

スケッチした pictorial symbols のディープラーニングのための派生基準形状を用いた学習形状の反復的生成

佐藤 信¹

概要: スケッチした形状のディープラーニングのために、学習形状の反復的な生成手法を提案する。提案手法では、SVG 形式により表現された pictorial symbols を基準形状とし、以下の手順を反復的にこなすことにより学習形状を生成する。始めに、基準形状から派生基準形状を生成する。次に、基準形状および派生基準形状に部分的な弾性変形をおこなうことにより、多様な類似性をもつラフ類似形状を生成する。そして、ラフ類似形状のなかから学習に用いる形状を選択する。学習形状の選択のために、基準形状および派生基準形状の特徴を学習した Restricted Boltzmann Machine(RBM) を用いる。

Iterative Generation of Learning Shapes with Derived Base Shapes for Deep Learning of Sketched Pictorial Symbols

MAKOTO SATOH¹

Abstract: This paper presents an iterative learning set generation method for learning sketched pictorial symbols with deep learning. In the method, pictorial symbols represented using SVG format are used as base shapes. Learning shapes are iteratively generated with the following steps: First a derived base shape are generated from a base shape. Then roughly similar shapes which have diversity-carrying similarities, are generated by applying partial elastic deformations to the base and the derived base shapes. Finally learning shapes are selected from the generated roughly similar shapes. The Restricted Boltzmann Machines(RBMs) that have learned the feature of the base and derived base shapes, is used for selecting the learning shapes.

1. はじめに

本稿では、スケッチした pictorial symbols [10] をディープラーニングにより学習することを目的として、予め与えた少数の基準形状に基づき学習形状を生成するための反復的な手法を提案する。

私たちの身の周りでは、各種の pictorial symbols が用いられている。その特徴は、特定の言語に依存せずに、情報を直感的に伝達することが可能なことである。そのことから pictorial symbols は、視覚的な言語としての役割を担っているといえる。例えば、地図記号および各種の図において用いられる簡潔な図形要素などの一般的なものから、UML ダイアグラムに用いられる図形要素および音符などのようにある特定の分野に関連するものまで、非常に多く

の種類の pictorial symbols が存在する。

計算機を用いて pictorial symbols を認識あるいは検索などをすることを目的として、その特徴をアルゴリズムにより捉えるための研究がおこなわれている。ここでは、基準形状から多様な類似性をもつスケッチ風の形状 (ラフ類似形状) を生成するための既提案手法 [8] を、派生基準形状を用いるように拡張をおこなう。

これ以降の構成について、簡単に説明する。2 節では、関連研究について比較をおこない、提案する手法の概要と特徴について述べる。そして、3 節では、派生基準形状を用いて学習形状を生成するための手法を提案する。提案手法の実装と実験結果の検討について 4 節で述べる。そして最後に、5 節で本稿のまとめと今後について述べる。

¹ 岩手大学
Iwate University, Ueda, Iwate 020-8551, Japan

2. 関連研究と提案手法の概要

2.1 バイアスとバリエーションとのトレードオフ

機械学習においては、学習アルゴリズムと同様に、学習データも重要な役割を果たす [1]。そのことを具体的に示すのが、学習誤差のバイアスとバリエーションとの間に存在するトレードオフの関係である。

機械学習では、学習データを入力として、学習誤差が小さくなるように学習モデルのパラメータを推定することにより学習をおこなう。学習誤差の成分は、バイアス、バリエーションおよび削除不可能な誤差により構成される [4]。バイアスとバリエーションとのトレードオフを考慮することにより、過学習に陥る危険性が小さくなるように学習をおこなうことが重要である [4] [9]。

ここで、学習データと学習モデルとの複雑性の関係に関連づけて、学習誤差のバイアスとバリエーションとの間のトレードオフについて定性的に説明する。学習データの複雑性と比較して学習モデルの複雑性が小さすぎる場合には、バイアスを小さくすることが困難であるがバリエーションを小さくすることができる。一方、学習データの複雑性と比較して学習モデルの複雑性が大きすぎる場合には、バイアスを小さくすることができるがバリエーションを小さくすることが困難である。このトレードオフを考慮すると、学習に用いないデータに対して頑健な学習をおこなうためには (すなわち、過学習の可能性を小さくするためには)、学習データの複雑性に対して適切な複雑性の学習モデルを用いて、十分な個数の学習データにより学習をおこなう必要があるといえる。

2.2 学習データ augmentation

機械学習に用いるための学習データの収集には、多くの時間と労力を必要とする。例えば、Eitz 等 [2] では、スケッチした形状の機械学習に用いる形状データを、インターネットを利用して収集している。そして、形状が例示された場合に、その形状をどのようにスケッチするのかについて、機械学習を用いて分析をおこなっている。

一方、学習データを収集するための労力を削減することなどを目的として、学習データの augmentation (増補) がおこなわれることがある。学習データの augmentation とは、学習データの個数をアルゴリズムを用いて増加させることにより、学習データの不足を補うことである。2.1 節で述べたように、過学習の可能性を減少させるためには、十分な個数の学習データを用いて学習をおこなう必要がある。そのため、過学習の可能性を減少させ、学習に用いていない入力データに対して頑健な学習をおこなうためには、学習データの augmentation が有効である。

形状画像の学習データに augmentation をおこなっている

る例としては、Schmidhuber [11], Krizhevsky 等 [7] がある。そこでは、基準とする形状画像を基にして学習データの augmentation をおこない、形状の微小変化および照明の変化に対して頑健な学習をおこなっている。具体的な augmentation の手法としては、形状の微小変化に対する頑健性のためには基準とする形状画像に微小な移動変換を用い、そして、照明の変化に対する頑健性のためには色情報の変換を用いている。また、Fu 等 [3] は、少数の学習データを用いて手描きの線図形の画像を学習するために、学習データに augmentation をおこなっている。そこでは、Simard 等 [12] の提案による画像の弾性変形手法を用いて、スケッチ風の学習データを生成することにより、学習データの augmentation をおこなっている。

2.3 形状のディープラーニング

形状を機械学習するために、ディープラーニングを用いる研究が多数発表されている。それらの研究では、手書き数字の画像あるいは風景画像のパッチなどを対象とするものが多数を占める。そして、比較的小さいピクセル数の画像を用いた研究がほとんどである。

一方、スケッチした線図形を対象とする研究は、少数である。手書き数字の画像あるいは風景画像のパッチを対象とする場合と比較して、線図形を対象とする場合には特有の工夫が必要になるのかどうかについては、ほとんど研究がおこなわれていない。特に、スケッチした形状を学習する場合には、形状の変化の度合いが大きいということに対応できる必要がある。形状の変化の度合いが大きいことの原因としては、手書きであるということ挙げることができる。そして、スケッチの作成過程では、スケッチの作成者がスケッチをしようとする形状をどのように捉えているのかということが、最終的なスケッチ形状に与える影響が大きいということ挙げることができる。

例えば、Fu 等 [3] では、スケッチした形状に augmentation をおこなった形状を訓練データとして機械学習をおこなっている。そこでは、学習手法としてディープラーニングを用いている。Hinton 等 [5] が示すように、ディープラーニングは、訓練データの特徴を抽出する性能に優れた手法である。Fu 等 [3] では、ディープラーニングの学習モデルのひとつである Deep Belief Network (DBN) [6] を用いている。DBN のパラメータの学習には、Restricted Boltzmann Machine (RBM) が用いられる。本稿の提案アルゴリズムでは、形状の特徴を捉えるために RBM を用いる。

2.4 提案手法の概要と適用分野

提案手法では、スケッチした pictorial symbols の機械学習に用いる学習形状を生成するための既提案手法 [8] に拡張をおこなう。既提案手法では、弾性変形を用いて多様な

類似性をもつスケッチ風の形状 (ラフ類似形状) を生成する。そこでの弾性変形には, Fu 等 [3] において用いられている Simard 等 [12] の手法を用いている。

今回の提案では, 基準形状から派生基準形状を生成し, それらの形状に弾性変形をおこなったラフ類似形状を生成する。そして, 基準形状および派生基準形状の特徴を学習した RBM を用いて, 学習形状を選択する。

学習形状を選択する場合には, 基準形状と派生基準形状とを補間する形状については, 学習形状として選択されることが望ましいといえる。そこで, そのような条件を満たす派生形状を選択するための手法についても提案をおこなう。

なお, pictorial symbols は, SVG 形式 [13] で表現されることが多いことから, 既提案手法と同様に, SVG 形式で表現した pictorial symbols を基準形状として用いる。

提案手法を用いると, 少数の形状データを基にして, 機械学習のための学習形状をアルゴリズムにより生成することが可能となる。それにより, 認識しようとする手描き形状のカスタマイズが容易になることから, 特定の適用分野において用いられる図形形状の認識に用いることなどが考えられる。

3. 派生基準形状を用いた学習形状の生成

3.1 アルゴリズムの各段階

提案するアルゴリズムは, 既提案手法 [8] を基準形状に加えて派生基準形状を用いるように拡張したものである。その概要を, 以下に示す。

Step1: 基準形状の生成 予め, pictorial symbols の形状が, SVG 形式の形状データとして与えられているとする。この SVG 形式のデータから, 形状を表現する画像を生成する。この形状画像は, 学習形状を生成するための基準形状として用いられる。

Step2: 形状特徴の学習 基準形状を基にして派生基準形状を生成する。そして, 基準形状および派生基準形状に小さな変位の弾性変形をおこなうことにより, スケッチ風の類似形状を画像データとして生成する。そして, 生成した画像の形状特徴を捉えるために, 生成した画像を Restricted Boltzmann Machine(RBM) を用いて学習する。

Step3: ラフ類似形状の生成 基準形状および派生基準形状に大きな変位の部分的な弾性変形をおこなうことにより, 多様な類似性をもつ粗く類似な形状 (ラフ類似形状) を生成する。

Step4: 学習形状の選択 生成したラフ類似形状のなかから, 基準形状の学習に適した形状を選択する。ここでの選択には, **Step2** で学習した RBM を用いる。

Algorithm 1 Derived base shape generation

```
Set a scaling factor  $c$  for generating a current candidate for a
derived base shape.
Set a scaling factor  $p$  for generating a previous candidate for
the derived base shape, to 1.0.
Transform a base shape using  $c$ . The transformed shape is the
current candidate for the derived base shape.
Set the previous candidate for the derived base shape to the
base shape.
Set a maximum iteration count of the following loop to  $N$ .
 $n \leftarrow 0$ 
while  $n < N$  do
  Verify the validity of the current candidate.
  if the verification succeed then
     $b \leftarrow c$ 
     $c \leftarrow c + (c - p)/2$ 
     $p \leftarrow b$ 
    Transform the base shape using  $c$ . The transformed
    shape is the current candidate.
    Transform the base shape using  $p$ . The transformed
    shape is the previous candidate.
  else if the verification failed then
     $c \leftarrow c - (c - p)/2$ 
    Transform the base shape using  $c$ . The transformed
    shape is the current candidate.
  end if
   $n \leftarrow n + 1$ 
end while
The previous candidate is the derived base shape.
```

Step2 から **Step4** までを, 必要に応じて繰り返し, 学習形状を生成する。これらの各段階のうち, **Step2** での派生基準形状の生成が, 既提案手法と大きく異なる部分である。それ以外については, 既提案手法においては基準形状を用いるものを, 提案手法では基準形状および派生基準形状を用いるように既提案手法を拡張しているのみである。既提案手法の詳細については, [8] に述べられているので, これ以降の節においては, **Step2** での派生基準形状の生成について述べる。

3.2 派生基準形状の生成

派生基準形状を生成するための手法を, アルゴリズム 1 に示す。このアルゴリズムでは, スケーリングによる形状変換を用いて, 基準形状から派生基準形状を生成している。スケーリングだけではなく, 補間形状を生成することが可能な任意の形状変換を用いることが可能である。

アルゴリズム 1 の概要を以下に示す。なお, アルゴリズムへの入力は, 基準形状の形状データ, スケーリングの倍率の初期値および繰り返し回数の最大値である。

Step1: 派生基準形状の候補形状生成 基準形状をスケーリングすることにより, 派生基準形状の候補形状を生成する。

Step2: 候補形状の妥当性確認 候補形状が派生基準形状

Algorithm 2 Validity verification of the current candidate in algorithm 1

Set a scaling factor t for generating a testing base shape:
 $t \leftarrow c + (1 - c)/2$.
Transform the base shape in algorithm 1 using t . The transformed shape is the testing base shape.
Apply small elastic deformations to the previous and current candidate for the derived base shape in algorithm 1, and the testing base shape:
the generated deformed shape sets are defined as $\mathcal{P}, \mathcal{C}$ and $\mathcal{T}$.
Construct a training set $\mathcal{A}$ for training RBM with $\mathcal{P}$ and $\mathcal{C}$.
Construct a test set $\mathcal{B}$ for testing RBM with $\mathcal{T}$.
Train RBM with the training set $\mathcal{A}$.
Test RBM with the training shapes $\mathcal{P}$: the average test error is defined as p .
Test RBM with the training shapes $\mathcal{C}$: the average test error is defined as c .
Test RBM with the test set $\mathcal{B}$: the average test error is defined as b .
if b is between p and c **then**
 The verification succeed.
else
 The verification failed.
end if

として適した形状であるかどうかについて判定をおこなう (3.3 節を参照)。

Step3: 形状変換係数の変更 判定結果に応じてスケールリングの倍率を変更し、新たな派生基準形状の候補を生成する。

Step4: 繰り返しの判定 繰り返し回数が最大値になっていけば、処理を終了する。そうでなければ、**Step2**に戻る。

3.3 派生基準形状の妥当性の確認

3.2 節では、アルゴリズム 1 を示すことにより、基準形状を基にして派生基準形状を生成するための手法について述べた。ここでは、アルゴリズム 1 に含まれる派生基準形状の妥当性の確認をおこなうための手法について述べる。その手法を、アルゴリズム 2 に示す。

その概要は以下のとおりである。なお、アルゴリズムへの入力、基準形状の形状データ、および派生基準形状の現在およびそのひとつ前の候補のそれぞれについての形状データおよびスケールリングの倍率である。これらは、アルゴリズム 1 においての値を用いる。

Step1: 補間形状の生成 派生基準形状の現在の候補形状と基準形状とを補間する形状を生成する。

Step2: 訓練形状の生成 基準形状および現在の派生基準形状の候補形状を用いて、訓練形状を生成する。

Step3: テスト形状の生成 補間形状を用いて、テスト形状を生成する。

Step4: 形状特徴の学習 基準形状および現在の派生基準形状の候補形状の形状特徴を学習するために、**Step2**で生成した訓練形状を入力として、RBM の学習をおこなう。

Step5: 形状の再構築誤差の計算 **Step4** で学習した RBM を用いて、訓練形状およびテスト形状の再構築誤差を計算する。

Step6: 妥当性の評価 **Step5** で計算した再構築誤差を用いて、派生基準形状の妥当性を評価する。

なお、RBM の再構築誤差の計算のためには、(1) 式を用いている。

$$e = \sqrt{\sum_i (v_i - \hat{v}_i)^2}. \quad (1)$$

ここで、ベクトル $\mathbf{v}$ および $\hat{\mathbf{v}}$ は、それぞれ、RBM の入力ベクトルおよび再構築ベクトルである。ここでの再構築ベクトルとは、RBM によりエンコードしたベクトルをデコードして得られるベクトルのことである。

4. 実装と結果の検討

3 節で提案したアルゴリズムを実装し、その有効性について検討した。

4.1 派生基準形状の生成例

3 節で述べたアルゴリズム 1, 2 を用いて派生基準形状を生成した。pictorial symbols の例として、図 1 に示すコーヒークップの形状を用いた。この形状は、SVG 形式のファイルを構文解析することにより生成した Bezier 曲線をラスターライズすることにより生成した画像形状である。なお、形状の画像サイズは 100×100 ピクセルである。その画像の中央の 50×50 の範囲に形状をラスターライズした。

この形状を基準形状として派生基準形状の候補形状を反復的に生成することにより、派生基準形状を生成した。反復の回数は 5 とした。各繰り返しで生成した派生基準形状の候補形状を、図 5 に示す。図 2, 3 は、1 回目の繰り返しで用いた RBM の訓練形状であり、それぞれ、基準形状、または派生基準形状の候補形状を基にして生成したラフ類似形状である。生成した形状数は、それぞれ 100 であり、基準形状、または派生基準形状の候補形状を含めて、合計 202 の形状を訓練形状として用いた。RBM の学習に用いたハイパーパラメータを、表 1 に示す。図 4 は、1 回目の繰り返しで用いた RBM のテスト形状である。テスト形状数は、101 である。

図 6 に、各繰り返しで用いた訓練形状およびテスト形状を示す。第 1, 2, $\dots$, 5 行は、それぞれ、第 1, 2, $\dots$, 5 回目の繰り返しで用いた形状である。それぞれの行の左側の 4 個の形状が訓練形状であり、残りの形状がテスト形状であ

る。なお、RBM の訓練形状およびテスト形状として、部分的な弾性変形を用いて生成したラフ類似形状に対して前処理をおこなったものを用いた。前処理としては、画像データのトリミングおよび2値化をおこなった。なお、訓練形状については、派生基準形状を基にして生成した訓練形状を示した。

各繰り返しのにおいて派生基準形状の候補形状を生成するために用いたスケーリングの倍率を、表2に示す。これらの値は、アルゴリズム1の、変数 c (current candidate) の値である。5回目の繰り返しの終了時では、coffee cup, sauser, および grab についての変数 p (previous candidate) の値は、それぞれ、 $(x, y) = (0.375, 1.0), (1.0, 1.0)$ および $(0.375, 1.0)$ である。この値がアルゴリズムにより選択した、派生基準形状を生成するためのスケーリングの倍率である。

なお、部分的な弾性変形を用いたラフ類似形状の生成については、[8]において詳細が述べられている。

4.2 派生基準形状の妥当性確認の例

提案のアルゴリズムでは、形状の特徴を学習したRBMによる形状の再構築誤差を用いることにより、派生基準形状の候補形状の妥当性の確認をおこなう。4.1節の例について、アルゴリズムの各繰り返して生成される形状の再構築誤差を図7に示す。第1, ..., 5行は、それぞれ、第1, ..., 5回目の繰り返しの各形状の再構築誤差である。それぞれの行は、左から右に、基準形状を基にして生成した訓練形状、派生基準形状を基にして生成した訓練形状、テスト形状の再構築誤差、そして、これらをひとつのグラフに描いたものである。

4.3 検討

図2, 3, 4と図1, 5とを比較することにより、訓練形状またはテスト形状として、基準形状に類似な形状が生成可能であることが分かる。

図6から、アルゴリズムの各繰り返しのにおいて、訓練形状およびテスト形状が変化する様子が分かる。図6と図5とを比較することにより、スケーリングの倍率にあわせて、基準形状に類似な形状を生成可能であることが分かる。

図7から、RBMを用いた形状の再構築誤差がアルゴリズムの繰り返しのにより変化する様子が分かる。ここでは、アルゴリズム2に記したように、青色およびオレンジ色のヒストグラムに対応する形状データを用いてRBMの学習をおこなっている。そして、学習したRBMを用いて、青色、オレンジ色、および緑色のヒストグラムに対応する形状データごとの再構築誤差の平均値を計算し、その値に基づいてスケーリングの倍率の更新をおこなっている。これらのヒストグラムからは、形状データの再構築誤差の分布が詳細に分かる。アルゴリズムにあわせて、各繰り返して



図1 SVGファイルから生成した基準形状

Fig. 1 A base shape generated from SVG file.

表1 RBMの訓練に用いたハイパーパラメータ

Table 1 Hyper-parameters for RBM training

learning rate	momentum	weight decay	dropout rate
0.0001	0.5	0.00001	0.9

の形状の再構築誤差の分布が変化する様子が分かる。派生基準形状の妥当性の確認が成功したのは、1および3回目の繰り返してである。

青色、オレンジ色、および緑色の各ヒストグラムの分布を調べると、それぞれの平均値付近の頻度が大きくなっている。これは、それぞれの基準とする形状に確率的な弾性変形をおこなうことによりラフ類似形状を生成しているためである。平均的な形状の特徴をよく捉えているといえる。

5. おわりに

スケッチした pictorial symbols をディープラーニングにより学習することを目的として、予め与えた少数の基準形状に基づき学習形状を生成するための反復的な手法を提案した。

提案では、基準形状から多様な類似性をもつスケッチ風の形状(ラフ類似形状)を生成するための既提案手法[8]を、派生基準形状を用いるように拡張をおこなった。また、基準形状と派生基準形状とを補間する形状を学習形状として選択することが可能なように、派生基準形状を選択する手法を提案した。

提案手法を実装することにより、与えられた pictorial symbols を基にして派生基準形状を生成することが可能であることを示した。また、基準形状と派生基準形状との補間形状を学習形状として選択することが可能な派生基準形状を生成することが可能であることを示した。

今後の課題には、提案手法の複雑な形状への適用、および、生成した学習形状を用いてのスケッチした pictorial symbols のディープラーニングに関する研究などを挙げるができる。



図 2 基準形状を基に生成したラフ類似形状の例

Fig. 2 Examples of roughly similar shapes generated with a base shape.

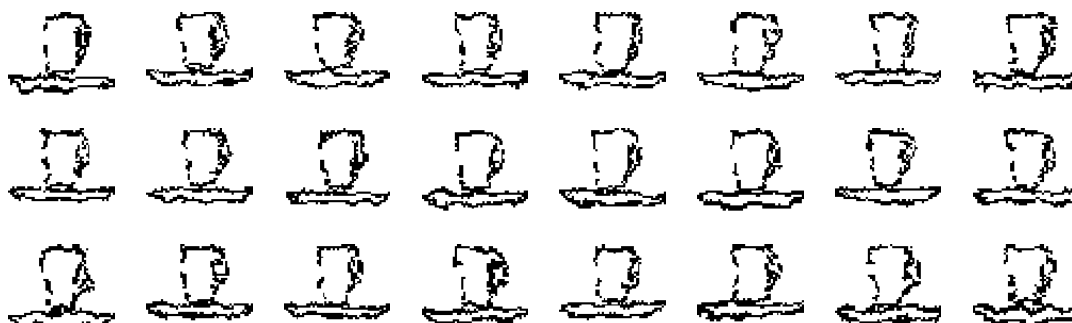


図 3 派生基準形状を基に生成したラフ類似形状の例

Fig. 3 Examples of roughly similar shapes generated with a derived base shape.

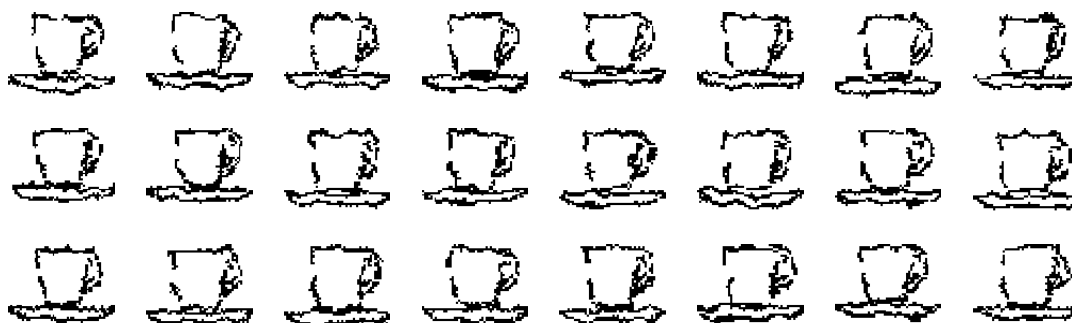


図 4 テスト形状を基に生成したラフ類似形状の例

Fig. 4 Examples of roughly similar shapes generated with a test shape.

表 2 アルゴリズム 1 の各繰り返しでのスケーリングの倍率 (x, y)

Table 2 Scaling factors at each iteration of algorithm 1

iteration	coffee cup	sauser	grab
1	0.500, 1.000	1.000, 1.000	0.500, 1.000
2	0.250, 1.000	1.000, 1.000	0.250, 1.000
3	0.375, 1.000	1.000, 1.000	0.375, 1.000
4	0.313, 1.000	1.000, 1.000	0.313, 1.000
5	0.344, 1.000	0.032, 1.000	0.344, 1.000

参考文献

- [1] Edwards, C.: Growing Pains for Deep Learning, *Commun. ACM*, Vol. 58, No. 7, pp. 14–16 (online), DOI: 10.1145/2771283 (2015).
- [2] Eitz, M., Hays, J. and Alexa, M.: How do humans sketch objects?, *ACM Trans. Graph.*, Vol. 31, No. 4, pp. 44:1–44:10 (online), DOI: 10.1145/2185520.2185540 (2012).
- [3] Fu, L. and Kara, L. B.: Technical Section: Neural Network-based Symbol Recognition Using a Few Labeled Samples, *Comput. Graph.*, Vol. 35, No. 5, pp. 955–966 (online), DOI: 10.1016/j.cag.2011.07.001 (2011).



図 5 各繰り返しの派生基準形状の候補形状

Fig. 5 Candidate shapes for derived base shape at each iteration: From left to right; Candidate shapes at the first, the second, ..., and the fifth iteration.



図 6 各繰り返して用いた学習形状とテスト形状の例

Fig. 6 Examples of training shapes and test shapes used in each iteration. From the first to the fifth row: shapes used in the first iteration, ..., and the fifth iteration respectively. In each row, the left four shapes are the training shapes, and the others are the test shapes.

- [4] Geman, S., Bienenstock, E. and Doursat, R.: Neural Networks and the Bias/Variance Dilemma, *Neural Comput.*, Vol. 4, No. 1, pp. 1-58 (online), DOI: 10.1162/neco.1992.4.1.1 (1992).
- [5] Hinton, G. E. and Salakhutdinov, R. R.: Reducing the Dimensionality of Data with Neural Networks, *Science*, Vol. 313, No. 5786, pp. 504-507 (2006).
- [6] Hinton, G. E., Osindero, S. and Teh, Y.-W.: A Fast Learning Algorithm for Deep Belief Nets, *Neural Comput.*, Vol. 18, No. 7, pp. 1527-1554 (online), DOI: 10.1162/neco.2006.18.7.1527 (2006).
- [7] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25* (Pereira, F., Burges, C., Bottou, L. and Weinberger, K., eds.), Curran Associates, Inc., pp. 1097-1105 (2012).
- [8] 佐藤 信: スケッチした pictorial symbols のディープラーニングのための部分的弾性変形を用いた形状学習データ生成, 情報処理学会研究報告, Vol. 2015-CG-160, No. 7, pp. 1-8 (2015).
- [9] Mitchell, T. M.: The Need for Biases in Learning Generalizations, *Readings in Machine Learning* (Shavlik, J. W. and Dietterich, T. G., eds.), Morgan Kaufman, pp. 184-191 (1980).
- [10] Modley, R. and Myers, W.: *Handbook of Pictorial Symbols: 3,250 Examples from International Sources*, Dover pictorial archive series, Dover Publications (1976).
- [11] Schmidhuber, J.: Multi-column Deep Neural Networks for Image Classification, *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, Washington, DC, USA, IEEE Computer Society, pp. 3642-3649 (online), available from <http://dl.acm.org/citation.cfm?id=2354409.2354694>

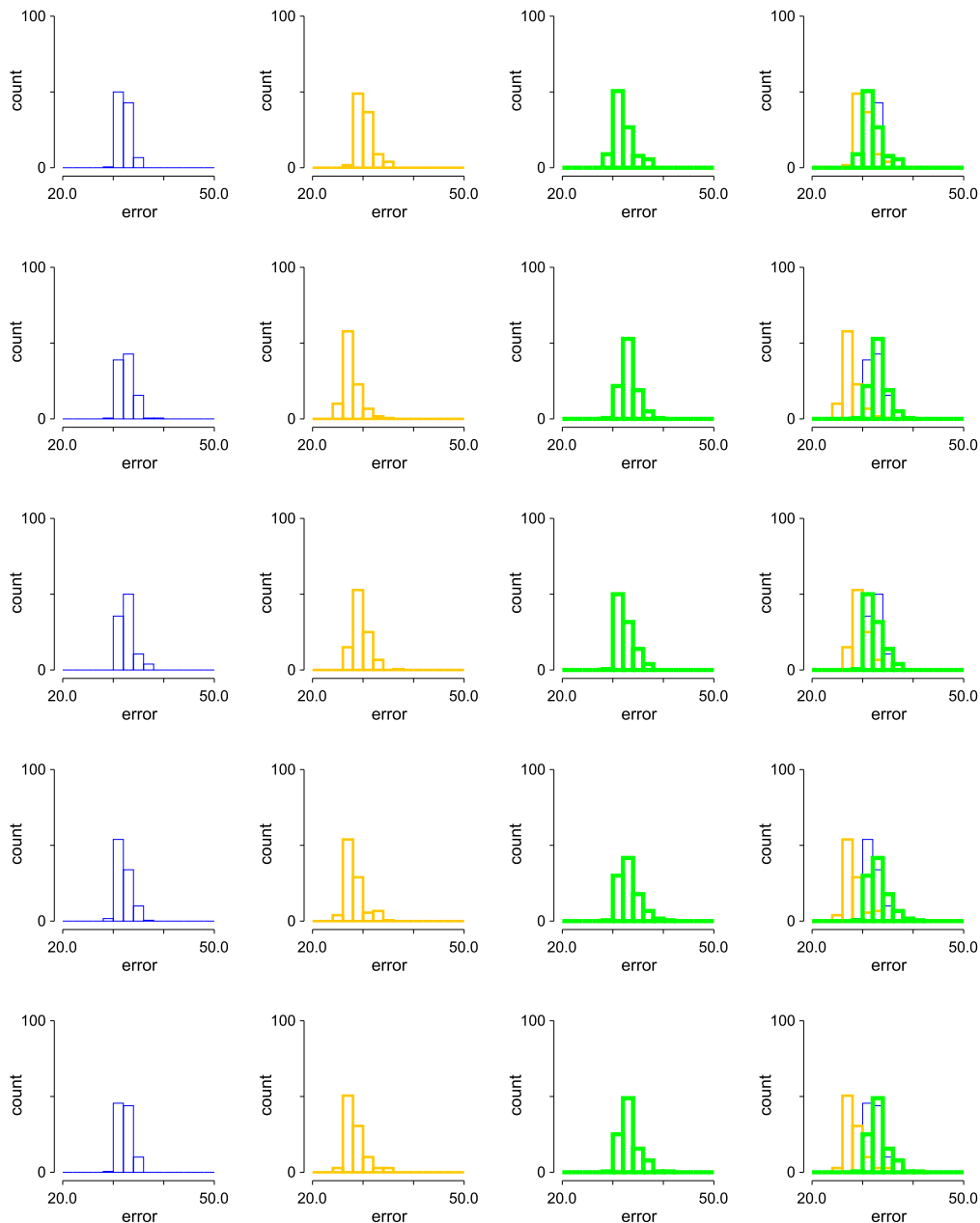


図 7 各繰り返しでの学習形状およびテスト形状の RBM を用いた再構築誤差

Fig. 7 Examples of RBM reconstruction errors of training and test shapes in each iteration. From the first to the fifth row: the reconstruction errors at the first, ..., and the fifth iteration respectively. In each row, from left to right: the reconstruction errors of the training shapes generated from a base shape, of the training shapes generated from a derived base shape, and of the test shapes. In the rightmost column, the left three graphs are put together into a whole.

(2012).

- [12] Simard, P. Y., Steinkraus, D. and Platt, J. C.: Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2, ICDAR '03*, Washington, DC, USA, IEEE Computer Society, pp. 958–963 (2003).
- [13] W3C: *Scalable Vector Graphics (SVG) 1.1 Specification (Second Edition)*, <http://www.w3.org/> (2011).