Distilling the Knowledge in a Neural Network

ソース

- https://arxiv.org/pdf/1503.02531

概要

- 蒸留(Distillation)を提案
- 蒸留は、大きくて複雑なモデル(Teacher)から小さくて軽量なモデル(Student)に知識を転送するプロセスのこと
- 需要として、本番環境にデプロイするときに計算量や計算時間の制約から大規模モデルは使えないが大規模モデルとあまり制度が変わらない計量モデルが欲しいなど
- アイディアとしては、大規模モデルが学習したことを計量モデルに効率よく与えるというもの
- 多クラス分類でラベルを使って学習するだけでなく、大規模モデルが出力するsoftmax関数の出力を学習に使う
    - こうすることで不正解のものの確率分布を学習させる
    - 論文中の例として、BMWが正解であるときにごみ回収車の確率が低くなっているがにんじんの確率はもっと低くなっているはずであり、このような分布を効率よく学習させる

蒸留

- 結論としては、softmaxの出力結果で各ラベル(不正解ラベルもすべて)に対して差分をとる

いま、クラスiである確率q_iがtemperature Tを用いて以下のようにかけるとする

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

このとき勾配は、

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}\left(q_i - p_i\right) \tag{1}$$

$$= \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right) \tag{2}$$

Tが十分大きいとして、

$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right) \tag{3}$$

$$\approx \frac{1}{NT^2}\left(z_i - v_i\right) \tag{4}$$

このとき、v_iが教師モデルが出力するクラスiである確率でありz_iが生徒モデルが出力するクラスiである確率であるため、softmaxの差分をとることに帰着する

英単語メモ

- cumbersome : 面倒な、煩雑な