

# QBoost を用いた画像認識

坂倉 佑季 (指導教員：工藤和恵)

## 1 はじめに

インターネットが広く普及し、IoT(モノのインターネット化)やAIが発展してきている現代において、膨大な量のデータが手軽に収集可能になり、大量の選択肢の中から最適なものを選ぶ「組合せ最適化問題」を高速に解く需要が高まってきている。実際に社会や産業界において、配送計画、勤務シフト、機械学習など数多くの組合せ最適化問題があり、幅広い応用分野がある。しかし、このような組合せ最適化問題はデータが多くなるにつれて選択肢の数が爆発的に増加し、従来のコンピュータでは解くのに非常に時間がかかる。そんな最適化問題を高速かつ高精度に解いてくれると期待されているのが、アニーリングマシンである。

そこで、様々な分野で応用されている画像認識を組み合わせた最適化問題に置き換え、アニーリングマシンで解くことによって、処理の高速化を実現したいと考えている。本研究では手始めに手書き数字認識を行う。

## 2 モデルと方法

### 2.1 Boosting

Boosting とは、性能の低い識別器を順番に学習し組み合わせることによって、性能の高い識別器を作る手法のことである [1]。最終的な強識別器は弱識別器の重み付き多数決によって構成され、それを式で表すと以下ようになる。

$$C(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^N w_i c_i(\mathbf{x}) \right) \quad (1)$$

この時、 $\mathbf{x}$  は入力画像ベクトル、 $w_i$  (実数値) は弱識別器の重み、 $c_i \in \{-1, 1\}$  は弱識別器の出力、 $C(\mathbf{x})$  は強識別器の出力を表している。

### 2.2 QBoost

Boosting の中で最もよく利用されている方法として、AdaBoost と呼ばれるものがあるが、AdaBoost において弱識別器に対する重み  $w_i$  は連続値となっており、本研究で用いるアニーリングマシンにそのまま載せることができないので、アニーリングマシン用の Boosting アルゴリズムである QBoost を用いることにする [2]。

QBoost では弱識別器の重みは 0 か 1 の値を取り、 $i$  番目の識別器を使用するかどうかを表しており、最適な重み  $w_i$  の組を求めることによって、どの識別器を使えば出来るだけ少ない数の弱識別器でより良い性能を得ることができるかを明らかにすることができる。

最適な重み  $w_i$  の組合せを決定するために、アニーリングマシンを用いる。アニーリングマシンに組合せ最適化問題を解かせるためには、解く問題を QUBO (Quadratic unconstrained binary optimization) という形式 (次式) に変換する必要がある。

$$H = \sum_i h_i w_i + \sum_{i < j} J_{ij} w_i w_j \quad (2)$$

問題を QUBO の形にするために識別器の出力と正解

との差の二乗を最小化する問題とする。 $d$  番目の画像ベクトル  $\mathbf{x}^{(d)}$  に対する教師データを  $y^{(d)}$  とすると、コスト関数は次式で与えられる。

$$\begin{aligned} H(\mathbf{w}) &= \sum_{d=1}^D \left( \frac{1}{N} \sum_{i=1}^N w_i c_i(\mathbf{x}^{(d)}) - y^{(d)} \right)^2 + \lambda \sum_{i=1}^N w_i \\ &= \sum_{i=1}^N \left( \lambda - \frac{2}{N} \sum_{d=1}^D c_i(\mathbf{x}^{(d)}) y^{(d)} \right) w_i \\ &\quad + \frac{1}{N^2} \sum_{i,j=1}^N \left( \sum_{d=1}^D c_i(\mathbf{x}^{(d)}) c_j(\mathbf{x}^{(d)}) \right) w_i w_j + \text{const} \end{aligned} \quad (3)$$

ここで、 $D$  は画像の枚数、 $N$  は弱識別器の数、 $\lambda$  は過学習を防ぐための正則化パラメータである。

そして、画像認識は以下のように行われる。例えば、図 1 のように「0」という画像が与えられた時、弱識別器が 0 であると判断した場合は「1」、0 でないと判断した場合は「-1」が出力される。そして、使用した弱識別器の中で多数決が取られ、強識別器の出力が決まる。

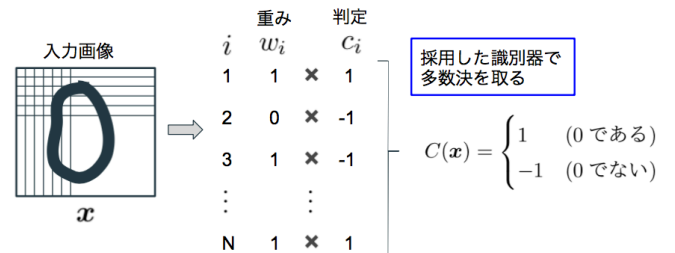


図 1: 方法

### 2.3 決定木

本研究では、弱識別器として決定木を用いた。決定木とは、変数とその値の組によって表現されたデータを、条件分岐によってグループに分類する手法である。条件分岐を繰り返すことで、データはツリー状にどんどん展開され、解くべき最小単位に分割される。

## 3 結果と考察

手書き数字データセット mnist[3] を用いて手書き数字認識を行なった。

まず最初に、正則化項のパラメータ  $\lambda = 3$ 、弱識別器の数は 35 個、木の深さは 1 とし、数字 0 と 1、数字 3 と 8 の二値分類を行った。それぞれ、訓練データを増やしていった時の正答率をグラフにしたものが図 2 と図 3 である。この時、訓練データとテストデータの割合は 8:2 になるようにした。

数字 0 と 1 の分類では、訓練データが 100 個ぐらいあれば正答率 9 割以上を得られており、精度が良いことがわかる。また、弱識別器の数を減らしても正答率は 9 割以上得られ、数字 0 と 1 だけの二値分類が簡単だったためだと考えられる。

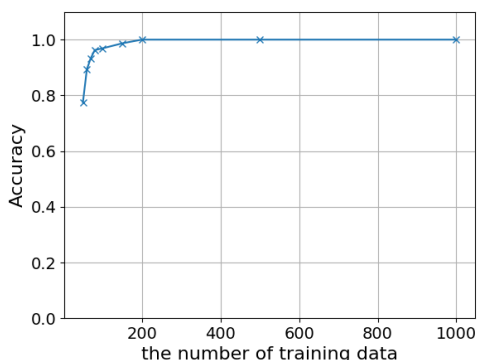


図 2: 数字 0 と 1 の分類結果 :  
正答率のデータ数への依存性

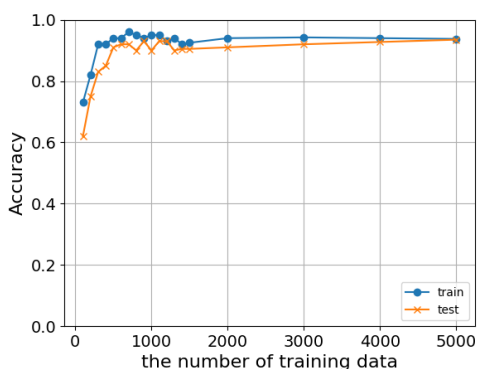


図 3: 数字 3 と 8 の分類結果 :  
正答率のデータ数への依存性

図 3 は青の線が訓練データ、オレンジの線がテストデータに対する正答率を表している。訓練データの数を増やすにつれて正答率が上がった。また、数字 0 と 1 の二値分類と比べると、3 と 8 の分類の方が正答率が低く 3 と 8 の分類の方が難しいことが分かる。

次に、数字 0 とそれ以外の 1~9 の二値分類を行った。これまでと同様に、正則化項のパラメータ  $\lambda = 3$ 、弱識別器の数は 35 個、木の深さは 1、訓練データとテストデータの割合は 8:2 とした。

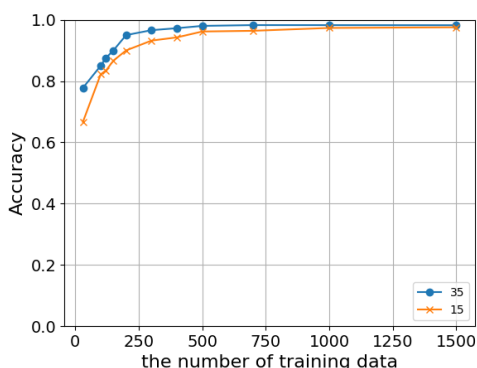


図 4: 数字 0 とそれ以外 (1~9) の分類結果。  
凡例の数字は弱識別器数。

図 4 は訓練データを増やしていった時の正答率をグラフにしたものである。青の線は弱識別器が 35 個で、オレンジの線は弱識別器 15 個の時の正答率を表して

いる。これもまた、訓練データを増やすと正答率も上昇する。また、弱識別器の個数を増やすと正答率が上がっており、弱識別器をまとめることでより精度のよい強識別器が作れていることが分かる。

次に、訓練データを 10000 枚、テストデータを 2000 枚、パラメータ  $\lambda = 5$  に固定して弱識別器の個数を変化させた時の正答率を見た。結果を図 5 に示す。

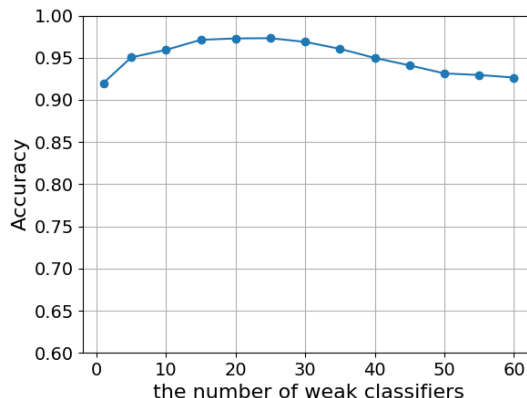


図 5: 正答率の弱識別器の個数への依存性

弱識別器の個数を増やしていくに連れ途中まではだんだんと正答率が上がっていくことが分かる。しかし、途中からは正答率が下がっていき、過学習が起きていると考えられる。そこで、過学習が起きている弱識別器 45 個の時に、パラメータ  $\lambda$  の値を調整した。すると、 $\lambda = 15$  あたりで使用する弱識別器の数は 34 個に減り、正答率 0.9765 が得られ、過学習を防ぐことができた。

#### 4 まとめと今後の課題

アンニリングマシン用の Boosting アルゴリズムである QBoost を用いて、画像の二値分類を行うことができた。訓練データや弱識別器の個数を増やすことによって、正答率が上がることを確認した。また、弱識別器の個数を増やしすぎると過学習を起こすが、パラメータ  $\lambda$  を調整することによって、過学習を防ぐことができた。今後は二値分類を組み合わせることで多値分類を行い、最終的には車のナンバープレートの画像認識を行いたいと考えている。

#### 謝辞

本研究は一部、未踏ターゲット事業の支援を受けている。

#### 参考文献

- [1] 西森秀稔, 大関真之, 「量子アンニリングの基礎」 (共立出版, 2018).
- [2] H. Naven, et al., "NIPS 2009 demonstration: Binary classification using hardware implementation of quantum annealing.", arXiv:0912.0779. (2009)
- [3] 手書き数字データセット (mnist), <https://www.kaggle.com/c/digit-recognizer/data> (2018 年 10 月 23 日アクセス)