コンピュータビジョン

担当: 井尻 敬

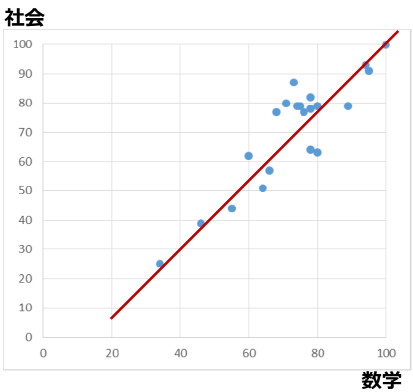
主成分分析

- データ群から最もばらつきの大きな軸を見つける
- データの次元圧縮に利用できる
- パターン認識，画像処理，そのほか様々な分野で使われる

主成分分析

ある21人のテスト点数とその散布図 (横:数学 縦:社会)が下図の通り

数学	社会
80	63
95	91
100	100
64	51
34	25
89	79
94	93
78	78
55	44
60	62
46	39
80	79
73	87
68	77
75	79
76	77
66	57
78	82
74	79
71	80
78	64



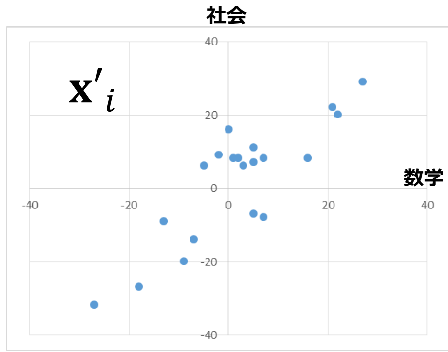
最もばらつきの大きな方向を考えてみる

※井尻が適当に作った 嘘 データ です

主成分分析

- 入力データ：$\mathbf{x}_i \in \boldsymbol{R}^2, i = 1,2, \ldots, N$
- 平均が原点となるよう平行移動する

$$\mathbf{x}'_i = \mathbf{x}_i - \frac{1}{N}\sum_i \mathbf{x}_i$$



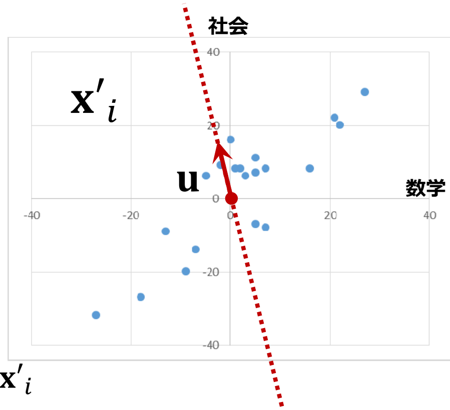
主成分分析

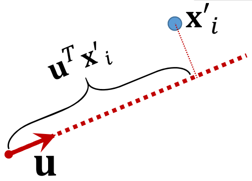
- ある単位ベクトル $\mathbf{u}$ を考える
- $\mathbf{u}$にデータ点を射影した距離の2乗平均は

$$\frac{1}{N}\sum_i (\mathbf{u}^T\mathbf{x}'_i)^2$$

これを最大化する $\mathbf{u}$ を探す！　※計算法後述

→最もデータがばらつく方向が分かる





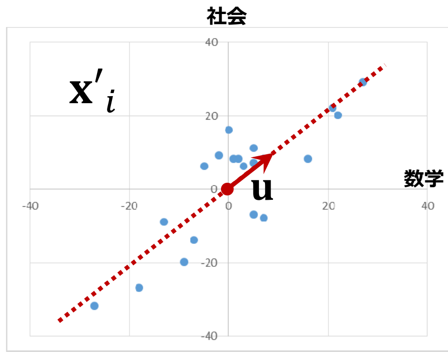
主成分分析

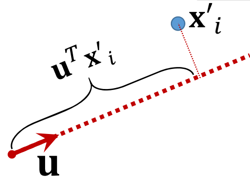
- ある単位ベクトル $\mathbf{u}$ を考える
- $\mathbf{u}$にデータ点を射影した距離の2乗平均は

$$\frac{1}{N}\sum_i (\mathbf{u}^T\mathbf{x}'_i)^2$$

これを最大化する $\mathbf{u}$ を探す！　※計算法後述

→最もデータがばらつく方向が分かる





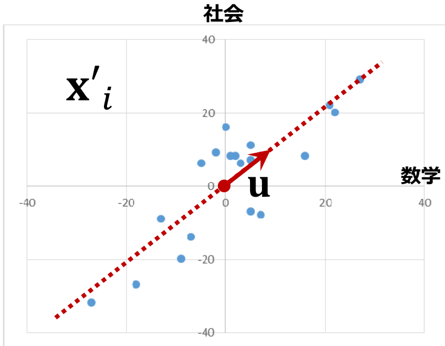
主成分分析

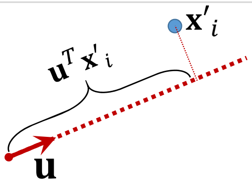
余談：『距離の平均』はゼロになる

- $\mathbf{u}$にデータ点を射影した距離の平均は以下の通り

$$\frac{1}{N}\sum_i \mathbf{u}^T\mathbf{x}'_i$$

※この値は0 → 証明せよ





主成分分析

例）右表のデータに対して，

$$\frac{1}{N}\sum_i (\mathbf{u}^T\mathbf{x}'_i)^2$$

を最大化する$\mathbf{u}$を計算すると

$$\mathbf{u} = (0.63, 0.78)$$

が得られた．この方向$\mathbf{u}$を**第一主成分**と呼ぶ

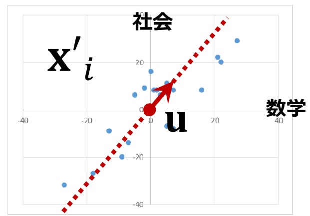
各データを$\mathbf{u}$ に射影する

(数学, 社会) の点が (80, 70)なら,

(数学, 社会) の平均値は(73, 71)なので

射影値 = (80-73)*0.63 +(70-71)*0.78

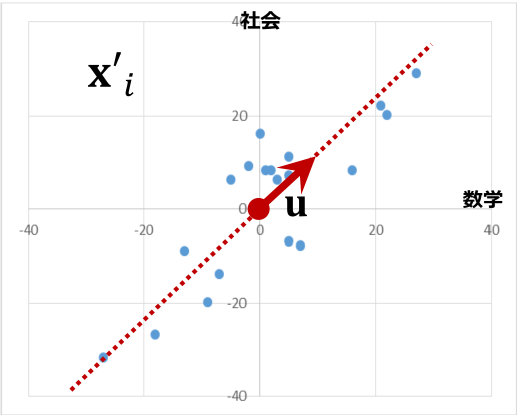
= 3.63



数学	社会	第1主成分得点
80	63	−1.7
95	91	29.6
100	100	39.8
64	51	−21.1
34	25	−60.3
89	79	16.5
94	93	30.5
78	78	8.8

この射影値を**第一主成分得点**と呼び，この例では『学力』に対応すると考えられる

主成分分析 – 小休止

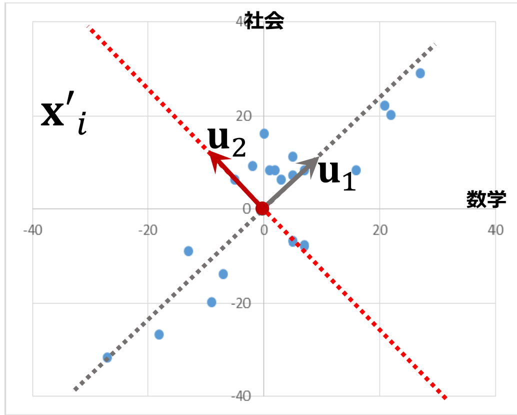


最もばらつきの大きい方向（**第一主成分**）を発見しその方向にデータを射影して**第一主成分得点**を取得した…

残ってる主な疑問

- **→ uと直交する方向にもデータはばらついているけど無視していいの？**
- **→ 射影によってデータ量が失われたのでは？**
- **→ ばらつき方向uはどうやって計算するの？**

主成分分析 - 第n主成分



データ点のばらつきが最も大きい方向を**第1主成分**，その方向への射影を**第1主成分得点**と呼ぶ

第1主成分と直交し，かつ，データ点のばらつきが最も大きい方向を**第2主成分**とよび，その方向への射影を**第2主成分得点**と呼ぶ

同様に**第n主成分・第n主成分得点**が定義される

※主成分は，主成分ベクトルや負荷量ベクトルなどとも呼ばれる

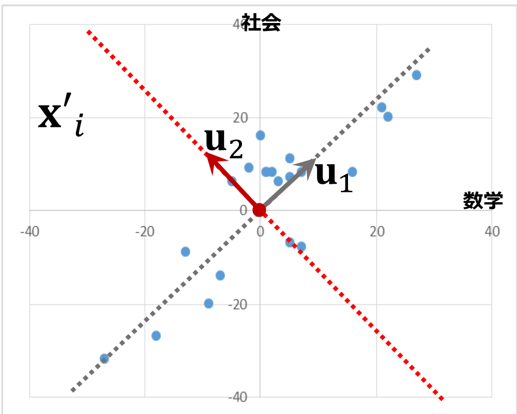
例）左図では・・・

第1主成分得点($\mathbf{u}_1$への射影)は『学力』を表現

第2主成分得点($\mathbf{u}_2$への射影)は『文系指向』を表現

しているように考えられるかも知れない（意味づけは解析者が実施）

主成分分析 - 第n主成分

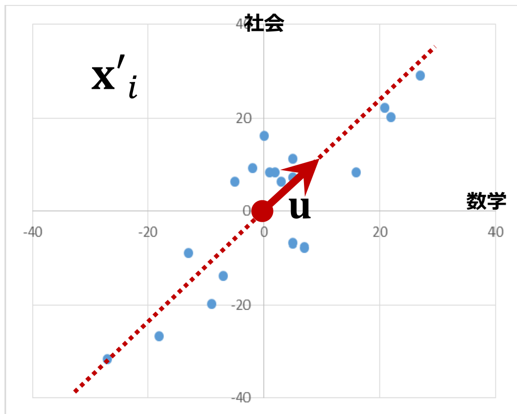


$$\mathbf{u}_1 = (0.63, 0.78)$$
$$\mathbf{u}_2 = (0.78, -0.63)$$

数学	社会
80	63
95	91
100	100
64	51
34	25
89	79
94	93
78	78
55	44
60	62
46	39
80	79
73	87
68	77
75	79
76	77
66	57
78	82
74	79
71	80
78	64

第1主成分得点	第2主成分得点
−1.7	−10.3
29.6	−4.4
39.8	−2.6
−21.1	−5.4
−60.3	1.6
16.5	−7.3
30.5	−2.3
8.8	0.7
−32.2	−2.8
−15.1	4.7
−41.8	1.1
10.8	−0.2
12.6	10.3
1.7	7.9
7.7	3.7
6.7	1.6
−15.2	−3.2
11.9	3.2
7.0	4.4
5.9	7.4
−2.2	−8.1

主成分分析 – 小休止



最もばらつきの大きい方向（**主成分**）を発見しその方向にデータを射影して**主成分得点**を取得した…

残ってる主な疑問

- **→ uと直交する方向にもデータはばらついているけど無視していいの？ → 場合による（n次元データには第n主成分まで存在する）**
- **→ 射影によってデータ量が失われたのでは？**
- **→ ばらつき方向uはどうやって計算するの？**

主成分分析 –
第1主成分の計算

入力点群：$\hat{\mathbf{x}}_i \in R^d, i = 1,2,\dots,N$

平均値：$\mathbf{m} = \frac{1}{N}\sum_i \hat{\mathbf{x}}_i$

平行移動：$\mathbf{x}_i = \hat{\mathbf{x}}_i - \mathbf{m}$

以下の最大値問題を求めたい

$$\underset{||\mathbf{u}||=1}{\mathrm{argmax}} \sum_i (\mathbf{u}^T\mathbf{x}_i)^2$$

主成分分析 –
第1主成分の計算

入力点群：$\hat{\mathbf{x}}_i \in R^d, i = 1,2,\dots,N$

平均値：$\mathbf{m} = \frac{1}{N}\sum_i \hat{\mathbf{x}}_i$

平行移動：$\mathbf{x}_i = \hat{\mathbf{x}}_i - \mathbf{m}$

以下の最大値問題を求めたい

$$\underset{||\mathbf{u}||=1}{\mathrm{argmax}} \sum_i (\mathbf{u}^T\mathbf{x}_i)^2$$

準備：

行列 $\mathbf{A} = \sum_i \mathbf{x}_i {\mathbf{x}_i}^T \in R^{d\times d}$ を考えると，この行列は対称行列であり，半正定置性を持つ．（→ 証明せよ）

$\mathbf{A}$ の固有値を $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$ とし，長さ1で互いに直交する固有ベクトルを $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d$ とする．

すると…

$$V^T\mathbf{A}\mathbf{V} = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$$

$$\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d)$$

と対角化できる．

主成分分析 –
第1主成分の計算

入力点群：$\hat{\mathbf{x}}_i \in R^d, i = 1,2,\dots,N$

平均値：$\mathbf{m} = \frac{1}{N}\sum_i \hat{\mathbf{x}}_i$

平行移動：$\mathbf{x}_i = \hat{\mathbf{x}}_i - \mathbf{m}$

以下の最大値問題を求めたい

$$\underset{||\mathbf{u}||=1}{\mathrm{argmax}} \sum_i (\mathbf{u}^T\mathbf{x}_i)^2$$

コスト関数を以下の通り変形する，

$$\begin{aligned}\textstyle\sum_i (\mathbf{u}^T\mathbf{x}_i)^2 &= \textstyle\sum_i \mathbf{u}^T\mathbf{x}_i{\mathbf{x}_i}^T\mathbf{u} \\ &= \mathbf{u}^T(\textstyle\sum_i \mathbf{x}_i{\mathbf{x}_i}^T)\mathbf{u}\end{aligned}$$

$\mathbf{A} = \sum_i \mathbf{x}_i{\mathbf{x}_i}^T$ と置いてさらに変形，

$$\begin{aligned}\mathbf{u}^T A\mathbf{u} &= \left(\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}\right)^T A\left(\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}\right) \\ &= (\mathbf{V}^T\mathbf{u})^T\, V^T\mathbf{A}\mathbf{V}\,(\mathbf{V}^T\mathbf{u}) \\ &= (\mathbf{V}^T\mathbf{u})^T\mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)(\mathbf{V}^T\mathbf{u}) \\ &\leq (\mathbf{V}^T\mathbf{u})^T\mathrm{diag}(\lambda_1, \lambda_1, \dots, \lambda_1)(\mathbf{V}^T\mathbf{u}) \\ &= \lambda_1(\mathbf{V}^T\mathbf{u})^T(\mathbf{V}^T\mathbf{u}) \\ &= \lambda_1\end{aligned}$$

等号成立は $\mathbf{V}^T\mathbf{u}=(1,0,0,\cdots,0)$ のとき，つまり $\mathbf{u}=\mathbf{v}_1$ のとき最大値となる．最大値は λ_1．

主成分分析 –
第2主成分の計算

入力点群：$\hat{\mathbf{x}}_i \in R^d, i = 1,2,\dots,N$

平均値：$\mathbf{m} = \frac{1}{N}\sum_i \hat{\mathbf{x}}_i$

平行移動：$\mathbf{x}_i = \hat{\mathbf{x}}_i - \mathbf{m}$

以下の最大値問題を求めたい

$$\underset{||\mathbf{u}||=1}{\mathrm{argmax}} \sum_i (\mathbf{u}^T\mathbf{x}_i)^2$$

ただし，$\mathbf{u}^T\mathbf{v}_1=0$を満たすものとする

先と同様にコスト関数を変形する，

$$\begin{aligned}\textstyle\sum_i (\mathbf{u}^T\mathbf{x}_i)^2 &= \mathbf{u}^T(\textstyle\sum_i \mathbf{x}_i{\mathbf{x}_i}^T)\mathbf{u} \\ &= \left(\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}\right)^T A\left(\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{u}\right) \\ &= (\mathbf{V}^T\mathbf{u})^T\mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)(\mathbf{V}^T\mathbf{u})\end{aligned}$$

ここで条件 $\mathbf{u}^T\mathbf{v}_1=0$ より $\mathbf{V}^T\mathbf{u} = (0, u_2, u_3, \dots)^T$ の形をしているので，

$$\begin{aligned}&= (\mathbf{V}^T\mathbf{u})^T\mathrm{diag}(0, \lambda_2, \dots, \lambda_d)(\mathbf{V}^T\mathbf{u}) \\ &\leq (\mathbf{V}^T\mathbf{u})^T\mathrm{diag}(0, \lambda_2, \dots, \lambda_2)(\mathbf{V}^T\mathbf{u}) \\ &= \lambda_2\end{aligned}$$

等号成立は $\mathbf{V}^T\mathbf{u}=(0,1,0,\cdots,0)$ のとき，つまり $\mathbf{u}=\mathbf{v}_2$ のとき最大値となる．最大値は λ_2．

主成分分析 –
第n主成分の計算

入力点群：$\hat{\mathbf{x}}_i \in R^d, i = 1,2,\ldots,N$

平均値：$\mathbf{m} = \frac{1}{N}\sum_i \hat{\mathbf{x}}_i$

平行移動：$\mathbf{x}_i = \hat{\mathbf{x}}_i - \mathbf{m}$

以下の最大値問題を求めたい

$$\underset{\mathbf{u}=1}{\operatorname{argmax}} \sum_i (\mathbf{u}^T \mathbf{x}_i)^2$$

ただし$\mathbf{u}^T\mathbf{v}_1=\mathbf{u}^T\mathbf{v}_2 = \cdots = \mathbf{u}^T\mathbf{v}_{n-1}=0$を満たす

先と同様に計算すると…

$\mathbf{u}=\mathbf{v}_\mathrm{n}$のときに最大値を取ることが分かる．

つまり…
第n主成分は，行列$\mathbf{A} = \sum_i \mathbf{x}_i \mathbf{x}_i{}^T$の第n固有ベクトルと等しくなる．

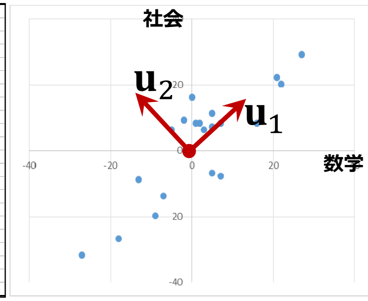
行列 Aに1/Nをかけると分散共分散行列が得られる

$$\frac{1}{N}\mathbf{A} = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i{}^T = \frac{1}{N}\sum_i (\hat{\mathbf{x}}_i - \mathbf{m})(\hat{\mathbf{x}}_i - \mathbf{m})^T$$

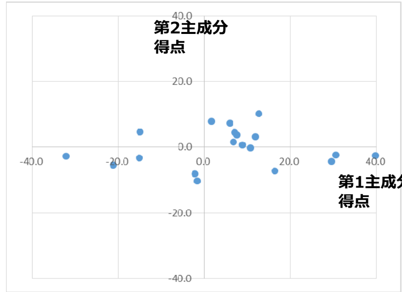
※対角成分に各軸方向の分散が並び，非対角成分に共分散成分が並ぶ

主成分分析 – 分散共分散行列を理解する

数学	社会
80	63
95	91
100	100
64	51
34	25
89	79
94	93
78	78
55	44
60	62
46	39
80	79
73	87
68	77
75	79
76	77
66	57
78	82
74	79
71	80
78	64



$$\mathbf{x}_i - \mathbf{m}$$



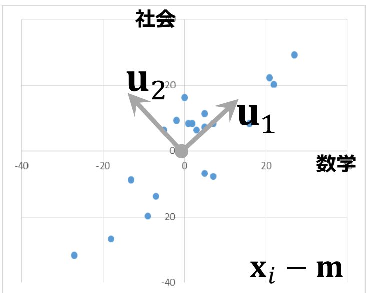
第1主成分	第2主成分
−1.7	−10.3
29.6	−4.4
39.8	−2.6
−21.1	−5.4
−60.3	1.6
16.5	−7.3
30.5	−2.3
8.8	0.7
−32.2	−2.8
−15.1	4.7
−41.8	1.1
10.8	−0.2
12.6	10.3
1.7	7.9
7.7	3.7
6.7	1.6
−15.2	−3.2
11.9	3.2
7.0	4.4
5.9	7.4
−2.2	−8.1

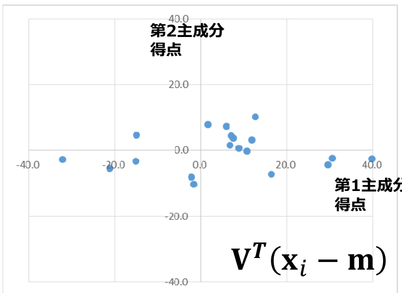
$$\mathbf{V}^T(\mathbf{x}_i - \mathbf{m}), \quad \mathbf{V} = (\mathbf{u}_1, \mathbf{u}_2)$$

得られた第1/2主成分は，ばらつきの大きな軸へ射影したものなので…
⇒ データ点群を平均を中心に回転したと考えてよい

主成分分析 – 分散共分散行列を理解する

数学	社会
80	63
95	91
100	100
64	51
34	25
89	79
94	93
78	78
55	44
60	62
46	39
80	79
73	87
68	77
75	79
76	77
66	57
78	82
74	79
71	80
78	64





第1主成分	第2主成分
−1.7	−10.3
29.6	−4.4
39.8	−2.6
−21.1	−5.4
−60.3	1.6
16.5	−7.3
30.5	−2.3
8.8	0.7
−32.2	−2.8
−15.1	4.7
−41.8	1.1
10.8	−0.2
12.6	10.3
1.7	7.9
7.7	3.7
6.7	1.6
−15.2	−3.2
11.9	3.2
7.0	4.4
5.9	7.4
−2.2	−8.1

元データの分散共分散行列

$$\sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$
$$= \mathbf{V}\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{d})\boldsymbol{V}^T$$
$$= \begin{pmatrix} 0.63 & 0.78 \\ 0.78 & -0.63 \end{pmatrix} \begin{pmatrix} 552.8 & 0 \\ 0 & 28.2 \end{pmatrix} \begin{pmatrix} 0.63 & 0.78 \\ 0.78 & -0.63 \end{pmatrix}^T$$

回転したデータの分散共分散行列

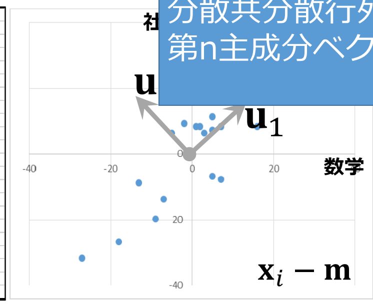
$$\sum_i \mathbf{V}^T(\mathbf{x}_i - \mathbf{m})(\mathbf{V}^T(\mathbf{x}_i - \mathbf{m}))^T$$
$$= \mathbf{V}^T\mathbf{V}\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{d})\boldsymbol{V}^T\,\mathbf{V}$$
$$= \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{d})$$
$$= \begin{pmatrix} 552.8 & 0 \\ 0 & 28.2 \end{pmatrix}$$

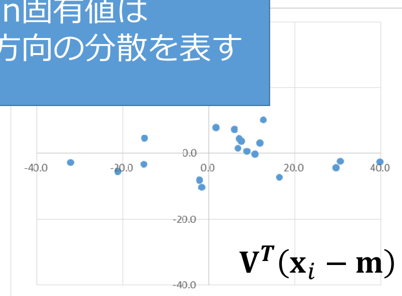
※先のデータの数値を入れて
計算したものを提示しています

主成分分析 – 分散共分散行列を理解する

分散共分散行列の第n固有値は
第n主成分ベクトル方向の分散を表す

数学	社会
80	63
95	91
100	100
64	51
34	25
89	79
94	93
78	78
55	44
60	62
46	39
80	79
73	87
68	77
75	79
76	77
66	57
78	82
74	79
71	80
78	64





第1主成分	第2主成分
−1.7	−10.3
29.6	−4.4
39.8	−2.6
−21.1	−5.4
−60.3	1.6
16.5	−7.3
30.5	−2.3
8.8	0.7
−32.2	−2.8
−15.1	4.7
−41.8	1.1
10.8	−0.2
12.6	10.3
1.7	7.9
7.7	3.7
6.7	1.6
−15.2	−3.2
11.9	3.2
7.0	4.4
5.9	7.4
−2.2	−8.1

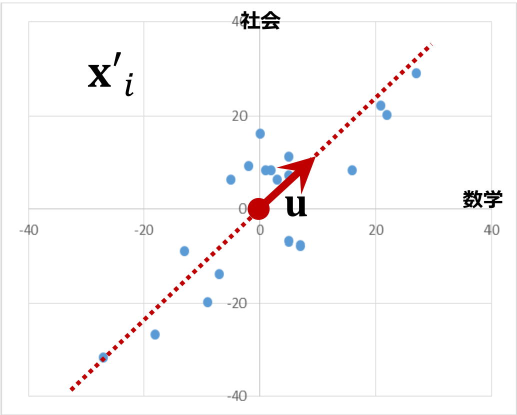
元データの分散共分散行列

$$\sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$
$$= \mathbf{V}\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{d})\boldsymbol{V}^T$$
$$= \begin{pmatrix} 0.63 & 0.78 \\ 0.78 & -0.63 \end{pmatrix} \begin{pmatrix} 552.8 & 0 \\ 0 & 28.2 \end{pmatrix} \begin{pmatrix} 0.63 & 0.78 \\ 0.78 & -0.63 \end{pmatrix}^T$$

回転したデータの分散共分散行列

$$\sum_i \mathbf{V}^T(\mathbf{x}_i - \mathbf{m})(\mathbf{V}^T(\mathbf{x}_i - \mathbf{m}))^T$$
$$= \mathbf{V}^T\mathbf{V}\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{d})\boldsymbol{V}^T\,\mathbf{V}$$
$$= \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{d})$$
$$= \begin{pmatrix} 552.8 & 0 \\ 0 & 28.2 \end{pmatrix}$$

主成分分析 – 小休止



最もばらつきの大きい方向（**主成分**）
を発見しその方向にデータを射影して
主成分得点を取得した…

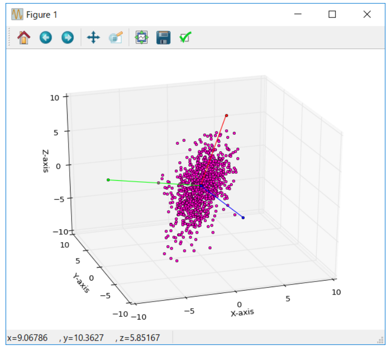
残ってる主な疑問

- **uと直交する方向にもデータはばらついているけど無視していいの？ →場合による（n次元データには第n主成分まで存在する）**
- **射影によってデータ量が失われたのでは？**
- **ばらつき方向uはどうやって計算するの？ →分散共分散行列の固有ベクトルを求めればok**

PCA_PLOT_3D.py

主成分分析 - 次元圧縮への応用

例）
3次元データ点群が下図の通り分布している
分布にはあまり偏りがないため，すべての主成分得点の数値が比較的大きな値に



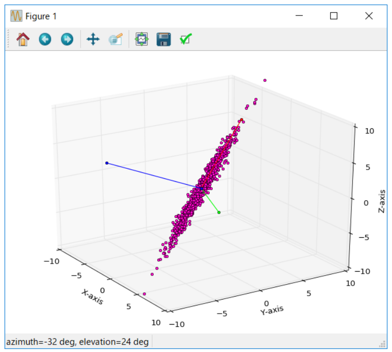
points		
x	y	z
0.86	-2.00	4.57
0.86	0.27	2.78
-1.19	0.73	-4.73
3.22	1.17	4.63
0.33	-1.07	-3.13
0.03	0.49	3.68
2.36	0.51	-1.73
-2.16	-0.07	-0.87
0.42	1.27	0.90
0.15	-1.02	-1.12
0.95	-0.20	0.01
2.26	-0.23	0.81
0.86	0.23	1.87
-2.28	-0.47	-3.74
0.67	-0.14	0.08
0.42	0.58	-0.15

pca		
1	2	3
-4.74	-0.42	-1.81
-2.94	0.12	0.40
4.85	0.03	0.67
-5.31	1.98	1.28
2.88	1.02	-1.12
-3.60	-0.90	0.68
1.05	2.70	0.43
1.33	-1.91	0.04
-0.99	0.20	1.35
0.98	0.35	-1.00
-0.30	0.88	-0.17
-1.40	1.94	-0.21
-2.06	0.35	0.33
4.13	-1.32	-0.45
-0.29	0.59	-0.10
0.01	0.44	0.63

PCA_PLOT_3D.py

主成分分析 - 次元圧縮への応用

例）3次元データ点群が下図の平面上に通り分布している
データ点は平面に乗っているため，第1主成分の寄与が大きく
第3主成分は寄与しない偏った分布



points		
x	y	z
1.30	-2.07	-2.85
0.61	0.36	1.33
-0.65	-0.33	-1.31
-1.61	-0.71	-3.04
-0.32	-2.74	-5.81
1.04	2.45	5.94
-0.49	-1.58	-3.66
-1.85	-0.36	-2.57
-0.74	-0.73	-2.20
0.02	2.57	5.16
0.27	1.55	3.36
-0.57	-2.86	-6.29
-0.59	-0.42	-1.42
-1.15	0.27	-0.61
-1.62	2.08	2.53
-0.01	1.02	2.02
0.73	-2.72	-4.70

pca		
1	2	3
-3.32	1.58	0.00
1.45	0.52	0.00
-1.30	-0.69	0.00
-3.08	-1.63	0.00
-6.37	-0.01	0.00
6.52	0.67	0.00
-3.95	-0.34	0.00
-2.53	-1.92	0.00
-2.29	-0.73	0.00
5.81	-0.40	0.00
3.77	0.00	0.00
-6.87	-0.24	0.00
-1.44	-0.61	0.00
-0.44	-1.29	0.00
3.13	-2.04	0.00
2.31	-0.22	0.00
-5.30	1.09	0.00

主成分分析 - 次元圧縮への応用

n次元データの次元を圧縮することを考える

- ***k***次元まで圧縮する
- 情報量の欠落を抑えられるいい感じの『***k***』を選択したい
（平面に縮退しているような軸は削除しつつも，分散の大きな軸は利用したい）

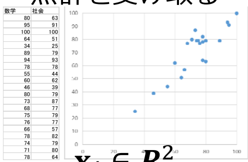
→ **寄与率**を利用する

$$\text{寄与率} = \frac{k\text{番目の方向までの分散}}{\text{全方向の分散}} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$$

※第k主成分方向の分散はλ_kとなる
例)寄与率が 0.8 以上になる最小のkを選択する

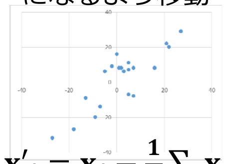
主成分分析 – まとめ

1.入力データ点群を受け取る



$\mathbf{x}_i \in \boldsymbol{R}^2$

2. 平均値が原点になるよう移動

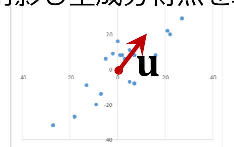


$\mathbf{x}'_i = \mathbf{x}_i - \frac{1}{N}\sum_i \mathbf{x}_i$

3. 分散共分散行列を計算し固有解析

$$\mathbf{A} = \sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

4. 各点を固有ベクトルに射影し主成分得点を取得



- 分散共分散行列の固有ベクトルが**主成分**ベクトルに対応
- 主成分ベクトルへ射影すると**主成分得点**が得られる
- 下例では**学力・文系指向**を説明（分析結果の意味付けはまた別の話）

数学	社会
80	63
95	91
100	100
64	51
34	25
89	79
94	93
78	78
55	44
60	62
46	39
80	79
73	87
68	77
75	79
76	77
66	57
78	82
74	79
71	80
78	64

第1主成分	第2主成分
−1.7	−10.3
29.6	−4.4
39.8	−2.6
−21.1	−5.4
−60.3	1.6
16.5	−7.3
30.5	−2.3
8.8	0.7
−32.2	−2.8
−15.1	4.7
−41.8	1.1
10.8	−0.2
12.6	10.3
1.7	7.9
7.7	3.7
6.7	1.6
−15.2	−3.2
11.9	3.2
7.0	4.4
5.9	7.4
−2.2	−8.1

主成分分析の画像処理応用

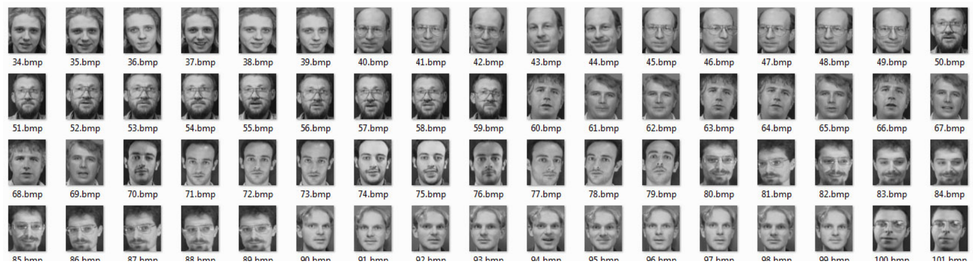
- 特徴ベクトルの次元圧縮
  - 特徴ベクトル群から寄与率の高い主成分のみ抽出し，低次元化してから計算（識別など）を行なう.
  - 情報量をあまり落とさずに，計算量・メモリ量などの削減が可能
- 画像の圧縮・編集・生成
  - 同じクラスタに属する画像群（例，顔画像）を仮定する
  - 画像群を高次元データと考え主成分を計算

  →寄与率の高い軸と主成分値のみを記憶する事で圧縮

  →主成分値を修正して画像を編集

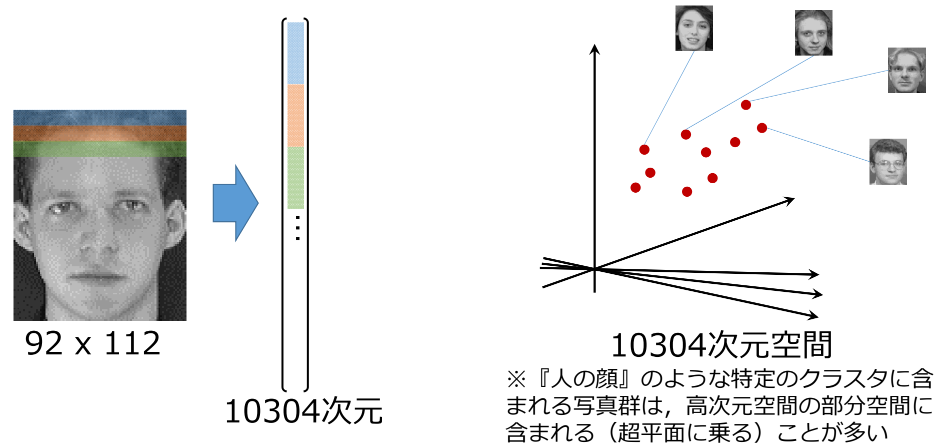
PCAによる画像の次元圧縮

- 例として顔データのPCA圧縮をしてみる
- AT&Tデータセットを利用 https://git-disl.github.io/GTDLBench/datasets/att_face_dataset/
- 40人 ＊ 10枚 ＝ 400枚の写真群 （PCAするには少し小さいが。。。）
- サイズは 92 x 112



PCAによる画像の次元圧縮

- 92 x 112 pixelの写真を，10304次元ベクトルに変換



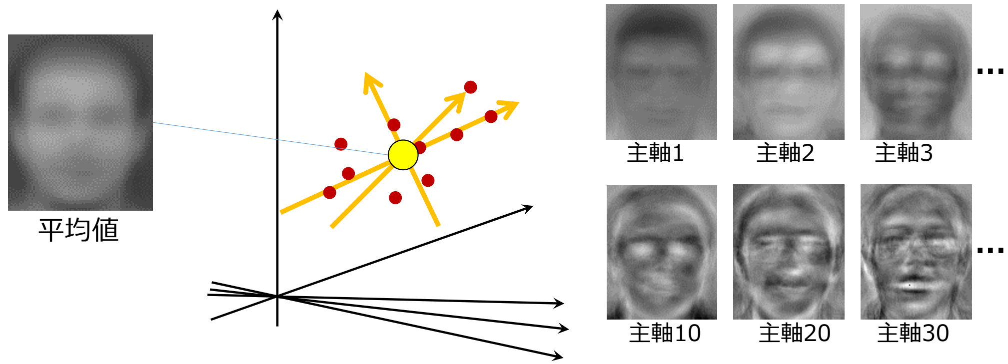
※『人の顔』のような特定のクラスタに含まれる写真群は，高次元空間の部分空間に含まれる（超平面に乗る）ことが多い

PCAによる画像の次元圧縮

- 分散共分散行列は10304 x 10304に
- 400個の固有値・固有ベクトルが取得できる

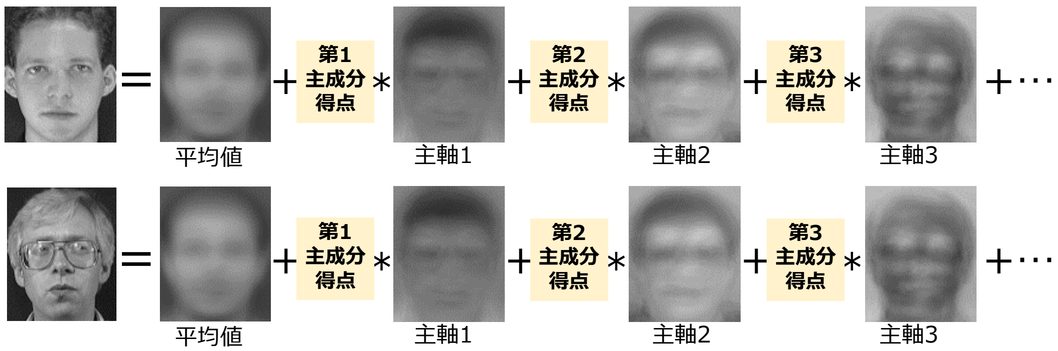
※$\sum_i(\mathbf{x}_i-\mathbf{m})(\mathbf{x}_i-\mathbf{m})^T$のrankは最大で$N$=400なので次元数分の軸は得られない

- 各軸は



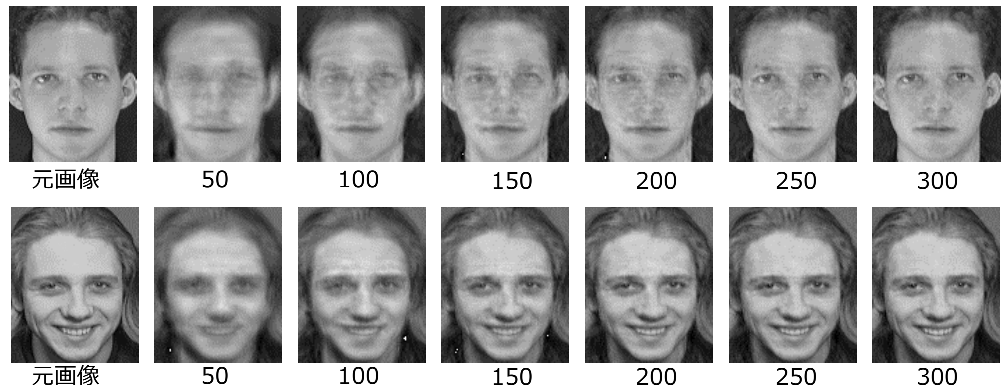
PCAによる画像の次元圧縮

- 元画像は，平均値 + Σ 主成分x主成分得点　の形で表現できる
- 後半の主成分は寄与が少ない(はず)ので，切り捨てても影響が少ない（のでは？）



PCAによる画像の次元圧縮

- 実際に50個，100個，…，300個の主成分を利用して再構築してみた



顔の向きもそろっているデータを利用するともっと速く収束すると思う

オートエンコーダ
自己符号化器

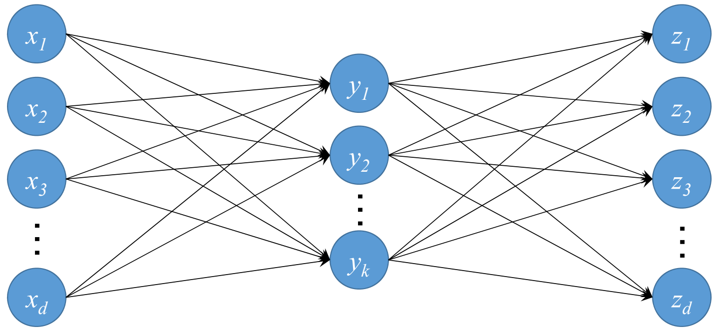
参考資料

- **深層学習**
- **(機械学習プロフェッショナルシリーズ)** 単行本
- 岡谷 貴之

オートエンコーダー（自己符号化器）とは

- ニューラルネットの一種
- 目的出力を伴わない入力だけの訓練データを利用した**教師なし学習**
- データをよく表す特徴の獲得を目指す

概要：下図のようなネットワークを考える

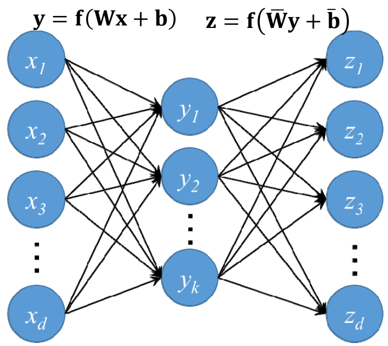


入力
$\mathbf{x} \in R^d$

中間層 $\mathbf{y}$
$\mathbf{y} = \mathbf{f}(\mathbf{W}\mathbf{x} + \mathbf{b})$
$\mathbf{W}$：重み係数
$\mathbf{b}$：バイアス項
$\mathbf{f}(\ \ )$: 活性化関数

出力層 $\mathbf{z}$
$\mathbf{z} = \mathbf{f}(\bar{\mathbf{W}}\mathbf{y} + \bar{\mathbf{b}})$
$\bar{\mathbf{W}}$: 重み係数
$\bar{\mathbf{b}}$: バイアス項
$\mathbf{f}(\ \ )$: 活性化関数

オートエンコーダの概要



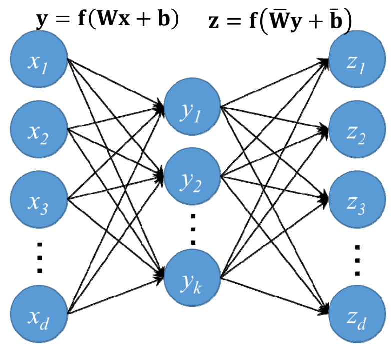
- N個の入力データ $\mathbf{x}_i \in R^d$
- 全入力$\mathbf{x}_i$に対し，その出力$\mathbf{z}_i$がなるべく等しくなるよう重み・バイアス項を学習する
- つまりデータ$\mathbf{x}_i$から，$\mathbf{W}, \mathbf{b}, \bar{\mathbf{W}}, \bar{\mathbf{b}}$を学習

※中間層の次元がdより小さい場合，$\mathbf{x}_i = \mathbf{z}_i$を必ず満たすことは不可能

- 全データに対して，入力と近い出力が得られるような学習が行えたら…

→ 元データ$\mathbf{x}_i$の情報をあまり落とさずに次元削減ができたことになる

オートエンコーダの概要



- N個の入力データ $\mathbf{x}_i \in R^d$
- 全入力$\mathbf{x}_i$に対し，その出力$\mathbf{z}_i$がなるべく等しくなるよう重み・バイアス項を学習する
- つまりデータ$\mathbf{x}_i$から，$\mathbf{W}, \mathbf{b}, \bar{\mathbf{W}}, \bar{\mathbf{b}}$を学習

※中間層の次元がdより小さい場合，$\mathbf{x}_i = \mathbf{z}_i$を必ず満たすことは不可能

- 全データに対して，入力と近い出力が得られるような学習が行えたら…

→ 元データ$\mathbf{x}_i$の情報をあまり落とさずに次元削減ができたことになる

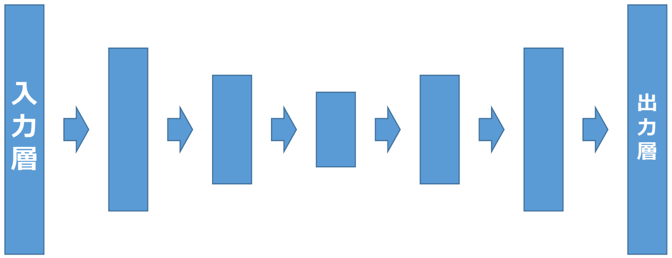
符号化

$\mathbf{y} = \mathbf{f}(\mathbf{Wx} + \mathbf{b})$

複合化

$\mathbf{z} = \bar{\mathbf{f}}(\bar{\mathbf{W}}\mathbf{y} + \bar{\mathbf{b}})$

多層自己符号化器

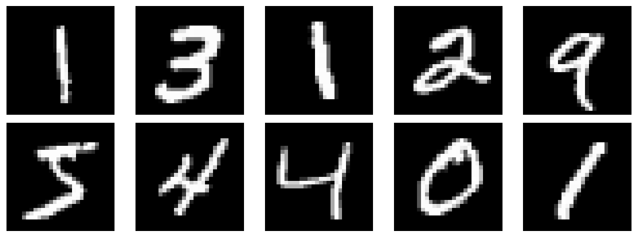


- 中間層と出力層のみでなく，複数の層を積み重ねた自己符号化器
- 複雑な分布を持ったデータの特徴抽出に利用される

自己符号化器の例

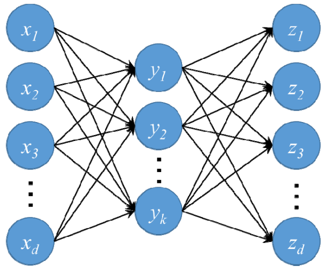
- **Mnist：** URL: http://yann.lecun.com/exdb/mnist/
  - パターン認識の勉強によく利用される**手書き数字画像**データセット
  - 数字は画像の中心に配置され，数字のサイズは正規化されている
  - 各画像のサイズは 28x28
  - データ数：トレーニング用：60000文字 / テスト用：10000文字

例)



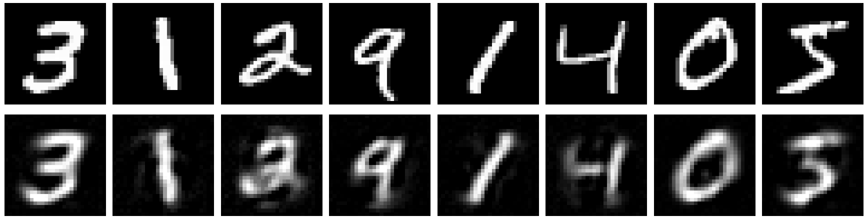
自己符号化器の例

- **Mnist** を自己符号化器で符号化してみる
  - データの次元：784 = 28x28
  - 中間層の次元：30
  - 訓練データ数：60000
  - 活性化関数 :恒等関数
  - epochs=50, batch_size=20



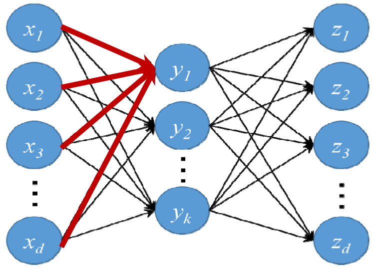
入力

出力

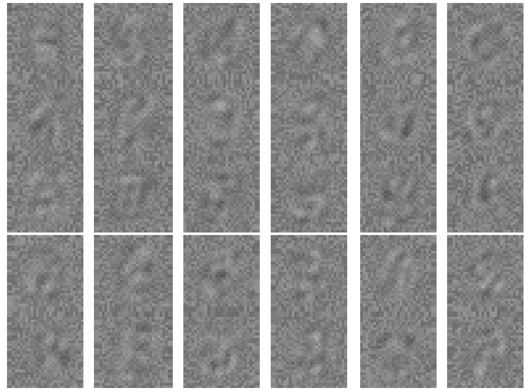


自己符号化器の例

- 自己符号化器を利用したときの興味は，戻せたかどうか？では無くて学習された重み係数（特徴量）



↑赤矢印部分の重みはd次元
これを画像に直すと…



まとめ

- オートエンコーダ（自己符号化器）とは…
  - 入力データになるべく似たデータを出力するニューラルネット
  - 目的出力を伴わない入力だけの訓練データを利用した教師なし学習
  - データをよく表す特徴の獲得を目指す
  - バイアス項 b=0，活性化関数を恒等写像とした場合主成分分析と実質的に同じ

- 応用例
  - 次元圧縮
  - 深層学習の前処理に利用