2020･08･21　ゼミ資料

# Pythonではじめる教師なし学習

# 11章 深層信念ネットワークを用いた特徴量検出

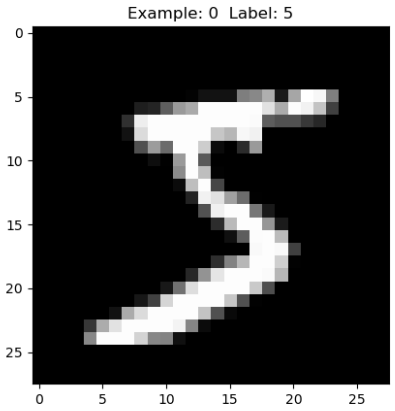
1116 17 9036
山口真哉

# やること

- 深層信念ネットワーク（DBN)
- DBNを用いて画像を大量に生成し, 分類問題を解く

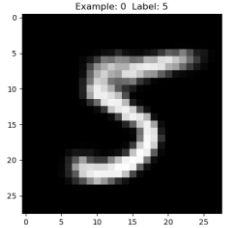
# モチベーション

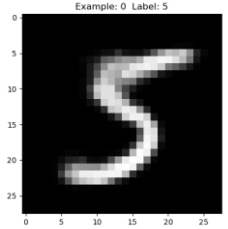
- ラベル付きのデータはあるが, 数が少ない場合がある.
- そのような場合に似たようなデータを生成して分類する時にかさ増ししたい.

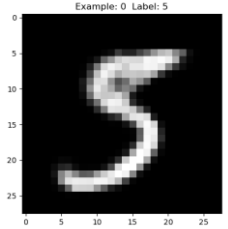



元のデータ

生成モデルで
新しく生成

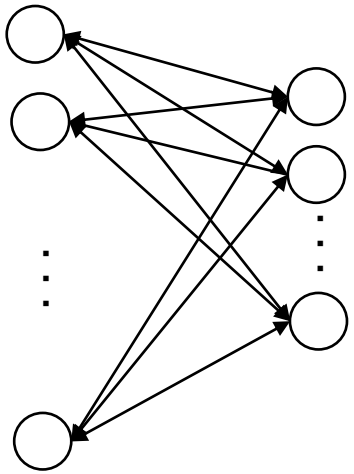


# RBMの復習

- 可視層(入力層)と隠れ層からなる浅い2層のニューラルネットワーク
- 層間の通信が一方向だけでなく、双方向に何度も通信する.
- これにより元の入力と類似するような生成モデルを構築する.

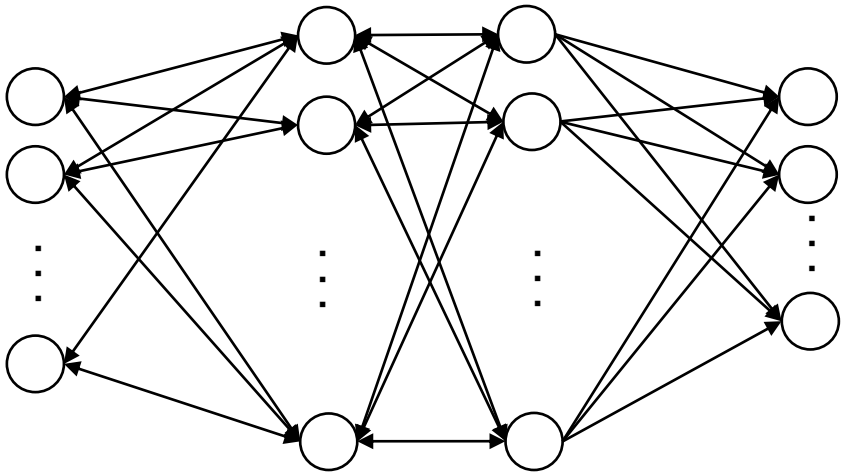


RBMの様子

extract from 間宮陸

# DBN

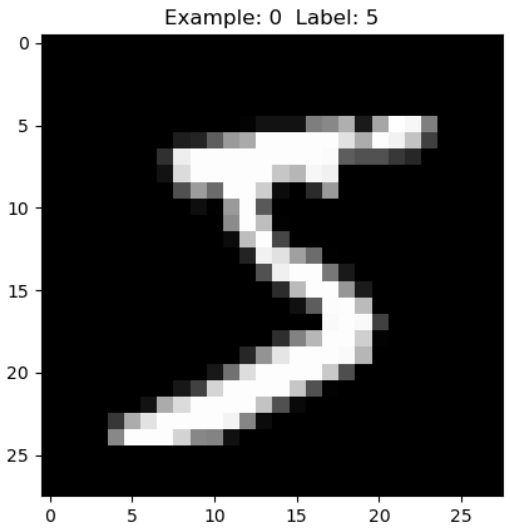
- DBN ・・・ RBMを重ねたもの.
- 今回は1-3層で両方向的な学習をする
- 4層目は構成したRBMの重みとバイアスを,
  元の画像と生成した画像の再構成誤差が最小になるように更新する.
- 4層目は全結合層的な役割を果たす.



DBNの様子

# データの準備

- mnist を使う.
- train size : 50000, vaild size : 10000, test size : 10000 に分割する.
- 状況を再現するためにtrainは5000番目までのみ使用する



# Case 1 (LightGBM only)

- LightGBMを使用し, 10種類の分類問題を解く
- ハイパーパラメータは右の通り
- trainデータの数は5000である.

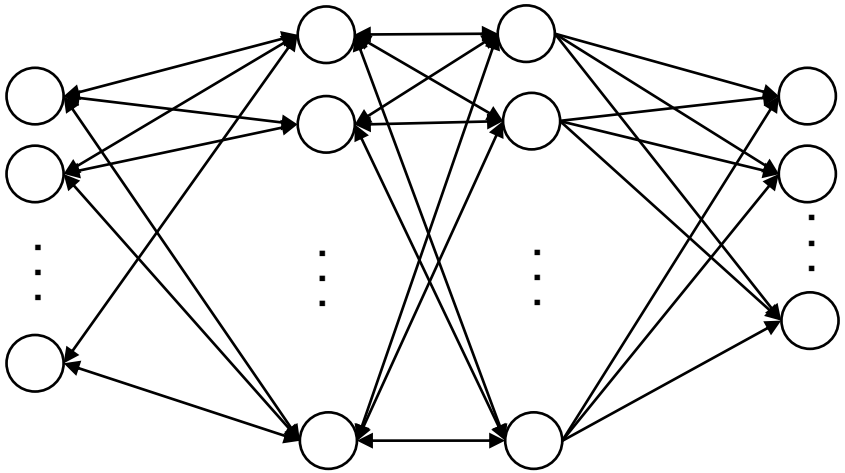
~結果~

- 対数損失はtrain 0.0018で valid 0.19であった.
- 正解率は94.37%になった.
- (ちなみにtrainデータを削らないと97.97%になった.)

```
params_lightGB = {
    'task': 'train',
    'num_class': 10,
    'boosting': 'gbdt',
    'objective': 'multiclass',
    'metric': 'multi_logloss',
    'metric_freq': 50,
    'is_training_metric': False,
    'max_depth': 4,
    'num_leaves': 31,
    'learning_rate': 0.1,
    'feature_fraction': 1.0,
    'bagging_fraction': 1.0,
    'bagging_freq': 0,
    'bagging_seed': 2018,
    'verbose': -1,
}
```

# Case 2 (DBN)

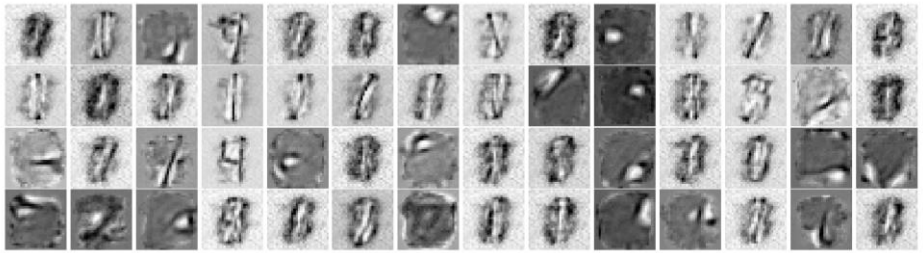
- この学習を20回して, 学習して生成した画像たちをtrain2データとする.
- LightGBMを使って, train2データを学習する.
- DBNの各層の次元は前から順番に784 -> 500 -> 500 -> 784である.
- エポック数は50でバッチサイズは200である.



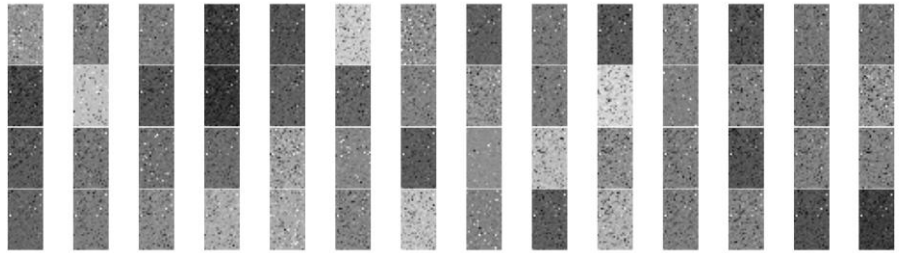
# Case 2 (DBN)

- 右の画像は各層が学習した特徴量
- 層が深くなるにつれ抽象的な特徴を学習している.

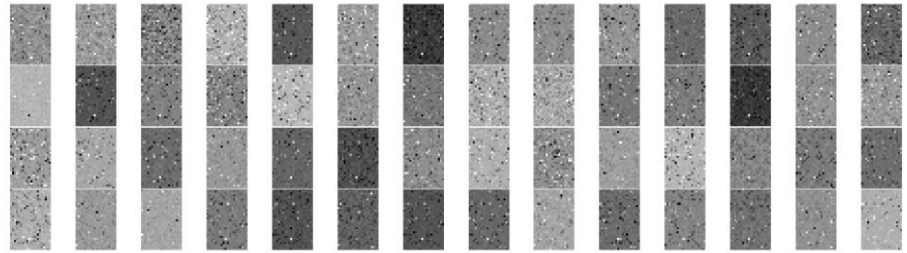
1層目



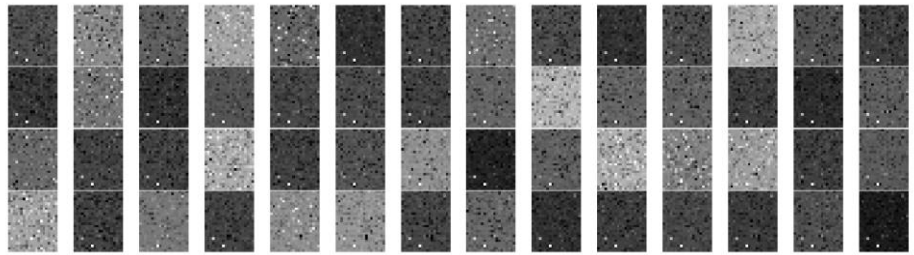
2層目



3層目

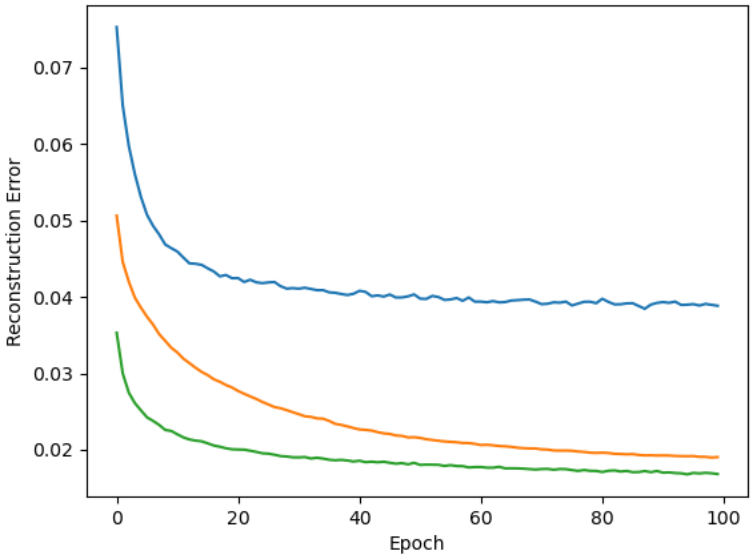


4層目

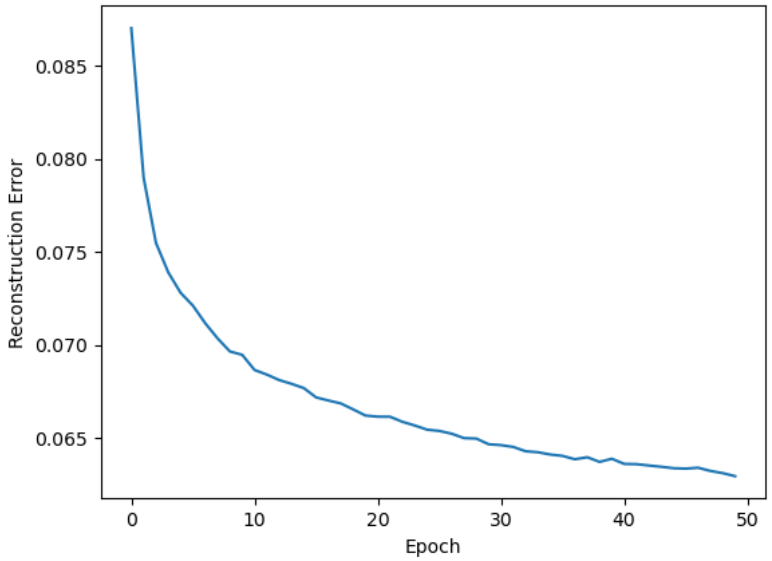


# Case 2 (DBN)

各層の再構成誤差の様子



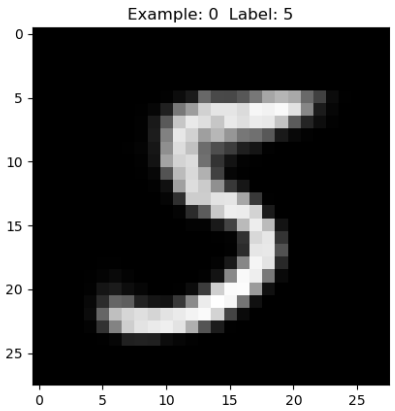

1層目 青
2層目 オレンジ
3層目 緑



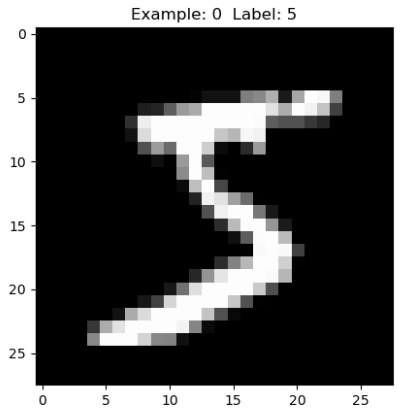

最終的な再構成誤差

# Case 2 (生成された画像)

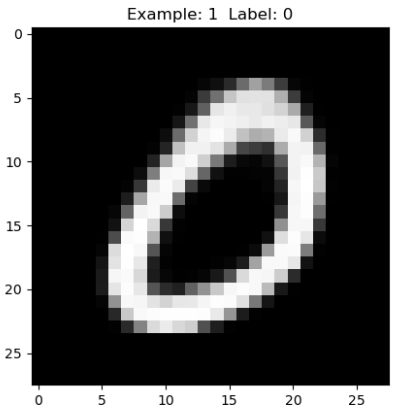
- うまく画像を生成できていることがわかる.
- 一番下の画像は同じ画像から生成した画像だが異なる画像になっている.



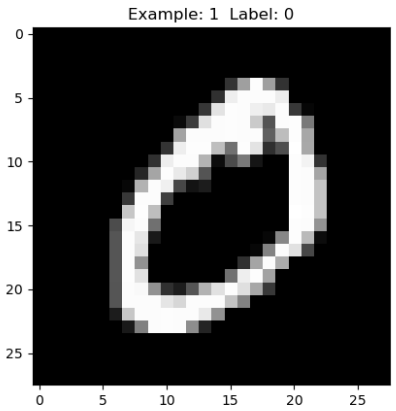

train後



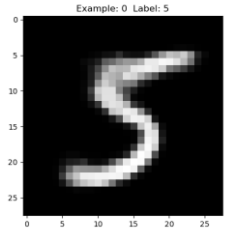

mnist

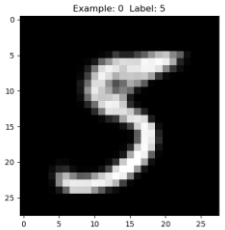


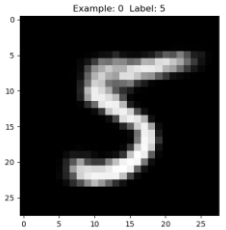

train後

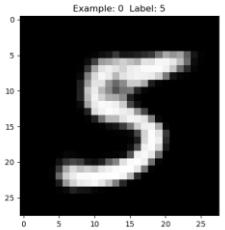


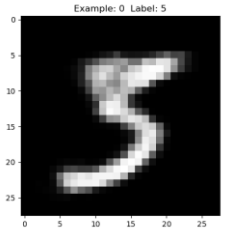

mnist

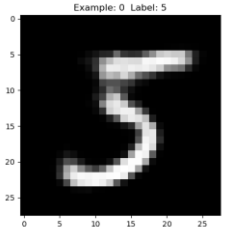


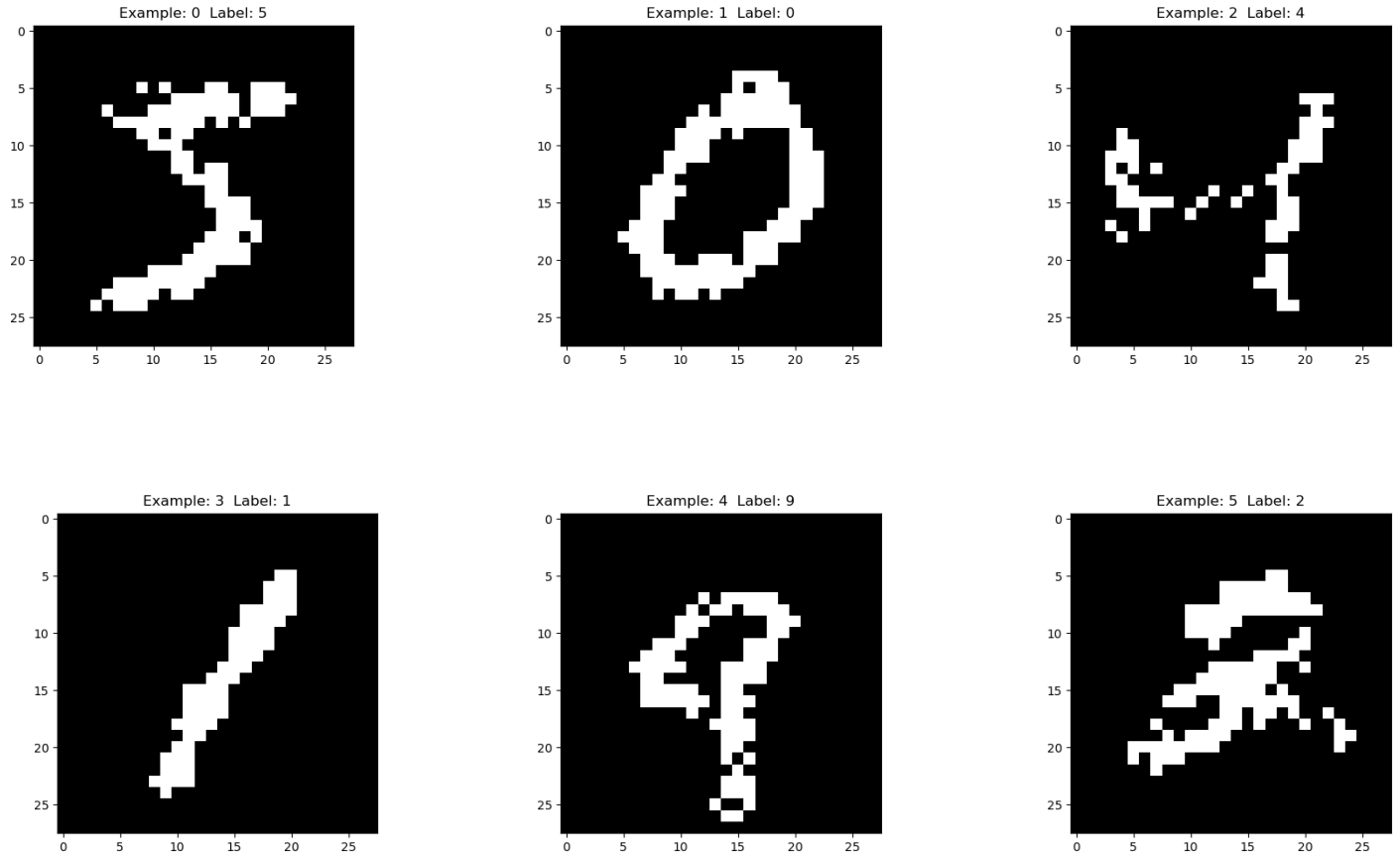



# Case 2 (第1層の様子)




第1層目ですでに読める.

# Case 2 (結果)

- 対数損失は, train 0.0043, valid 0.1625になった.
- 正解率は95.51%になった.
- LightGBMのみの94.37%より良い結果となった.

# まとめ

- RBMを積み重ねたDBNを勉強した.
- 画像を生成することで教師ありの手助けができる.
- mnistの例だとデータをかさ増しすることでうまくいった.