情報システム工学科

2009 年 4 月 12 日

塩貝亮宇

Supporting Online Material for Reducing the Dimensionality of Data with Neural Networks

T1076006 塩貝亮宇

1 Details of the pretraining

各 RBM の事前学習を高速化するために、全てのデータセットをミニバッチに分割し、含まれる 100 個のデータベクトルと各ミニバッチ化した後の重みを更新した。ミニバッチのサイズで割りきれないデータセットについて、残りのデータベクトルは最後のミニバッチに含められる。全てのデータセットにおいて、各隠れ層は全ての訓練セット 50 通りについて事前学習される。各ミニバッチは学習率 0.1 で論文内の式 (1) と平均値を使いミニバッチ化した後の重みを更新する。

更に、前の更新 0.9 倍が各重みに加えられ、そして重みの値 0.00002 倍が広範の重みのペナルティとして差し引かれる。

重みは平均 0.0、標準偏差 0.1 の正規分布から小さな標本値が抽出されて初期化される。

- 学習率 0.1
- 前の更新 0.9 倍
- 重みの値 0.00002 倍
- 平均 0、分散 0.1 の正規分布から

2 Details of the fine-tuning

fine-tuning について、1000 データベクトルを含む大きなミニバッチ上で共役勾配法を使った。々は Carl Rasmussen の書いた ‘minimize‘コード (1) を使う。(three line search,3 線形探索) は各世代内の各ミニバッチについて形成される。

適切な世代数を決定し、オーバフィッティングについてチェックするために、訓練データの極一部で各 (自動符号化,autoencoder) を fine-tune し、そして、それの残りについてテストした。私たちは、そのとき全ての訓練セットにおいて fine-tuning を置換した。

合成曲線と手書き数字について、私たちは fine-tune の 200 世代 ; 顔については 20 世代、ドキュメントについては 50 世代を用いる。

(軽視する, 非常に小さい,slight) オーバフィッティングは顔について観測されるが、他のデータセットにオーバーフィッティングは存在しない。

オーバフィッティングは訓練データが終わりへ向うことを意味するが、その (復元,reconstraction) はまだ訓練セットで改善されている最中で、しかし、(検証セット,validation set) より悪いものが与えられている。

我々は学習率と (モーメンタム,momentum)、(重みの低下,weight-decay) の様々なパラメータについて実験し、そして、RBM のもつ更なる世代についての訓練に挑戦する。

私たちは fine-tuning 後の最終的な結果にどんな重要な違いも観測しなかった。

これは (正確な値,precise weight) を (,greedy pretraining) により重みを発見し、fine-tuning を開始するための良い領域を発見する限りは長さが同じで

それは fine

によって気にされない。

- CG 法を使って 1000 データベクトルを含む

- Carl Rasmussen's 'minimize'コードを使う
- fine-tuning の 200 世代を使う
- 顔について 20 世代とドキュメントと 50 世代

3 How the curves were generated

合成曲線を生成するために、その前の点の x 座標より少なくと 2 つ良い各点の x 座標を制約した。
私たちは $[2, 26]]$ の範囲内で (横たわる、 のままである,to lie) 全ての座標についても制約を与える。
その三点は、論文中の図 2 で示される 28x28 画素のイメージを生成するために印付けられる 3 次スプラインで定義される。
(inking procedure) の詳細については (2) に記述されており、その (matlab) コードは (3).

- 少なくとも 2 つ良い x 座標の各点を制約
- $[2, 26]$ の範囲内で制約を与える
- 28x28 画素のイメージを生成するために 3 次スプラインを使う

4 Fitting logistic PCA

ロジスティック主成分分析をフィットさせるために入力もロジスティック出力素子も直接に結合された (線形コード素子,linear code unit) である (自動符号化,autoencoder) を使い、そして、共役勾配法を使ってクロスエントロピー誤差を最小化する。

- ロジスティック PCA をフィットさせるために autoencoder を使う
- CG 法を使ってクロスエントロピー誤差を最小化

5 How pretraining affects fine-tuning in deep and shallow autoencoders

図 S1 では事前学習のパフォーマンスとランダムに初期化された曲線データセット上の (自動符号化, autoencoder) を比較した。
図 S2 はパラメータが同じ数を持つ (自動符号化,autoencoder) の deep と shallow のパフォーマンスを比較した。
それら全てについて比較すると、平均 0、標準偏差 0.1 の正規分布から抽出される小さな乱数により重みが初期化される。

- 図 S1、autoencoder と pretraine を比較
- 図 S2、deep と shallow のパフォーマンスを比較
- 重みは平均 0 標準偏差 0.1 の正規分布で初期化

クロスエントロピー

$$
\begin{aligned}
H(p,q) &= E_p[-\log q] && (1)\\
&= H(p) + D_{KL}(p||q) && (2)\\
H(p,q) &= \begin{cases} -\sum_x p(x)\log q(x) \\ -\int_X p(x)\log q(x)dx \end{cases} && (3)\\
D_{KL}(p||q) &= \begin{cases} \sum_i P(i)\log \frac{P(i)}{Q(i)} \\ \int_{-\infty}^{\infty} p(x)\log \frac{p(x)}{q(x)}dx \end{cases} && (4)
\end{aligned}
$$

6 Details of finding codes for the MNIST digits

MINIST 数字について、全ての訓練手続きはオリジナルの画素の (intensity, 明度) が正規化されて $[0,1]$ の区間内に納まる。

それらは圧倒的多数の値と、それ故に、ロジスティックやガウシアンより更に良くモデル化された。

図 S3 と図 S4 は PCA によって生成された 2 次元コードと 2 符号素子だけを伴う自動符号化による 2 次元コードを可視化するための代替手段だ。それら可視化の代替手段は実際の数字画像の多くを表わす。

私たちは最初の隠れ層内に 1000 素子をともなう (自動符号器,autoencoder) を用いて私たちの実験結果を獲得した。

これは RBM についての問題を引き起こさない画素数を越えるといった事実だ。

-これは 1 つの隠れ層の自動符号器としての画素を単純にコピーしないだろう。

後述する実験では 1000 を 500 に減少させたときを示し、自動符号化のパフォーマンスにはとても小さな変化がある。

- 画素の明度は正規化されて $[0,1]$ の区間に納まる
- 訓練セット $60,000$ 画像
- 検証データ $10,000$ 画像
- 隠れ素子 1000 個で自動符号器の結果を獲得
- 後の実験では 1000 から 500 個に素子を減少させる

7 Details of finding codes for the Olivetti face patches

我々は 40 人の異なる人物のそれぞれの 64x64 画素の 10 枚を含む顔パッチを Olivetti 顔画像データセットから獲得した。私たちは 25x25 画素の $165,600$ 枚のデータセットを復元した。回転率は $(-90$ から $+90)$、拡大率は (1.4 から 1.8) で (トリミング,cropping) し、オリジナル 400 画像から (副標本、subsampling) を抽出。トリミングされた画像の明度は変換され、全ての画素がゼロ平均を持つ。そして、全てのデータセットは平均画素の分散を 1 とするための 1 つの数字によって拡大化される。

そのデータセットは $124,200$ 訓練画像に分割されるが、最初の 30 人を含み、そして、$41,400$ テストデータには残りの 10 名が含められる。

その訓練セットは更に $103,500$ 訓練画像と $20,700$ の検証画像、25 の (互いに素な,disjoint) 組と 5 人を含んでいる。

2000 の 2 値特徴の最初の層を事前学習したとき、各実値画素の明度は素子の分散によるガウス分布でモデル化される。

事前訓練する特徴のこの最初の層は振動を避けるために、より小さな学習率を必要とする。

学習率には 0.001 がセットされ、そして、事前訓練は 200 世代について (開始した,proceed)。

私たちは画素より更に良い (特徴検出器,feature detector) を使う。なぜならば、(実値,real value) の画素の明度はバイナリ特徴の振舞いよりはるかに情報が含まれているからだ。その高層の事前訓練は全ての他のデータセットについて実行される。

知覚的に重要な復元するための自動符号器の能力は、顔の高頻出の詳細は画素の完全な二乗誤差を反映していない。

これは知覚を持つ類似点の評価について良く知られている画素の二乗誤差が不適切な例である。

- 40 人の異なる人物の 64x64 画素の 10 枚の顔パッチを Olivetti 顔データから獲得
- 25x25 画素の 165, 600 枚を復元
- 回転率は −90 から +90、拡大率は 1.4 から 1.8 でトリミングし、オリジナル画像から 400 枚の副標本を抽出
- 学習率は 0.001 にセットし、200 世代について事前訓練を開始

8 Details of finding codes for the Reuters documents

Reuters Corpus Volume 2 内の 804, 414 の newswire story は手動で 103 トピックに分類される。

そのコーパス (言語資料) は 4 つのメジャーグループに覆われる。

- 法人/工業
- 経済
- 政治/社会
- 市場

ラベルは事前訓練や fine-tuning の間中には使われない。

データはランダムに 402, 207 訓練ストーリーと 402, 207 テストストーリーに分割され、そして、訓練セットはさらに 302, 207 訓練ドキュメントと 100, 000 の (検証ドキュメント,varidation document) にランダムに分割される。

(,common stopwords) はドキュメントから取り除かれ、そして、残りの単語は (,common ending) を取り除くことにより茎付けされる。

しかし、トップ層内の 2000 のロジスティック素子。

事前訓練の間中、2 つ以上の代替へ向けたロジスティックを一般化するための ‘softmax‘を使うことで計算される可視素子の (談笑される活動,confabulated activity):

$$\hat{p}_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{5}$$

ここで、$\hat{p}_i$ は単語 i と正のバイアス項により特徴付けられた活動で生成される重み付けられた入力 x_i のモデル化された確率である。

‘softmax‘の使用により学習則に影響は与えることはない (4) が、単語 (over, について) の確率分布からドキュメントは N の観測者を含むといった事実について許すために、単語 i から特徴 k へ向う重み w_{ik} には特徴から重みへ向う重み w_{ki} の N 倍がセットされる。

Latent Semantic Analysis(LSA) を伴う比較について、各単語のカウント数 c_i を $\log(1+c_i)$ によって置き換えた LSA の版を使った。

この標準的な前処理トリックは頻出単語の影響の (重みの降下,down-weighting) により LSA のパフォーマンスをわずかに改善する。

- Latent Semantic Analysis
- softmax を使って計算される可視素子の (談笑活動,confabulated activity)
- 単語 i から特徴 k へ向う重み w_{ik}
- 特徴 k から単語 i へ向う重み w_{ki}
- 単語のカウント数 c_i
- LSA は $\log(1+c_i)$ により置き換えられる

9 A comparison with Local Linear Embedding

復元処理での Local Linear Embedding をともなう自動符号化は比較することが難しい、なぜならば、様々な別の非パラメトリックな次元を減少させる手続きのような、LLE はテスト画像を復元する単純な方法を提供しないからだ。

我々は、そのためにドキュメントを修復する処理で自動符号化をともなう LLE と比較する。

LLE は使うために気の効いた手法ではない、なぜならば、それぞれ新しい問い合わせドキュメントについて高次元空間の最近傍の発見に関与するからである。

しかし、ドキュメント回復は、少なくとも、低次元コードがどう良いか評価する方法を私たちに与える。

LLE フィッティングは訓練ケースの数の二次方程式をとり、そのため、私たちは 11,314 訓練ドキュメントと 7,531 テストドキュメントだけの ‘20 newsgroup‘コーパスを使う。

そのドキュメントは Usenet newsgroup collection からポストされている。

そのコーパスは 20 の異なるニュースグループに公平にムラ無く分割されており、各対応するトピックに分離される。

そのデータは前処理され、データによって組織化され、そして (6) によって得られる。

そのデータはデータによって分割され、だから訓練とテストセットは時間内に分割される。

いくつかのニュースグループは互いに関係が非常に近く、’e.g.soc.religion.christian’ と’talk.religion.misc’、他のもう一方はとても異なっており、’e.g.rec.sport.hockey’ と’comp.graphics’.

私たちはドキュメントから’common stopwords’ を取り除き、そして、(common ending, 現代の終端語) による残りの単語もまた茎付けた。

(Reuters corpus, ロイターコープス) に関して、私たちは訓練データセット内の最頻出単語 2000 だけを考えた。

LLE は各データポイントの K 近傍が与えられなければならず、そして、私たちは最近傍を見つけるために (数えたベクター, count vector) の間の角度の余弦を使った。

LLE のパフォーマンスは K について選択された値に依存するため、私たちは $K = 5, 10, 15, 20, 25, 30$ と試した。そして、ベストな K 値を使った結果を常にレポートする。

私たちは、(1 つの, ある,one) 結合された構成要素が形成されるための全てのデータセットについて十分な広さだった K についてもまた検査した。($K = 5$ について、21 ドキュメントの切断された構成要素が存在された。)

テストフェーズの間中、各問い合わせドキュメント q について、

訓練セットからの K 近傍 (count vector, カウントベクター) を (識別する, 同一視する,identify) し、そして、

その近傍から (count vector, カウントベクトル) の復元ついてベストな重み $\boldsymbol{w}$ を計算した。

私たちは、それの高次元 K 近傍の低次元コードから q について低次元コードを生成するための同様の重み $\boldsymbol{w}$ を使う。LLE コードには (8) が使われる。

2 次元コードについて LLE($K = 10$) のパフォーマンスは LSA より良いが私たちの自動符号化よりは悪い。

高次元コードについて LLE($K = 25$) のパフォーマンスは LSA のパフォーマンスにとても似ており、(図 S5 参照) そして、私たちの自動符号化よりとても悪い。

私たちは LLE を適用する前にドキュメントカウントベクトルの二乗距離を正規化することを試みるが、これは助けにならない。

10 Supporting text

同様の事前訓練手続きはクラス化手続きでの一般化を改善できることを表わすために、私たちは MNIST 数字認識処理の'(permutation invariant, 順列不変量)' 版を使った。

学習プログラムを与えられる以前は、全ての画像はその同じランダムな順列の画素を経験する。

これは例えば、画像のアフィン変換や (共有される局所受容野,local receptive fields with shared weight) などの (配置、幾何学, geometry) について、事前情報を使用することから学習プログラムを防ぐ。

(順列不変タスク,permutation invariant task) では、Support Vector Machine は 1.4% に至る (10)。

784-800-10 網について backpropagation1.6% で訓練を受けたランダムに初期化されたニューラルネットについて最良の結果を発表した (11)。

事前訓練はオーバーフィッティングを減少させ、そして、学習を高速化する。

だから、1.2% に至るより大きな 784-500-500-2000-10 ネットを使うことが出来た。

事前学習された (自動符号器,autoencoder) とみなす同様の手法で全 60,000 訓練ケースの 100 世代について 784-500-500-2000 ネットを事前学習した。

しかし、トップ層に 2000 のロジスティック素子を持つ。その事前学習は、クラスラベルについてどんな情報も持たない。

私たちは、そのときトップ層へ向けた'softmax' された出力素子 10 個と事前訓練により見つかった過度に摂動する重みを避けるためにとてもやさしい学習率をともなうクロスエントピー誤差で単純な勾配降下法を使った fine-tune された全てのネットワークを結合した。

最後の層を (除いては,for all but)、バイアスについて 0.1、重みについて 0.03 の学習率だ。

学習を高速化させるために各重みの更新には前の重み増加量の 0.8 倍が加えられる。

そのバイアスと出力素子 10 個の重みについて、事前訓練から破壊された情報の危険はない、故に、それらの学習率は 5 倍大きく、それらもまた 5×10^{-5} 倍の (規模の二乗?,squared magniture) のペナルティを持つ。

fine-tuning の 77 世代後、訓練データでのクロスエントロピー誤差の平均は事前に指定した閾値を (下回った,fell below)、そして、(fine-tuning, 微修正) は停止された。

あの点でのテストエラーは 1.17%。事前訓練のパフォーマンスにより決定される閾値と訓練ケース 50,000 上での (微修正,fine-tuning) と

検証セットで最小の分類誤差を与えて訓練するクロスエントロピー誤差を決定するための検証セットとして残っている訓練ケース 10,000 が使われる。

私たちは 1000 のデータベクトルに含まれるミニバッチで共役勾配法を使用する全てのネットワークにもまた fine-tune した。(,three line searche) は各世代の各ミニバッチについて形成された。

48 世代後の fine-tuning、テストエラーは 1.14% になった。
fine-tunng の停止基準は後述する同様の手法で決定される。
そうした分類器の訓練についての Matlab コードは http:www.cs.toronto.edu/ hinton/MatlabForSciencePaper.html で入手できる。

- オーバーフィッティング; 少数の個別パターンを多数のパラメータを持つ複雑な関数で誤差ゼロで近似する
- MNIST 数字認識処理の permutation invariant 版
- 784-800-10 網は backpropagation1.6%
- 784-500-500-2000 網は 1.2%
- 784-500-500-2000 網 60, 000 訓練ケースを 100 世代
- 最後の層を除いてバイアス項 0.1, 重み 0.30 の学習率
- 前の重みの増加量の 0.8 倍が加えられる
- 1.17%のテストエラー
- データベクトル 1000 個に含まれるミニバッチに共役勾配法を使用する
- 48 世代後の fine-tuning でテストエラーは 1.14%
- アフィン変換:平行移動を伴う線形写像 (回転、拡大、剪断)

10.1 How the energies of images determine their probabilities

画像のエネルギーから、それらの確率をどのように決定するか:
モデルに可視ベクトル $\boldsymbol{v}$ を代入した確率、$\boldsymbol{v}$ は

$$p(\boldsymbol{v}) = \sum_{\boldsymbol{h}\in H} p(\boldsymbol{v},\boldsymbol{h}) \tag{6}$$

$$= \frac{\sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v},\boldsymbol{h}))}{\sum_{\boldsymbol{u},\boldsymbol{g}} \exp(-E(\boldsymbol{u},\boldsymbol{g}))} \tag{7}$$

ここで、H は全てのとりうることが可能な隠れ素子の 2 値ベクトルの組である。
$\log p(\boldsymbol{v})$ の勾配へ続けるために w_{ij}(に関しては, w.r.t.) 定常分布に至るまで、その特徴状態と画素状態の間を交互にの更新する必要がある。
この定常分布での $v_i h_j$ の期待値は (1 ステップ談笑,one-step confabulation) についての期待値の代替に使われる。
加えて、より一層に遅くなるため、この最尤推定学習の手続きは、サンプリングするノイズの影響をより一層に (被る,suffer from)。
学習信号内でのノイズの減少を促進するために、
「データ」層で既に学習された特徴検出器 (または画素) のバイナリ状態は特徴検出器の次層が学習するとき、それらの活動の真値確率によって置換されるが、新しい特徴検出器は常に (,they) を運ぶことができる情報の量を制限するために (統計的バイナリ状態,stochastic binary state) を持つ。

10.2 The energy function for real-valued data

線形可視素子とバイナリ隠れ素子が使われるとき、線形素子が素子の分散にともなうガウシアンノイズを持つ線形素子を仮定すれば、エネルギー関数と更新則はかなり単純である。

分散が 1 でないと仮定すると、そのエネルギー関数は:

$$E(\boldsymbol{v},\boldsymbol{h}) = \sum_{i\in pixels}\frac{(v_i-b_i)^2}{2\sigma_i^2} - \sum_{j\in features} b_j h_j - \sum_{i,j}\frac{v_i}{\sigma_i}h_j w_{ij} \tag{8}$$

ここで σ_i は可視素子 i についてのガウシアンノイズの標準偏差だ。

隠れ素子についての統計的な更新則は σ_i によって分割された各 v_i を除いたものと同じままである。

可視素子 i についての統計的な更新則は平均

$$b_i + \sigma_i \sum_j h_j w_{ij} \tag{9}$$

と分散

$$\sigma_i^2 \tag{10}$$

のガウシアンからサンプリングするためである。

10.3 The differing goals of Restricted Boltzmann machines and autoencoders

(自動符号器,autoencoder) とは異なり、事前訓練アルゴリズムの狙いは正確には各訓練画像を復元するためでない、しかし、画像の分布と同じである (談笑分布,confabulation distribution) を作るためには画像の分布と同じでなければならない。

このゴールは例え、全ての A について

$$p(A) = \sum_B p(B)p(A|B)$$

を提供する確率 $p(B|A)$ となる画像 B とみなす (confabulated, 談笑される) 訓練セット内で確率 $p(A)$ で発生する画像 A であっても十分に満足されるだろう。

- 事前訓練アルゴリズムは訓練画像を復元するためのものではない
- (confabulation distribution, 談笑分布) と画像の分布は同じ分布

11 Supporting figures

図 S1:訓練データ曲線で fine-tuning 中のテスト画像毎の (平均二乗誤差復元?,average squared reconstruction error)

左図：deep 784-400-200-100-50-25-6 自動符号器が事前訓練後に進捗を急速させるが、事前訓練無しでは進捗がない。

右図：shallow 784-532-6 自動符号器は事前訓練無しで学習できるが、事前訓練は fine-tuning をより高速化し、そして事前訓練は fine-tuning の反復 10 回より少ない回数をとる。

図 2：テストデータ上での画像毎の平均二乗誤差による復元は曲線データセットで fine-tuning 中を示される。

A 784-100-50-25-6 自動符号器は同数のパラメータを有する shallower 784-108-6 自動符号器よりわずかに良く機能する。

両方の自動符号化は事前学習された。

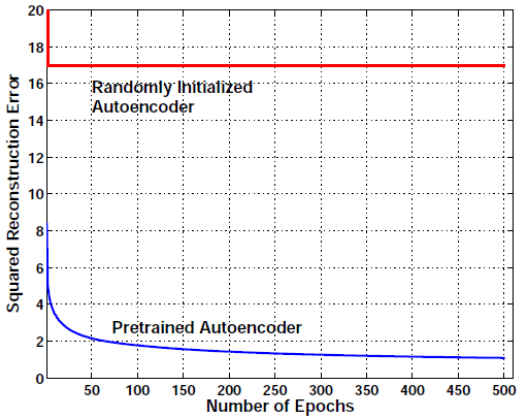

図 1: deep 784-400-200-100-50-25-6

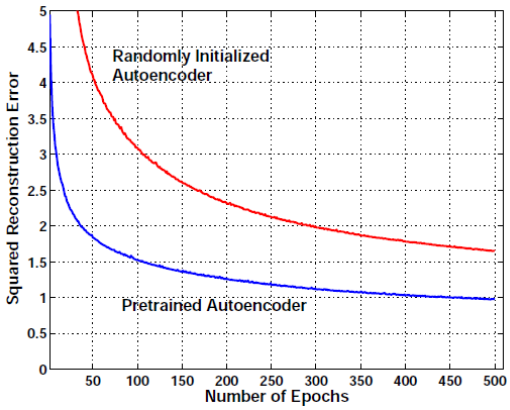

図 2: shallow 784-532-6

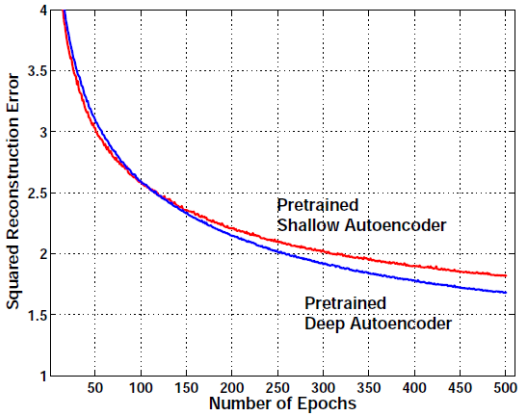

図 3: A 784-100-50-25-6 autoencoder

図 3：2 次元コードの可視化代替手段は全 60,000 訓練画像の最初の 2 つの主要な構成をとることによって生み出される。数字の 5,000 画像 (500 毎クラス) はランダムオーダーでサンプリングされる。まだ表示されていないどんな画像でも重複せずに各画像は表示される。

図 4：2 次元コードの可視化代替手段は全 60,000 訓練画像に向けた 784-1000-500-250-2 自動符号器により生成された。数字の 5,000 画像 (500 毎クラス) はランダムオーダーでサンプリングされる。まだ表示されていないどんな画像でも重複せずに各画像は表示される。

図 S5: テストセットからドキュメントが問い合わされた際の精度曲線は他のテストセットドキュメントを取り戻すために使われ、全ての取り得る 7,531 の可能な問い合わせ全体を平均化した。

- 図 S1 左：deep 784-400-200-100-50-25-6 自動符号器
- 図 S1 右：shallow 784-532-6 自動符号器
- 図 S2 :平均二乗誤差による復元
- 図 S3 :2D コードの可視化
- 図 S4 :2D コードの可視化
- 図 S5 :精度曲線

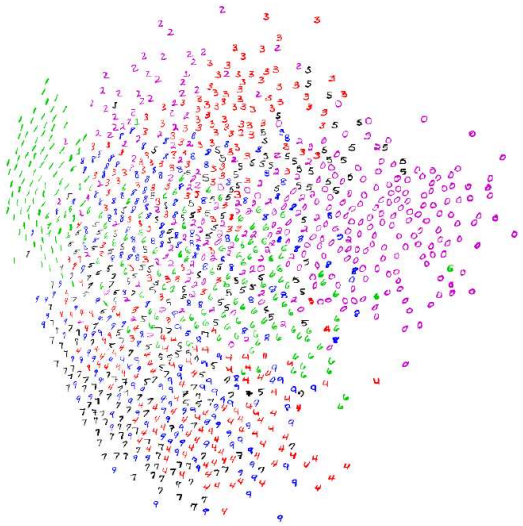

図 4:

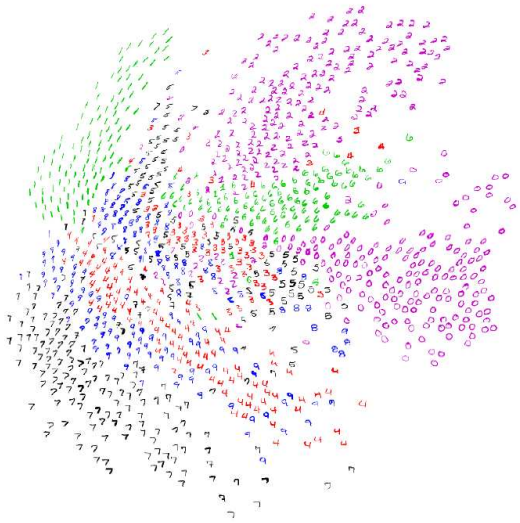

図 5: a 784-1000-500-250-2 autoencoder

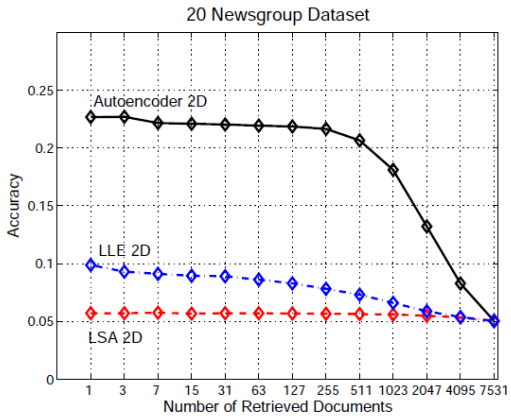

図 6: accuracy curve

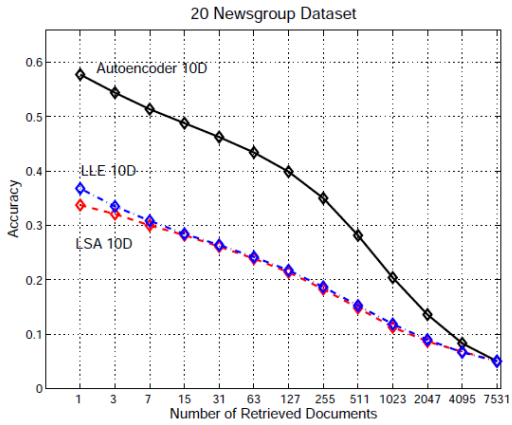

図 7: accuracy curve

12 References and Notes

参考文献

[1] For the conjugate gradient fine-tuning,we used Carl Rasmusses's 'minimize' code available at http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/