

進捗報告

1 今週やったこと

- cifer-10 の Augmentation 操作と GA の紐付け
- 論文調査 (Fast AutoAugment)

1.1 cifer-10 の Augment と GA の紐付け

cifer-10 データセットに対してクラス分類をする際の Data Augmentation の GA を使った自動最適化の実験のスク립トを実装していた。モデルとしては事前学習済み ResNet を用いている。実装している操作は ShearX/Y, TranslateX/Y, Rotate, AutoContrast, Invert, Equalize, Solarize, Posterize, Contrast, Color, Brightness, Sharpness, Cutout の 15 種。GA の遺伝子にはひとまず各操作を使うか使わないかのバイナリしか持たせていないが、これに操作の順序および確率と強度を扱えるような枠組みを実装したいと考えている。

また、DEAP のオペレーション周りの拡張についても一応試してみた。交叉について一様交叉の枠組みを用いてガウス分布からサンプリングした値としきい値を用いて交叉するように改変した。これについては特に本質的な意味はなく、正しく拡張オペレーションが動いたという確認だけ。前回同様レポは https://github.com/1g-hub/DEAP_CLS に。

1.2 論文調査 (Fast AutoAugment)

前回紹介した強化学習を用いた Data Augmentation の自動化の研究 AutoAugment を高速化する論文を発見したので読んだ [1]。AutoAugment との違いは、探索手法に強化学習ではなく density matching の考え方を用いたベイズ最適化を取り入れている点。また、訓練の仕方が独特で、訓練データを Kfold してその中でモデルのパラメータチューニングに使う部分と操作の探索に使う部分とに分けて、問題を訓練データと評価データの density matching と置き換えて解いている。この 2 つの新技术を用いて従来の AutoAugment と同程度の性能を保ったまま 1000 倍以上早くなっている。現在はこの Fast AutoAugment を GA でできればいいなあと考えている。もともとの AutoAugment はグーグルのマシンで cifer-10 を縮小したやつでも 5000 時間とか... 現在拡張しようと思っているレポは <https://github.com/JunYeopLee/fast-autoaugment-efficientnet-pytorch> で、サブポリシーの最適化の部分のアルゴリズムを GA に差し替える予定。

2 来週のタスク

GA の遺伝子型の改良。

参考文献

- [1] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *CoRR*, abs/1905.00397, 2019.