知識の蒸留

～モデルの圧縮・高速化～

2019/1/5

眞野　宏伸

ディープラーニングの課題

- ディープラーニング技術の発展

  画像、自然言語、音声等のさまざまな領域において、旧来手法をはるかに凌ぐ性能を示すモデルを学習することが可能。

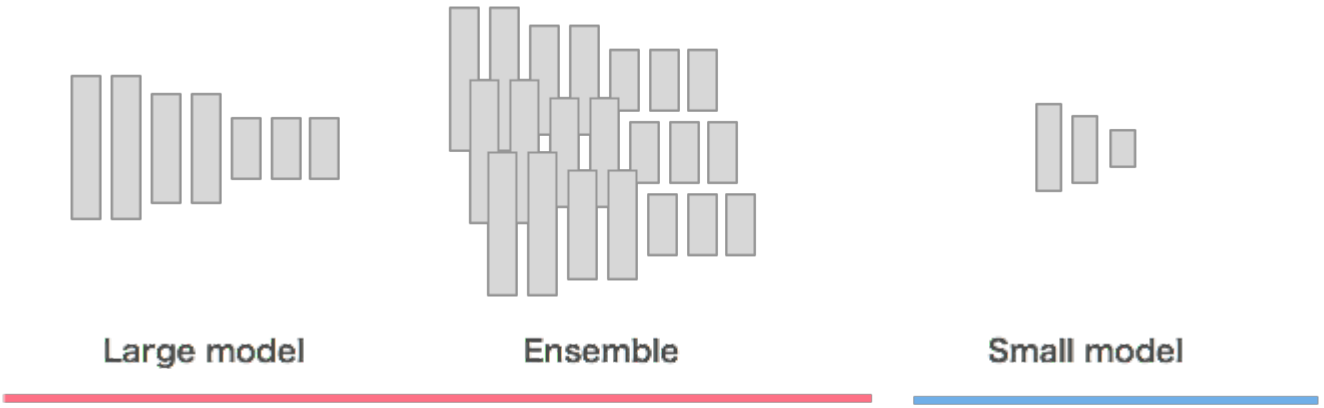
実応用する際には大きな課題

- 大規模なデータに対して、大規模なモデルを学習させる
- モデルをより深く、より大きくすることで、性能が向上

大規模な計算リソースが必要

ディープラーニングの課題

- 精度 VS 処理速度



精度 - 高い
計算コスト - 大きい

精度 - 低い
計算コスト - 小さい

モデルが小さいと精度が低い

ディープラーニングの課題

例として

- 画像認識モデルをスマートフォンアプリで動かす
- 画像認識モデルを移動型ロボットに搭載して高度な制御を実現する
- 音声認識・生成モデルをスマートスピーカに搭載する

デバイスは小型、計算リソースに制約

モデルの圧縮が必要

モデル圧縮方法

- 効率的なマイクロアーキテクチャ

  訓練に先立って、精度と計算のトレードオフを改善するように、モデルの構成要素を工夫する

- 量子化

  訓練後に、重みの浮動小数点数の精度を下げる

- 低ランク近似

  訓練後の重みを特異値分解により、低ランク近似する

- Pruning

  訓練後に、寄与度の小さい重み、チャネル、レイヤーを削除する

- 蒸留

  一度訓練したモデルが学習した知識を、別の軽量なモデルに継承させる

モデル圧縮方法

- 効率的なマイクロアーキテクチャ

  訓練に先立って、精度と計算のトレードオフを改善するように、モデルの構成要素を工夫する

- 量子化

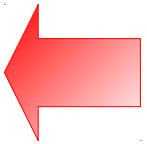
  訓練後に、重みの浮動小数点数の精度を下げる

- 低ランク近似

  訓練後の重みを特異値分解により、低ランク近似する

- Pruning

  訓練後に、寄与度の小さい重み、チャネル、レイヤーを削除する

- 蒸留  今回は蒸留を説明

  一度訓練したモデルが学習した知識を、別の軽量なモデルに継承させる

知識の蒸留：仕組み

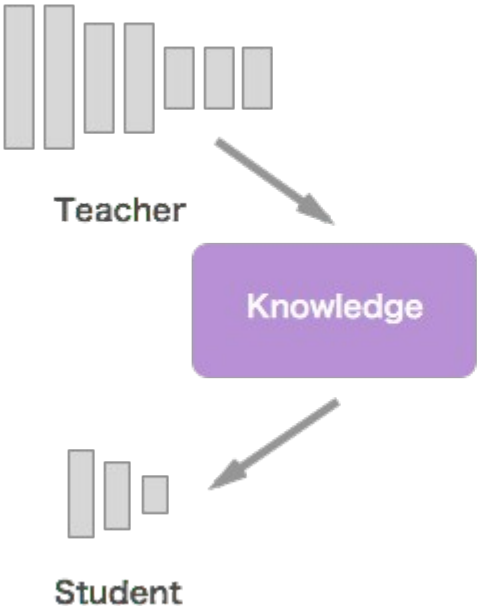
- 学習時

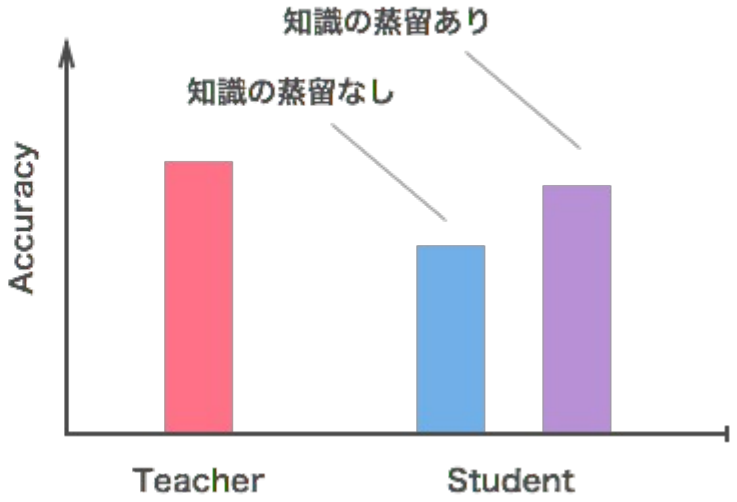
  大きい **教師モデル** から得た知識を 小さい **生徒モデル** に学習させる

  訓練データを丸暗記するのではなく、未知のデータに対する認識精度が良くなるように訓練を行う
- 推論時：

  限られた計算資源の中で動作

  予測精度だけでなく、処理速度も重要





知識の蒸留：継承される知識

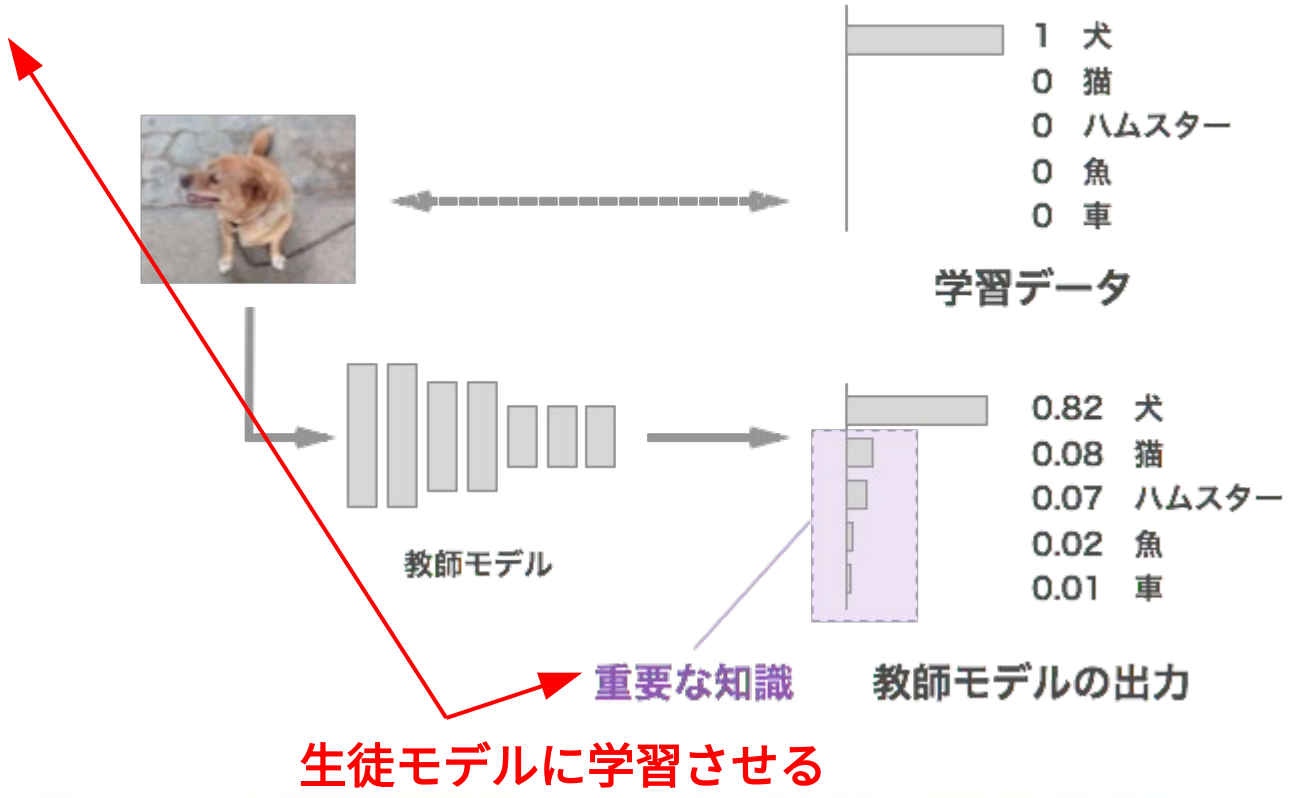
画像分類タスクを例として

- 正解ラベル

  正解である犬は **1 それ以外は 0** しか情報がない

- 教師モデルの推論

  正解である犬以外にも **0 以外**の情報がある



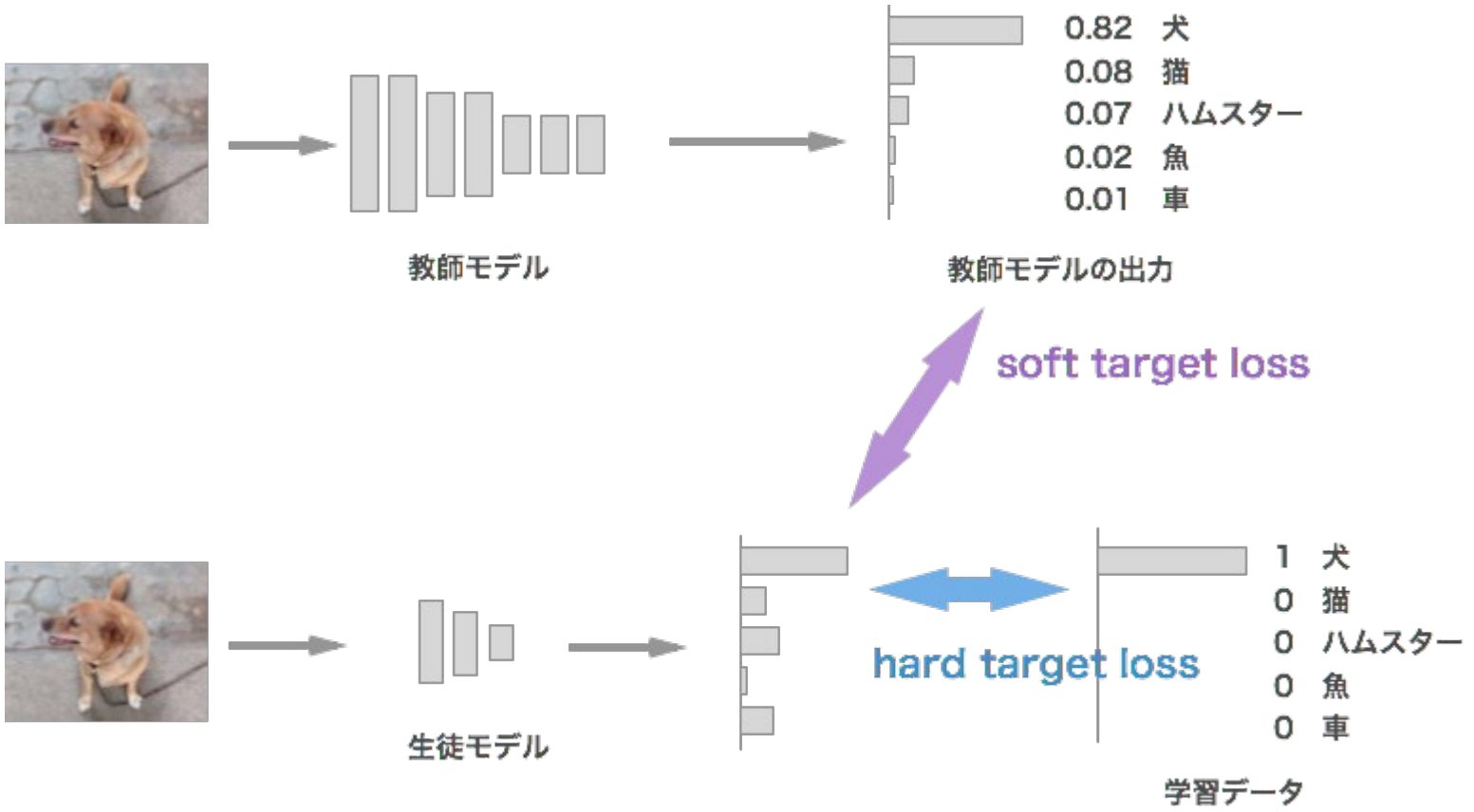
知識の蒸留：生徒モデルの学習

- Soft target loss

  教師モデルの出力の分布と近くなるような損失

- Hard target loss

  学習データの正解ラベルを利用した通常の損失（使用しないことも可能）



知識の蒸留：期待できる効果

- 精度の向上

  知識の蒸留なしで通常の学習を行った場合と比べ、高い精度が期待できます。

  教師モデルに匹敵する精度や、場合によっては教師を超えるような精度も報告されています。

- 正則化効果

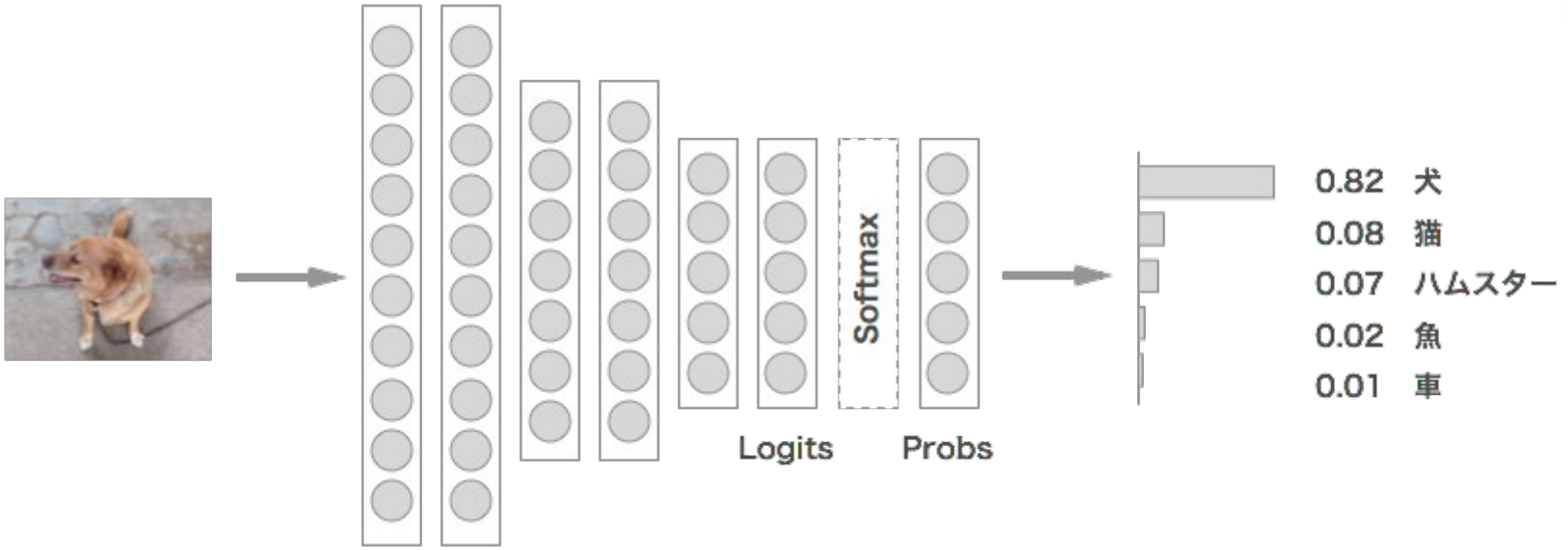
  知識の蒸留でソフトターゲットを加えると強い正則化効果があることも報告されています。

  トレーニングデータの 3% のみをつかって学習を行ったところ、知識の蒸留なしでは過学習（早期停止が必要）してしまうところを、知識の蒸留ありではきちんと収束したと報告されています。

- 膨大な知識を学ぶ

  単一のモデルで学習に時間のかかるような、クラス数が非常に多い＆膨大な学習データがあるようなケースでも、問題を分割して複数の教師モデル (スペシャリスト) を学習させておき、それらの知識を利用することで、効率的に学習することが可能です。

知識の蒸留：損失と出力



- Softmax に入力する手前の変数を Logits
- Softmax 入力後の合計して 1 になる変数を Probs
- 教師モデルの Logits を v Probs を p
- 生徒モデルの Logits を z Probs を q

知識の蒸留 :soft target loss

- L2 Loss

  一番シンプルな方法として、 Logits の差分の L2 ノルムの最小化があります。

$$Loss_{L2}=\frac{1}{2}\|z-v\|^2$$

- Softmax with Temperature

  温度付きのソフトマックス関数を使う方法です。教師と生徒の出力について、ソフトマックス関数を使う代わりに、 Logits を温度パラメータ T で割った値を入力とした温度付きのソフトマックスを提案しています。これを、教師モデルと生徒モデルのそれぞれに適用します。

$$q_i=\frac{\exp(\frac{Z_i}{T})}{\sum\exp(\frac{Z_i}{T})} \qquad q_i=\frac{\exp(\frac{v_i}{T})}{\sum\exp(\frac{v_i}{T})}$$

- あとは、生徒と教師の温度付きソフトマックスの出力に対してクロスエントロピーを取って損失を計算します。 $Loss_{softmax}=\sum_i p_i\log(q_i)$

知識の蒸留 :Softmax with Temperature

- Softmax with Temperature

  式から分かるように、温度 T を上げることによって、 soft target 内の正解クラス以外の類似クラスに対する値が増幅されます。

  それにより、 soft target に期待する効果がより現れやすくなります。

  なお、 soft target で学習する生徒モデルも、蒸留による学習時には、最終層の Softmax を同じ温度 T にして学習します 。

  温度 T を上げることで知識を教師モデルから生徒モデルに抽出することが、「蒸留」という命名の由来だと思われます。