特　別　研　究　報　告

題　目

サポートベクタに基づく
深層学習の訓練データ削減の検討

指　導　教　員

橋本 昌宜 教授

報　告　者

武田 滉弥

平成31年2月12日

大阪大学 基礎工学部 情報科学科

特別研究報告

報告者　武田滉弥

特別研究報告

題目

サポートベクタに基づく
深層学習の訓練データ削減の検討

指導教員
橋本 昌宜 教授

報告者
武田 滉弥

平成 31 年 2 月 12 日

大阪大学 基礎工学部 情報科学科

平成 31 年度 特別研究報告

サポートベクタに基づく
深層学習の訓練データ削減の検討

武田 滉弥

内容梗概

近年，機械学習の分野ではネットワークの層を深くすることで汎用性と精度を高める深層学習が多く用いられている．深層学習により従来のモデルと比べて高い表現能力を実現できるが，その分膨大な量の訓練データと学習時間が必要である．そこで，訓練データを削減するためにサポートベクタに注目した．サポートベクタとはサポートベクタマシンの分類問題において分類境界に最も近いデータのことである．あるデータ分布を分類する超平面を引くことを考えたとき，その超平面を引くために必ずしも全ての訓練データが必要ではない．分類の境界近辺のデータ，すなわちサポートベクタは非常に重要であるが，境界から遠くにあるデータは分類に大きな影響力はないと考えられる．そこで，本研究では，サポートベクタに注目した訓練データ削減手法を深層学習に適用し，その効果を確認する．評価では，深層学習のモデルである ResNet と CIFAR-10 データセットを用いて，分類精度劣化なく訓練データ数を約 12% 削減し，ニューラルネットワークの学習時間を約 9.5% 削減可能であることを確認した．

目 次

表 目 次

図 目 次

1 序論

深層学習の登場によりニューラルネットワークを中心とする機械学習分野は全盛期を迎えている．以前の多層パーセプトロンと呼ばれるニューラルネットワークが一つの隠れ層による非線形変換の組合せで任意関数の近似を行なっていたのに対して，深層ニューラルネットワーク（Deep Neural Network; DNN）では隠れ層の数を増やし，繰り返し非線形変換を行うことでニューラルネットワークの表現能力を飛躍的に向上させている．大規模画像認識の競技会である ILSVRC [1] では，図 1 のように初期の DNN である AlexNet [2] が提案されてから毎年対数的な精度改善を達成している．これにより学習可能な水準のパラメータ数で複雑な関数の近似が可能となり，ニューラルネットワークの実問題への応用が期待されている．しかし，数億から数十億個以上のパラメータが用いられる最新のニューラルネットワークを学習するにはパラメータ数に比例した訓練データが必要であり，膨大な計算資源と時間が要求される．さらに図 2 に示す ILSVRC で優勝を収めたネットワークモデルの規模に対する年度別推移からわかるようにニューラルネットワークの規模は指数的な増加を見せており，訓練データの増加による問題は深刻化すると予想される．

本研究では，この問題に取り組むべく，DNN の精度を損なわない訓練データの削減手法を提案する．提案手法の基本アイディアとして訓練データにおける重要度の違いに着目し，ニューラルネットワークの学習を重要度の高い訓練データのみで行うことで学習の効率化を図る．これを実現するために本研究ではサポートベクタマシン（Support Vector Machine; SVM）[3] におけるサポートベクタの役割に注目する．サポートベクタは SVM の空間分離において境界の超平面を定める際に用いられるデータ群を指す．そのためサポートベクタは同一クラスの訓練データの最外部に位置する特徴を持っており，ニューラルネットワークの学習においても重要な役割を果たすことが予想される．サポートベクタに基づく訓練データ削減の既存研究では手書き数字認識データセットである MNIST に対して初期の成果が得られているが，より複雑な問題に対する学習可能性については明らかにしていない．

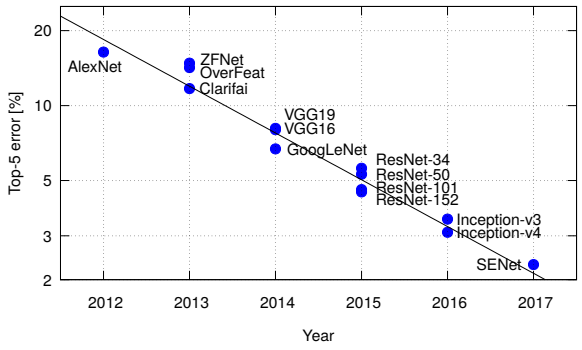


図 1: ImageNet に対する年度別の誤差推移

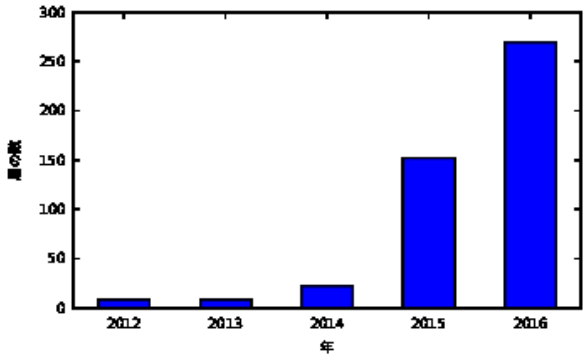


図 2: ILSVRC 優勝モデルの年度別の規模推移

そこで本研究では最新のニューラルネットワークの一つである ResNet [4] と 画像分類用のデータセットである CIFAR-10 を用いて DNN の学習におけるサポートベクタの効果を明らかにすることを目的する．

本稿の構成は以下の通りである．まず第 2 節では，機械学習と深層学習の概念について説明し，本研究で用いられるニューラルネットワークと SVM の基礎について述べる．次に第 3 節では，サポートベクタを用いた訓練データの削減手法について説明し，複数の 2 次元データに対する予備実験を行うことによってニューラルネットワークの学習におけるサポートベクタの効果を視覚的に確かめる．さらに第 4 節では最新のニューラルネットワークである ResNet と画像データセット CIFAR-10 を用いた評価実験の結果を示し，考察を述べる．最後に第 5 節で結論について述べる．

2　関連研究

本章では提案手法を理解するために必要な関連研究として，機械学習とサポートベクタマシンについて述べる．まず初めに機械学習で主に用いられるネットワークモデルであるニューラルネットワークの構成要素と基本原理について説明し，その後機械学習手法の 1 つであるディープラーニングの仕組みについて述べる．続いて SVM について説明する．最後に，既存の訓練データ削減手法について述べる．

2.1　機械学習

機械学習とは，あらかじめ人間がプログラムにすべての動作を書き込んでおくのではなく，機械が自ら学習してデータの規則性や法則性を解析することができるようになる手法である．これにより，何度も学習を重ねトレーニングすることで特定の動作を行うことが可能になる．例として，ウマの画像とシマウマの画像を識別することを考える．与えられた画像をウマかシマウマか見極めるために，ウマの画像とシマウマの画像を大量に読み込ませて学習を重ねる．このとき，体の縞模様の有無に注目しなさいという指示を与えて学習させることで，新たなウマもしくはシマウマの画像を読み込ませた際，縞模様があるかないかに注目して識別するようになる．このように，あらかじめ答えのわかったデータを読み込ませて学習を行うことを教師あり学習といい，主流な機械学習手法の一つである．逆に教師なし学習というものもあるが，本研究への関連性は低いため割愛する．

2.2　ニューラルネットワーク

ニューラルネットワークとは，人間の脳の構造を模倣して作られたネットワークである．図 3 のように，ノードとシナプスから構成される．ノードとは人工ニューロンとも呼ばれ，

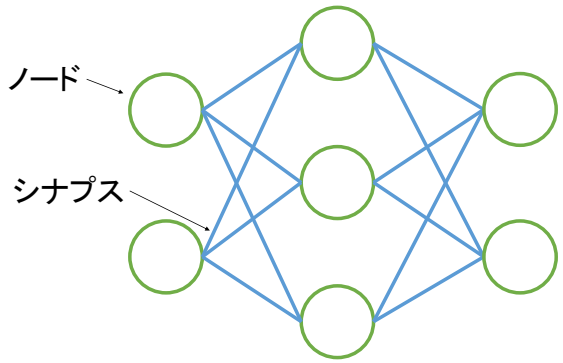


図 3: ニューラルネットワークの構造

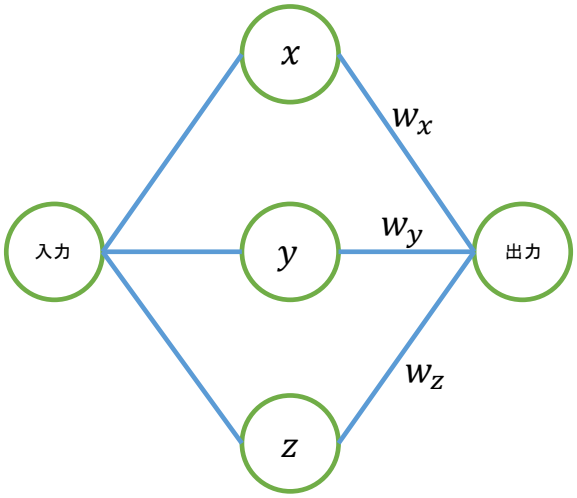


図 4: 買い物に行くかどうかを判断するモデル

人間の脳にある神経細胞を真似て作られた数式的なモデルであり，任意の演算を行うことができる．ノードはお互いに結合することで計算結果を渡すことができるようになっており，この結合に用いられているのがシナプスである．このようにして，脳の神経回路網に似たネットワークを成している．

ニューラルネットワークが人間の脳の思考回路を真似ることができていることを説明するために，例として「週末に買い物に行くか」を考える．買い物に行くかどうかのファクターとして，

1. x:家から近いか
2. y:天気は良いか
3. z:割引セールをやっているか

の 3 つがあるとする．これらの出力は 0 か 1 で表せるとし，例えばお店が家から近ければ 1，遠ければ 0 である．残り 2 つも同様とする．ここで，それぞれのファクターは等価値でないことに注目する．例えば，どれだけ家から近くても天気が悪ければ絶対に行かないかもしれないし，どれだけ家から遠くて天気が悪かったとしても，割引セールをやっていれば絶対に行くかもしれない．このように，ファクターの重要度を重みという．ここで，x, y, z の重みをそれぞれ，$w_x = 2, w_y = 4, w_z = 8$ とすると，買い物に行くかどうかを判断するモデルを図 4 に示す．このとき，買い物に行くかどうかを表す式を，

$$xw_x + yw_y + zw_z > 5 \tag{1}$$

と定義する．この 5 を閾値といい，閾値を変えることで判断基準を変えることができる．閾値が 5 の場合は，家から近く天気も良い場合と，割引セールをやっている場合に買い物に行くことになる．つまり，家から近く天気も良ければ割引セールをやっていなくても行くし，割引セールをやっていれば他のファクターは無視するということになる．次に閾値を 10 とし，

$$xw_x + yw_y + zw_z > 10 \tag{2}$$

とすると，天気が良くて割引セールをやっていれば買い物に行くことになり，家から近いかどうかは結果に一切影響しない．このように，閾値によって買い物に行くかどうかの判断基準が変わる．この例では簡単なネットワークを考えたが，ノードを増やしより複雑なネットワークを用いれば，さらに高度な判断を行うことができる．

2.3 深層学習

深層学習とは，深層ニューラルネットワーク (DNN) を用いた機械学習手法の一つである．DNN が通常のニューラルネットワークと比べて異なる点は，層が深く重なっているという点である．通常のニューラルネットワークでは中間層が 1 層しかないのに対し，DNN には中間層が 2 層以上あり，これが DNN の特徴と言える．通常のニューラルネットワークと比べて必要な学習データや学習時間，消費電力が大きい分，より高い精度で学習することができる．

通常，機械学習では特徴量と呼ばれる識別の指標となるファクターを事前に設定しなければならない．しかし，深層学習では，各層で行われる非線形変換の式を更新することによって自動的に特徴を抽出できる．ここでは例として，渦巻状のモデルを識別する結果を図 6 に示す．図の背景がオレンジの部分が正，青が負，白が 0 に識別されることを表現していて，点がその座標におけるデータを表している．左側の入力から層を進むごとに空間

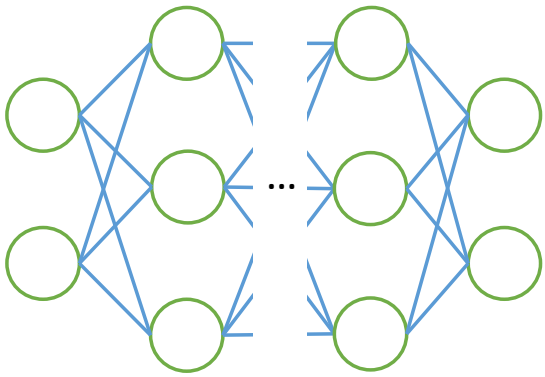


図 5: 深層ニューラルネットワークの構造

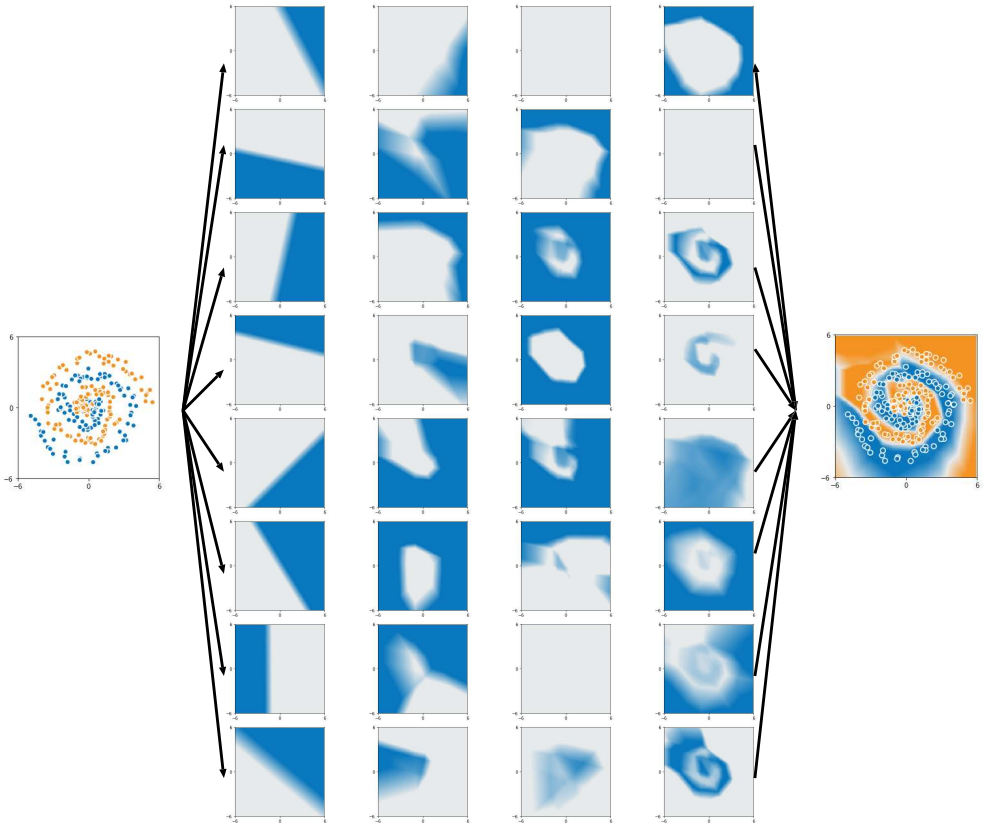


図 6: 特徴検出の大まかな流れ

の分割が複雑な曲線で行われるようになっていき，最終的には渦巻状のモデルを表現になることがわかる．

ディープラーニングで画像や音声などのデータを分析するとき，データには多くの特徴が含まれている．これを損失無く抽出するには特徴一つごとに分割するのが理想であるが，特徴が等間隔に並んでいるとも限らず，そもそも初めは何が特徴なのかすらわかっていないため非常に難しい．そのためなるべく特徴を損失させずデータを分割することが重要となる．ここで画像識別を例に考える．もしカンガルーかどうかを判定する基準が仮に腹袋の有無だとすると，もしその腹袋の部分がデータ分割の境界線上にあって二つに分かれてしまうと判定ができなくなる．このように，データの分割の方法次第では特徴が失われてしまう可能性がある．これを解決するために，オーバーラップという手法を用いる．これはデータの分割の際に，少しずらしてデータ同士をある程度かぶせて分割する手法である．簡単のために「あいうえお」という文字列で説明すると,「あいう」と「えお」という分け方ではなく,「あいう」「いうえ」「うえお」という分け方をするという具合である．これにより特徴の損失を抑えることができるため，データの分析精度の上昇に貢献する．

データの分割をして第 1 層の各ノードにデータが渡ると，ノードごとに処理が行われて

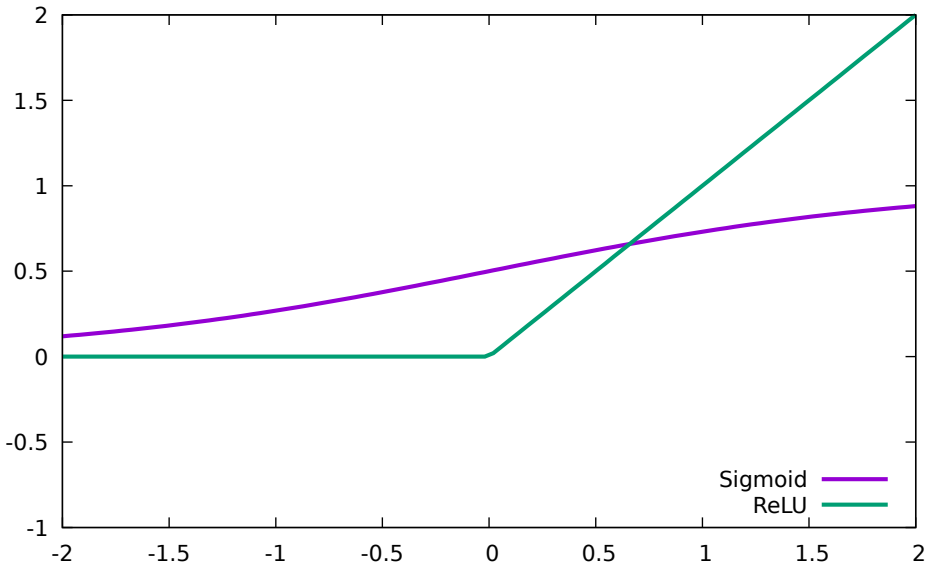


図 7: 活性化関数

次の層のノードにデータが伝播する．このとき，各ノードはまだ精度の高い結果を出すのに必要な特徴と不必要な特徴 (誤差) を持っている状態である．次の層へすべての情報を渡すと誤差ももれなく伝えてしまうし，誤差を伝えないために渡す情報を削り過ぎると特徴まで失われてしまう．次の層へすべての情報を渡すと誤差ももれなく伝えてしまうし，かといって誤差を伝えないために渡す情報を削り過ぎると特徴まで失われてしまう．ここで，次の層へデータを渡す際に施される計算式を活性化関数という．活性化関数は元のデータを少し変化させることで次の層へよりはっきりと特徴を伝えることができる他，過学習を抑えられるなどの効果がある．活性化関数にはいくつか種類があるが，ここでは主要なシグモイド関数と ReLU 関数の二つについて説明する．

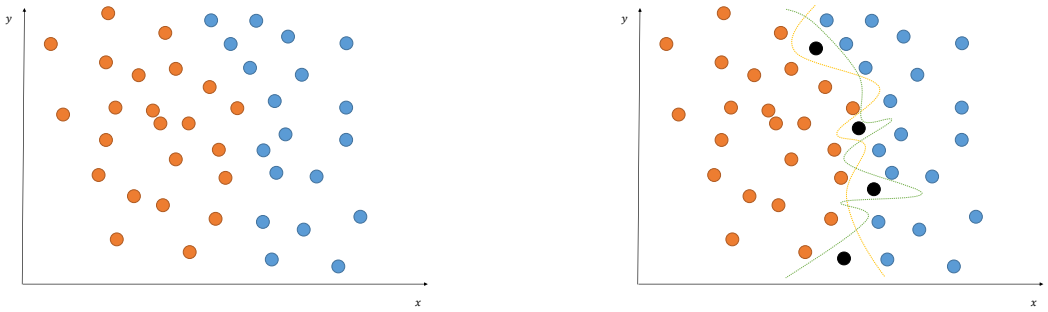
図 7 に示すシグモイド関数は，入力の値が大きいほど 1 に近づき，小さいほど 0 に近づく関数である．特徴としては単調増加関数であり，元となる入力がどれだけ大きくても 1 にはならず，どれだけ小さくても -1 にならないことが挙げられる．シグモイド関数を式 (3) に示す．

$$h_{\mathrm{sigmoid}}(x) = \frac{1}{1+e^{-x}} \tag{3}$$

シグモイド関数はある程度元の入力の特徴を生かすことができるが，その反面計算量が多くなってしまうという欠点がある．また，-1 や 1 に近づけば近づくほど値の変化がゆるやかになるという特徴から，シグモイド関数を微分したときの勾配 (傾き) は

$$h'_{\mathrm{sigmoid}}(x) = (1 - h_{\mathrm{sigmoid}}(x))h_{\mathrm{sigmoid}}(x) \tag{4}$$

となり，それらの値の近辺でほぼ 0 である．つまり，それらの値では特徴が平坦になりや



(a) 訓練データ　　(b) 識別超平面の違いによる分類結果の違い

図 8: 同じデータに対する異なる分類の例

すいため，層が深くなるにつれ特徴が失われやすく学習が進まなくなることがある．したがって計算量の問題とも併せて，層の少ない通常のニューラルネットワークではよく用いられているが，DNN ではあまり用いられない．

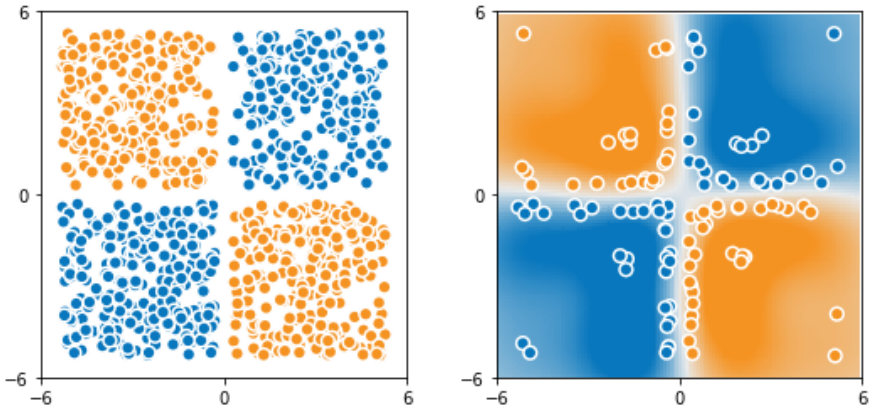
シグモイド関数の欠点を踏まえ，DNN では図 7 に示す ReLU (Rectified Linear Unit) 関数を用いることが多い．ReLU 関数は入力が 0 以下のときは 0 を，0 以上のときは入力の値をそのまま出力する関数である．シグモイド関数と比べてシンプルな処理であり，式は以下のようになる．

$$h_{\mathrm{ReLU}}(x) = \max(0, x) \tag{5}$$

ReLU 関数は線形であるため，微分しても勾配が 0 になることはない．これにより，先ほどシグモイド関数で課題となっていた特徴の損失により学習が進まなくなる問題に陥ることはない．計算も非常にシンプルであることから，DNN では ReLU 関数を用いることが多い．

2.4　サポートベクタマシン

サポートベクタマシン (SVM) [3] とは，教師あり学習によるパターン認識モデルの 1 つであり，分類問題や回帰問題に対して用いることのできる手法である．SVM の考え方を述べるにあたり，図 8(a) のような分布図を分類する問題を考える．図 8(a) は，それぞれを横軸を x，縦軸を y とした分布を表したグラフである．このように分布するオレンジのデータと青のデータを 1 つの超平面で分類することを考える．分割方法は無数にあるが，任意の分割方法が可能である訳ではない．以下に，例として 2 つの引き方を図 8(b) に示す．図 8(b) 示す 2 つの分割はどちらも正しくオレンジと青を分類できているため，不正解とは言えない．仮にどちらの線も分類にふさわしい線としたとき，黒で示された入力データを考える．これらの入力データは，一方の線ではオレンジ側に属し，他方の線では青側に属してしまうため，これらの線は分類としてふさわしいとは言えない．このように，分類する境界線付



(a) 訓練データ　　(b) サポートベクタ

図 9: XOR に対するサポートベクタでのデータ削減の様子

近のデータはいわばどちらに分類するか曖昧なデータということになる．この曖昧なデータをできるだけ正しく分類するための方法が SVM である．

SVM の構成要素にはサポートベクタとマージンがある．サポートベクタは分類する境界線付近のデータのことで，マージンは分類する境界線とデータとの距離のことである．前述のいわゆる曖昧なデータは境界線に近い，つまりマージンの小さなデータということである．マージンの小さなデータが多いと分類の精度が下がるため，マージンをできるだけ大きくすることでそういったデータを減らすことができる．これはマージン最大化と呼ばれる．また，誤った分類を防ぐためには境界線付近のデータが正しく分類されれば良いため，サポートベクタに注目して分類を行えば良いと言える．この考え方は後述の提案手法で用いられる．サポートベクタによるデータ削減の様子を図 9 に示す．

ここまでは簡単のために綺麗に分類できるデータを例に考えたが，ここからは図 10 のようなやや特殊なデータが与えられた場合を考える．このデータに対して先ほどと同じ線で分類すると誤った識別が起きてしまう．誤って識別されているデータに対して無理やり正しい分類をしてしまうと，これらの貴重なデータを失ってしまうことになる．このように，元のデータに過剰に適合してしまい，新しく与えられたデータへの予測精度が落ちてしまうことを過学習 [5] という．過学習を起こさず与えられたデータに対して柔軟に予測ができることを汎化性 [6] があるといい，誤った識別をある程度許すことが汎化性を高めることに繋がる．汎化性において，誤った識別を許さない手法をハードマージンといい，誤った識別をある程度許す手法をソフトマージンという [7]．ソフトマージンでは誤った識別をある程度許すが，それにペナルティを設けることでマージン最大化と誤った識別の許容のバランスを取っている．このペナルティはコストとも言い，コストパラメータ C の値を調整することで誤った識別をどの程度許容するか決めることができる．ここで，マージン最大化

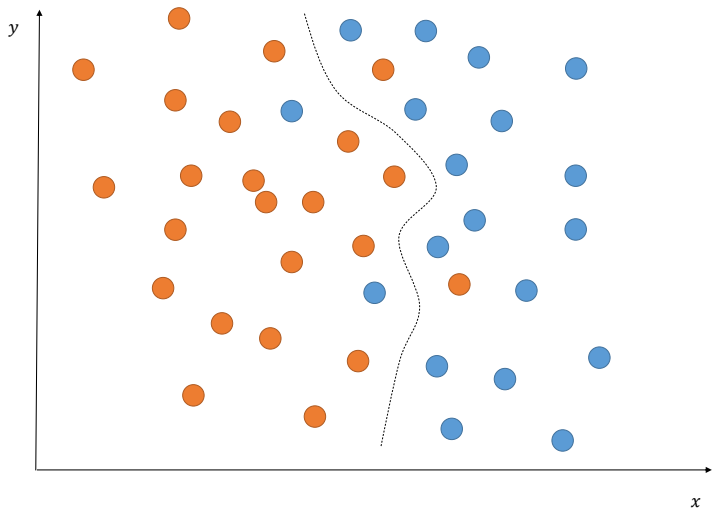
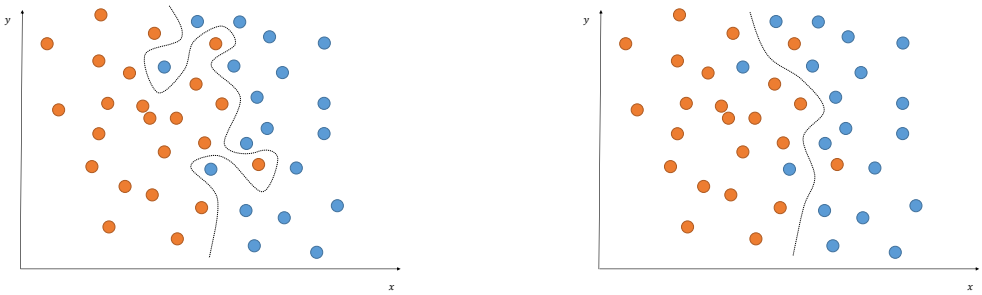


図 10: 誤った分類が生じる例



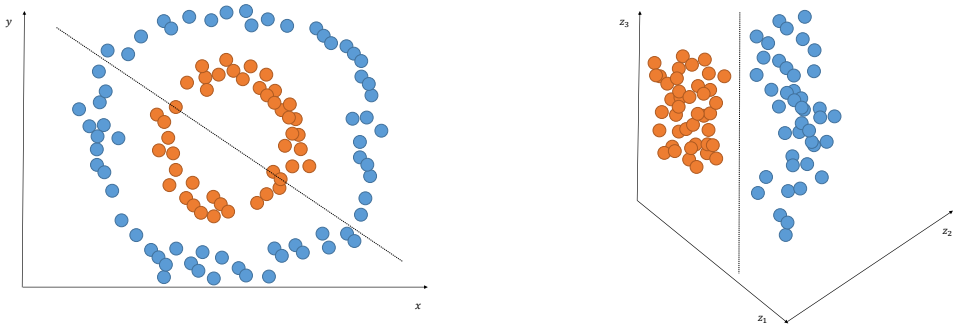
(a) ハードマージン　　(b) ソフトマージン

図 11: ハードマージンとソフトマージン

は以下の最小化問題に置き換えられる．

$$\min \left\{ \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \xi_i \right\} \tag{6}$$

ξ はスラック変数といい，そのデータが正しく識別されていれば 0，正しく識別されているがマージン内に入っていれば $0 < \xi \leq 1$，誤識別の場合は $\xi > 1$ となる．つまり，誤識別が多いほど第二項が大きくなるため，最小化しようとすれば ξ を小さくなるような調整が行われることになる．先ほどコストパラメータ C の値を調整することで誤った識別をどの程度許容するか決めることができると述べたが，これは C が大きいほど第二項の値が大きくなるためであり，$C \to \infty$ の場合をハードマージンという．ソフトマージンの場合は C の値を調整することで誤った識別をどれだけ許容するかを変えることができる．



(a) 線形分類不能なデータの例　　(b) 線形分類可能なデータに変換

図 12: カーネル法による非線形データの変換

SVM では，クラス分類に RBF (Radial Basis Function) カーネルと多項式カーネルの 2 つのカーネル法が広く用いられる．カーネル法とは SVM の提案が発端となって 1990 年代から発展したデータ解析法であり，非線形データを扱う際に便利な方法である．例として，図 12(a) のように円状に分布した非線形なデータは線形分類できない例を考える．

$$(z_1, z_2, z_3) = (x^2, y^2, \sqrt{2}xy) \tag{7}$$

カーネル法を用いた座標変換により，線形分類不能であったデータを線形分類可能なデータへと変換することができる．本研究の評価で用いる RBF カーネルと多項式カーネルのうち，RBF カーネルは以下の式で表される．

$$K_r(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{8}$$

RBF カーネルは頻繁に使われるカーネル関数であり，SVM のハイパーパラメータとして C と γ を用いることが多い．続いて多項式カーネルを表す式は以下のとおりである．

$$K_p(x_i, x_j) = (x_i x_j + r)^d \tag{9}$$

これは d 次の多項式で表されるカーネルであり，r がハイパーパラメータとして追加されている．これらの 2 つのカーネル関数を用いたとき，それぞれ訓練データの削減にどれほどの効果があるのかを 3 節と 4 節で述べる．

2.5　訓練データ削減手法

2 つの訓練データ削減に関する既存研究について考察する．まず 1 つ目は，Nguyen らが行った 2 次元データに対してサポートベクタを用いて訓練データ削減と分類精度がどのように変わるかを検証した研究である [8]．同研究では，2 次元データであるガウシアン，正弦，楕円の 3 つの異なるデータ分布に対する分類問題に対し，サポートベクタによってデー

図 13: MNIST データセット

表 1: Dahiya らが行った実験の結果

訓練データ数	データ削減方法	SVM カーネル	精度 (%)
60,000	なし	なし	97.62
20,000	サポートベクタ	RBF	97.66
10,000	サポートベクタ	多項式	97.48
20,000	ランダム	なし	95.32
10,000	ランダム	なし	94.67

タ削減を行ったときの分類精度とのトレードオフについて述べている．サポートベクタによりどれだけのデータ削減が可能かの上限を導出しており，あらかじめ定めた許容範囲内であれば適切な量の訓練データ削減を行うことができるとの結果が出ている．2 次元データの分類問題において，サポートベクタによる訓練データの削減は有効であることがわかる．

次に 2 つ目の既存研究として，Dahiya らが行ったニューラルネットワークに対してサポートベクタを用いた訓練データの削減 [9] について述べる．同研究では，図 13 に示す MNIST と呼ばれる 0 から 9 までの手書き数字の画像データセットを分類する問題 [10] に対し，サポートベクタによって訓練データを削減しつつ精度を保っている．その実験結果を表 1 に示す．同研究では訓練データ 60, 000 枚に対し，RBF カーネルの SVM を用いて 20,000 枚に減らした上で，精度劣化が発生しなかったと報告している．また，多項式カーネルの SVM を用いた場合は訓練データを 10,000 枚まで減らすことに成功しており，こちらもほとんど精度の劣化はなかった．対照実験としてサポートベクタの代わりにランダムに同じ数だけ抽出した場合も実験しており，これをサポートベクタを抽出した場合と比較すると，2 から 3% 精度が向上している．このことから，無作為に訓練データ数を削減すると精度が落ちる問題でも，サポートベクタを用いてデータを抽出することで精度劣化を起こさず分類可能になることがわかる．

3　深層学習の訓練データ削減

本節では，サポートベクタを用いて深層学習の訓練データの削減について述べる．2 節で述べたとおり，深層学習には大量の訓練データと学習時間が必要になるため，サポートベクタを用いて効率的に訓練データを抽出し，比較的重要度の低いデータを除くことで訓練データの削減を目指す．

3.1　サポートベクタを用いた訓練データ削減

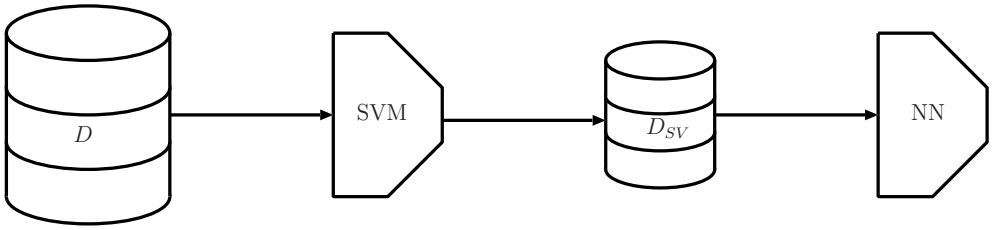
本手法では，サポートベクタを用いて訓練データをすることで，ニューラルネットワークの学習時間を削減する．提案手法と通常のニューラルネットワークの学習を図 14 に示す．サポートベクタを用いた訓練データの削減手法の手順は以下のとおりである．

1. データセット D を SVM で学習し，サポートベクタ D_{SV} を抽出
2. D_{SV} を訓練データとして用いて対象のニューラルネットワークを学習

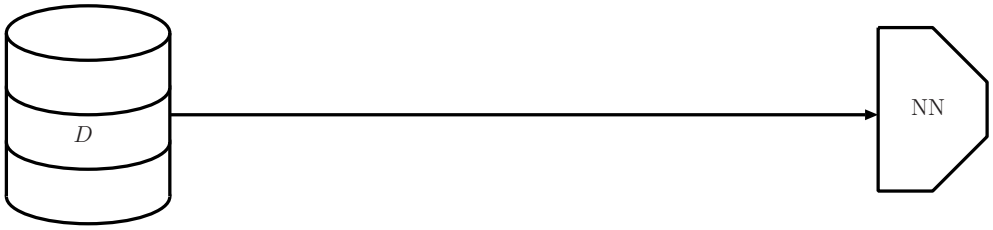
サポートベクタによるデータ削減を二次元データに適用した例を図 15 に示す．サポートベクタを用いるにあたって，以下の点に着目する．

- 訓練データはどれだけ削減できたか
- 学習時間はどれだけ減少したか
- 分類精度への影響はどうか
- 上記 3 項目について，同じ数だけランダムにデータ抽出を行った場合はどうか

サポートベクタを用いたとき，訓練データがわずかにしか削減されなければあまり意味のないものとなる．訓練データの削減量に見合わない分類精度の劣化が起きるのであれば尚更である．また，訓練データを大幅に削減できたとしても，分類精度がある程度の水準で維持できていなければならない．これらの点に注目しながら訓練データの削減を検討していく．しかし，もしこれらの条件をクリアし，サポートベクタだけを抽出して訓練データを削減しつつも精度がほとんど落ちないことが確認できたとしても，ランダムにデータ抽出を行ってデータ削減を行った場合と結果がほとんど変わらない場合，サポートベクタを用いた意味がないため望ましくない結果となる．扱う問題が簡単なものであったり，もともと与えられている訓練データが過剰である場合はこのようなことが起こる可能性が十分に考えられるため，留意したい．また，サポートベクタを用いた際に分類精度が維持されず劣化してしまった場合，図 15(c) のようにサポートベクタとして抽出されなかったデータを補うことで訓練データを補強することを考える．これにより，本来の空間形成を担って



(a) サポートベクタを用いたニューラルネットワークの訓練データ削減



(b) 通常のニューラルネットワークの学習

図 14: サポートベクタによる訓練データ削減

いたデータの一部が補われるため，訓練データがサポートベクタだけだったときと比較して精度の向上が期待できる．

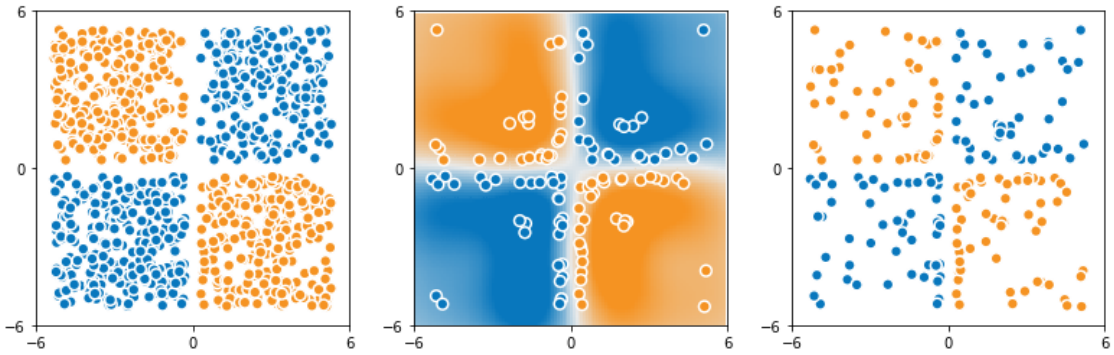
3.2 予備実験

以上で述べたサポートベクタを用いた訓練データ削減手法の有効性を確認するために，予備実験として，TensorFlow Playground [1] と同等の機能を持つプログラムを用いて評価を行う．使用するプログラムでは 2 次元データをニューラルネットワークで学習することができ，活性化関数，中間層の数，ニューロンの数を任意に変更可能である．評価で使用する 2 次元データは，2 種類の回帰問題としてガウシアン (R_GAUSS)，直線 (R_PLANE) と，4 種類の分類問題としてガウシアン (C_GAUSS)，スパイラル (C_SPIRAL)，円 (C_CIRCLE)，XOR (C_XOR) である．データ削減量と精度を評価し，分類結果を視覚的に分析する．評価では，活性化関数は ReLU を用いた．中間層は 3 層で，それぞれの層のニューロン数は 5 つである．詳細な結果については 4 節で述べるが，6 種類の 2 次元データに対してサポートベクタによるデータ削減を行ったところ，6 つ全てにおいてほとんど精度の劣化は起きなかった．

予備実験では，以下の 3 つの方法で抽出した訓練データで分類・回帰精度と学習時間の評価を行う．

(A) 全訓練データを使用した学習

[1] https://playground.tensorflow.org/



(a) 元データ　(b) サポートベクタ　(c) サポートベクタ+ランダムに抽出したデータ

図 15: サポートベクタによるデータ削減と無作為抽出データによるサポートベクタの補強

表 2: 予備実験結果

データセット	(A) 全訓練データ		(B) サポートベクタ		(C) ランダム抽出	
	データ数	ロス	データ数	ロス	データ数	ロス
C_GAUSS	1,000	0.001	126	0.001	126	0.001
R_PLANE	2,400	0.004	292	0.001	292	0.006
R_GAUSS	2,400	0.192	263	0.011	263	0.563
C_SPIRAL	1,000	0.009	137	0.039	137	0.105
C_CIRCLE	1,000	0.001	129	0.002	129	0.177
C_XOR	1,000	0.003	117	0.001	117	0.005

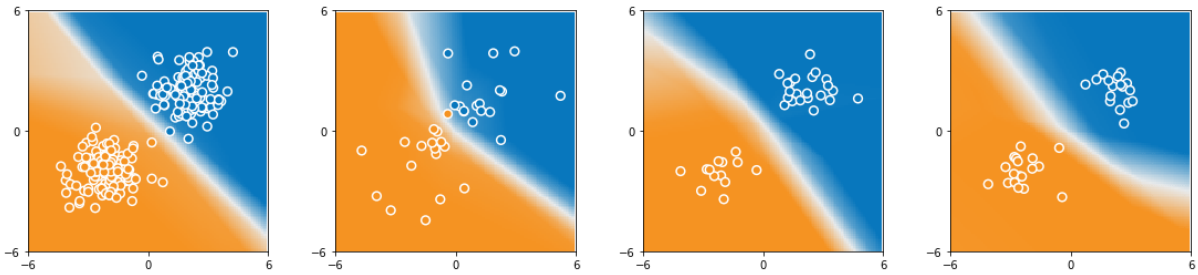
(B) サポートベクタだけを使用した学習

(C) サポートベクタと同数のデータをランダムに抽出して学習

学習は 100 エポック行った．ただし，C_SPIRAL は他の関数と比べて複雑であり，学習に時間がかかるため 500 エポック行った．中間層はいずれも 3 層で，各層にはニューロンが 5 つずつ割り当てられている．分類精度はロスで評価し，ロスが小さいほど高精度である．

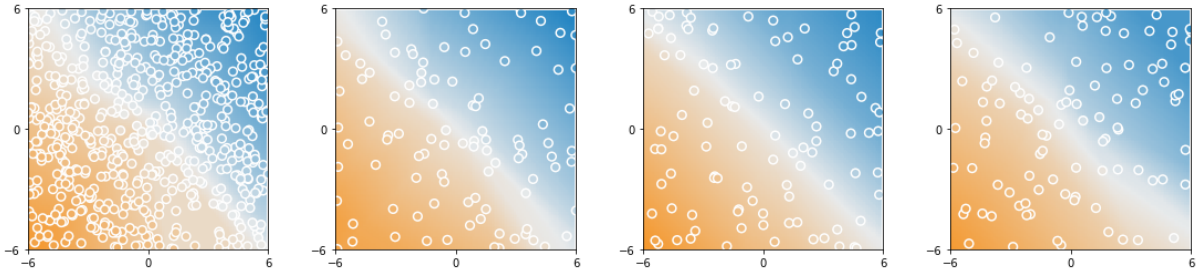
C_GAUSS はサポートベクタのみで学習を行っても精度劣化は起きなかった．しかし，分類が比較的簡単であるため，ランダム抽出したものであっても精度劣化はほとんど見られず，サポートベクタが有効であったかは判断しかねる．図 16 より，元データのように綺麗な直線で分類されてこそいないが，分類自体はどれも適切に行われているため問題ないと言える．

R_PLANE も C_GAUSS と同様に精度劣化は起きなかった．また同様に，ランダム抽出したものであっても精度劣化はほとんど確認できなかったのでサポートベクタの有効性につ



(a) (A) 全訓練データ　(b) (B) サポートベクタ　(c) (C) ランダム 1　(d) (C) ランダム 2

図 16: C_GAUSS データセットにおける評価



(a) (A) 元データ　(b) (B) サポートベクタ　(c) (C) ランダム 1　(d) (C) ランダム 2

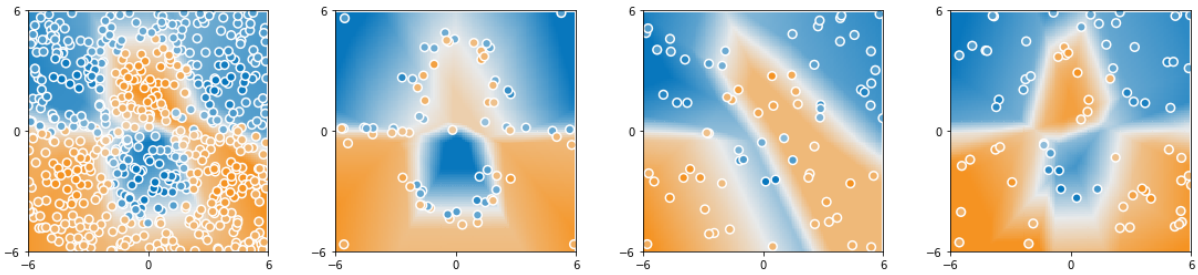
図 17: R_PLANE データセットにおける評価

いては不明である．図 17 を見ても，どれも同じように分類が行われているため C_GAUSS と同様に問題ないと言える．

R_GAUSS はサポートベクタのみで学習を行っても精度劣化は起きなかった．また，ランダム抽出したものでは精度劣化が起こり上手く学習が行えなかったものが見られた．比較的複雑なデータセットである R_GAUSS においては，サポートベクタの有効性が示せた．図 18 を見てみると，サポートベクタだけでも元データと同じように分類が行えており問題ない．また，ランダム抽出だと不適切な分類を行ってしまう場合もあった．

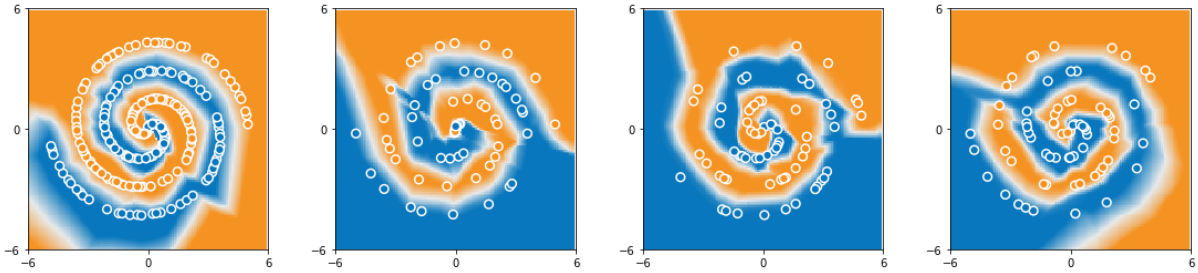
C_SPIRAL では，わずかではあるが精度の劣化が見られたが，致命的とまでは言えず許容できる範囲内であった．また，ランダム抽出したものであっても大きな精度劣化は見られず，サポートベクタが有効であったかは定かではない．図 19 を見ても，どれも同じように分類が行われているため問題ないと言える．

C_CIRCLE はサポートベクタのみで学習を行っても精度劣化は起きなかった．しかし，これも C_GAUSS と同様に分類が比較的簡単であるため，ランダム抽出したものであっても精度劣化はほとんど見られず，サポートベクタが有効であったとは断言できない．図 20 を見てみると，ランダム抽出のときやや角ばった分類になってしまうこともあったが，分類自体はいずれも許容範囲内で問題なく行えていた．



(a) (A) 元データ　(b) (B) サポートベクタ　(c) (C) ランダム 1　(d) (C) ランダム 2

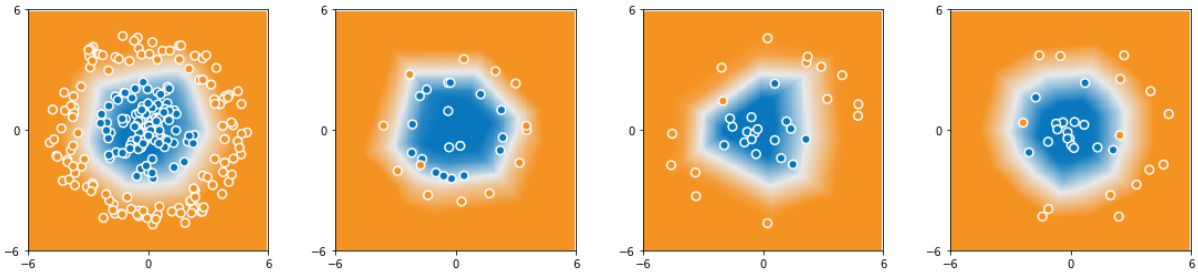
図 18: R_GAUSS データセットにおける評価



(a) (A) 元データ　(b) (B) サポートベクタ　(c) (C) ランダム 1　(d) (C) ランダム 2

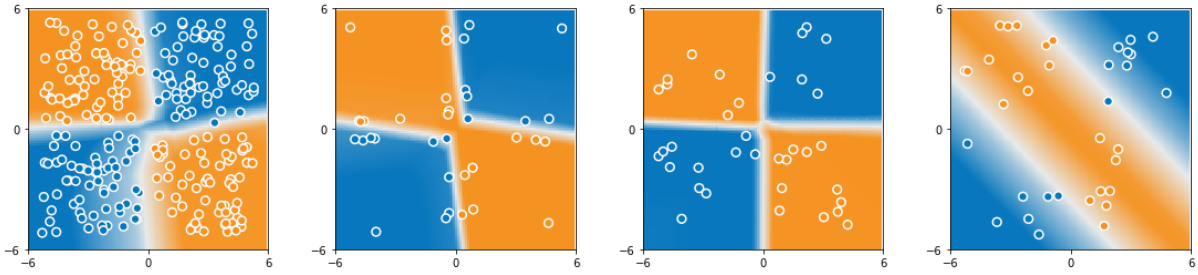
図 19: C_SPIRAL データセットにおける評価

C_XOR も精度劣化はなかった．また，ランダム抽出したものであっても精度劣化はほとんど確認できなかったが，一度だけ分類に大きく失敗してしまった場合があった．そのため，サポートベクタは多少なりとも有効であったのではないかと考えられる．



(a) (A) 元データ　(b) (B) サポートベクタ　(c) (C) ランダム 1　(d) (C) ランダム 2

図 20: C_CIRCLE データセットにおける評価



(a) (A) 元データ　(b) (B) サポートベクタ　(c) (C) ランダム 1　(d) (C) ランダム 2

図 21: C_XOR データセットにおける評価

4 評価

本節では，3 節で提案したサポートベクタによる深層学習の訓練データの削減手法を最新のニューラルネットワークの一つである ResNet 18 [4] に適用し，訓練データの削減量と分類精度の観点から評価を行う．ResNet は画像認識で高認識精度を達成しているネットワークの一つである．評価には，CIFAR-10 [11] データセットを用いる．

CIFAR-10 は，画像データセットに対する分類問題である．CIFAR-10 とは訓練データ 50,000 枚とテストデータ 10,000 枚の合わせて 60,000 枚ある画像データセットであり，図 22 に示すように，各画像は 32×32 ピクセルで RGB の 3 チャンネルのカラー画像である．それぞれの画像にはクラスを表すラベルがついており，クラスラベルには飛行機，自動車，鳥，ネコ，シカ，イヌ，カエル，ウマ，船，トラックの 10 種類がある．今回はそのうち 5 ラベルを分類する場合と，10 ラベル全てを分類する場合の 2 パターンを行う．5 ラベルを分類する際に使用するラベルは飛行機，鳥，シカ，カエル，船の 5 つである．いずれも学習回数は 100 エポックである．

多クラス分類において，一対一分類器と一対多分類器では異なるサポートベクタが選択されるため，両方の多クラス分類識別に対して評価を行う [12]．一対一分類器では，クラス C_1 からクラス C_N の N クラスの分類を ${}_NC_2$ 個の識別器を用いて分類する．一対多分類器では，N クラスの分類をある特定のクラスに分類されるかそれ以外のクラスのどれかに分類されるかの 2 クラス分類問題を解く識別器を N 個利用する．

評価では，(A) 全訓練データを使用する場合，(B-1) 一対一分類器でサポートベクタのみを使用する場合，(B-2) 一対多分類器でサポートベクタのみを使用する場合の 3 つに対して，分類精度と学習時間の関係を確認する．5 クラスに対する実験結果を表 3 に示す．表 3 より，提案手法が従来の全訓練データを使用した手法と同等の分類精度を達成しつつ，計算時間を削減していることが分かる。例えば，計算精度に着目すると，(A) の計算精度が 78.300% であるのに対し，(B-1) と (B-2) の計算精度はそれぞれ 78.580% と 78.340% である．また，

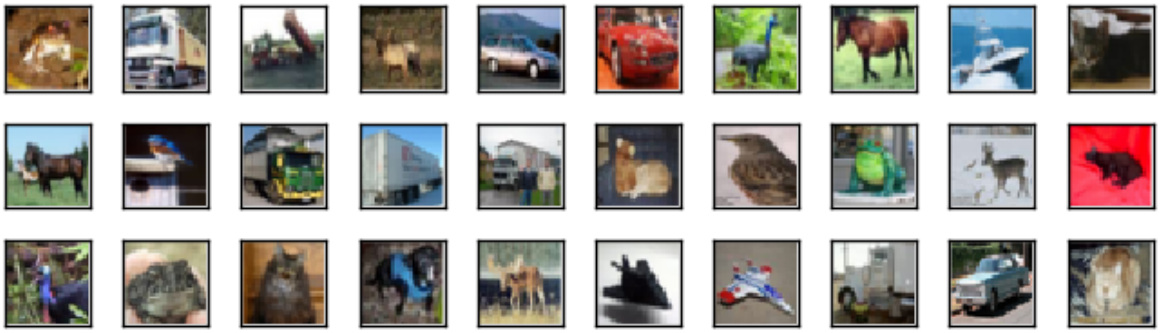


図 22: CIFAR-10 データセットの一部

計算時間に関しては，(A) が 1058.243 秒必要とするのに対して，(B-1) と (B-2) は 961.395 秒，957.629 秒要している．換言すれば，サポートベクタのみを用いることで，(B-1) では計算時間を 9.16% $(= 1 - 961.395/1058.243)$，(B-2) では 9.50% $(= 1 - 957.629/1058.243)$ 削減した．10 クラスに対する実験結果を表 4 に示す．5 クラスに対する評価と同様に，提案手法が従来手法と同等の計算精度を維持しつつ，計算時間を削減した．表 4 より，(B-1) では 2.54% $(= 1 - 2062.906/2116.688)$，(B-2) では 2.66% $(= 1 - 2060.405/2116.688)$ 計算時間を削減した．以上より，5 ラベルと 10 ラベルの両評価において，提案手法が精度劣化を起こさずに計算量を削減し得ることを実験的に確認した．

ここで，表 3 と表 4 を比較すると，10 ラベルにおける計算時間削減効果は，5 ラベル時の効果よりも低いことが読み取れる．提案手法では，サポータベクタのみを用いることで訓練に必要なデータ数を削減しており，これが実行結果の削減に貢献していると考えられる．従来手法，(B-1)，(B-2) が必要とする訓練データ数に着目すると，表 3 より，5 ラベル時では (B-1) と (B-2) によりデータ数がそれぞれ 11.86% $(= 1 - 22,035/25,000)$ と 12.10% $(= 1 - 21,975/25,000)$ 削減されている．同様に，表 4 より，10 ラベル時の訓練データ数削減率は 4.26% $(= 1 - 47,870/50,000)$ と 4.40% $(= 1 - 47,799/50,000)$ である．計算時間の削減率と同様に，訓練データ数の削減率においても，5 ラベル時の値が 10 ラベル時よりも高いことが読み取れる．

上記の実験結果を踏まえ，CIFAR-10 に対してサポートベクタを用いたときの訓練データ削減について考察する．3.2 節での予備実験では，サポートベクタを決定する際に元のデータの中から 86∼88% 削減できた．しかし CIFAR-10 の訓練データに対してサポートベクタを抽出した場合，5 ラベルの場合は元データの中から削減できたのは約 12% であった．元のデータの大部分を占める割合となり，訓練データの削減という点で予備実験から期待されるほどの効果は得られなかった．この原因として，カラー画像を識別する問題は予備実験で扱ったものと比べて非常に複雑な問題であることが考えられる．予備実験で扱ったような比較的単純なデータセットであれば，サポートベクタとして抽出されたデータ数が 12∼14%であっても精度に劣化は見られなかった．しかし CIFAR-10 のような比較的複雑なデータセットであれば，分類平面の形成を担うデータが多いためサポートベクタとして抽出された訓練データの数は約 88%と全体の大部分を占める結果となった．予備実験の結果と比べると期待されるほどの訓練データの削減とはならなかったものの，確実に訓練データを削減できた．しかも精度劣化をほとんど起こしていないため，訓練データ数と分類精度のトレードオフを向上させたといえ，サポートベクタを使用することで精度劣化を起こさず CIFAR-10 の訓練データを削減することができたと言える．対照実験として行ったランダム抽出データの結果と比べると，約 1%ではあるが精度も向上している．また，訓練データの削減に伴って学習にかかった時間も減少しており，その点でも効果があったと言える．10 ラベルの場合は 5 ラベルの場合よりサポートベクタとして抽出されたデータは多く，約

表 3: CIFAR-10 に対するサポートベクタ適用結果 (ラベル数 5)

訓練データ	データ数	精度 (%)	時間 (秒)
(A) 全訓練データ	25,000	78.300	1058.243
(B-1) サポートベクタ (一対一)	22,035	78.580	961.394
(B-2) サポートベクタ (一対多)	21,975	78.340	957.629
(C-1) ランダム抽出	22,035	77.400	962.348
(C-2) ランダム抽出	21,975	77.260	959.526

表 4: CIFAR-10 に対するサポートベクタ適用結果 (ラベル数 10)

訓練データ	データ数	精度 (%)	時間 (秒)
(A) 全訓練データ	50,000	77.560	2116.688
(B-1) サポートベクタ (一対一)	47,870	77.420	2062.906
(B-2) サポートベクタ (一対多)	47,799	76.970	2060.405
(C-1) ランダム抽出	47,870	76.580	2052.179
(C-2) ランダム抽出	47,799	76.290	2057.681

95% が抽出された．わずか 5% ではあるが，こちらも訓練データを削減することができた．5 ラベルの場合と同様に精度劣化もほとんど起きていないため，サポートベクタによる効果が確認できたと言える．

5 結論

本研究では，サポートベクタを用いることで深層学習における訓練データの削減を試みた．サポートベクタによるデータ削減を行った既存研究はいくつかあるが，それらと異なる点はサポートベクタを CIFAR-10 に適用している点である．まず初めに 6 種類の 2 次元データに対してサポートベクタによる訓練データの削減を行う予備実験を行い，訓練データを 86∼88%削減しつつ精度を維持できることを確認した．これを踏まえ，CIFAR-10 のカラー画像を識別する問題に対して同様にサポートベクタを用いて訓練データの削減を試みたところ，精度をほとんど劣化させることなく訓練データを 5 クラス分類においては約 12%，10 クラス分類においては約 5% 削減することができた．予備実験の結果ほどの効果は得られなかったものの，CIFAR-10 に対しても同様にサポートベクタを用いることで訓練データを削減ができることを確認した．

謝辞

本研究を行うにあたり，終始丁寧な御指導を賜りました大阪大学情報科学研究科橋本昌宜教授に深謝の意を表します.

本研究に際し，貴重な御助言，御指導を賜りました同研究科粟野皓光准教授に深く感謝申し上げます.

本研究において多大な御指導，御指摘を頂きました同研究科劉載勲助教に心から感謝致します.

研究室での生活を様々な面で支えてくださった秘書村上麻子氏に感謝致します.

日々の研究生活のサポートや助言，また本研究に対する助言だけでなく論文執筆においても多大なるご助力をいただきました大西一輝氏に心より感謝申し上げます.

研究室での生活において御世話になりました集積システム設計学講座の皆様方に感謝致します.

最後に，日頃の生活を支えてくださった家族に深く感謝致します.

参考文献

[1] J. Deng, W. Dong, R. Socher, L. J. Li, , K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 770–778.

[5] S. Lawrence, C. L. Giles, and A. C. Tsoi, "Lessons in neural network training: Overfitting may be harder than expected," in *Proceedings of National Conference on Artificial Intelligence*, Jul. 1997, pp. 540–545.

[6] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in deep learning," in *Mathematics of Deep Learning, Cambridge University Press, to appear*, 2018.

[7] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[8] X. Nguyen, L. Huang, and A. D. Joseph, "Support vector machines, data reduction, and approximate kernel matrices," in *Proceedings of Machine Learning and Knowledge Discovery in Databases*, Sep. 2008, pp. 137–153.

[9] K. Dahiya and A. Sharma, "Reducing neural network training data using support vectors," in *Proceedings of Recent Advances in Engineering and Computational Sciences*, Mar. 2014, pp. 1–4.

[10] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN-SVM classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, Apr. 2012.

[11] A. Krizhevsky, "Convolutional deep belief networks on CIFAR-10," 2010, unpublished.

[12] J. Weston and C. Watkins, “Multi-class support vector machines,” Royal Holloway University of London, Tech. Rep., May 1998.