

第六章

Denoising
AutoEncoder

17nm712t 張天偉

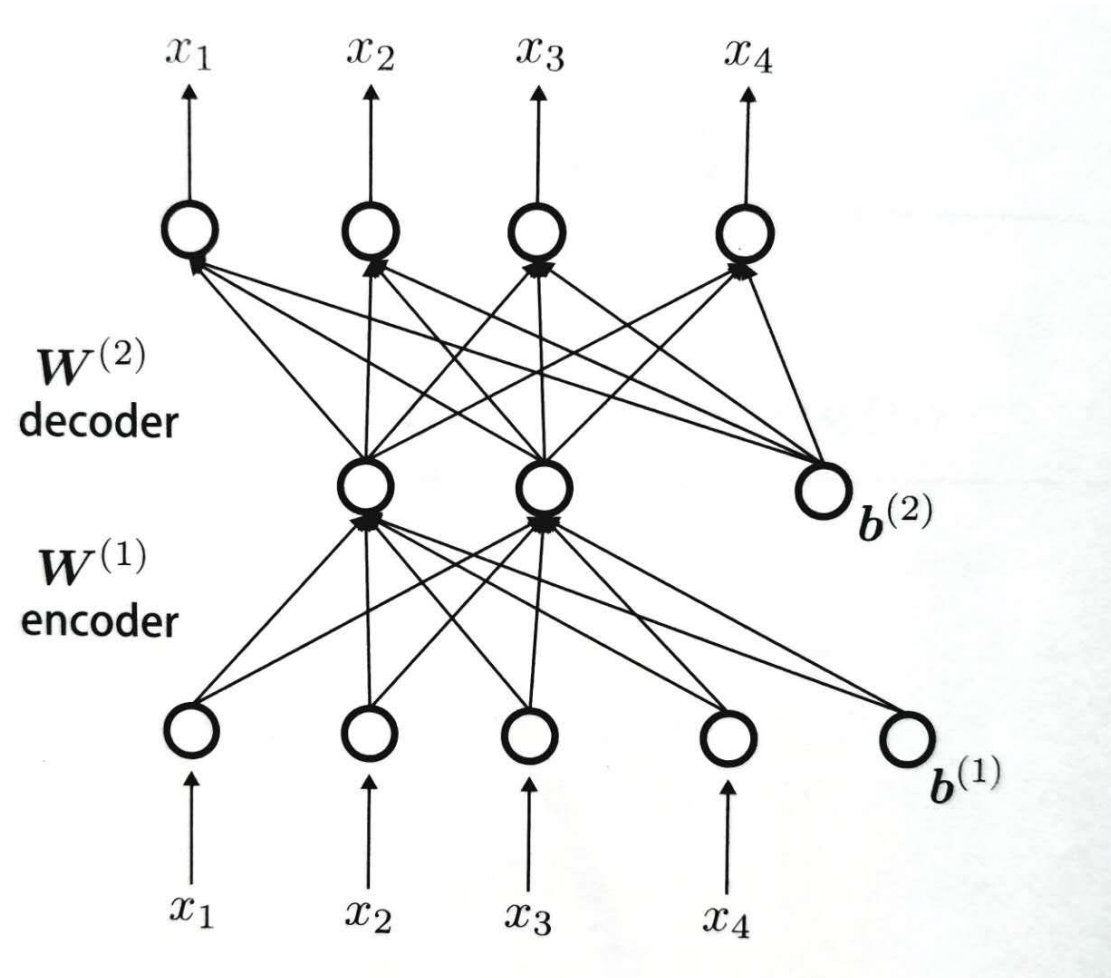
本章ではIrisデータを利用して、簡単な分類問題をChainerで作ってみました。ただ、これは古典的ニューラルネットワークであり、Chainerでの書き方を紹介したものでした。

本章もChainerの書き方の練習として、AutoEncoder (AE)を作ってみます。

6.1 AutoEncoder (AE)

- Autoencoder (AE)とは、出力データが入力データをそのまま再現する3層のニューラルネットワークです。そのため、入力層から中間層への変換器はencoder、中間層から出力層への変換器はdecoderと呼ばれます。
- 中間層のノード数は入力データの次元数以上だとencoderが恒等関数になるので、中間層のノード数は入力データの次元数よりも小さくないと意味はありません。つまり、中間層の出力は入力データを次元縮約したものを表現していると言います。

Irisデータを2次元に縮約してみます。ネットワークで表すと、下記の図で表しています



ポイント：

ChainクラスのMyAEの損失関数の部分で教師信号が入力データ x になっているのがポイントです。

サンプリングを動かしたイメージ

```
[ 0.97688258 0.96485734]
[ 0.95665681 0.95572305]
[ 0.96989977 0.95636898]
[ 0.96133351 0.9403652 ]
[ 0.98012078 0.96305418]
[ 0.97939819 0.95634186]
[ 0.97365934 0.94559944]
[ 0.97219628 0.95699167]
[ 0.95453691 0.93674636]
[ 0.96249604 0.95490026]
[ 0.98007059 0.96913481]
[ 0.97122967 0.94481689]
[ 0.96066582 0.95609951]
[ 0.97046411 0.95611405]
[ 0.9884131 0.98446035]
[ 0.99037296 0.9743467 ]
[ 0.98434579 0.97244054]
[ 0.97491103 0.96202779]
[ 0.97716689 0.96649206]
[ 0.9820233 0.95975178]
[ 0.96594226 0.95890081]
[ 0.97768289 0.95571011]
[ 0.98589718 0.96927965]
[ 0.95296764 0.93552393]
[ 0.96475196 0.92292583]
[ 0.94964933 0.94809675]
[ 0.9649989 0.94387782]
[ 0.97482693 0.963202 ]
[ 0.9731313 0.9665755]
[ 0.96313393 0.93876362]
[ 0.95724928 0.94168043]
[ 0.96501303 0.96185309]
[ 0.98966086 0.96959901]
[ 0.9903605 0.97668523]
[ 0.96249604 0.95490026]
```

Ln: 18 Col: 25

```
[ 0.97041488 0.96835089]
[ 0.97689199 0.97628194]
[ 0.96249604 0.95490026]
[ 0.96271235 0.94448465]
[ 0.97172272 0.95983803]
[ 0.97695994 0.96373385]
[ 0.8995223 0.93633258]
[ 0.97138482 0.94649804]
[ 0.96394098 0.93591118]
[ 0.97434437 0.93178868]
[ 0.95381486 0.94883537]
[ 0.98226774 0.95822561]
[ 0.96832728 0.94768548]
[ 0.98040712 0.96692467]
[ 0.97032887 0.96091139]
[ 0.42297709 0.42892247]
[ 0.46248719 0.36426395]
[ 0.34214824 0.33270624]
[ 0.32998392 0.35000366]
[ 0.31323993 0.33563912]
[ 0.39930254 0.27400717]
[ 0.44526738 0.28342107]
[ 0.56804025 0.51181811]
[ 0.3780525 0.39387292]
[ 0.46936697 0.32818681]
[ 0.39394924 0.45071357]
[ 0.46819496 0.35534433]
[ 0.33617228 0.48994517]
[ 0.36245465 0.27134117]
[ 0.59269685 0.51159006]
[ 0.45375091 0.45945033]
[ 0.42922238 0.23730576]
[ 0.49216717 0.44902399]
[ 0.18481642 0.29004982]
[ 0.44662938 0.44310355]
[ 0.3717311 0.18052536]
[ 0.4679749 0.47684228]
[ 0.2025288 0.22411731]
[ 0.36961907 0.30014384]
[ 0.43700799 0.44611236]
```

Ln: 39 Col: 14

[0.42433619 0.43695414]
[0.29046646 0.34831405]
[0.27022675 0.2426281]
[0.37957025 0.28878814]
[0.56677651 0.60336715]
[0.43441746 0.45007497]
[0.47251022 0.49979806]
[0.48520017 0.46829328]
[0.21952379 0.14984334]
[0.43775615 0.21242154]
[0.52722311 0.29235464]
[0.38245395 0.35423291]
[0.24472651 0.37186384]
[0.54050475 0.3705126]
[0.39303696 0.35894355]
[0.37906134 0.27752343]
[0.41145357 0.29961985]
[0.43389359 0.43395883]
[0.52987373 0.52477652]
[0.42114255 0.33008313]
[0.53965187 0.3781552]
[0.48456329 0.35496244]
[0.44557574 0.41113526]
[0.62289464 0.61937058]
[0.46787083 0.37809363]
[0.13198605 0.03915319]
[0.18463194 0.10718343]
[0.11632597 0.09563267]
[0.17304552 0.09629312]
[0.12593585 0.06682214]
[0.06888115 0.06158322]
[0.26593369 0.1271911]
[0.09731811 0.08635426]
[0.08940494 0.09365067]
[0.15456077 0.06799281]
[0.26765949 0.16763985]
[0.15066862 0.12679929]
[0.1550031 0.12086043]
[0.14735046 0.1005125]
[0.14582458 0.07570535]

Ln: 79 Col: 15

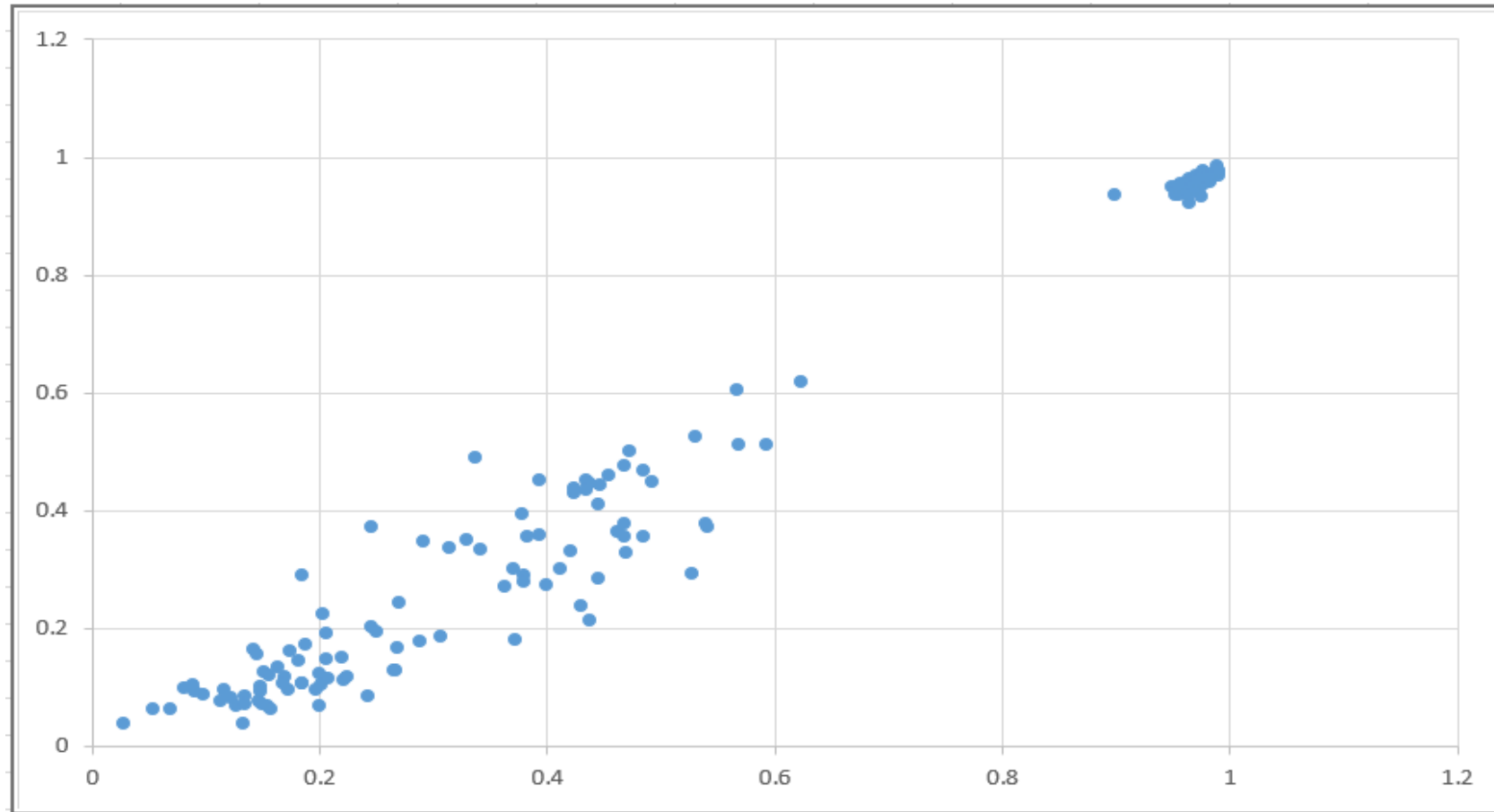
[0.15066862 0.12679929]
[0.1550031 0.12086043]
[0.14735046 0.1005125]
[0.14582458 0.07570535]
[0.20091653 0.10401532]
[0.19906002 0.12373728]
[0.15704438 0.06323111]
[0.02802464 0.03736967]
[0.14198542 0.16329035]
[0.14840955 0.09339112]
[0.22137007 0.11054417]
[0.05309421 0.06121218]
[0.2061235 0.19094405]
[0.19657955 0.09660652]
[0.16942465 0.11771309]
[0.2452125 0.201554]
[0.28840131 0.17826241]
[0.12238899 0.08103782]
[0.17424387 0.16053972]
[0.08902252 0.10441676]
[0.20789555 0.11563277]
[0.1136629 0.07525021]
[0.24976575 0.19441178]
[0.16734788 0.10725468]
[0.08020949 0.09821722]
[0.2005074 0.06762961]
[0.22479171 0.11821833]
[0.30664176 0.18536371]
[0.18145847 0.14502749]
[0.13433427 0.08343807]
[0.18778604 0.17117205]
[0.18463194 0.10718343]
[0.13358873 0.0702585]
[0.14890006 0.07195672]
[0.16294518 0.13482589]
[0.14480689 0.1567623]
[0.20584705 0.14670235]
[0.24194402 0.08499151]
[0.2673226 0.12904647]

>>>

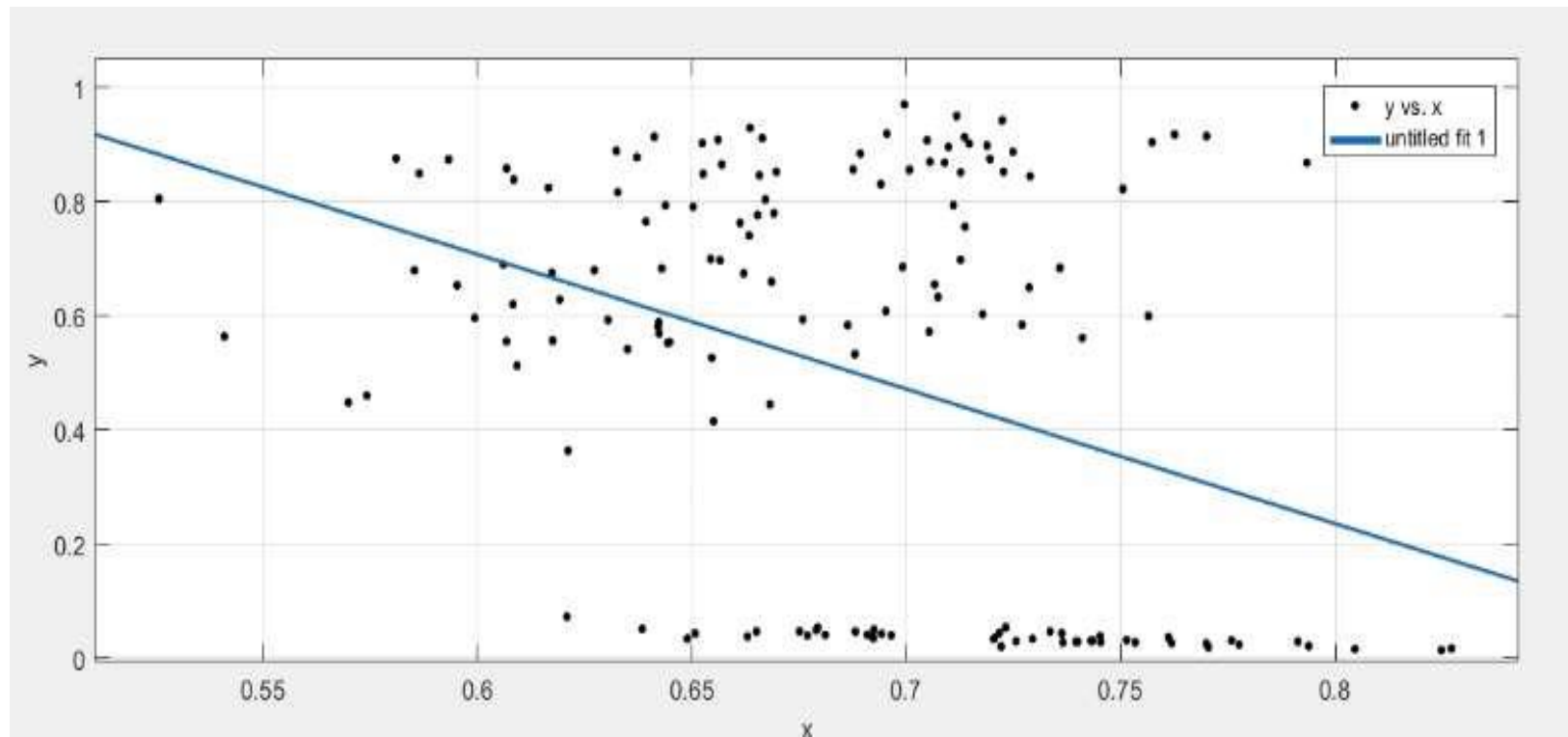
Ln: 119 Col: 16

2次元への次元縮約のデータに位置

excelで作成した分布図：



Matlabで作成した分布図：



6.2 Denoising AutoEncoder (DAE)

- AEでは中間層のノード数が入力層のノード数(入力ベクトルの次元数)よりも大きいと、encoderが恒等関数になってしまうという問題があり、中間層のノード数は入力ベクトルの次元数よりも小さくしないといけません。
- AEを次元縮約として使うのならこれで問題はありませんが、次元縮約とは逆に入力ベクトルをより高次元の空間に射影したい場合にAEが使えません。

Denoising AutoEncoder (DAE)とは、高次元に射影することも可能にしたAEの改良版です。また、高次元への射影ではなく、通常の次元縮約にも DAEを利用できます。

ChainerでDAEを実現するもの簡単です。適当に訓練データにノイズを入れたのを入力、ノイズを入れないものを教師データとして、通常のNNで実現できます。