ジャーナルクラブ
『知識の蒸留』

2021/5/19

長谷川凌太郎

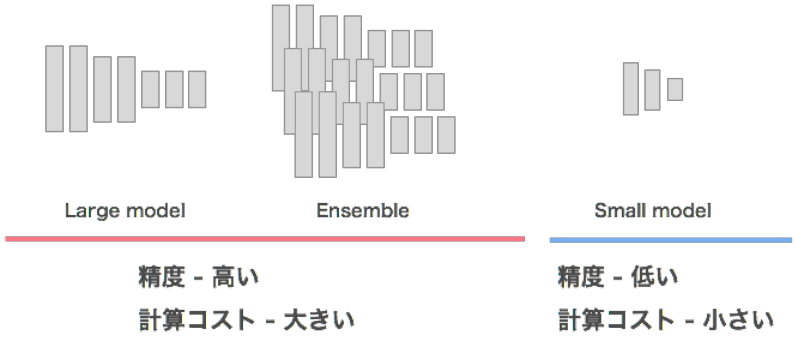
今回参考にしたもの

- 元論文
  - Distilling the Knowledge in a Neural Network
  - https://arxiv.org/pdf/1503.02531.pdf
- 解説記事
  - Deep Learningにおける知識の蒸留
  - 他の解説記事よりも情報量が多いため参考にしました。
  - http://codecrafthouse.jp/p/2018/01/knowledge-distillation/

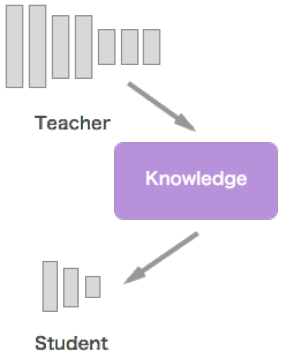
  - Distilling the Knowledge in a Neural Network
  - 論文の解説
  - https://paperdrip-dl.github.io/distillation/2018/12/23/Distillating-Knowledge-in-Neural-Networks.html

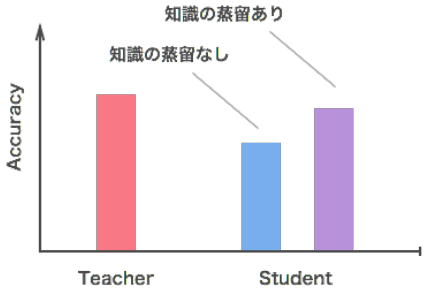
知識の蒸留(knowledge distillation)とは

深層学習において
・一般に深くてパラメータ数の多いモデルの方が精度が上がりやすい。
・複数モデルの予測結果（アンサンブル）を組み合わせる方が精度が上がりやすい。
・処理スピードが求められる場面では計算コストの小さい軽量なモデルが求められる。



そこでこのギャップを埋めるために大きいモデルやアンサンブルしたモデルを**教師モデル**として用意し、その知識を軽量な**生徒モデル**の学習に利用することで、軽量でありながら教師モデルに匹敵する精度のモデルを得ることが期待できる。





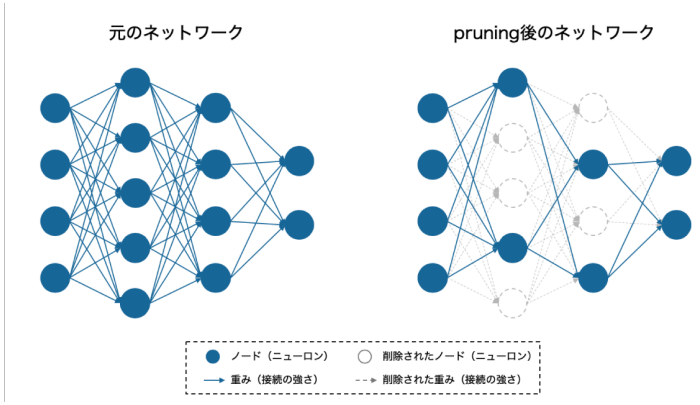
期待できる効果

Speech recognition

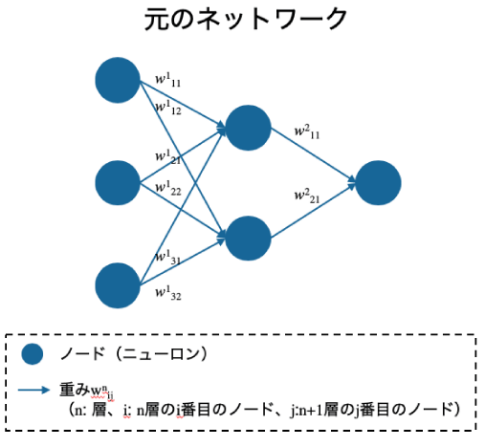
System	Test Frame Accuracy	WER
Baseline 10xEnsemble Distilled Single model	58.9% 61.1% 60.8%	10.9% 10.7% 10.7%

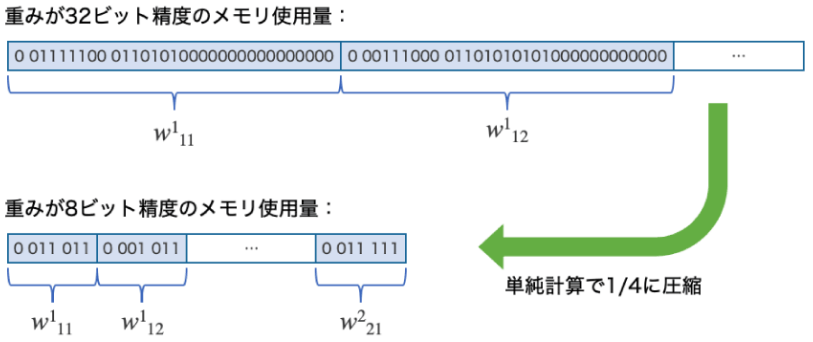
- 精度の向上
  - 蒸留なしと比べて高い精度を期待できる。場合によっては教師を超える。
- 正則化効果 Soft Targets as Regularizers
  - 強い正則化効果があるという報告がある。トレーニングデータの3%のみを使って学習したところ、蒸留なしでは過学習してしまう(44.5%)が、蒸留ありでは収束した(57%)。(speech recognition, baseline 58.9%)
- 膨大な知識を学ぶ Training ensembles of specialists
  - クラス数の多い&膨大な学習データがあるケースで、問題を分割して複数の教師モデルを学習させておき、それらを使うことで効率化。
  - 訓練時間:数週間→数日

モデル圧縮



- 枝刈り Pruning
  - ノードや重みを削除することでパラメータ数を減少させる。計算する回数が削減し、メモリ使用量が少なくなる。
- 量子化 Quantize
  - 重みなどのパラーメータをより小さいビットで表現することで、モデルの軽量化を図る手法。使用するビットを制限することでネットワークの構造を変えずにメモリ使用量を削減できる。Float型(32bit)が主流。8ビットの量子化で1%の性能低下。
- 蒸留 Distillation





https://laboro.ai/activity/column/engineer/ディープラーニングを軽量化するモデル圧縮/

知識の蒸留の派生

- 複数モデルのアンサンブルを教師とする
  - Model Distillation
  - Ensemble Distribution Distillation https://www.slideshare.net/DeepLearningJP2016/dlensemble-distribution-distillation-218038516
- 単一の教師で複数のデータ変形を利用
  - Data Distillation http://codecrafthouse.jp/p/2018/01/knowledge-distillation/#id61
- 生徒同士で教え合う
  - Deep Mutual Learning http://codecrafthouse.jp/p/2018/01/knowledge-distillation/#id60
- 教師と同じ構造を持った生徒で精度を上げる(生まれ変わり)
  - Born Again Neural Networks http://codecrafthouse.jp/p/2018/01/knowledge-distillation/#id59

知識の蒸留の仕組み

- 以前使用したスライドを使って説明

論文紹介

『Distilling Knowledge in a Neural Network』

- まだ理解が足りておらず内容をきちんと説明できる自信がないため、知識の蒸留の大枠のみを解説する。
- 下記の解説記事を参考に
- https://paperdrip-dl.github.io/distillation/2018/12/23/Distillating-Knowledge-in-Neural-Networks.html
- 元論文
- https://arxiv.org/pdf/1503.02531.pdf

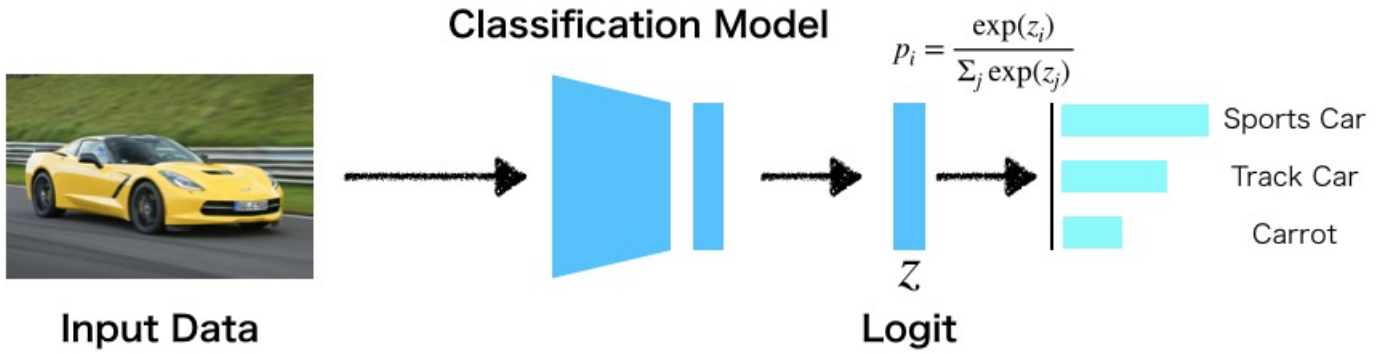
知識の蒸留

- Distilling Knowledge 知識の蒸留とは：
- 温度つきSoftmax cross-entropy による大規模モデルから小規模モデルへの知識の転移
- →巨大で複雑なモデルの知識を小さくシンプルなモデルへと転移させることで、小さいモデルで大きいモデルと同等の精度で推論を実行できるようにする。
- TeacherとStudentの関係性。

知識の蒸留

- 通常の分類モデル
  - 対数尤度pをSoftmax cross-entropyで最大化する。(下の式を最小化)

$$L = -\sum_{i=0}^{n} q_i \log p_i$$



解説記事より抜粋

復習

機械学習/2020-06-30/ML9_Taki.pdfより抜粋

Softmax関数

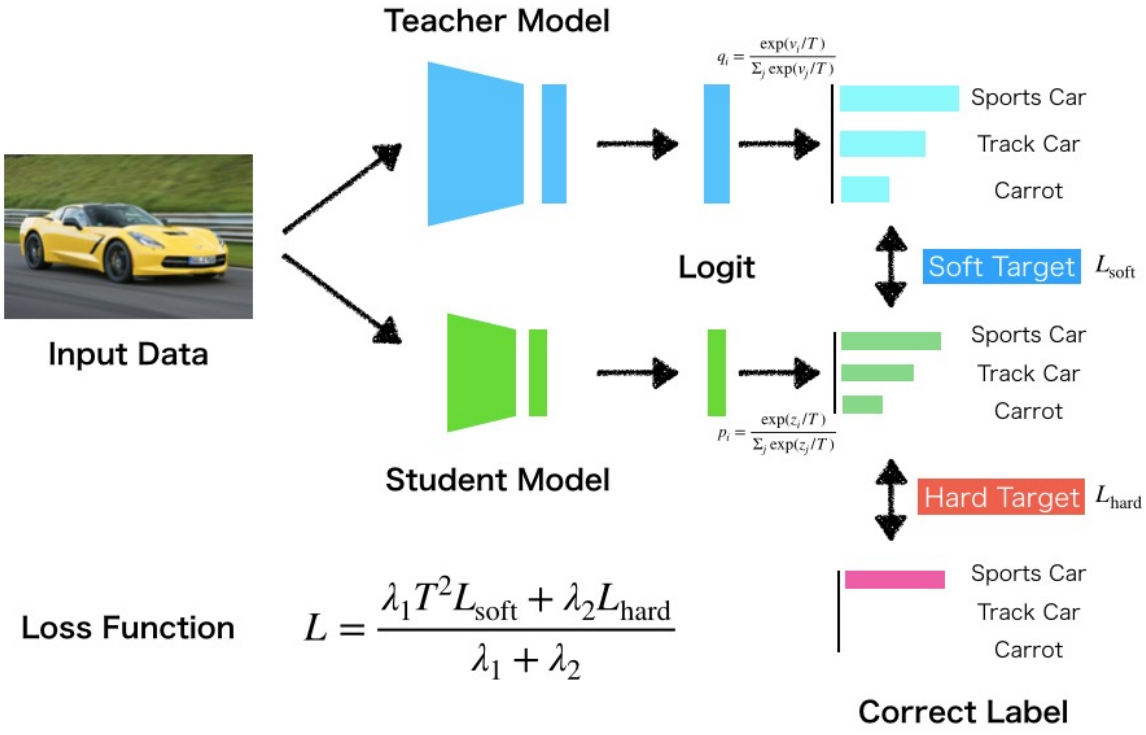
$$P(y = k|\vec{x}) = \frac{\exp(u)}{\sum_{k'} \exp(u_{k'})}$$

Categorical Cross-Entropy誤差関数

$$E(\vec{\theta}) = -\sum_{n}\sum_{k=1}^{K} y_{nk} \log P(y = k|\vec{x}_n)$$

知識の蒸留

- 知識の蒸留：
- Teacherの出力（予測確率）を使ってStudentの学習を行う。

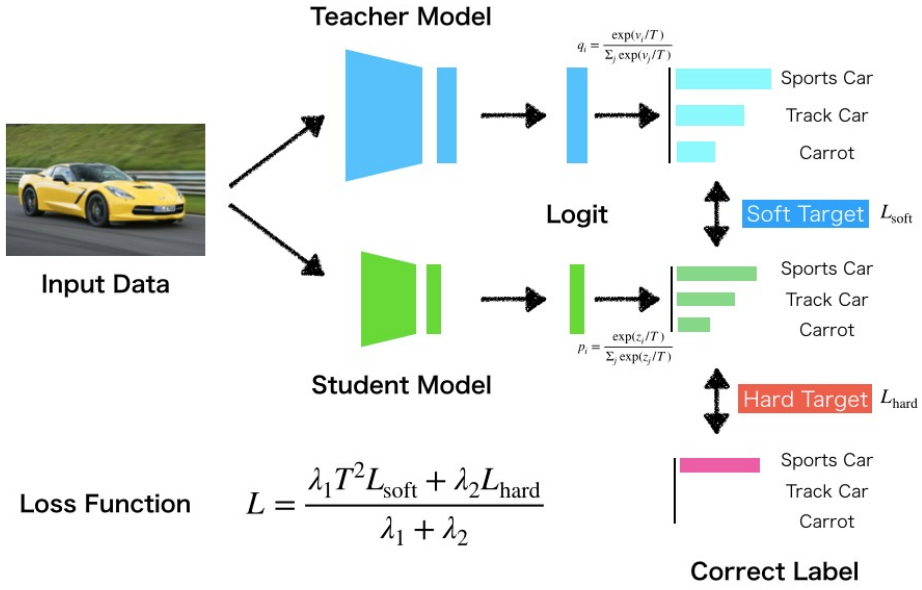


２つの損失項で最適化される。

Soft target loss: Teacherの予測確率と
Studentの予測確率を近づける項

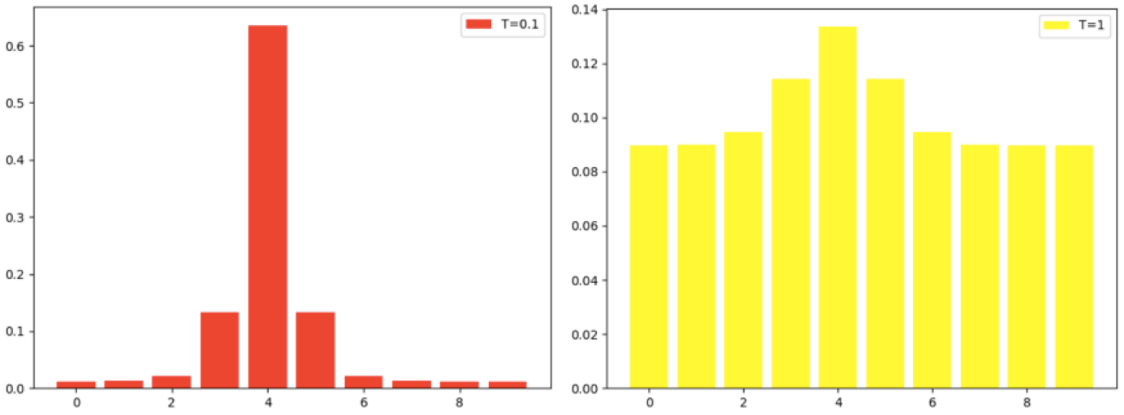
Hard target loss:正解ラベルと予測確率を
近づける項(普通のクロスエントロピー)

知識の蒸留



特殊なパラメータ：**温度T**

Teacherから得られる予測確率は入力xの指数関数exp(x)から計算されるので正解ラベルについて高く、相対的に不正解ラベルについてはとても小さいものになる(確率の偏り)
なので**温度付きsoftmax**を導入して温度パラメータTを**T>1**に設定することで不正解ラベルに対する予測確率を強調して学習する



予測確率teacher(温度付きsoftmax): $q_i = \frac{\exp(v_j/T)}{\sum_j \exp(v_j/T)}$

予測確率student(温度付きsoftmax): $p_i = \frac{\exp(z_j/T)}{\sum_j \exp(z_j/T)}$

$$L_{soft} = -\sum_{i=0}^{n} q_i \log p_i$$

$$L_{hard} = -\sum_{i=0}^{n} y_i \log h_i$$

予測確率student model: $h_i = \frac{\exp(z_j)}{\sum_j \exp(z_j)}$

$$L = \frac{\lambda_1 T^2 L_{soft} + \lambda_2 L_{hard}}{\lambda_1 + \lambda_2}$$

瀧先生コードだと、$\lambda_1 + \lambda_2 = 1$

知識の蒸留

- 精度(MNIST)

	# of layers	# of hidden units per layer	Test error cases
Teacher	2	1200	67
Student	2	800	146
Student (Distilled)	2	800	**74**

少ないパラメータで近い汎化性能が得られる

具体的な手法

- 出力を利用する方法
  - 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
    - $Loss_{L2} = \frac{1}{2}\|z - v\|_2^2$
  - Softmax with Temperature
    - 上記で説明済み。Logitsを温度パラメータTで割った値を入力にしたソフトマックスを教師モデルと生徒モデルに適用してクロスエントロピーを取る。
  - KL Divergence
    - 教師と生徒の出力の分布間の損失としてKL Divergenceを利用する方法
    - 教師の出力の分布pと生徒の出力の分布qが一致した時にゼロとなる指標なので、より自然な表現
    - $Loss_{KLD} = KL(q||p) = \sum_i p_i log \frac{p_i}{q_i}$　もしくは $Loss_{KLD} = KL(p||q) = \sum_i q_i log \frac{q_i}{p_i}$
- 中間層の情報を利用する方法
  - 教師モデルの中間層をヒントとして生徒モデルの学習に利用(L2 Loss)
- 特権情報（追加の情報）を利用する
  - 教師の訓練時のみ追加の情報（特権情報）を使い、そこから得た知識を生徒の学習に利用

研究案

- アンサンブルした教師モデルの知識の多様性をキープして生徒モデルに蒸留する手法
- 蒸留時のファシリテーターモデル（進行役・監視役）の作成
  - リーダーシップ論、ファシリテーション論の応用で精度向上？
- 蒸留時のフィルタの違いを見る
  - teacher/student only/distilled studentでの畳み込みフィルタの特徴の違い
  - →蒸留される知識、継承される文化を観察できるのでは？