高次元入力データのための誤差逆伝搬を用いたGBDT実装の検討

藤野 知之[1,a)]　柏木 啓一郎[1,b)]

概要：勾配ブースティング法（GBDT）はさまざまな利用シーンで用いられ，特にデータベースのようなテーブルデータやIoTのセンサデータを用いた機械学習に頻繁に用いられている．一方，映像や画像・自然言語・音声といったメディアデータの分類においてはニューラルネットワークを用いた深層学習が一般的によく用いられており，GBDTはそのアルゴリズム的な構造から，メディアデータのような高次元データにおいて精度面で深層学習に劣る．本研究では，GBDTにニューラルネットワークの学習アルゴリズムである誤差逆伝播法の考え方を導入し，高次元データを高精度で扱えるよう拡張を施した．これにより，メディアデータの機械学習において，従来深層学習一択であったユーザーの選択肢を広げるとともに，GBDTの適用可能な問題範囲を広げる可能性を示した．本稿では，アルゴリズムの提案，実装そして画像データセットによる性能評価について述べる．

1. 背景

GBDT(Gradient Boosting Decision Tree; 勾配ブースティング法) は深層学習と並んで広く普及している機械学習技術であり，様々な産業・用途で用いられている [3]．GBDTが広く利用されている理由としてスケーラブルで大量のデータを扱えるライブラリの存在がある [1], [8]．これらのライブラリの登場によりクラウド上で膨大なデータに対しても学習・推論を行うことができるようになり，ビッグデータの解析手法として用いられている．

一方，近年の機械学習・AI技術の躍進の原動力は深層学習である [14]．深層学習は，ネオコグニトロンが画像の一般物体認識において極めて有効であることを示したAlexNet[11] に端を発し，現在では音声処理や自然言語処理においてもニューラルネットワークをベースとした深層学習が主流となってきている．画像認識の成功のみならず，特に近年ではBERTやGPT-2などの登場により自然言語処理の分野においても高い有効性を示し，注目を集めている．

深層学習がGBDT等他の機械学習アルゴリズムと異なる点として，特徴量抽出を学習の過程で獲得していくことが挙げられる．従来の機械学習は，データの特性に基づいて事前に適切な特徴量抽出という処理を施し，抽出した特徴量を学習器に入力し，分類させる方法が一般的であった．例えば画像の分類においてはSIFT, SERF, HOGなどさまざまな有力な特徴量があったり，顔認識に特化したHaar-like特徴量があったり，とデータやタスクに合わせて特徴量抽出手法が数多く提案されてきた [9]．他方深層学習では，直接データそのものを入力しニューラルネットワークの前段部分において特徴量抽出を学習の過程で獲得するため，特徴量抽出の事前処理を必要としない．特徴量の学習による獲得により，従来は事前処理として扱われていた特徴量抽出においても学習による最適化が施され，データ・タスクにより適した特徴量抽出が選択されていると考えられる．

GBDTは深層学習を除けば極めて高い精度を達成している機械学習手法の一つと言える [17]．しかしながら，GBDTは特徴量の獲得という点はほとんど考慮されておらず，入力データの次元を独立に扱いその相関を考慮しない．特徴量抽出は入力データの次元間の相関情報を汲み取り，分類しやすいような特徴量空間にマッピングをする処理であるため，次元を独立に扱う従来のGBDTでは特徴量の獲得は見込めない．著者らは先行研究 [19] において画像に代表される高次元データをGBDTで取り扱うための幾つかの工夫を提案したが，深層学習の精度には及んでいない．これは，GBDTには深層学習のような特徴量の獲得という性質がないことに起因するのではないかと仮説を立てることができる．

[1] NTTソフトウェアイノベーションセンタ
NTT SIC, 3-9-11, Midori-cho, Musashino-shi, Tokyo 185–8585, Japan
[a)] tomoyuki.fujino.bn@hco.ntt.co.jp
[b)] keiichirou.kashiwagi.yk@hco.ntt.co.jp

©2021 Information Processing Society of Japan

本研究の目的は，GBDT の適用範囲を広げ，画像・音声・自然言語といったメディアデータにも GBDT を適用可能にし，データサイエンティストに深層学習以外の選択肢を提供することである．選択肢を増やす利益の一例として，学習・推論に用いるデバイスのコストを抑えられる可能性について述べる．深層学習は行列の積和計算と非線形関数による処理からなり，GPU による並列計算に適した処理となっている．一方で GBDT は同様に並列化が可能であるが，処理のベースはツリーの分岐処理と重みの和の計算である．特に分岐処理は GPU が苦手とする計算であり，CPU であったりより単純に閾値を用いた電気回路などの得意とする範疇である．本研究を進めることにより，汎用の CPU や安価な電気回路を用いて効率よく機械学習の推論を行える可能性がある．

本稿では，GBDT に誤差の逆伝播則を取り入れ，特徴量の獲得を行う拡張を行うことにより GBDT の精度改善を図る．2 章では提案手法の基となる GBDT と誤差逆伝播法や本研究に類似のアプローチを行なっている関連研究について述べ，3 章では基となる GBDT の基本的なアルゴリズムに触れ，4 章にてニューラルネットワークの学習に用いられる誤差逆伝播法の考え方に基づいた多段構成の GBDT を学習させるアルゴリズムを提案，最後にアルゴリズムの実装を行い画像データセットを例として検証を行う．

2. 関連研究

2.1 Boosting

Boosting は識別率の低い単純な弱識別器を組み合わせて強力な学習器を作るアンサンブル学習の一手法であり，Freund らによって提案された Adaboost[4] に端を発する．Friedman らは Adaboost が弱識別器を適切に決定することで目的関数の最小化を行なっていることを明らかにし [5]，Adaboost の目的関数よりも外れ値に強い Cross Entropy Loss を最小化する Logit Boost を提案した [6]．Logit Boost を基に，弱識別器として決定木を用い，目的関数にさらに正則化項を加えたものが GBDT である [2]．XGBoost[1] や LightGBM[8] といったライブラリは目的関数や正則化項を選択的に選べたり，外部から与えることができ，Logit Boost や Adaboost を包含する一般的な GBDT モデルと言える．

2.2 誤差逆伝播法

ニューラルネットワークの学習において従来より広く使われているアルゴリズムが誤差逆伝播法（BP; BackPropagation）である [7]．誤差逆伝播法はニューラルネットワークの出力値と正解ラベルデータを目的関数で評価し，ニューラルネットワークの末尾から順にネットワークのパラメータの勾配を計算し，勾配降下法によりパラメータを更新する手法である．ニューラルネットワークは誤差逆伝播法により複雑なネットワーク構造であっても末尾から順を追って最適化を行うことができる．深層学習はこの誤差逆伝播法を用いて深層な構成のネットワークを学習可能にしている．近年では勾配降下法としてオンライン学習が可能な確率的勾配降下法 [12] が用いられることが多く，深層学習の発展とともに様々な効率の良いアルゴリズムが提案されている．

2.3 類似研究

決定木のアンサンブルとニューラルネットワークを組み合わせる試みは様々なアプローチで行われている．[13] では深層画像認識アルゴリズムである ResNet を基に Residual に相当する項を GBDT に導入したり，汎関数勾配を用いて入力データの特徴量を求めるような最適化を行なっている．[15] では決定木をニューラルネットで表すことができるように変形し，勾配降下法による最適化を可能にした．[16] は深層学習の多層構造を決定木アンサンブルに適用し，ランダムフォレストの多層構造化を実現している．これらの先行研究は本研究のベースとなっており，特に [16] の多層構造は本研究においても取り入れている．一方で，本研究では誤差逆伝播法を模したブースティングによる最適化方法を提案しており，特徴量の獲得という深層学習の長所を GBDT にも取り込もうと試みている点に違いがある．

3. 従来方法：GBDT

本章では XGBoost 等で現在一般的に使われている GBDT のアルゴリズムについて説明する．N をサンプル数，D を入力の次元，C をラベルのクラス数とすると，学習データセットは $\{\boldsymbol{x}_i, y_i\}_{i=0}^{N}$ と表される，ただし，$\boldsymbol{x}_i \in \mathcal{R}^D$ は入力ベクトル，$y_i \in \{1, 2, ..., C\}$ はラベルである．ブースティングでは以下の式のようにラベルの推定値 $\hat{y}_i$ を複数の弱識別器 $f_m(\boldsymbol{x})$ の和で推定する．

$$\hat{\boldsymbol{y}}_i = F(\boldsymbol{x}) = \sum_{m=1}^{M} f_m(\boldsymbol{x}_i) \tag{1}$$

ここで太字の $\hat{\boldsymbol{y}}_i$ は C 次元ベクトルの推定値を表し，y_i の one-hot エンコーディング形式 $\boldsymbol{y}_i$ の予測値とする．また，M は弱識別器の数である．

GBDT で多クラス分類問題を解く場合，出力クラス数分木の集合を構築するのが一般的である．このことを式 (1) に当てはめると，あるクラス c に対して，

$$\hat{y}_{i,c} = F_c(\boldsymbol{x}_i) = \sum_{m=1}^{M} f_{m,c}(\boldsymbol{x}_i) \tag{2}$$

とクラスの添字を付与することで表現する．

この F_c はブースティングによって形成される多入力 1 出力の関数であり，内部的には複数の決定木の集合である．この木の集合は必ず一まとまりで扱われる単位であり，本

©2021 Information Processing Society of Japan

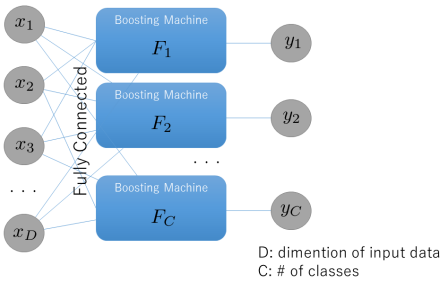

図 1 従来の GBDT のネットワーク表記

稿ではこの関数をブースティングマシンと呼ぶ．

多クラス分類問題をデータ次元とブースティングマシンによってネットワーク状に示したのが図 1 である．

ここで目的関数 l が各データ点の予測値と真値の間に定義されており，l を学習データ点全体で和をとった物を L とする．l は典型的にはクロスエントロピーロス関数などが用いられる．あるブースティングマシンに $M-1$ の木が存在し，M 個目の決定木を追加するとした時，目的関数 L を新規に追加する決定木 $f_{M,c}$ 近傍のテイラー展開を用いて以下のように近似する．

$$L_{(M)} = \sum_{i=1}^{N} l(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i^{(M-1)} + f_{M,c}(\boldsymbol{x}_i)) + \sum_{m=1}^{M} \Omega(f_{m,c}) \tag{3}$$

$$\approx \sum_{i=1}^{N} \{l(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i^{(M-1)}) + f_{M,c}(\boldsymbol{x}_i) g_{i,c} + \frac{1}{2} f_{M,c}(\boldsymbol{x}_i)^2 h_{i,c}\} + \sum_{m=1}^{M} \Omega(f_{m,c}) \tag{4}$$

ただし $g_{i,c} = \frac{\partial L_{(M-1)}}{\partial F_c}$，$h_{i,c} = \frac{\partial L^2_{(M-1)}}{\partial^2 F_c}$ で，それぞれ目的関数のツリー数 $M-1$ における推定値による勾配と 2 階微分値であり，前回のブースティング結果により定まる値である．GBDT の学習はテイラー近似された L を最小化するように L に対して $f_{M,c}$ で微分をとることで，

$$\frac{\partial L_{(M)}}{\partial f_{M,c}} = \sum_{i=1}^{N} (g_i + h_i f_{M,c}) \tag{5}$$

と書き下すことができる．ただし，簡略化のため正則化項は省略した．その極値をとる値は

$$\hat{f}_{M,c}(\boldsymbol{x}) = -\frac{\sum_i g_{i,c}}{\sum_i h_{i,c}} \tag{6}$$

という最適解を得る．実際には，$f_{M,c}$ は決定木であるため木の葉の重みを (6) で定める．これにより，木 $f_{M,c}$ の分岐構造が与えられた時，適切な重みを葉に設定できるようになったが，以下では適切な重みが設定できることを前提として分岐構造を探索することを考える．

木の分岐の探索においては (4)(6) より以下のようなゲイン値を得ることができる．

$$\mathrm{Gain}_T = \frac{\sum_{i \in T} g_i^2}{2 \sum_{i \in T} h_i} \tag{7}$$

ここで T はツリー内のある分岐を通過するデータ点の集合である．このゲイン値は分岐を設定することにより目的関数値を下げることが可能かを定量的に計算したものである．分岐を設定しその左右に適切な重みの葉を割り当てた場合と，分岐せずに全体を葉とした場合のゲイン値の比較を行い，ゲイン値が大きくなる場合，その分岐が目的関数値を下げるのに貢献すると言える．

分岐の探索方法として，次元とデータ点に対してグリッドサーチを行うのが一般的である．すなわち分岐に含まれるデータ点集合 $\{\boldsymbol{x}_t; t \in T\}$ を全ての次元 d でソートし，可能な分岐点ごとに分岐後の左右のデータ集合のゲインを計算し，現在のゲイン値を最も上回る分岐位置を探索する．

このグリッドサーチの際，全ての次元は独立に探索され，最もゲイン値が高くなる次元が一つ選択される．このことは前述の通り，GBDT が入力空間における次元間の相関性を掴むのを難しくしていると考えられる．

4. 提案方法：BP Boost

前章において一般的な GBDT のアルゴリズムについて述べた．本章では本稿の提案である BP Boost(Back Propagated Boosting; 誤差逆伝播ブースティング法) を提案する．

4.1 入力次元の相関関係

GBDT を幾何学的に捉えるとブースティングマシンが入力空間を四角形の部分空間に分割するアルゴリズムと捉えることができる．分岐を決めるグリッドサーチでは入力の次元は独立に扱われ入力の相関関係は考慮されない．しかし，ブースティングマシンは複数の決定木の出力の和であり，入力空間内の矩形の貼り合わせによって相関関係を考慮した関数近似になっている．

そこで，ブースティングマシンを多段にカスケードする BP Boost を提案する（図 2）．前段のブースティングマシンは入力の相関関係を考慮した特徴量の変換を行い，後段のブースティングマシンは特徴量を基に分類問題を解く．この構造は深層学習の多層ニューラルネットワークの構造を基にした．

BPBoost では 2 段だけでなく複数段のカスケードに対応する．これもまた多層ニューラルネットワークを基にした着想である．複数回の非線形変換を行った方がより入力の相関関係を捉えた特徴量を得ることができるという仮説に基づいている．また，通常の GBDT ライブラリでは全ての入力次元が全てのブースティングマシンに入力する全結合であるが，BPBoost では各”層”の任意の次元をブース

©2021 Information Processing Society of Japan

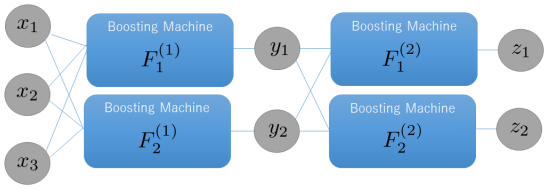

図 2 提案方法の概略図

ティングマシンに入力できるようにする．これにより，画像における畳み込みニューラルネットワークのように局所的な入力の相関関係を捉えるようなブースティングマシンの設計も可能となる．

4.2 BPBoost の学習アルゴリズム

本節では多層構造をとる BPBoost の学習アルゴリズムについて述べる．ニューラルネットワークが多層構造でも学習可能であるのは誤差逆伝播法による勾配計算と勾配降下法による．ブースティングも勾配情報 g_i，h_i を用い弱識別器をフィットさせることにより，近似的にニュートン法を行っている [5]．

本提案のコンセプトはカスケードされたブースティングマシンの最適化をブースティングにより行うことである．逆方向（出力側から入力側への方向）に g_i を伝播させ，各ブースティングマシンで弱識別器を追加することで，全体としてニュートン法による識別器の最適化が行われる．

図 2 の簡単な構成の多段ブースティングマシンを基に学習アルゴリズムを説明する．現在までに各ブースティングマシンに $M-1$ 個の弱識別器がすでに学習されているものとする．まず，学習処理を行うためには順方向処理を行う．すなわち入力 $\boldsymbol{x}$ に対して，中間出力 $\boldsymbol{y}$ 及び最終出力 $\boldsymbol{z}$ を定める処理である．これは単純に各ブースティングマシンの推論を $\boldsymbol{x} \to \boldsymbol{y} \to \boldsymbol{z}$ の順番で行うことによる．

最後段のブースティングマシン $F_c^{(2)}$ は入力 $\boldsymbol{y}$，出力 $\boldsymbol{z}$ が定まれば，式 (4) よりゲイン値 (7) が計算でき，通常のブースティングの学習アルゴリズムが適用可能である．

次に，前段のブースティングマシン $F_c^{(1)}$ の学習について述べる．$F_1^{(1)}$ に新たに M 個目の決定木 $f_{M,1}^{(1)}$ を加えるとする．すると，$f_{M,1}^{(1)}$ により目的関数 L は，

$$L_{(M)} = \sum_{i=1}^{N} \sum_{c} l_c(t_{i,c}, F_c^{(2)}(F_1^{(1)}(\boldsymbol{x}) + f_{M,1}^{(1)}(\boldsymbol{x}), y_2)) \tag{8}$$

と入れ子の関数で表記することができる．まず，内側の関数 $F_{c,i}^{(2)}$ について 1 次のテイラー展開を行うと，

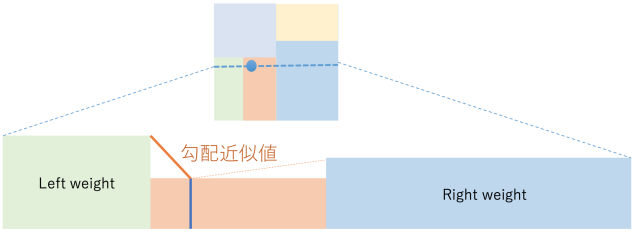

図 3 勾配の近似計算

$$\begin{aligned} F_c^{(2)}(F_1^{(1)}(\boldsymbol{x}) + f_{M,1}^{(1)}(\boldsymbol{x}), y_2) =& F_c^{(2)}(y_1, y_2) + \frac{\partial F_c^{(2)}}{\partial y_1} f_{M,1}^{(1)}(\boldsymbol{x}) \\ =& F_c^{(2)}(y_1, y_2) + \Delta_{c,1}^{(2)} f_{M,1}^{(1)}(\boldsymbol{x}) \end{aligned} \tag{9}$$

となる．ただし $\Delta_{c_{out},c_{in}}^{(R)} = \frac{\partial F_{c_{out}}^{(R)}}{\partial y_{in}}$ は記号による略記である．さらに外側の入れ子を 2 次のテイラー展開すると，

$$\begin{aligned} L_{(M)} =& \sum_{i=1}^{N} \sum_{c} l_c(t_{i,c}, y_c) \\ &+ \left(g_{i,1}\Delta_{1,1}^{(2)} + g_{i,2}\Delta_{2,1}^{(2)} \right) f_{M,1}^{(1)}(\boldsymbol{x}) \\ &+ \frac{1}{2} \left(h_{i,1}\Delta_{1,1}^{(2)2} + h_{i,2}\Delta_{2,1}^{(2)2} \right) f_{M,1}^{(1)}(\boldsymbol{x})^2 \end{aligned} \tag{10}$$

となる．式 (10) は式 (4) の g_i を $g_{i,1}\Delta_{1,1}^{(2)} + g_{i,2}\Delta_{2,1}^{(2)}$ に h_i を $h_{i,1}\Delta_{1,1}^{(2)} + h_{i,2}\Delta_{2,1}^{(2)}$ に置き換えた式と言える．ここから，以下のような勾配と 2 階微分の逆伝搬則を得る．

$$\begin{aligned} g_{i,c_R}^{(R)} &= \sum_{F_c \in F_{in}(c_R)} g_i^{(R+1)} \Delta_{c,c_R}^{(R+1)} \\ h_{i,c_R}^{(R)} &= \sum_{F_c \in F_{in}(c_R)} h_i^{(R+1)} \Delta_{c,c_R}^{(R+1)2} \end{aligned} \tag{11}$$

4.3 ブースティングマシンの勾配

前述の $\Delta_{c_{out},c_{in}}^{(R)}$ はブースティングマシンの入力に対する勾配の情報である．しかしながら，ブースティングマシンは決定木の和であり，微分可能な関数ではない．そこで，何らかの方法により微分値の近似を得ることが必要になる．

ブースティングマシンは決定木の和であるため，2 次元の場合，四角形の足し合わせのような形状をしている．あるデータ点の近傍について表したのが図??である．データ点近傍のある次元に注目した時，関数の形状は階段関数になる．本稿では，データ点近傍の左右の段差に注目し，線形近似を採用した．左右の分岐点とデータ点との直線の傾きのうち勾配の大きい方をそのデータ点に対する勾配とする．ただし，分岐点とデータ点が近すぎる場合，傾きが無限大に大きくなってしまうのを避けるため，上限値と下限値を設定し $[-1, 1]$ 内に収まるようにした．

5. 実装と検証

前述の提案方法の検証を行うために実装を行なった．実

©2021 Information Processing Society of Japan

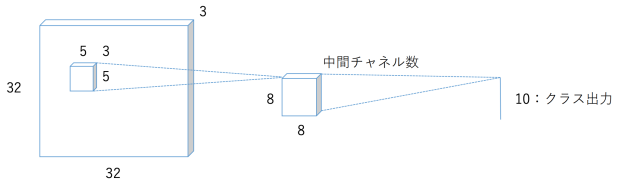

図 4 多層の画像認識ブースティングモデル

装は C++言語を用い，数値計算ライブラリとして Eigen を用いた．

5.1 初期値の設定

各ブースティングマシンに木が存在する場合に，前述の誤差逆伝播によるブースティングが適用可能であるが，初期状態では各層の入出力値が定まらず，勾配計算ができない．通常の GBDT では出力値を 0 として初期状態の勾配計算を行うことで処理を開始するが，提案方式では出力値を 0 とすると次の層のブースティングマシンの入力が全て 0 になり学習が開始できない．

そこで，最終層の出力値は 0 として勾配を計算し，それ以外の層については区間 [-1,1] の一様乱数によって勾配情報を初期化した．これにより，ランダムな木構造がブースティングマシンに追加され，学習の開始が可能となる．

5.2 学習の単位

深層学習の誤差逆伝播法による学習では，全てのパラメータを一気に更新することが一般的である．一方，提案方法では，式 (11) は特定のブースティングマシンのみ変化させた時の式であり，前段のブースティングマシンの出力は後段の複数の出力に対して影響を与えてしまう．

そこで，本稿においては，同じ層を同時に更新する層毎学習とブースティングマシン毎に学習を行うマシン毎学習の 2 種類について実装を行なった．層毎学習においては，ある層のブースティングマシンについての勾配情報を全て計算しその層の学習を行なった後，再度勾配情報を計算し直し別の層の学習に移ることとした．また，マシン毎学習ではマシン毎に勾配情報の計算を行い，学習し，次のマシンに移り勾配計算を行う，という反復処理を行うこととした．

5.3 検証

検証には画像認識タスクでよく用いられる CIFAR-10 データセットを用いた [10]．このデータセットは 32x32 ピクセルの RGB 画像の 10 クラス分類問題である．3072 次元の入力次元を持つ．学習用データセットは 50000 点で，テスト用データセットは 10000 点である．

提案方法で用いる多層のブースティングモデルとして，最もシンプルなモデルとして図 4 のようなモデルを作成した．これは非常に小さな畳み込みニューラルネットワークを模したもので，中間層は 64 個のデータ点である．入力層と中間層の間は 5x5x3 のカーネル状の入力をスライドさせた．この際ストライドを 4 ピクセル開けることで中間層は 8x8 となる．中間層と出力層は全結合の通常の GBDT とした．

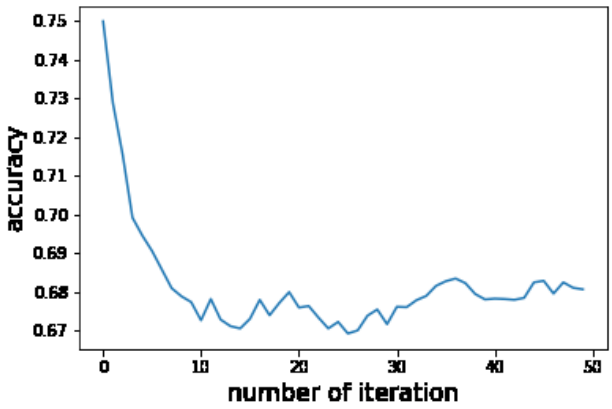

図 5 学習の経過

十分に学習した後のテストデータセットの誤識別率を取得した．従来技術との比較を以下の表に示す．

アンサンブル法	CIFAR-10 error rate
gcForest	38.22% [16]
XGBoost	40.69 % [19]
SVM	50.12 % [18]
KNN	66.14 % [18]
提案方法（層毎学習）	59.68 %
提案方法（マシン毎学習）	67.32 %

表 1 CIFAR-10 分類精度の比較

KNN 以上の識別精度を得ていることから，学習が行われていることは確認できたものの，その精度は他の決定木のアンサンブル手法に比較して劣っていることが見て取れる．

図 5 はあるマシン毎学習の経過である．全てのマシンを 1 度更新する処理を 1 イテレーションとして，横軸にイテレーション数，縦軸にテストデータセットの識別精度を表したものである．現状，学習が進まなくなってしまっており，過学習が起こってしまっていることがこの図から見て取れる．原因の究明と改善が必要である．

6. まとめと今後の展望

本稿においては，GBDT にニューラルネットワークで用いられる誤差逆伝播法に類似した学習アルゴリズムを適用し，実装し，検証を行なった．検証において，本提案方法により学習が可能であることが示されたものの，当初の仮説通りの精度を得るには至らなかった．本稿執筆時におけ

©2021 Information Processing Society of Japan

る課題点として以下があげられる．

- 今回の検証では動作の確認のため小規模な構造のモデルでしか試すことができなかった．より多くの中間層を持つようなモデルにすることで精度の向上が図れる可能性がある．
- 今回の検証では 2 層までしか検証できておらず，多層に拡張することで精度の向上が図れる可能性がある．
- 勾配の近似計算アルゴリズムについて今回は極めて単純な線形補完のみ試したが，より良い近似方法を検討したい．
- 初期状態によって学習の進展が変わってくることがわかった，適切な初期状態の設定が課題である．
- 本提案方法において決定木を弱識別器に使用するのは微分可能性の観点から適切でない可能性がある．可微分な弱識別器の利用を検討したい．

本稿の提案方法は様々な課題を抱えながらも，GBDT に新たな可能性を開くはじめの一歩となるような新しいアルゴリズムと言える．前述の課題を検討することで機械学習の新たな可能性を探究したい．

参考文献

[1] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. ACM.

[2] Corinna Cortes, Mehryar Mohri, and Dmitry Storcheus. Regularized gradient boosting. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pp. 5449–5458. Curran Associates, Inc., 2019.

[3] A. Ferreira and M. Figueiredo. Boosting algorithms: A review of methods, theory, and applications. *Ensemble Machine Learning: Methods and Applications*, Vol. 3, pp. 35–85, 01 2012.

[4] Yoav Freund and Robert E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In Paul Vitányi, editor, *Computational Learning Theory*, pp. 23–37, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[5] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, Vol. 29, No. 5, pp. 1189–1232, 2001.

[6] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, Vol. 28, No. 2, pp. 337–407, 2000.

[7] ROBERT HECHT-NIELSEN. Iii.3 - theory of the backpropagation neural network**based on " nonindent " by robert hecht-nielsen, which appeared in proceedings of the international joint conference on neural networks 1, 593-611, june 1989. 1989 ieee. In *Neural Networks for Perception*, pp. 65–93. Academic Press, 1992.

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pp. 3146–3154. Curran Associates, Inc., 2017.

[9] Yassin Kortli, Maher Jridi, Ayman Al Falou, and Mohamed Atri. A comparative study of cfs, lbp, hog, sift, surf, and brief techniques for face recognition. In *Pattern Recognition and Tracking XXIX*, Vol. 10649, p. 106490M. International Society for Optics and Photonics, 2018.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, Vol. 25, pp. 1097–1105, 2012.

[12] Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Advances in neural information processing systems*, Vol. 27, pp. 1017–1025, 2014.

[13] Atsushi Nitanda and Taiji Suzuki. Functional gradient boosting based on residual network perception. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 3819–3828. PMLR, 10–15 Jul 2018.

[14] Le Cun Yan, B Yoshua, and H Geoffrey. Deep learning. *nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

[15] Yongxin Yang, Irene Garcia Morillo, and Timothy M. Hospedales. Deep neural decision trees, 2018.

[16] Zhi-Hua Zhou and Ji Feng. Deep forest. *arXiv preprint arXiv:1702.08835*, 2017.

[17] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Vol. 2006, pp. 161–168, 06 2006.

[18] Yehya Abouelnaga, Ola S. Ali, Hager Rady, and Mohamed Moustafa. Cifar-10: Knn-based ensemble of classifiers. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1192–1195, 2016.

[19] 藤野知之, 千々和大輝, 税所修, 柏木啓一郎. 高次元データ多クラス識別問題における gbdt ライブラリの実装と改善. *DPSWS 2020*, 2020.

©2021 Information Processing Society of Japan