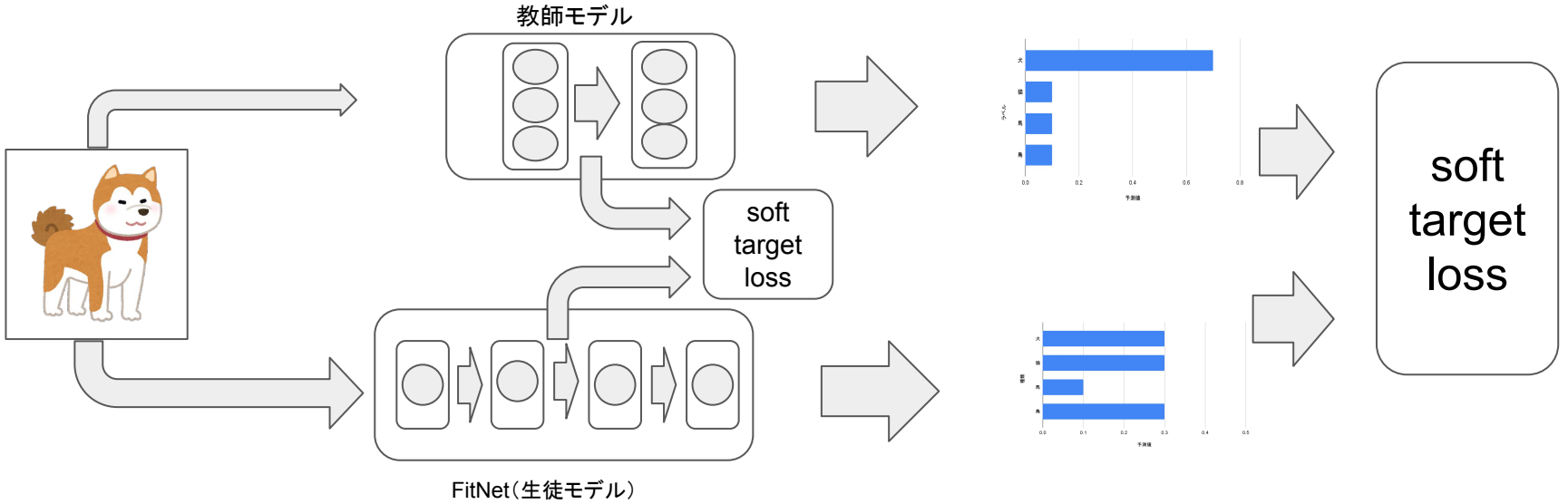
背景

- 生徒モデルが教師モデルと同等のパフォーマンスを達成

- この手法では層を深くすることによる利点を捨てていることになる
- 層を深くする利点
  - 表現学習において
    - より抽象的で安定した表現が得られる
  - 経験的に
    - SOTAを達成しているImageNetのレイヤー数は19と22になっていて深層になっている
  - 理論的に
    - 中間表現の幅が浅い層よりも深い層の方が広くなる

提案手法

- 教師モデルよりも隠れ層が深く，パラメータの数が少ない生徒モデルに知識を移す手法を提案
  - 生徒モデルのことをFitNetという
  - 教師モデルの隠れ層の出力を学習のヒントとして活用する



提案手法

- 学習アルゴリズム

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: $\mathbf{W^*_{Guided}} \leftarrow \underset{\mathbf{W_{Guided}}}{\text{argmin}}\ \mathcal{L}_{HT}(\mathbf{W_{Guided}}, \mathbf{W_r})$
5: $\{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\} \leftarrow \{\mathbf{W_{Guided}}^{*1}, \ldots, \mathbf{W_{Guided}}^{*g}\}$
6: $\mathbf{W_S^*} \leftarrow \underset{\mathbf{W_S}}{\text{argmin}}\, \mathcal{L}_{KD}(\mathbf{W_S})$

提案手法

- 学習アルゴリズム

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 guided layer g.
pre-training the student network up
hint layer (line 4). The second stag

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: $\mathbf{W^*_{Guided}} \leftarrow \underset{\mathbf{W_{Guided}}}{\text{argmin}}\ \mathcal{L}_{HT}(\mathbf{W_{Guided}}, \mathbf{W_r})$
5: $\{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\} \leftarrow \{\mathbf{W_{Guided}}^{*1}, \ldots, \mathbf{W_{Guided}}^{*g}\}$
6: $\mathbf{W_S^*} \leftarrow \underset{\mathbf{W_S}}{\text{argmin}}\ \mathcal{L}_{KD}(\mathbf{W_S})$

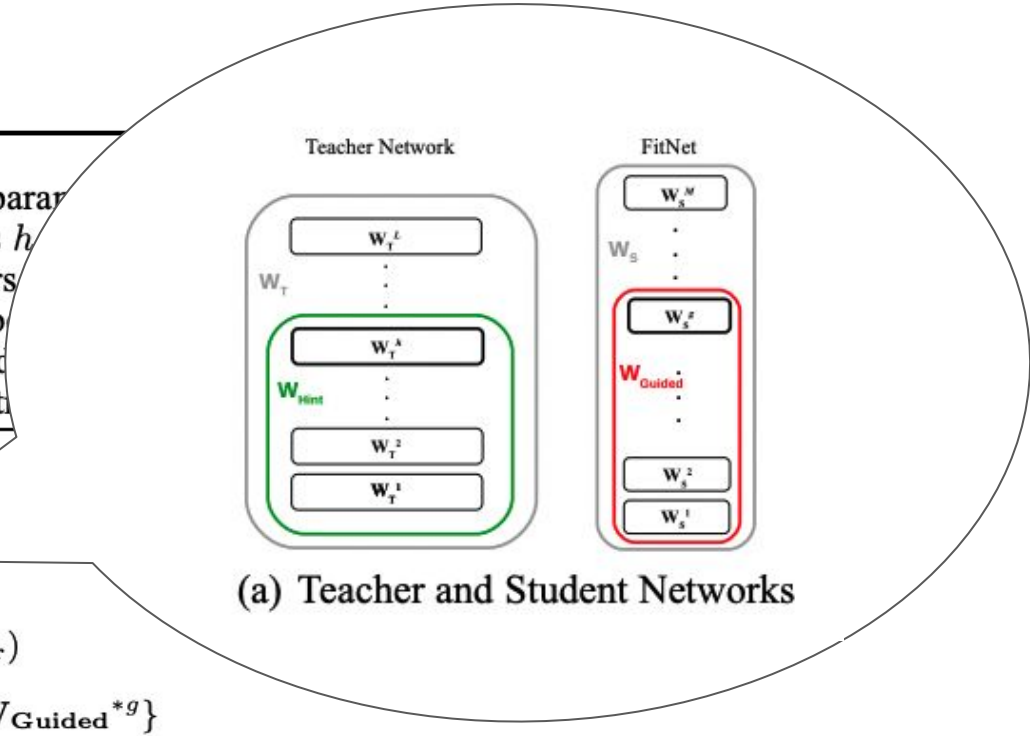
W_T：教師モデルの隠れ層のパラメータ
W_s：生徒モデルの隠れ層のパラメータ
h：ヒントとして使う教師モデルの隠れ層の層数
g：ヒントを使って調整される生徒モデルの隠れ層の層数

提案手法

- 学習アルゴリズム

Algorithm 1 FitNet Stage-Wise Training.
The algorithm receives as input the trained para
parameters $\mathbf{W_S}$ of a FitNet, and two indices h
tively. Let $\mathbf{W_{Hint}}$ be the teacher's parameters
parameters up to the guided layer g. Let $\mathbf{W_r}$ b
pre-training the student network up to the guid
hint layer (line 4). The second stage is a KD t

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: $\mathbf{W^*_{Guided}} \leftarrow \underset{\mathbf{W_{Guided}}}{\text{argmin}}\ \mathcal{L}_{HT}(\mathbf{W_{Guided}}, \mathbf{W_r})$
5: $\{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\} \leftarrow \{\mathbf{W_{Guided}}^{*1}, \ldots, \mathbf{W_{Guided}}^{*g}\}$
6: $\mathbf{W_S^*} \leftarrow \underset{\mathbf{W_S}}{\text{argmin}}\ \mathcal{L}_{KD}(\mathbf{W_S})$



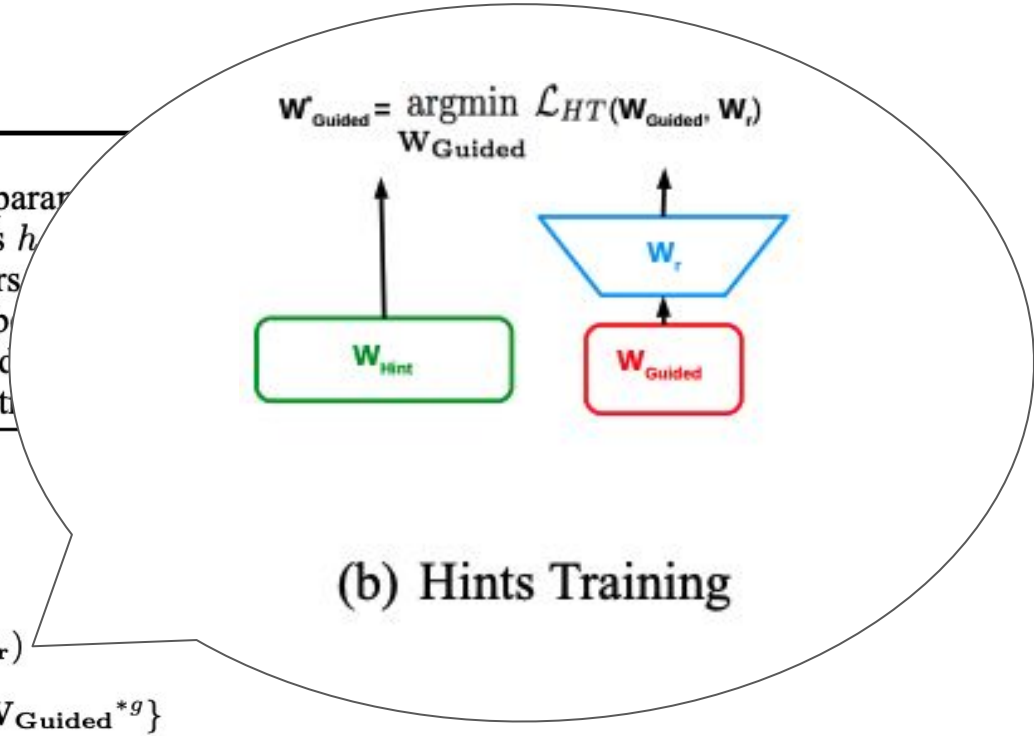
(a) Teacher and Student Networks

提案手法

- 学習アルゴリズム

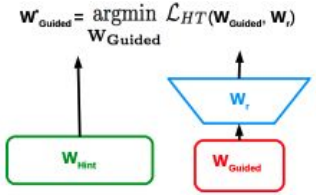
Algorithm 1 FitNet Stage-Wise Training.
The algorithm receives as input the trained para
parameters $\mathbf{W_S}$ of a FitNet, and two indices h
tively. Let $\mathbf{W_{Hint}}$ be the teacher's parameters
parameters up to the guided layer g. Let $\mathbf{W_r}$ b
pre-training the student network up to the guid
hint layer (line 4). The second stage is a KD t

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: $\mathbf{W^*_{Guided}} \leftarrow \underset{\mathbf{W_{Guided}}}{\text{argmin}}\ \mathcal{L}_{HT}(\mathbf{W_{Guided}}, \mathbf{W_r})$
5: $\{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\} \leftarrow \{\mathbf{W_{Guided}}^{*1}, \ldots, \mathbf{W_{Guided}}^{*g}\}$
6: $\mathbf{W_S^*} \leftarrow \underset{\mathbf{W_S}}{\text{argmin}}\ \mathcal{L}_{KD}(\mathbf{W_S})$



提案手法

- FitNet（生徒モデル）の一部のパラメータを更新する
- 式3を最小化するように生徒モデルの隠れ層のパラメータを調整する
  - 教師モデルの中間表現を真似するように調整



(b) Hints Training

$$\mathcal{L}_{HT}(\mathbf{W_{Guided}}, \mathbf{W_r}) = \frac{1}{2}||u_h(\mathbf{x}; \mathbf{W_{Hint}}) - r(v_g(\mathbf{x}; \mathbf{W_{Guided}}); \mathbf{W_r})||^2, \quad (3)$$

x：モデルに入力するベクトル

W_{Guided}：ヒントによって調整される生徒モデルの隠れ層のパラメータ

W_{Hint}：ヒントに使う教師モデルの隠れ層のパラメータ

r：モデルの出力の次元数を合わせるための回帰関数

W_r：回帰関数のパラメータ

u_h：入力ベクトル x に対してパラメータ W_{Hint}による出力を計算する関数

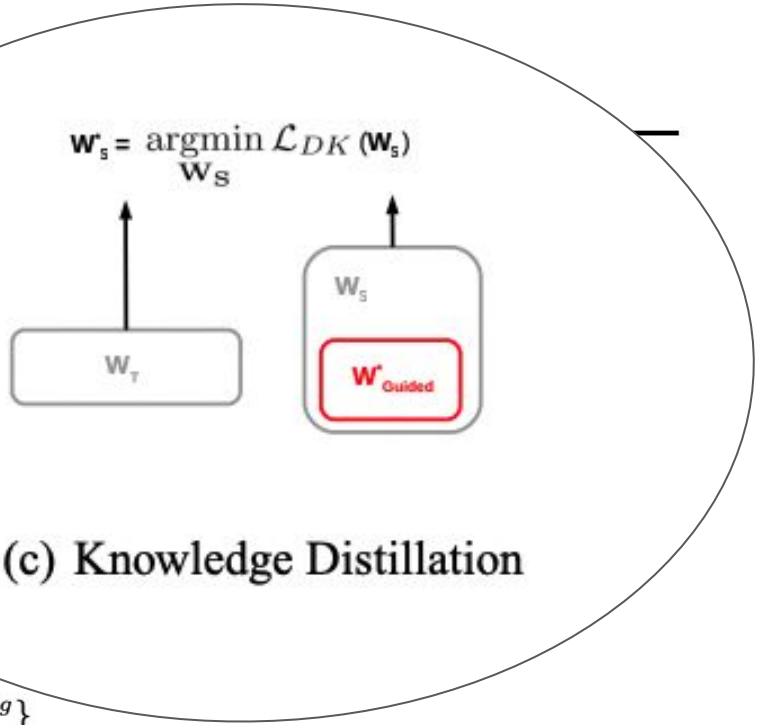
v_g：入力ベクトル x に対してパラメータ W_{Guided}による出力を計算する関数

提案手法

- 学習アルゴリズム

Algorithm 1 FitNet Stage-Wise Training
The algorithm receives as input the tra
parameters $\mathbf{W_S}$ of a FitNet, and two
tively. Let $\mathbf{W_{Hint}}$ be the teacher's p
parameters up to the guided layer g.
pre-training the student network up
hint layer (line 4). The second stage

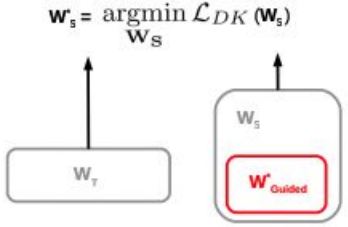
Input: $\mathbf{W_S}, \mathbf{W_T}, g, h$
Output: $\mathbf{W_S^*}$
1: $\mathbf{W_{Hint}} \leftarrow \{\mathbf{W_T}^1, \ldots, \mathbf{W_T}^h\}$
2: $\mathbf{W_{Guided}} \leftarrow \{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\}$
3: Intialize $\mathbf{W_r}$ to small random values
4: $\mathbf{W^*_{Guided}} \leftarrow \underset{\mathbf{W_{Guided}}}{\text{argmin}}\ \mathcal{L}_{HT}(\mathbf{W_G}$
5: $\{\mathbf{W_S}^1, \ldots, \mathbf{W_S}^g\} \leftarrow \{\mathbf{W_{Gui}}$ $, \ldots, \mathbf{W_{Guided}}^{*g}\}$
6: $\mathbf{W_S^*} \leftarrow \underset{\mathbf{W_S}}{\text{argmin}}\ \mathcal{L}_{KD}(\mathbf{W_S})$



(c) Knowledge Distillation

提案手法

- FitNet（生徒モデル）全体のパラメータを調整する
  - 既存手法の知識蒸留（KD）を使う



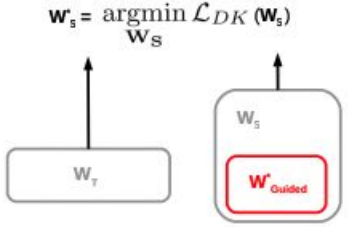
(c) Knowledge Distillation

$$\mathrm{P}_{\mathrm{T}}^{\tau}=\mathrm{softmax}\left(\frac{\mathbf{a}_T}{\tau}\right),\quad \mathrm{P}_{\mathrm{S}}^{\tau}=\mathrm{softmax}\left(\frac{\mathbf{a}_S}{\tau}\right). \tag{1}$$

a_T：教師モデルの入力ベクトル x に対する出力
a_S：生徒モデルの入力ベクトル x に対する出力
P_T^{τ}：温度τで正規化した教師モデルの出力
R_S^{τ}：温度τで正規化した生徒モデルの出力
τ：確率分布を調整するためのハイパーパラメータ（温度）

提案手法

- FitNet（生徒モデル）全体のパラメータを調整する
  - 式2を最小化するように更新する
  - FitNetが正解ラベルを予測できるようにするために行う



(c) Knowledge Distillation

$$\mathcal{L}_{KD}(\mathbf{W_S}) = \mathcal{H}(\mathbf{y_{true}}, \mathrm{P_S}) + \lambda \mathcal{H}(\mathrm{P_T^\tau}, \mathrm{P_S^\tau}), \tag{2}$$

P_T^τ：温度τで正規化した教師モデルの出力

R_S^τ：温度τで正規化した生徒モデルの出力

P_S：温度$\tau = 1$で正規化した生徒モデルの出力 $(P_S = softmax(a_s))$

y_{true}：入力ベクトル x に対する正解ラベル

λ：重み

実験

- ベンチマークデータセットによる評価
  - 使用するデータセット（全て画像分類のタスク）
    - CIFAR-10，CIFAR-100
    - SVHN
    - MNIST
    - AFLW+ImageNet
- 結果の分析
  - ネットワークを深くして行った時の学習の影響
  - モデルのパフォーマンスと効率の比較

実験：ベンチマークデータセットによる評価

- CIFAR-10
  - 10クラスのカラー画像の分類をする

- 教師モデル：畳み込み＋maxoutレイヤーを3層繋げたモデル
- FitNet（生徒モデル）：畳み込み＋maxoutレイヤーを17層繋げたモデル
  - FitNetのパラメータ数は教師モデルの⅓
- FitNetの11層目の出力が教師モデルの2層目の出力に近づくようにヒントを与える

実験：ベンチマークデータセットによる評価

- CIFAR-10
- FitNetが教師モデルよりもいい精度を出している
- 既存手法よりもいいパフォーマンスを出している
- 隠れ層の深さがパフォーマンスを向上させていることがわかる

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

実験：ベンチマークデータセットによる評価

- 実験設定
  - CIFAR-100
    - 100クラスのカラー画像分類
  - モデルの構造や学習方法はCIFAR-10と同じ

- 結果
  - FitNetの方が教師モデルよりも精度が良い
  - SOTAに匹敵する精度を出した

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

実験：ベンチマークデータセットによる評価

- SVHNデータセット
  - 32×32のフルカラー数字画像データセット
- 教師モデル：畳み込み＋maxoutレイヤーを2層繋げたモデル
- FitNet（生徒モデル）：畳み込み＋maxoutレイヤーを11層繋げたモデル
  - FitNet（生徒モデル）のパラメータ数は教師モデルの32%

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

- FitNetは教師モデルとほとんど同じ精度を達成

実験：ベンチマークデータセットによる評価

- MNIST：10クラスの手書き数字データセット
- 目的
  - 他の手法と比較して提案手法の有効性を検証
  - ヒントを与えることの有効性を検証

- 教師モデル：畳み込み+maxoutレイヤーを2層重ねたモデル
- FitNet（生徒モデル）：畳み込み+maxoutレイヤーを4層重ねたモデル
  - FitNetのパラメータ数は教師モデルの8%
  - FitNetの4層目の出力に教師モデルの2層目の出力をヒントとして与えた

実験：ベンチマークデータセットによる評価

- MNIST
- FitNet（生徒モデル）を以下の3種類の学習方法で学習
  - 誤差逆伝播法(BP)
  - 知識蒸留(KD)
  - 提案手法(HT)

- HTで学習したモデルが一番精度がよかった

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

実験：ベンチマークデータセットによる評価

- AFLW+ImageNetデータセット
  - 顔画像の判定を行う
  - AFLWデータセットは顔認識のためのデータセット
    - AFLWデータセットから顔だけを切り抜いた画像25000枚
    - ImageNetデータセットから顔が映っていない画像25000枚
- 目的：モデルの構造を変えてもHTがうまくいくことを検証したい

- 教師モデル：畳み込み+ReLUを3層繋げたモデル
- FitNet1（生徒モデル1）：畳み込み+ReLUを7層繋げたモデル
  - 教師モデルの6.6%のパラメータ数
- FitNet2（生徒モデル2）：畳み込み+ReLUを7層繋げたモデル
  - 教師モデルの40%のパラメータ数

実験：ベンチマークデータセットによる評価

- AFLW+ImageNetデータセット
  - FitNet（生徒モデル）を知識蒸留（KD）と提案手法（HT）で学習させた
- KDよりHTの方が有効であることがわかる
- HTが他のモデルでも有効になっていることがわかる

モデル	誤分類率
教師モデル	4.21%
FitNet1（KD）	4.58%
FitNet1（HT）	2.55%
FitNet2（KD）	1.95%
FitNet2（HT）	1.85%

実験：結果の分析

- 分析１：層が深いネットワークの学習
  - ネットワークの計算量を制限してモデルのパフォーマンスを評価
  - 既存の学習手法と提案手法の比較

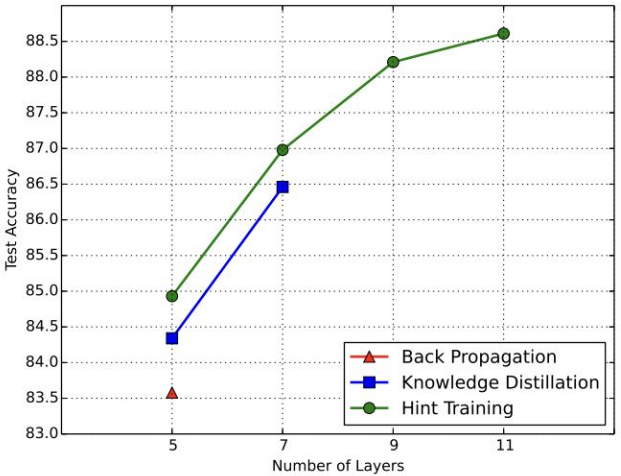
- 分析２：モデルのパフォーマンスと効率の比較
  - パラメータ数を減らした時の精度と推論時間を比較

実験：結果の分析1

- 実験設定
  - 計算量の制限がある中で生徒モデルの層を深くしていき，パフォーマンスを比較
  - 使用するデータセットはCIFAR-10
  - 教師モデル：畳み込み+maxoutレイヤーを3層繋げたモデル
  - FitNet（生徒モデル）：畳み込み+maxoutレイヤーを繋げたモデル
- FitNetの学習方法
  - 誤差逆伝播法（BP：教師モデルを使わずに学習）
  - 知識蒸留（KD：ヒントを使わずに学習）
  - 提案手法（HT)

実験：結果の分析1

- 結果：計算量が30Mの時
  - BPでは5層より深くすると，学習がうまくいかなかった
  - KDでは7層より深くした時，学習がうまくいかなかった
  - HTでは13層まで深くしても学習できた
- 提案手法（HT）が他の手法と比べて最適化に適している

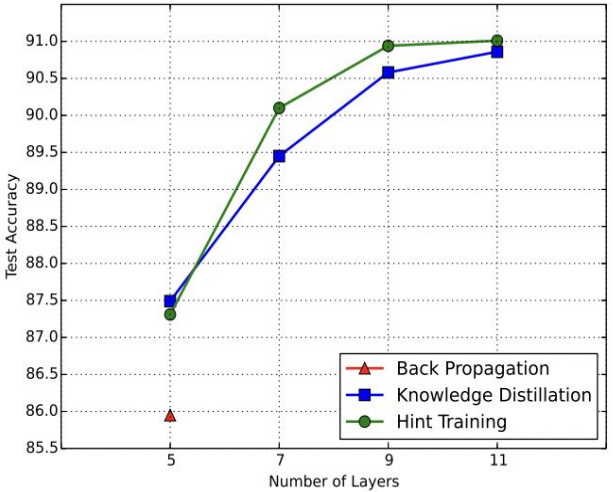


(a) 30M Multiplications

実験：結果の分析1

- 結果：計算量が107Mの時
  - BPでは5層より深くすると，学習がうまくいかなかった
  - 7層の時，HTよりKDの方が精度がよかった
- KDよりHTの方がいい正規化をしていると言える

- この実験から以下のことが言える
  - モデルの層が深くなるほど，HTはBPやKDより有効
  - 計算量が制限されている中では，層が深いモデルの方がパフォーマンスがいい



(b) 107M Multiplications

実験：結果の分析2

- 実験設定
  - 生徒モデルの推論時間とパラメータの圧縮率を比較
  - 推論に使うデータセットはCIFAR-10のテストデータ

- 使うモデル
  - 教師モデル：畳み込み+maxoutレイヤーを5層繋げたモデル
  - FitNet1（生徒モデル）：畳み込み+maxoutレイヤーを11層繋げたモデル
  - FitNet2：畳み込み+maxoutレイヤーを11層繋げたモデル
  - FitNet3：畳み込み+maxoutレイヤーを13層繋げたモデル
  - FitNet4：畳み込み+maxoutレイヤーを19層繋げたモデル

実験：結果の分析2

- 全てのFitNetで推論速度が向上した
- FitNet1では大きくパラメータ数を減らしたが，教師モデルに匹敵する精度を達成
- それ以外のFitNetでは，高速化と精度の向上が達成できた
- モデルの層を深くして，パラメータの数を減らしても，教師モデルよりも早く正確になっている

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.

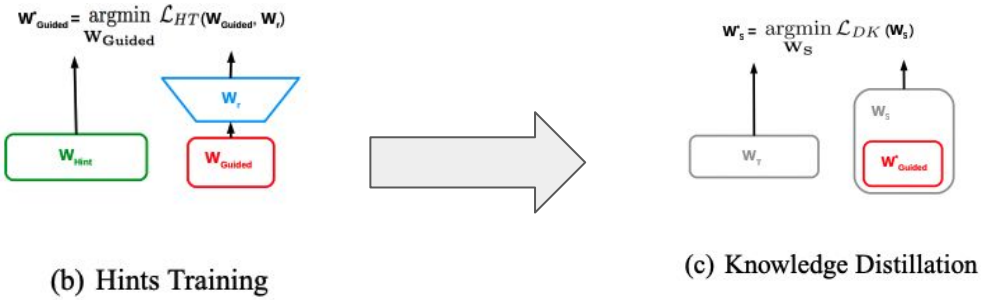
議論

- ヒントレイヤーの選び方がたまたま良いものだったのでは？
- 出力について直接役に立つようなヒントを与えればいいのでは？

- ヒントとして使う値を変化させて検証
  - 理想的な出力を得るためのヒントを与える
  - ヒントと予測ラベルについて同時に最適化する
  - 全てのレイヤーにヒントを与えて学習させる
  - 中間表現のクラス分類をヒントとして用いる

議論

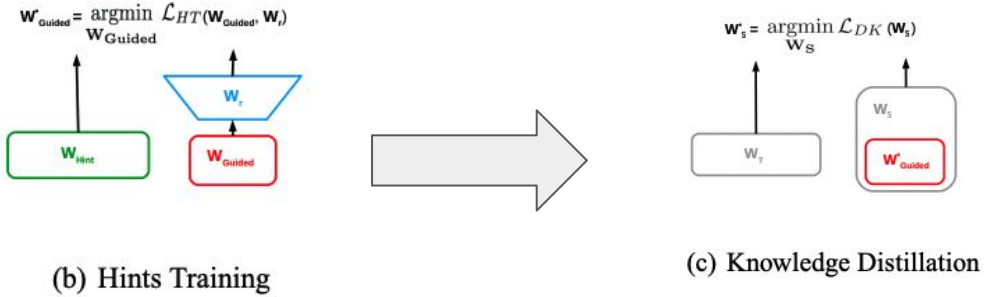
- HTでの学習がKDでの学習の役に立っているとは限らない
  - ヒントで調整したレイヤーは局所解になるように初期化されている
  - この初期化が2段階目の学習（KD）に置いて役に立つとは限らないため
- 理想的な出力を得るためのヒントを追加で与えた
  - 学習はうまくいかなかった



(b) Hints Training　　(c) Knowledge Distillation

議論

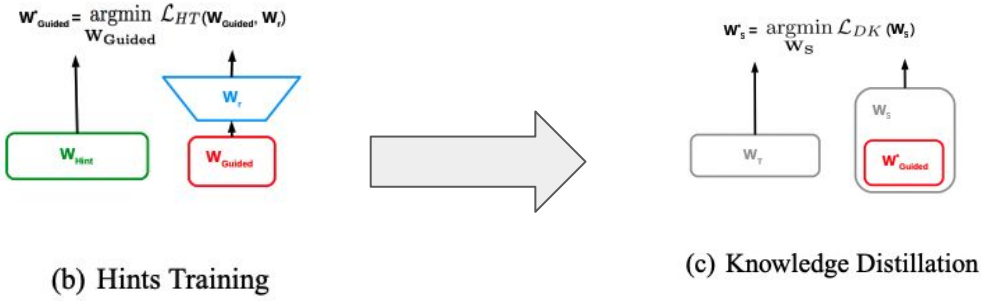
- 教師あり事前学習という観点から
  - (b)の段階で隠れ層を事前学習しているのと同じだと言える
  - 予測をするためにより多くの隠れ層や非線形関数を通すので
  - 入力の特徴を隠れ層が捨ててしまうことがある
- →(b)の最適化が(c)の最適化の助けになるとは限らない



(b) Hints Training　　(c) Knowledge Distillation

議論

- ヒントと予測ラベルについて同時に最適化する
  - ヒントとして教師モデルの隠れ層の出力の合計値を用いる
    - 出力を合計する隠れ層の組み合わせを色々試す
  - FitNetの出力層にヒントを与える
    - 隠れ層にもヒントを追加で与えた
- 学習がうまくいくような隠れ層の組み合わせは見つけることができなかった



(b) Hints Training　　(c) Knowledge Distillation

議論

- 全てのレイヤーにヒント（中間表現のクラス分類）を与えて学習させた
  - Deeply Supervised Networks（DSN）を使って最適化
  - 19層のモデルの時精度が88.2%
  - FitNetよりも精度が低い
- クラス分類に直接役に立つヒントは正規化として強すぎる

Algorithm	# params	Accuracy
Compression		
FitNet	~2.5M	**91.61%**
State-of-the-art methods		
Maxout		90.65%
Network in Network		91.2%
Deeply-Supervised Networks		**91.78%**
Deeply-Supervised Networks (19)		88.2%

DSNのCIFAR-10に対する精度

まとめ

- 教師モデルよりも層が深く，パラメータの少ないモデルに知識を移す手法を提案
  - 教師モデルの隠れ層の出力を学習のヒントとして活用する

- 既存手法よりも最適化，高速化の点で優れている

- ヒントとして教師モデルの中間層の出力の方がラベルよりも優れていることを示した