

# ロボットインテリジェンス レポート課題 A

## 「ニューラルネット学習シミュレーション」

03-140299 東京大学機械情報工学科 3 年 和田健太郎

2015 年 1 月 13 日

### 1 はじめに

レポート課題として課題 A を選択し、3 層フィードフォワード型のニューラルネットとバックプロパゲーション学習をシミュレーションするプログラムを作成し、識別実験を行った。実験に利用したデータ群は The MNIST database of handwritten digits であり、このデータは過去に様々な分類器において識別能力を図るために利用されている。[1] 図 1 に実際に利用した学習データの一部を示す。

MNIST dataset

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 4 | 6 | 9 | 7 | 3 | 7 | 2 | 5 | 1 | 5 | 4 | 9 | 3 | 3 | 9 | 0 | 5 | 7 |
| 3 | 3 | 0 | 5 | 8 | 9 | 3 | 5 | 6 | 0 | 6 | 2 | 8 | 1 | 0 | 2 | 7 | 5 | 3 | 7 |
| 0 | 9 | 2 | 3 | 6 | 4 | 7 | 9 | 1 | 3 | 2 | 8 | 8 | 3 | 5 | 4 | 0 | 1 | 2 |   |
| 2 | 3 | 3 | 6 | 1 | 5 | 3 | 2 | 4 | 1 | 0 | 7 | 4 | 1 | 0 | 5 | 5 | 1 | 2 |   |
| 4 | 8 | 7 | 4 | 1 | 3 | 8 | 5 | 0 | 8 | 7 | 2 | 6 | 9 | 5 | 4 | 4 | 1 | 5 | 1 |
| 0 | 5 | 2 | 2 | 2 | 4 | 2 | 8 | 7 | 0 | 6 | 8 | 7 | 4 | 8 | 4 | 5 | 7 | 0 | 4 |
| 5 | 0 | 2 | 9 | 7 | 5 | 6 | 6 | 3 | 9 | 2 | 0 | 2 | 3 | 2 | 0 | 4 | 6 | 1 | 5 |
| 8 | 6 | 0 | 6 | 7 | 5 | 7 | 0 | 3 | 4 | 0 | 9 | 4 | 2 | 1 | 5 | 8 | 0 | 5 | 3 |
| 7 | 5 | 7 | 0 | 6 | 3 | 3 | 7 | 3 | 1 | 4 | 4 | 9 | 2 | 4 | 5 | 3 | 3 | 3 | 3 |
| 5 | 3 | 9 | 7 | 7 | 4 | 7 | 4 | 6 | 5 | 6 | 5 | 9 | 2 | 7 | 3 | 1 | 1 | 1 |   |
| 7 | 0 | 4 | 8 | 3 | 5 | 2 | 6 | 7 | 6 | 2 | 9 | 7 | 7 | 1 | 0 | 1 | 4 | 6 | 1 |
| 4 | 5 | 0 | 7 | 1 | 1 | 4 | 1 | 2 | 7 | 4 | 4 | 9 | 7 | 7 | 2 | 8 | 9 | 7 | 5 |
| 1 | 1 | 0 | 6 | 6 | 1 | 2 | 1 | 8 | 8 | 4 | 3 | 2 | 2 | 7 | 2 | 8 | 0 | 0 | 0 |
| 6 | 4 | 0 | 9 | 6 | 4 | 2 | 0 | 2 | 0 | 7 | 0 | 2 | 5 | 1 | 8 | 1 | 3 | 1 | 1 |
| 1 | 7 | 8 | 3 | 2 | 8 | 2 | 2 | 3 | 7 | 7 | 9 | 6 | 4 | 8 | 0 | 8 | 6 | 6 |   |
| 0 | 8 | 0 | 6 | 5 | 2 | 6 | 8 | 3 | 9 | 2 | 6 | 4 | 8 | 9 | 6 | 6 | 4 | 6 | 2 |
| 0 | 8 | 2 | 5 | 9 | 2 | 4 | 5 | 1 | 0 | 5 | 1 | 4 | 8 | 7 | 4 | 4 | 3 | 0 |   |
| 2 | 8 | 4 | 5 | 6 | 1 | 2 | 4 | 0 | 7 | 0 | 8 | 0 | 1 | 9 | 3 | 3 | 2 | 8 |   |
| 2 | 0 | 1 | 5 | 4 | 4 | 2 | 3 | 1 | 1 | 0 | 7 | 7 | 9 | 8 | 4 | 5 | 1 | 8 | 4 |
| 4 | 0 | 3 | 1 | 7 | 6 | 0 | 5 | 0 | 3 | 5 | 8 | 6 | 1 | 4 | 0 | 0 | 1 | 7 |   |

図 1: MNIST の画像データ

また、ノイズを加えた場合の性能変化、ノイズ耐性、中間ニューロンの役割、オートエンコーダを利用した画像特徴抽出による識別性能変化について考察した。

### 2 ニューラルネット学習シミュレーション

実験に利用した MNIST は、 $28 \times 28$  ピクセルのグレースケール手書き数字画像の 70000 件のデータセットである。データセットを 3 分割し 3 分の 2 を学習データ、3 分

の 1 をテストデータとし、以下の様なパラメータに関してモデルの性能を交差検定によって調べた。

なお、パラメータの検証は以下の順番で行い、一部を覗いてそれぞれ最良であったものを次の検証で利用することで、性能の向上を目指した。最初のパラメータ値は、学習率 0.2、慣性項の係数 0、隠れ層の数入力 0.1 倍、学習の際のノイズ 0 とした。

- 学習の際の繰り返し数
- 学習率
- 慣性項の係数
- 隠れ層の数
- 学習の際のノイズ率

図 2 が学習の際の繰り返し数を変化させ、各試行での性能を調べたものである。この結果から、性能は繰り返し回数に従って良くなるということがわかる。今回の実験では結果として現れなかったが、繰り返し回数をさらに増やしていくと、性能の向上が打ち止めになり、過学習によって逆に性能が下がるということが予想される。この後の学習では、繰り返しの数を 150000 として解析した。

図 3 が学習率を 0.02 から 0.4 まで変化させ、性能の変化を調べたものである。この結果から、0.25 以降は変化が小さいことがわかる。最良の性能を出したのは、学習率が 0.3 の時であった。また、0.2 のときのように性能が大きく下がるパラメータ値もあるため、パラメータを変化させて性能を検証することの必要性がわかる。

図 4 が慣性項の係数を 0.0 から 0.38 まで変化させ、性能の変化を調べたものである。これは学習率を 0.3 としたときに調べたものであり、0.12 のとき最も性能が高かった。

図 5 は隠れ層のニューロン数を入力層に対する比率として変化させたときに性能がどう変わるかを調べたもの

である。グラフから、隠れ層のニューロン数は今回のデータでは0.15あたりが最適であるとわかる。

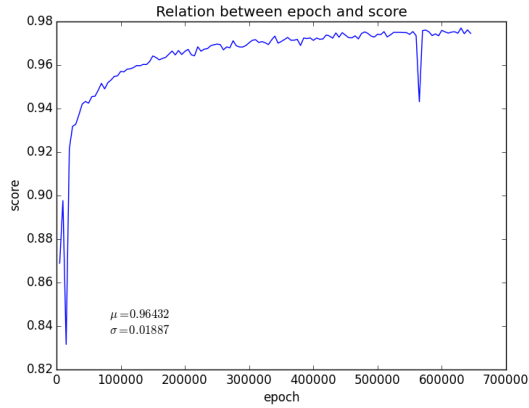


図 2: 学習の際の繰り返し数と性能の関係

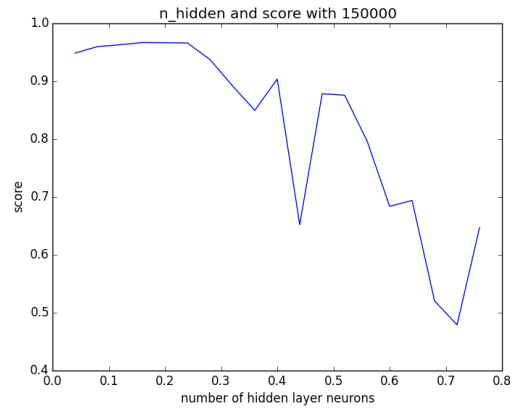


図 5: 隠れ層のニューロン数と性能の関係

図 6 は学習の際のノイズの確率を変化させ、それに伴う性能の変化を調べたものである。

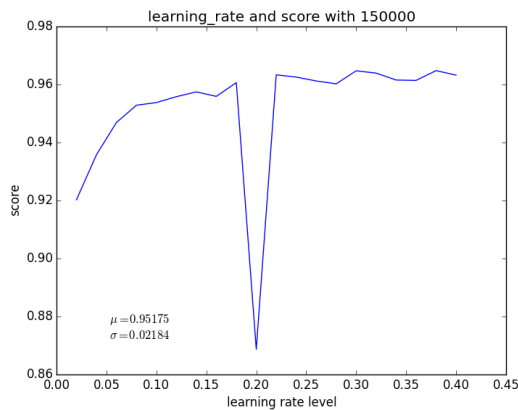


図 3: 学習率と性能の関係

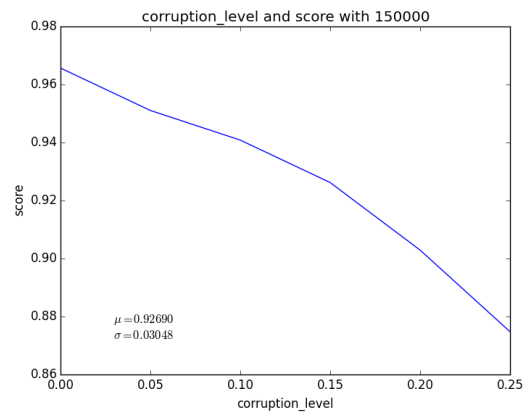


図 6: 学習の際のノイズと性能の関係

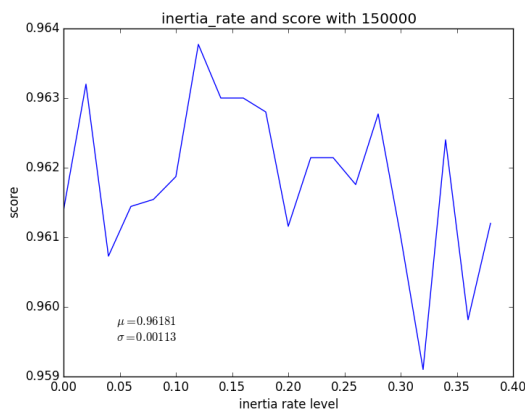


図 4: 慣性項の係数と性能の関係

### 3 ノイズによる性能変化

ある確率で画像のピクセル値をランダムな値に変化させるということによってノイズをテストデータに発生させ、予測性能を調べることでノイズの影響を検証した。

図 1 がノイズなし、図 7 がノイズを 10%、図 8 がノイズを 25% 発生させたデータを、画像としてプロットしたものである。ノイズを 25% 発生させた場合でも、人であれば十分に手書き文字を判別することができる程度である。

ノイズを変化させ識別器の性能を調べたのが図 9 である。グラフより、テストデータに対するノイズの量を増

やすにつれて性能も低下する事がわかる. ( $\mu$ : スコア平均,  $\sigma$ : スコア標準偏差)



図 7: ノイズを 10%発生させたデータ

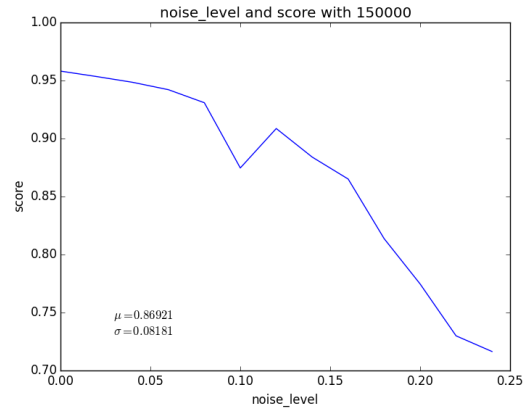


図 9: ノイズと性能の関係

図 10 は学習データにランダムなノイズを 10%加えた識別器に、ノイズ量を変えたテストデータを識別させた場合のスコアの変化を示したものである。図 9 と比較すると、テストデータのノイズを増やしても性能を保っており、スコアの標準偏差も小さいためノイズ耐性が高いといえる。つまりモデルのノイズ耐性は、学習データにノイズが存在するかどうか大きく影響されることがわかる。

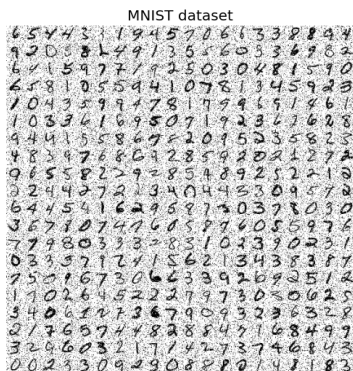


図 8: ノイズを 25%発生させたデータ

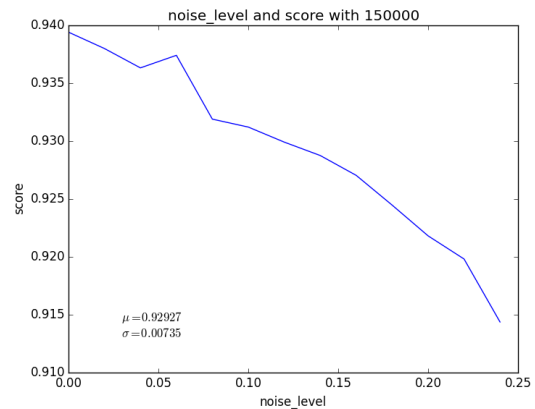


図 10: ノイズと性能の関係 (10%のノイズを発生させたデータで学習させた場合)

## 4 中間層のニューロンの役割

中間層のニューロンの数を変化させ、ニューロン数と識別性能の関係を表したのが図 5 である。このとき、性

能が最も高かった場合と低かった場合についてその重みを図 11 と図 12 にプロットした。

それぞれの図を比較すると、性能が高かったモデルの重みの方がよりエッジがはっきりしており、模様のようなものが見える。重みのうち、値が大きくなっている部分は各入力刺激（入力画像）に対する受容野のような働きをしており、結果から、その働きが顕著になっている方がより正確な出力を出すことができると言える。

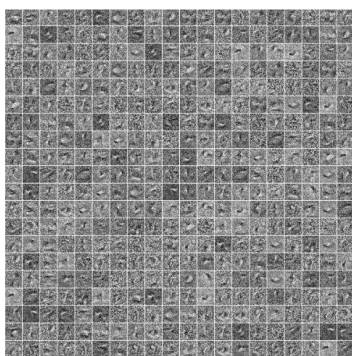


図 11: 識別性能が高い場合 (中間層 0.16) の重み

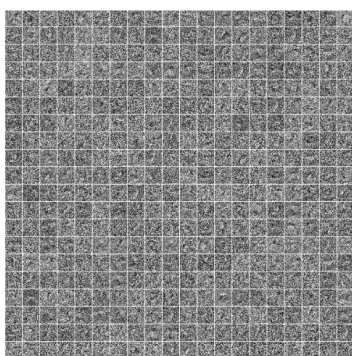


図 12: 識別性能が高い場合 (中間層 0.72) の重み

## 5 画像特徴抽出と性能変化

画像の特徴抽出としてオートエンコーダを用い、その中間層の値を入力として、性能を調べた。パラメータは以下の様に設定した。

- オートエンコーダの中間層の数: 484

- パーセプトロンの中間層の数:  $484 * 0.16$
- 学習率: 0.3
- 慣性項の係数: 0.12
- 学習データへのノイズ: 0.0
- 学習の繰り返し回数: 150000

精度は 0.947 であった。パーセプトロンのパラメータを同じにした場合、オートエンコーダによる前処理なしの場合の精度 0.963 より低くなっているが、パラメータの調整によってより精度が高くなることが期待される。

また各画像データのサイズは、前処理なしの場合には  $28 \times 28 = 784$  であったが、オートエンコーダによる前処理を行った場合は  $22 \times 22 = 484$  であり、約 6 割のデータ量で大きく精度を落とすことなく予測ができている。これより、識別器への入力データに対するオートエンコーダの圧縮機としての機能 [2] が十分に働いていると言える。

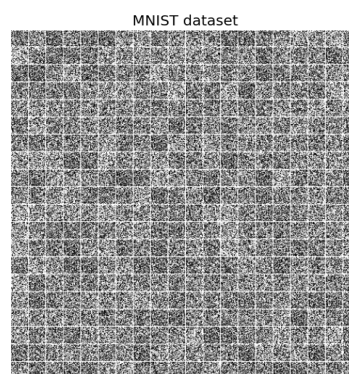


図 13: アートエンコーダの中間層 (パーセプトロンへの入力データ)

## 6 今後の課題

今回の実験では、モデルのパラメータと性能の関係の検証に関して、1 つのパラメータのみを変化させるという手法を取った。より信頼性の高い方法として、Grid Search を用いた方法が挙げられ、組み合わせも含めて最適なパラメータ値を選択することで、より精度の高いモデルの構築が可能だと言える。

また 5 において、オートエンコーダのパラメータ調整を行う必要や、前処理ありの場合も前処理なしの場合と

同じパラメータのパーセプトロンモデルを利用して、同様に最適であるのかということを調べる必要がある。

## 7 講義の感想

知的な機械を実現するということを考えさせられる授業で、現状実現している部分とその限界、そしてそれを解決するためのアプローチについて丁寧に説明があり、自分で考えながら聞くことができたので力になったと思う。必要なところでは黒板を使って説明があるが、個人的にはもう少し黒板を使った授業の部分を増やしてほしいと感じた。

## 参考文献

- [1] Yann LeCun, Corinna Cortes, Christopher J.C. Burges, “MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges”, <http://yann.lecun.com/exdb/mnist/>
- [2] <http://deeplearning.net/tutorial/dA.html>