最適化から見たディープラーニングの考え方

得居　誠也

機械学習において，人手で設計した特徴量にもとづく手法が性能の限界を迎えつつあるなか，計算機性能の進歩とデータセットの大規模化によって，深層学習（ディープラーニング）は圧倒的な認識性能を次々に叩き出し，産業界を巻き込み注目を集めている．本稿では，教師あり学習とニューラルネットの基本的な定式化からはじめ，深層学習において高い性能を実現するための最適化，モデリング，正則化の技術について広く紹介する．

キーワード：機械学習，深層学習，ニューラルネット，確率的勾配降下法，正則化

1. はじめに

深層学習は，この 5 年間にパターン認識の様々な分野で成功している．音声認識では，混合ガウスモデルによる方法に代わって，2011 年に深層ニューラルネットを用いた手法が単語誤識別率を 10 ポイント以上改善している [1]．画像認識では，大規模一般物体認識のコンテスト ImageNet Large Scale Visual Recognition Challenge (ILSVRC) において，2012 年に深層学習を用いたチームが，2 位以下に正解率 10 ポイントの差をつけ圧勝した [2, 3]．さらにここ 1 年半の間に，強化学習 [4] や機械翻訳 [5] など，より広い分野においても，従来の手法を追い越そうとしている．

深層学習は，深いニューラルネットを中心とした，無名の変数からなる深い階層を持つ予測器や確率モデルを用いた機械学習の総称である．無名の変数とは，それぞれにあらかじめ特定の意味づけがされておらず，学習によって役割が自動的に決まるという意味だ．本稿では，まず機械学習の問題とニューラルネットを定式化したあと，深層学習で用いられる種々の手法を，最適化，モデルの構成，および正則化を軸に紹介する．

2. 統計的な教師あり学習

本稿では，教師あり学習と呼ばれる機械学習の問題設定を扱うので，まずその目的を簡単に説明しよう．入力変数を $x \in \mathcal{X}$，それに対する予測を $y \in \mathcal{Y}$ と書く．便宜上，これらの組を $z = (x, y), \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ と書く．教師あり学習の目標は，与えられたデータ集合 $S = \{z_i\}_{i=1}^N \subset \mathcal{Z}$ を用いて，予測関数 $f : \mathcal{X} \to \mathcal{Y}$ を自動的に獲得することである．ここで，あらゆる予測関数を考慮することは難しいので，パラメータ付けられた予測関数の集合 $\mathcal{F} = \{f_\theta | \theta \in \Theta\}$ を考え，その中で最適な $f_\theta \in \mathcal{F}$ を探索するのが一般的である．探索空間 $\mathcal{F}$ をモデルと呼ぶことにする．学習に用いる各データ z_i を訓練事例，その集合 S を訓練集合と呼ぶ．

この問題を定量的に扱うためには，予測関数 f がどれくらい間違っているかを測る尺度が必要である．そこで，入力と真の予測の組 $z = (x, y)$ を用いて，損失関数 $\ell(f, z)$ を定義する．損失関数は，$f(x)$ が y に近いほど小さな値を取るように選ぶ．訓練集合に対して正しく予測する関数を得るには，損失の平均 $E_S(f) = (\sum_{z \in S} \ell(f, z))/N$ を最小化すればよい．$E_S(f)$ を f の経験損失という．

さて，教師あり学習の真の目的は，訓練集合に含まれない新しい入力に対しても正しい予測を行うことである．この観点で最適化問題を書き直すのが，統計的な教師あり学習である．入力と真の予測の組 $z = (x, y)$ が独立同分布に従って生成されると仮定する．このとき，損失の期待値 $E(f) = \mathbb{E}_z[\ell(f, z)]$ が定義できる．この期待値 $E(f)$ を汎化誤差という．統計的教師あり学習の目標は，汎化誤差 $E(f)$ を最小化する f を獲得することである．

学習手法が出力する予測関数 f の汎化誤差は，四つの項に分解して解釈できる [6]．ここで，あらゆる予測関数における汎化誤差の最小解を $f^\star$，モデル $\mathcal{F}$ における最小解を $f^\star_\mathcal{F}$，モデル $\mathcal{F}$ における経験損失の最小解を $\hat{f}_\mathcal{F}$ とする．このとき $E(f)$ は，汎化誤差の下限 $E(f^\star)$，近似誤差 $E(f^\star_\mathcal{F}) - E(f^\star)$，推定誤差 $E(\hat{f}_\mathcal{F}) - E(f^\star_\mathcal{F})$，最適化誤差 $E(f) - E(\hat{f}_\mathcal{F})$ に分解できる（図 1）．このうち $E(f^\star)$ 以外の 3 項が学習手法

とくい　せいや
株式会社 Preferred Networks
〒 113–0033 東京都文京区本郷 2–40–1 本郷東急ビル 4 階
tokui@preferred.jp

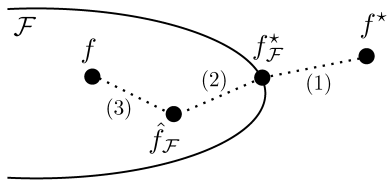

図 1 汎化誤差の分解を模式的に表した絵．(1)，(2)，(3) の各端点における汎化誤差 E の差がそれぞれ近似誤差，推定誤差，最適化誤差に対応する．

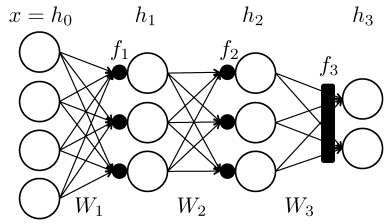

図 2 3 層ニューラルネットの概略図．白丸は各層の値 h_i を，エッジは結合重み行列の掛け算を，黒い節点はバイアスの足し算と非線形関数を表す．非線形関数として，中 2 層のように各成分ごとに独立な非線形変換を施すこともあれば，図の最終層のように複数成分にまたがった変換を行うこともある．

によって変わる．近似誤差はモデル $\mathcal{F}$ の表現力を反映する．推定誤差は，実際に最適化する目的関数が E ではないことで生じる誤差の増分であり，主にデータが有限個しかないことに由来する．推定誤差が大きい状態は過学習と呼ばれ，この項を小さくおさえることは機械学習における重大な関心事である．最後に，最適化誤差はアルゴリズムが経験損失の最小解に到達できないことで発生する誤差の増分である[1]．

深層学習では，近似誤差の小さなモデルをもとに，推定誤差と最適化誤差の双方を最小化することを考える．画像や音声の認識においては，今まで人手で設計された特徴量を用い，線形予測器やカーネル法を用いて凸な損失関数を最小化していた．これは，最適化が比較的容易なモデルを出発点として，特徴量を工夫することで近似誤差を小さくするアプローチである．深層学習はその意味で，従来のパターン認識のアプローチと異なっている．

3. ニューラルネットの定式化と勾配計算

ニューラルネットはモデルの一種で，予測関数を複数の非線形変換の合成で表す（図 2）．入力 $x = h_0$ に対して，$f_1, \ldots, f_L$ を非線形関数として

$$h_i = f_i(z_i),\ z_i = W_i h_{i-1} + b_i\ (i = 1, \ldots, L) \quad (1)$$

のように定義する．ここで，W_i は結合重み行列，b_i はバイアスベクトルであり，ともに学習すべきパラメータである．各 h_i の成分をユニットと呼ぶ．各非線形関数 f_i は活性化関数と呼ばれ，パラメータを持たない関数である．通常，ユニットごとの非線形関数を用いることが多いが，多値分類などの出力層では全成分にまたがった変換を行うこともある．

損失関数は，ニューラルネットの出力 h_L に対して定義され，その設計は学習したい課題によって工夫を要する．教師あり学習における典型的な課題は，回帰と分類である．回帰問題の場合，目標とする値 $y \in \mathbb{R}$ に対して，二乗誤差 $\ell(h_L) = (y - h_L)^2$ を用いるのが一般的だ．多値分類問題では，正解ラベルの 1-hot ベクトル[2] y を目標として，交差エントロピー誤差 $\ell(h_L) = \sum_i (y_i \log h_{L,i} + (1 - y_i) \log(1 - h_{L,i}))$ をよく用いる．この場合，h_L は確率ベクトルである必要があるため，最終層の活性化関数としてソフトマックス関数 $f(x) = (\exp(x_i)/\sum_j \exp(x_j))_{i=1}^d$ を用いる．ここで，d は出力層のユニット数である．これはユニットごとの実数値関数に分解できない活性化関数の一例である．

ニューラルネットの最適化手法は，勾配にもとづく方法が主流である．あるデータに対する損失 $\ell = \ell(h_L)$ の勾配は，式 (1) に連鎖律を適用して次のように計算できる．

$$\nabla_{z_i} \ell = f_i'(z_i) \nabla_{h_i} \ell, \qquad \nabla_{h_{i-1}} \ell = W_i^\top \nabla_{z_i} \ell, \quad (2)$$

$$\nabla_{W_i} \ell = \nabla_{z_i} \ell\, h_{i-1}^\top, \qquad \nabla_{b_i} \ell = \nabla_{z_i} \ell. \quad (3)$$

$\nabla_{h_L} \ell$ を出発点として，式 (2) を用いて $\nabla_{h_i} \ell$ を $i = L - 1, \ldots, 2$ の順に計算し，層ごとに式 (3) を用いてパラメータの勾配を求める．このアルゴリズムを誤差逆伝播法 [7] という．

活性化関数としては，区分的に微分可能な連続関数を用いることが多い．微分不可能な点を含む場合，その点での微分係数としては左右の微分係数かその間の値を代わりに用いる．主要な活性化関数を図 3 に示す．中間層の活性化関数として，従来よく用いられていたのは成分ごとのシグモイド関数で，$\sigma(x) = 1/(1 + \exp(-x))$ や $\tanh(x)$ が代表的だ．これらは滑らかであり，値域

[1] ただし，あくまで経験損失ではなく E によって評価されるため，最適化誤差は負の値を取りうる．

[2] ラベル i の 1-hot ベクトルとは，$\{0, 1\}$ 値ベクトルであり，次元がラベル数に一致し，第 i 成分のみが 1 となっているものである．

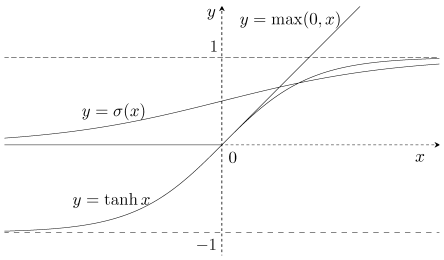

図 3　主要な活性化関数のプロット

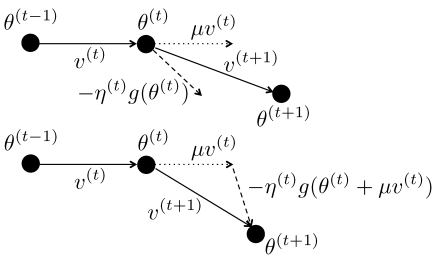

図 4　モーメンタム法（上）と加速法（下）の更新方向

が有界であるため，数値的に不安定になりにくい利点がある．一方で，入力 x が大きい値を取るときに微分係数が小さくなるため，層数が多い場合，式 (2) において次々に微分係数をかけることで，値が指数的に減少する問題がある．そこで，最近は関数 $\max(0, x)$ をよく用いる [8] ．この関数は，Rectified Linear Unit (ReLU) や rectifier などと呼ばれている．ReLU は，値域が有界でない欠点はあるものの，正の値を取るユニットについて勾配が減衰せずに伝播するため，最適化において有利である．

4. ニューラルネットの最適化

ここからいよいよニューラルネットの学習の話題に入る．本節および次の節では，最適化誤差を最小化する方法を述べる．

深層学習においてもっとも基本的な最適化手法は，確率的勾配降下法である．t 回目の反復時のパラメータ全体をまとめて $\theta^{(t)}$ としたとき，データ集合の中から一部分をランダムに取り出し，それらの平均損失の勾配 $g(\theta^{(t)})$ を用いて次のようにパラメータを更新する．

$$\theta^{(t+1)} = \theta^{(t)} - \eta^{(t)} g(\theta^{(t)}). \tag{4}$$

ここで，$\eta^{(t)}$ は t 回目の反復におけるステップ幅である．機械学習では，ステップ幅を学習率ともいう．

目的関数の曲率が大きいとき，二階微分に頼らずに曲率の小さな方向を見つけ出す方法として，次のモーメンタム法をよく用いる．

$$\begin{aligned} v^{(t+1)} &= \mu v^{(t)} - \eta^{(t)} g(\theta^{(t)}), \\ \theta^{(t+1)} &= \theta^{(t)} + v^{(t+1)}. \end{aligned} \tag{5}$$

これとよく似た手法として，Nesterov の加速法と呼ばれる次の更新式もよく使う [9].

$$\begin{aligned} v^{(t+1)} &= \mu v^{(t)} - \eta^{(t)} g(\theta^{(t)} + \mu v^{(t)}), \\ \theta^{(t+1)} &= \theta^{(t)} + v^{(t+1)}. \end{aligned} \tag{6}$$

どちらも，$v^{(t)}$ が実際の更新に用いる差分ベクトルを表し，μ で指数的に減衰する重み付けで過去の勾配の平均を取ったものである．モーメンタム法や加速法は，直近の勾配で一貫して現れる方向を増幅し，勾配の符号がころころ変わる方向については減速するように働く．よって，目的関数の曲率が大きな方向について，差分ベクトル $v^{(t)}$ は小さな値を取り，二次最適化の手軽な模倣とみなすことができる．モーメンタムと加速法の更新方向を図 4 に示した．加速法はモーメンタム法と比べて，勾配を計算する位置が更新後のパラメータにより近くなっている．

さて，ニューラルネットを含む深層学習の最適化では，目的関数は一般に非凸であり，局所解や鞍点などの構造を持つ．勾配法では局所解を抜け出すことは難しく，また鞍点にとらわれる可能性もある．

局所解の存在に問題があるのかは，注意深く考える必要がある．というのも，深層学習において，ほとんどの局所解は十分よい解であるという予想があるのだ．ラメルハートが誤差逆伝播法を導入した当時，局所解であっても十分高い性能を発揮することが実験的に示されていた [7]．同様のことは，ニューラルネットの様々なデータに対する成功により，実験的にはよく確かめられている．

もう少し理論的な視点からの考察もある [10, 11]．解析の対象はガウス確率場から生成された関数である．ヘッセ行列の負の固有値の個数を指数というが，停留点は指数によって分類できる．そこで [11] では，高次元のガウス確率場から生成された関数は，停留点のほとんどが鞍点であり，停留点での値は指数の単調増加関数上に集中していることが示されている．局所解は指数が 0 の点であるから，ほとんどの局所解は近い値を取り，それらはほとんどの鞍点よりもよい解である．この解析は理論的に扱いやすい関数のクラスを対象としているが，実際の認識課題においても停留点が同じように分布することが実験的に確かめられている [10]．

ほとんどの停留点が鞍点であることは，直感的に次のように理解できる [10]．目的関数がランダムであり，

停留点のヘッセ行列の要素が平均 0 の同じ分布に従うならば，その固有値の分布はウィグナーの半円分布に従う．これは 0 を平均・中央値とする半楕円形をした分布で，特に正負の値が半々の確率で生成される．このとき，停留点の指数が 0 となる確率は，次元を高くするにつれて指数的に減少する．

鞍点付近では勾配が小さくなるため，勾配法は鞍点にとらわれやすい．ニューラルネットのモデルはときに 1 億次元という高次元空間になり，停留点のほとんどは鞍点と予想されるため，最適化の大きな障害となる．

鞍点から素早く抜け出すには，勾配の小さな方向について加速することが重要である．言い方を変えると，勾配の大きさによらず，目的関数が減少する方向に等速で進むことができれば，鞍点にとらわれない．次に示す RMSprop [12] はこの発想にもとづく．

$$\begin{aligned} \bar{g}^{(t+1)} &= \lambda g^{(t)} + (1-\lambda) g(\theta^{(t)})^2, \\ \theta^{(t+1)} &= \theta^{(t)} - \eta^{(t)} \frac{g(\theta^{(t)})}{\sqrt{\bar{g}^{(t+1)}}}. \end{aligned} \tag{7}$$

ここで，この式を含め，以降は積や平方根などの演算はすべて要素ごとに行うこととする．式 (7) は，勾配を直近のスケールで正規化したような勾配法となっている．

よく似た手法として次の ADADELTA [13] がある．

$$\begin{aligned} \bar{g}^{(t+1)} &= \lambda \bar{g}^{(t)} + (1-\lambda) g(\theta^{(t)})^2, \\ v^{(t+1)} &= -\sqrt{\bar{v}^{(t)}} \frac{g(\theta^{(t)})}{\sqrt{\bar{g}^{(t+1)}}}, \\ \theta^{(t+1)} &= \theta^{(t)} + v^{(t+1)}, \\ \bar{v}^{(t+1)} &= \lambda \bar{v}^{(t)} + (1-\lambda)(v^{(t+1)})^2. \end{aligned} \tag{8}$$

少々ややこしいが，$\bar{g}^{(t)}$ や $\bar{v}^{(t)}$ は移動平均であり，それぞれ直近の二乗勾配と更新幅を表す．更新則 (8) では，正規化された勾配に $\sqrt{\bar{v}^{(t)}}$ をかけることで，ニュートン法のようにパラメータのスケールに応じて更新幅を自動的にスケールさせている．これはたとえば，方向ごとに曲率が大きく異なる場面で威力を発揮する．RMSprop はプラトーに強いが，曲率の意味で関数が大きくゆがんでいる場合には ADADELTA のほうが素早く降下方向へと加速する．これらの新しい勾配法は，理論解析に乏しいものの，様々な目的関数において特別なチューニングなしによい性能を発揮する [14]．

プラトーの問題を捉える別のアプローチとして，自然勾配法がある [15]．自然勾配法では，情報幾何学にもとづいて定まる行列 G^{-1} を勾配にかける．行列 G は，モデル $\mathcal{F}$ に導入される自然な可微分多様体構造におけるリーマン計量である．この方法は，座標系によらないパラメータ間の本質的な距離構造を考慮した上で，目的関数がもっとも素早く降下する方向を探していると解釈できる．多層ニューラルネットのモデルにおいては，この多様体が特異点を持ち，この特異点がプラトーをなす．自然勾配法は，計量を考えることで特異点に由来するプラトーを自然に解消する方法である．

自然勾配法は，素朴な計算方法では反復ごとに行列 G を再計算する必要があり，効率が悪い．そこで行列 G を逐次的に推定する手法なども提案されている．上で述べた RMSprop などと比べて，自然勾配法は新しい手法ではないが，深層学習の学界ではこれを再評価する流れがあり，注目されている [16]．自然勾配法の詳細や具体的な効用，解析については [15] を参照されたい．

5. 最適化誤差を小さくするモデル

最適化誤差を小さくするには，前述のような最適化手法の高度化ももちろん重要だが，より最適化しやすい予測関数を用いるという手もある．モデルを変更して最適化誤差を縮める際には，近似誤差を小さく保つ必要がある．いくつか例を挙げよう．

5.1 畳み込みネット

画像認識では全連結のニューラルネットの代わりに畳み込みネット [17] を使う（図 5）[3]．3 節では，行列積で層を定義したが，畳み込み層では二次元畳み込み演算で出力を計算する．これに対して，密な行列をかける層を全連結層という．畳み込み演算は，入出力をベクトルに展開して行列積で書くこともできる．このとき，入力ベクトルにかけられる行列の各行は，図の左側の小さな箱に相当する要素だけが非ゼロであり，非ゼロのパラメータは各行の間で共有される．よって，畳み込み層は全結合層に対して，パラメータに制約を入れたものと解釈することができる．

畳み込み層のパラメータに加えられた制約は大きく，全連結層のほうがずっと自由度が高いが，画像認識においては畳み込み層を用いてもよい解が得られる．つまり，近似誤差はあまり下がらない．これは，画像が空間方向に対称性と局所性を持ち，畳み込み層が低次の特徴を表現するのに十分なためである．パラメータ

[3] 畳み込みネットは，畳み込み層のほかにプーリングと呼ばれる特殊な活性化関数も用いる．これも画像の空間方向への対称性を用いた手法だが，本稿ではその解説は省略する．

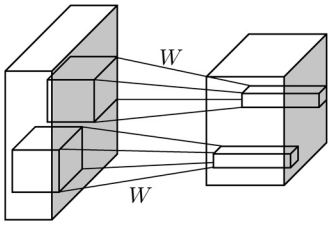

図 5 畳み込み層の概略図．入力と出力はともに 3 階のテンソルで表される．テンソルの各軸は，画像の縦・横方向およびチャンネルの種類に対応する．各チャンネルは，入力がカラー画像なら R，G，B に対応し，中間層ではその位置における何らかの特徴を表す．畳み込み演算は，入力の各矩形をベクトルに展開し，それぞれに同じ重み行列 W を作用させ，位置ごとにベクトルを出力する．

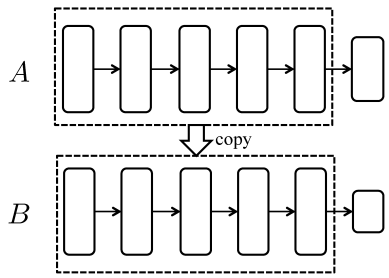

図 6 ニューラルネットの転移学習の例．課題 S を学習したニューラルネット A の途中の層までを別のニューラルネット B にコピーし，残りの層をランダムに初期化して別の課題 T を学習させる．

が少ないほうが最適化誤差を小さくしやすいため，畳み込みネットのほうがより低い汎化誤差を達成する．

畳み込みネットを含め，上記のようなパラメータ制約は推定誤差をおさえる効果も持つ．この点については 6 節でまた触れる．

5.2 転移学習

さて，最適化誤差をおさえる別のアプローチとして，大規模データセットを用いて学習したニューラルネットを，別のデータセットに転用するという方法がある（図 6）．これは，目標とする課題 T において学習データが十分でない場合，特に有用である．T に似た性質を持つ学習課題 S であって，大規模なデータセットが手に入るものを考える．このとき，まず課題 S を大規模なニューラルネット A に学習させる．次に A の途中の層までを残して以降を取り除き，パラメータをランダムに初期化した層を後ろに取り付ける．こうしてできた新しいニューラルネット B を，課題 T のデータセットで学習する．これは転移学習と呼ばれる方法の一種で，課題 T と S において有効な特徴量が似通っている場合に効果がある．後段で課題 T を学習する際には，A からコピーしたパラメータを固定する場合と，それらも含めて全体を最適化しなおす場合がある．T のデータセットが小さい場合は，コピーしたパラメータを固定したほうが，過学習のリスクが少ない．逆に T のデータセットが大きい場合，全体を最適化しなおしたほうが汎化性能がよくなる．

課題によっては，転移学習を用いることで，最適化誤差が小さくなる．最適化の観点から考えると，ニューラルネット A は B の初期パラメータを与えるだけだが，この初期値には T のデータセットのみを使っては到達できない可能性がある．その場合には最適化の観点からも転移が有効である．

一方，転移の方法を間違うと，最適化に悪影響を及ぼすこともある．ニューラルネットの学習は，層をまたいだパラメータ間の相互依存を引き起こす場合がある．この場合，A から B を初期化する際に，相互依存の関係にある層の片方を取り去ってしまい，後段の学習でコピーしたパラメータを固定してしまうと，最適化が困難になる [18]．このような相互依存は，入出力から遠い中間層でよく発生する．

この転移学習の例は，深層学習における表現の重要性も示している．表現とは，特徴量と似た意味で使われる用語だが，特にニューラルネットの中間層のように，ベクトル全体として意味をなす場合をさす．複数の課題の間で表現を共有する試みは，ほかにも化合物の活性予測で成功している [19]．

5.3 モデル圧縮

最後に，深層学習のもっとも興味深い点である，深さの効果を考える．深層学習の本質はその深さにあると信じられているが，深いニューラルネットが高い性能を発揮する理由はまだよくわかっていない．

Universal Approximation 定理によれば，2 層の浅いニューラルネットであっても，中間層が十分多くのユニットを持てば，任意の連続関数を再現できる（厳密な定式化は [20] を参照せよ）．しかし「現実的なパラメータ数で，深いニューラルネットと同じくらいよい予測器を表現できるか」はまた別の問題である．

この命題は，少なくとも多値分類問題に対しては，モデル圧縮のアプローチによる実験によって，肯定的に解決された [21]．モデル圧縮では，多値分類問題をまず深いニューラルネット A に学習させる．次に，学習データに付与されたラベルに加えて，A が出力する各ラベルの確率を教師情報として加え，それを模倣するように浅いニューラルネット B を学習させる．このとき，A の出力は，手で付与されたラベルと比べ，不正

解ラベル間の関係をも表現する，より豊かな情報となる．たとえば，一般画像認識の例で，狐の画像に対して「犬」ラベルと「車」ラベルをつける確率を考えると，ともに不正解という意味では低い値を取るが，相対的には「犬」ラベルのほうにより大きな値を付与するべきだろう．

モデル圧縮を用いると，浅いニューラルネットを学習データから直接学習する場合とくらべて，より高い認識性能を達成する．モデル圧縮によって，深いニューラルネットと同じ性能の浅いニューラルネットを学習できるとする報告もある [22]．このことは，現実的なユニット数においても，浅いニューラルネットの表現力は十分高く，近似誤差は小さいということを示す．問題となるのは推定誤差や最適化誤差であり，ここにニューラルネットの深さが関わってくるのだろう．

6. 正則化と最適化の関係

今度は推定誤差をおさえる方法を考える．推定誤差が大きい状態（過学習）を避けるためにアルゴリズムに手を加えることを正則化という．正則化は，汎化誤差の小さな予測器に求められる性質を事前知識にもとづいて考え，それを学習手法に反映することで，過学習を避けるアプローチである．

過学習の典型的な例は，学習データによりよく適合するために，一部のパラメータがつじつま合わせをする場合である．たとえば，二つの入力ユニット x，y と一つの出力ユニット z を考える．ここでは簡単のため，z は線形のユニット，すなわち非線形関数を用いないユニットを考える．$x=1, y=1$ のときに $z=1$ を出力したい場合，$z=0.5x+0.5y$ としてもよいし，$z=100x-99y$ としてもよい．ここで，データに少しだけノイズがのり，$x=1.1$，$y=0.8$ となった場合，前者は $z=0.95$ という頑健な値を出力するが，後者は $z=30.8$ と大きく異なる値を出力する．後者が頑健でない原因は二つある．一つは，パラメータが大きな値を取ってしまったことである．もう一つは，二つのパラメータが相互依存の関係にあることである．つまり，$z=100x-99y$ という式の背後には，x と y が同じ値を取るという仮定が入り，二つのパラメータ 100，-99 の間にはお互いの役割に関して依存性が生じる．

パラメータの絶対値を小さく保つための正則化として，たとえば荷重減衰やノルム制約がある．荷重減衰では，目的関数にパラメータの二乗和 $\|\theta\|^2$ を足す．ノルム制約では，適当な半径 $R>0$ を決めておいて，各反復後にパラメータを超球 $\{\|\theta\| \leq R\}$ に射影する．ノルム制約を用いた勾配法は，この超球をパラメータの定義域としたときの射影勾配法と捉えることもできる．これらの正則化は，深層学習に限らず機械学習の幅広い場面で使われている．

ニューラルネットの学習に特有の正則化としては，ドロップアウト [23] をよく用いる．これは，最適化の際に，各ユニットに確率 α でランダムに 0 をかけるという手法である．学習時にユニットがランダムに脱落するため，ユニット間に相互依存ができるようなパラメータでは学習時の誤差が大きくなり，結果として相互依存の少ないパラメータが得られる．ドロップアウトは非常に強力な正則化手法だが，得られる勾配が不安定になるため，最適化は一般に難しくなる．

5 節で，畳み込み層の導入によって最適化が簡単になると述べたが，これは同時に正則化の効果も持つ．これは，次のように直感的に理解できる．畳み込み層を用いると，各フィルタは一つの画像に対して繰り返し適用される．一回の適用ごとに，出力側から誤差逆伝播によって勾配が伝播し，各位置での勾配の総和が実際の更新方向となる．よって，畳み込み層のフィルタを学習する際には，実効的なデータ数が増す．データ数が大きい場合，経験損失は汎化誤差をよく近似するようになるため，推定誤差が小さくなる．畳み込み層に限らず，モデルの中で複数回パラメータを同じように使う場合，同様の議論が可能である．その場合，パラメータを共有する回数が多いほど，正則化の効果は強くなる．画像認識における畳み込みネットではこの効果が非常に強いため，多くの場合では畳み込み層においてはドロップアウトが必要ない．

このように，深層学習においても正則化によって推定誤差を小さくおさえることが重要だが，一方でこれらの手法は最適化誤差にもよい影響を及ぼすことがある．たとえば [2] は，荷重減衰によって経験損失が小さくなったと報告している．正則化の効果は，よいパラメータの性質についての事前知識を用いて，目的関数にバイアスを入れることである．このとき，変化するのは局所解の位置だけではなく，反復最適化においてパラメータが通る経路全体がバイアスの効果を受ける．その結果，鞍点やプラトーを回避するように経路が変更されれば，より経験損失の小さな解に到達できる．たとえば荷重減衰が最適化を補助するという結果から，原点に近いパラメータ領域のほうが，原点から遠いパラメータ領域よりも目的関数の形状が素直で，最適化がしやすいと予想される．

勾配法をもとにした最適化手法の発展が目的関数のミクロな構造に着目しているとすれば，正則化による最適化の補助は目的関数のマクロな構造にもとづくといえる．

7. おわりに

本稿では，深層学習の基本的な考え方を紹介し，その根幹をなす最適化問題の性質とそれに対するアプローチについて，実際に成功をおさめている手法にからめて駆け足で眺めた．深層学習の要点は，多層のニューラルネットにより階層的な表現が学習される点にあるが，その理論的な背景はいまだ謎に包まれている．一方で，深層学習の研究自体はまさに日進月歩の様相を呈している．理解が性能に追いつき，それがさらなる性能の向上に役立つ日が来ると期待している．

謝辞 大野健太氏，岡野原大輔氏，丸山宏先生には，拙稿に目を通していただき，貴重なご意見をたくさんいただいたので，ここに感謝の意を表する．

参考文献

[1] F. Seide, G. Li and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” In *Interspeech 2011*, pp. 437–440, 2011.
[2] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” In *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, 2012.
[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” *CoRR*, abs/1409.0575, 2014.
[4] V. Mnih, K. Kavukcuoglu and D. Silver, “Playing Atari with deep reinforcement learning,” In *Deep Learning Workshop*, 2013.
[5] I. Sutskever, O. Vinyals and Q. V. V. Le, “Sequence to sequence learning with neural networks,” In *Advances in Neural Information Processing Systems 27*, pp. 3104–3112, 2014.
[6] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” In *Optimization for Machine Learning*, MIT Press, pp. 351–368, 2011.
[7] D. E. Rumelhart, G. E. Hinton and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, **323**, pp. 533–536, 1986.
[8] X. Glorot, A. Bordes and Y. Bengio, “Deep sparse rectifier neural networks,” In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, pp. 315–323, 2011.
[9] I. Sutskever, J. Martens, G. E. Dahl and G. E. Hinton, “On the importance of initialization and momentum in deep learning,” In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, **28**, pp. 1139–1147, 2013.
[10] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli and Y. Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,” In *Advances in Neural Information Processing Systems 27*, pp. 2933–2941, 2014.
[11] A. J. Bray and D. S. Dean, “Statistics of critical points of gaussian fields on large-dimensional spaces,” *Physical Review Letters*, **98**, p. 150201, 2007.
[12] T. Tieleman and G. Hinton, “Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural Networks for Machine Learning*, 2012.
[13] M. D. Zeiler, “ADADELTA: An adaptive learning rate method,” *CoRR*, abs/1212.5701, 2012.
[14] T. Schaul, I. Antonoglou and D. Silver, “Unit tests for stochastic optimization,” In *International Conference on Learning Representations*, 2014.
[15] 甘利俊一，『情報幾何学の新展開』，サイエンス社，2014.
[16] R. Pascanu and Y. Bengio, “Revisiting natural gradient for deep networks,” In *International Conference on Learning Representations*, 2014.
[17] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition,” In *Proceedings of the IEEE*, **86**, pp. 2278–2324, 1998.
[18] J. Yosinski, J. Clune, Y. Bengio and H. Lipson, “How transferable are features in deep neural networks?,” In *Advances in Neural Information Processing Systems 27*, pp. 3320–3328, 2014.
[19] G. E. Dahl, N. Jaitly and R. Salakhutdinov, “Multi-task neural networks for QSAR predictions,” *CoRR*, abs/1406.1231, 2014.
[20] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals, and Systems*, **2**, pp. 303–314, 1989.
[21] C. Bucila, R. Caruana and A. Niculescu-Mizil, “Model compression,” In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pp. 535–541, 2006.
[22] J. Ba and R. Caruana, “Do deep nets really need to be deep?” In *Advances in Neural Information Processing Systems 27*, pp. 2654–2662, 2014.
[23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *CoRR*, abs/1207.0580, 2012.