

遺伝的アルゴリズムを用いたスパース自己符号化器の生成及びそれらを用いたアンサンブル学習

Generation of sparse self-coders using genetic algorithms and ensemble learning with them

Basic Format of Student Research Symposium

渡辺 龍也[†] 李 磊[†]

Ryuya Watanabe[†] Lei Li[†]

[†] 法政大学大学院理工学研究科応用情報工学専攻

Major in Applied Informatics, Graduate School of Science and Engineering, Hosei university

1. はじめに

近年では AutoML という自動的に AI を構築するサービスの関心が高まっており、様々な研究が行われている[1]。本稿では組み合わせ問題に有効な遺伝的アルゴリズムを用いてスパース自己符号化器の構成を最適化し、それらの学習器から得られるデータの多様性を担保にアンサンブル学習[2]を行った。

2. 関連研究

2.1 遺伝的アルゴリズム

遺伝的アルゴリズムとは生物の自然選択に基づく遺伝や進化の仕組みを用いた最適値探索の手法である。

遺伝的アルゴリズムは以下のステップからなる。

STEP1: ランダムに M 個の個体を生成し、初期個体集合 $P(0)$ を作り、世代数を $t=0$ とし、繰り返し回数 T を設定する

STEP2: 個体集合 $P(t)$ 内の個体について、その適応度を算出する

STEP3: 個体集合 $P(t)$ に選択演算子を適用し、 $P'(t)$ を生成する。

STEP4: $P'(t)$ に交叉演算子を適用し、 $P''(t)$ を生成する。

STEP5: $P''(t)$ に突然変異演算子を適用し、 $P'''(t)$ を生成する。

STEP6:

$t < T$ ならば、 $t = t+1$ として STEP2 へ戻る。そうでなければ、計算を終了する。

これまでに得られた最大適応度の個体を解とする。

2.2 自己符号化器

砂時計型ニューラルネットワークと呼ばれる構造をしている。基本的に隠れ層のノード数が入力層よりも少ないネットワークを使い、入力信号を教師信号として誤差逆伝搬していくことで、隠れ層に入力層を圧縮表現するノードが得られる手法[4]。特に中間層が 1 より大きいものを深層自己符号化器と呼ぶ。スパース自己符号化器とはこれの損失関数にスパース項と呼ばれる値を追加してゼロでない特徴集合を得る手法である。本手法では元の特徴数より大きい自己符号化器での学習において、スパース項を追加することによって恒等学習することを避けている

2.3 アンサンブル学習

アンサンブル学習とは多様性のある複数の学習モデルによる多数決をとることで未知のデータへの予測能力を向

上させる手法である。通常アンサンブル学習に用いる学習モデルは複数のモデルを使用することで多様性を実現している。

3. 実験結果

以下にカルフォルニアの住宅価格の中央値データを用いた実験結果の一例を示す

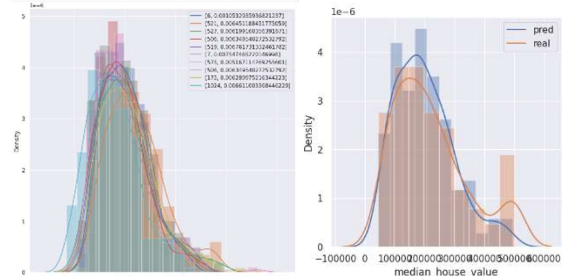


図 1 実験結果一例

自己符号化で作成したデータを用いた個々の学習結果は元のデータに対しておおむね改善がなされていた。また、作成されたデータを用いて行ったアンサンブル学習は最も良い結果を示した。これによってこのデータによって作成されたデータに多様性があり、アンサンブル効果が得られることが判明した。

4. 今後の課題

本実験で用いた自己符号化器は単層であるため多層での効果の検証行っていない。また、遺伝的アルゴリズムの探索パラメータも 2 つのため、組み合わせの探索範囲は不十分と言わざるを得ない。今後はこれらの課題に対処し、改善を図ることが望まれる。

参考文献

[1] FENG Xuanang: "Genetic Algorithm-based Optimization of a Deep Neural Networks Ensemble and Its Applications", Department of Complex Systems Science Graduate School of Informatics Nagoya University, 2020-03-25

[2] G. W. Cottrell and P. Munro, Principal component analysis of image via back-propagation. In Proceedings of SPIE 1001 Visual Communications and Image processing '88, pp. 1070-1076. 1988.