

**Анализ на главните
компоненти (Principle
Component Analysis (PCA)).
Невронни мрежи базирани на
анализ на главните
компоненти**

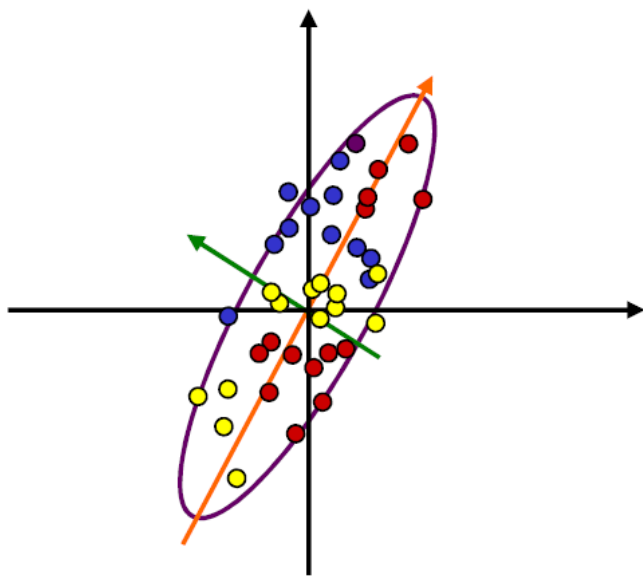
Какво представлява анализа на главните компоненти(РСА)

- РСА: Статистическа процедура
 - Намалване на размерите на входните вектори
 - Твърде много функции на входовете / информативни признаци, а и някои от тях са зависими едни от други
 - Извличане на важни (нови) характеристики на/от данните, които са функции на оригиналните входни функции / информативни признаци
 - Минимизиране на загубите на информация при процеса
 - Реализира се чрез формиране на нови интересни функции / нови информативни признаци
 - Като линейни комбинации от оригинални характеристики (апроксимаци от първи ред)
 - Новите функции е необходимо да бъдат линейно независими (за да се избегне дублиране и излишък на информация)
 - Новите функции е желателно да бъдат различни едни от други, колкото е възможно повече (за максимална вариация).

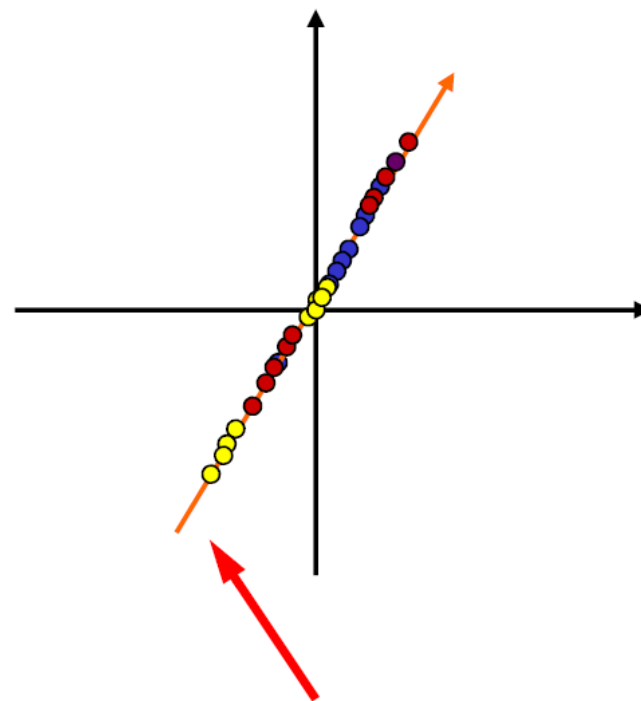
Какво представлява анализа на главните компоненти(РСА)

■ Пример

Да се намери пространство в по-ниско измерение което най - добре да представлява смисъла на данните в по-малко квадранти.



Пълно n -мерно
пространство
(в случая $n = 2$)



m -мерно
подпространство
(в случая $m = 1$)

Линейна Алгебра

- Два вектора $\mathbf{x} = (x_1, \dots, x_n)$ и $\mathbf{y} = (y_1, \dots, y_n)$ се казва че са *ортогонални* един към друг ако

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^n x_i y_i = 0.$$

- Набор вектори $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}$ от n -та размерност са *линейно независими* един от друг ако не съществува набор от реални числа a_1, \dots, a_k които не са всички нули, така че:

$$a_1 \mathbf{x}^{(1)} + \dots + a_k \mathbf{x}^{(k)} = 0$$

в противен случай, тези вектори са линейно зависими и всеки един от тях може да бъде представен като *линейна комбинация* от другите

$$\mathbf{x}^{(i)} = -\frac{a_1}{a_i} \mathbf{x}^{(1)} - \dots - \frac{a_k}{a_i} \mathbf{x}^{(k)} = \sum_{j \neq i} \frac{a_j}{a_i} \mathbf{x}^{(j)}$$

- Вектор x е *вектор на собствените стойности* на матрицата A ако съществува константа $\gamma \neq 0$, при която $Ax = \gamma x$
 - γ представлява *собствена стойност на матрицата A* (по x)
 - Матрицата A може да има повече от един вектор на собствените стойности, всеки с отделна собствена стойност на матрицата
 - Собствените вектори на матрицата отговарят на определени отличими собствени стойности на матрицата и са линейно независими едни от други
- Матрицата B се нарича *обратна* матрица на A ако $AB = I$
 - I е единичната матрица
 - Обозначаваме B като A^{-1}
 - Не всяка матрица има обратна (например когато един ред/стълб може да се изрази като линейна комбинация от останалите редове/стълбове)
- Всяка матрица A има уникална псевдообратна матрица A^* , за която са изпълнени следните отношения:

$$AA^*A = A; \quad A^*AA^* = A^*; \quad A^*A = (A^*A)^T; \quad AA^* = (AA^*)^T$$

- Пример за PCA: 3-мерен x се трансформира в 2-мерен y

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} a & b & c \\ p & q & r \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} ax_1 + bx_2 + cx_3 \\ px_1 + qx_2 + rx_3 \end{bmatrix}$$

↑
↑
↙

2-мерен вектор Трансформационна матрица W 3-мерен информативен вектор

$$WW^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Ако редовете на W са единични вектори и са ортогонални (например, $\mathbf{w}_1 \cdot \mathbf{w}_2 = ap + bq + cr = 0$), то тогава W^T е псевдообратна матрица на W .

- Генерализация

- Трансформира се n -мерен \mathbf{x} към m -мерен \mathbf{y} ($m < n$), като псевдообратната матрица W е с размерност $m \times n$
- Трансформацията е с: $\mathbf{y} = W\mathbf{x}$
- Обратната трансформация е: $\mathbf{x}' = W^T\mathbf{y} = W^TW\mathbf{x}$
- Ако W минимизира “загубата на информация” при трансформацията, то тогава $\|\mathbf{x} - \mathbf{x}'\| = \|\mathbf{x} - W^TW\mathbf{x}\|$ също трябва се минимизира
- Ако W^T е псевдообратна на W , тогава $\mathbf{x}' = \mathbf{x}$: следва, че имаме перфектна трансформация (без загуба на информация)

- Как да се намери W за определен набор входни вектори ???

- Нека $T = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ е набора (извадката) от входни вектори
- Преобразуваме ги в усреднени центрирани към нулата вектори, като извадим усреднения вектор $(\sum \mathbf{x}_i) / k$ от всяко \mathbf{x}_i .
- Изчислява се корелационната матрица $S(T)$ за тези усреднени и центрирани към нулата вектори, което представлява $n \times n$ матрица (наричана в литературата ковариантна-вариантна матрица)

- Намират се m вектори на собствените стойности $S(T)$: $\mathbf{w}_1, \dots, \mathbf{w}_m$ отговарящи на m най-големи собствени стойности $\gamma_1, \dots, \gamma_m$
- $\mathbf{w}_1, \dots, \mathbf{w}_m$ представляват първите m главни компонента от T
- $W = (\mathbf{w}_1, \dots, \mathbf{w}_m)$ в случая е търсената трансформираща матрица
- m нови информативни признака извлечени от трансформацията с W трябва да са линейно независими и с **максимална вариация**
- Това се базира на следните математически резултати:

Нека \mathbf{b} е произволен вектор, за когото е изпълнено $\|\mathbf{b}\|=1$. Тогава вариацията на $\mathbf{b} \cdot \mathbf{x}$, където $\mathbf{x} \in T$, е максимална когато \mathbf{b} е избрано да е собствен вектор на $S(T)$, който отговаря на най-големите собствени стойности на $S(T)$.

- Пример

$$T = \{(1.3, 3.2, 3.7), (1.4, 2.8, 4.1), \\ (1.5, 3.1, 4.6), (1.2, 2.9, 4.8), (1.1, 3.0, 4.8)\}.$$

усреднен вектор $(1.3, 3.0, 4.4)$

$$x_1 = (0.0, 0.2, -0.7), x_2 = (0.1, -0.2, -0.3), \dots,$$

ковариантна матрица
$$S = \frac{1}{5} \begin{pmatrix} 0.10 & 0.01 & -0.11 \\ 0.01 & 0.10 & -0.10 \\ -0.11 & -0.10 & 0.94 \end{pmatrix}$$

собствени стойности

$$\gamma_1 = 0.965, \quad (-0.823, -0.542, -0.169)$$

$$\gamma_2 = 0.090 \quad (0.553, -0.832, -0.026)$$

$$\gamma_3 = 0.084 \quad (-0.126, -0.115, 0.985)$$

вектори на собствените стойности

За $m=1$, се взема в предвид най-голямата собствена стойност от 0.965 и това води до $W=W_1$, където:

$$W_1 = (-0.823 \quad -0.542 \quad -0.169).$$

Оригиналните 3-мерни вектори се преобразуват в едномерни

$$y_1 = W_1 x_1 = (-0.823, -0.541, -0.169)^T (0, 0.2, -0.7) = 0.101$$

$$y_2 = W_1 x_2 = 0.0677$$

За $m=2$, се използват едновременно γ_1 и γ_2 :

$$W = W_2 = \begin{pmatrix} -0.823 & -0.542 & -0.169 \\ 0.553 & -0.832 & -0.026 \end{pmatrix}.$$

Оригиналните 3-мерни вектори биват преобразувани в 2-мерни

$$y_1 = W_2 x_1 = \begin{pmatrix} 0.1099 \\ -0.1462 \end{pmatrix} \quad y_2 = W_2 x_2 = \begin{pmatrix} 0.0677 \\ 0.2295 \end{pmatrix}$$

Трябва да се отбележи, че $\gamma_1/(\gamma_1 + \gamma_2 + \gamma_3) = 0.965/1.139 = 0.84$

От тук като следствие може да се заключи, че 84% от вариацията в обучаващата извадка може да се представи с единичен вектор на собствените стойности, т.е. само с използването на W_1 .

За да се запазят и прихванат над 84% от входните характеристики на данни е необходимо преобразуването на входните векторни данни в двумерно пространство с $y = W_2x$.

Пример

Данни: Момиче на 12 години отговаря по 9 точкова възходяща скала (от 1 до 9) за възприятията си от 7 свои познати. Класирането е по следните 5 описания: “естествен”, “интелигентен”, “добър”, “приятен” и “справедлив”. Да се направи групиране на данните.

Таблица:

| | естествен | интелигентен | добър | приятен | справедлив |
|--------------|-----------|--------------|-------|---------|------------|
| съученичка 1 | 1 | 5 | 5 | 1 | 1 |
| сестра | 8 | 9 | 7 | 9 | 7 |
| съученичка 2 | 9 | 8 | 9 | 9 | 8 |
| баща | 9 | 9 | 9 | 9 | 9 |
| учител | 2 | 9 | 1 | 1 | 9 |
| съученик | 5 | 7 | 7 | 7 | 9 |
| съученичка 3 | 9 | 6 | 9 | 9 | 7 |

Резултантна корелационна матрица

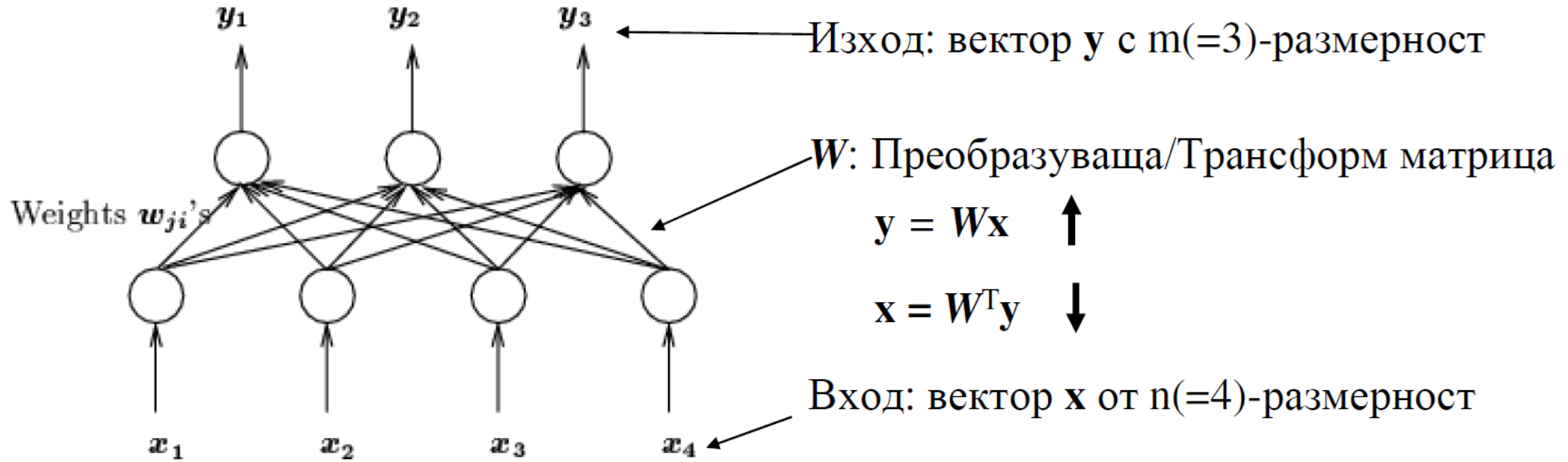
Correlation Matrix^a

| | | естествен | интелигентен | добър | приятен | справедлив |
|-----------------|--------------|-----------|--------------|-------|---------|------------|
| Correlation | естествен | 1,000 | ,338 | ,854 | ,969 | ,484 |
| | интелигентен | ,338 | 1,000 | -,101 | ,280 | ,737 |
| | добър | ,854 | -,101 | 1,000 | ,886 | ,125 |
| | приятен | ,969 | ,280 | ,886 | 1,000 | ,472 |
| | справедлив | ,484 | ,737 | ,125 | ,472 | 1,000 |
| Sig. (1-tailed) | естествен | | ,229 | ,007 | ,000 | ,136 |
| | интелигентен | ,229 | | ,415 | ,271 | ,029 |
| | добър | ,007 | ,415 | | ,004 | ,394 |
| | приятен | ,000 | ,271 | ,004 | | ,142 |
| | справедлив | ,136 | ,029 | ,394 | ,142 | |

a. Determinant = ,001

Матрицата е симетрична. Анализират се само корелационните коефициенти > 0,5. Съответните им нива на значимост от долната половина на таблицата в случая имат нива на значимост Sig. <0,05. Това показва, че тези корелационни зависимости са статистически значими и трябва да участват в анализа. Останалите са незначими. В частност за тази извадка най-голям е корелационният коефициент между “приятен” и “естествен” (0.969) и той е значим.

- Архитектура на PCA невронна мрежа



- Обучава се W така, че да може да преобразува примерни прости входни вектори \mathbf{x}_l от n към m размерен изходен вектор \mathbf{y}_l .
- Преобразуването трябва да минимизира загубата на информация:

Трябва да се намери W , което да минимизира:

$$\sum_l \|\mathbf{x}_l - \mathbf{x}_l'\| = \sum_l \|\mathbf{x}_l - W^T W \mathbf{x}_l\| = \sum_l \|\mathbf{x}_l - W^T \mathbf{y}_l\|$$

където \mathbf{x}_l' е “противоположната” трансформация $\mathbf{y}_l = W\mathbf{x}_l$ чрез W^T

- Обучение на W за НМ с PCA

- Обучение без учител (unsupervised learning):
зависи само от входните образци \mathbf{x}_l
- Формира се от грешка: ΔW зависи от $\|\mathbf{x}_l - \mathbf{x}_l'\| = \|\mathbf{x}_l - \mathbf{W}^T \mathbf{W} \mathbf{x}_l\|$
- Стартира се с произволно избрани тегла и се променя W съответно с:

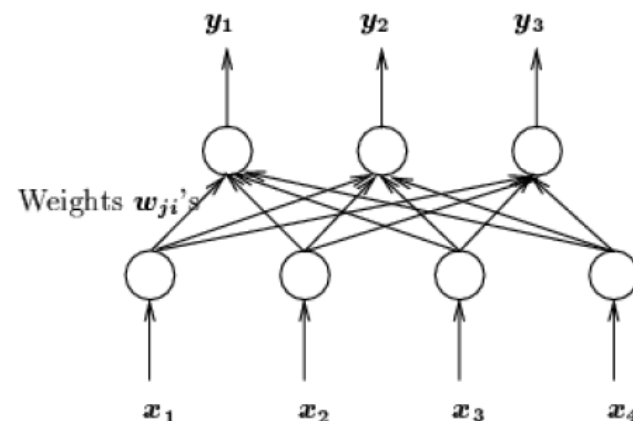
$$\Delta W = \eta (\mathbf{y}_l \mathbf{x}_l^T - \mathbf{K}_l W) \text{ където } \mathbf{K}_l = \mathbf{y}_l \mathbf{y}_l^T$$

- Това е само една от многобройните възможности за K
- Правилото за обновление на теглата се трансформира в:

$$\Delta W = \eta_l (\mathbf{y}_l \mathbf{x}_l^T - \mathbf{y}_l \mathbf{y}_l^T \mathbf{W}) = \eta_l \mathbf{y}_l (\mathbf{x}_l^T - \mathbf{y}_l^T \mathbf{W}) = \eta_l \mathbf{y}_l (\mathbf{x}_l^T - \mathbf{W}^T \mathbf{y}_l)$$

↗ колона
 ↘ вектор
 ред
 вектор

↑ трансформираща
 грешка



- Пример (при входи като предходните)

$\eta_l = 1.0$, с начални стойности на теглата $(0.3, 0.4, 0.5)$

За първия вход, $y = (0.3, 0.4, 0.5) \cdot (0, 0.2, -0.7) = -0.27$

$$\Delta W = (-0.27(0.00, 0.20, -0.70) - (-0.27)^2(0.30, 0.40, 0.50))$$

$$W = (0.30, 0.40, 0.50) + \Delta W = (0.28, 0.32, 0.65)$$

За следващия вход (x_2) , $y = -0.23$, и

$$W = (0.278, 0.316, 0.652) + \Delta W = (0.240, 0.346, 0.687)$$

Последващите презентации на x_3 , x_4 и x_5 променят W на:

$(0.272, 0.351, 0.697)$ След x_3

$(0.238, 0.313, 0.751)$ След x_4

$(0.172, 0.293, 0.804)$ След x_5

$(-0.008, 0.105, 0.989)$ След втора епоха

$(-0.111, -0.028, 1.004)$ След трета епоха

евентуална сходимост към първи главен компонент $(-0.823 \quad -0.542 \quad -0.169)$

- Бележки

- НМ с РСА **апроксимират** главните компоненти (като може да съществува остатъчна грешка от процеса)
- Те намират главните компоненти чрез обучение, без използване на методи от статистиката
- Проектират входната информация в подпространство с максимизация на вариацията на данните, които не са обвързани
- Имаме стабилизация на теглата с плавно редуциране параметъра на скоростта на обучение η
- Възможно е използването на следните подходи за подобряване на резултатите от процедурата по обучение.
 - Вместо да се използва функцията за изхода $y = Wx$, може да се използва нелинейна функция S и да се минимизира:

$$\mathcal{E} \{ ||x - W^T S(Wx)||^2 \}$$

- Ако S е диференцируема, може да се използват градиентните методи за намаляване
- Например: нека S е монотонна $S(-x) = -S(x)$ е.g., $S(x) = x^3$

Пример за приложение на НМ с PCA за разпознаване на лица

- Нека имаме PCA мрежа с 2429 19x19 черно-бели изображения
- Можем ли да получим добра реконструкция от само 3 компонента?



- PCA участва във фазата на първична обработка на данните с втори последващ класификационен слой
 - При разпознаване на лица с 3 компонента PCA се запазва 79% точност на определянето лице/не-лице от тестовите данни
- Този пример дава и добро ниво на визуализация

Прилагане на PCA към лица и какви бази са научени.



Главните компоненти от обект лице от изображенията
(“собствени стойности на лицата”)



reconstructed with 2 bases



reconstructed with 10 bases



reconstructed with 100 bases



reconstructed with 506 bases



mean



principal basis 1



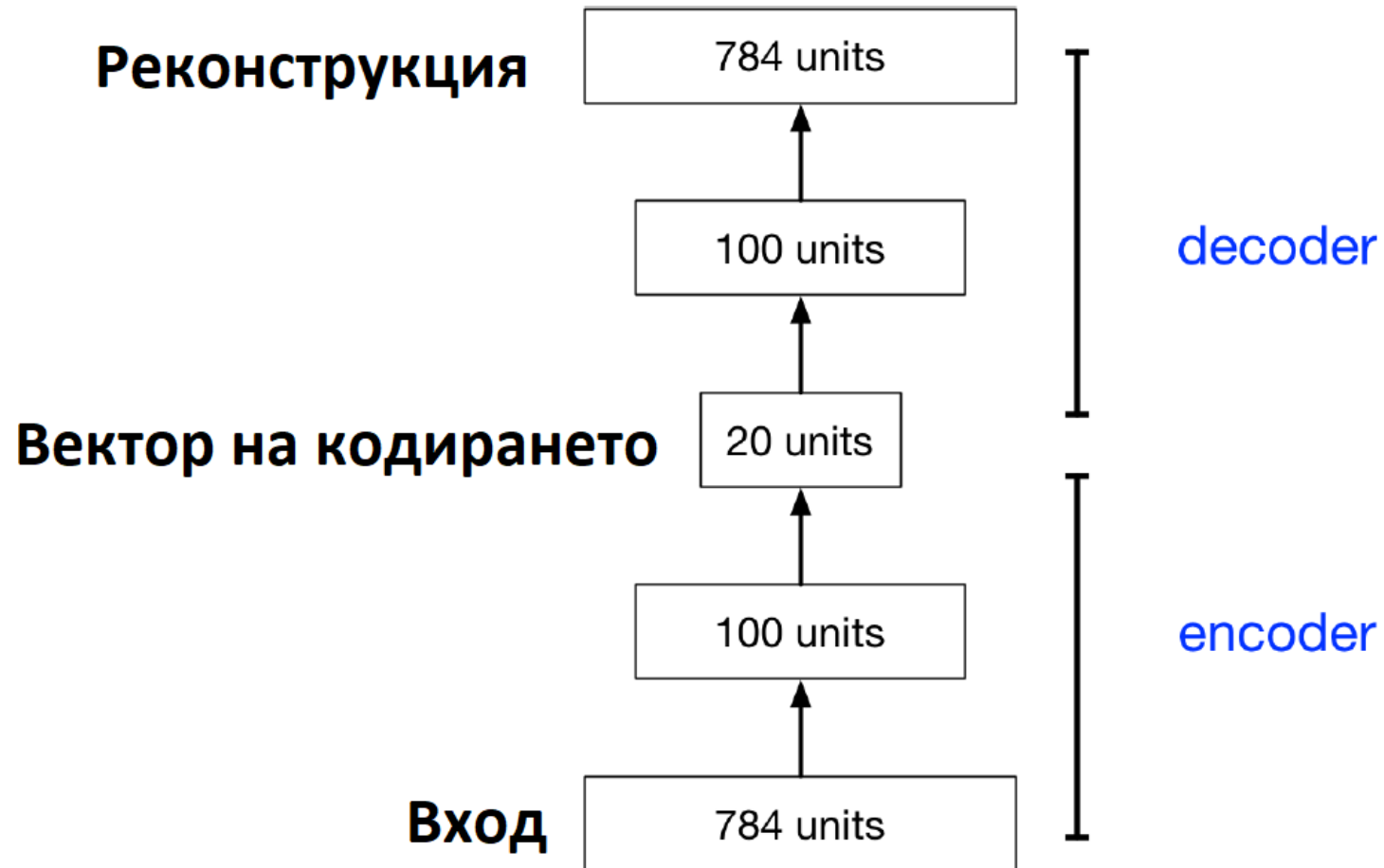
principal basis 2



principal basis 3



- **Автоенкодерите** са многослойни невронни мрежи със задача да приемат входен вектор x и да реализират прогноза за x .
- Добавя се **ограничителен слой**, чиято размерност е много по-малка от размерността на входния слой.

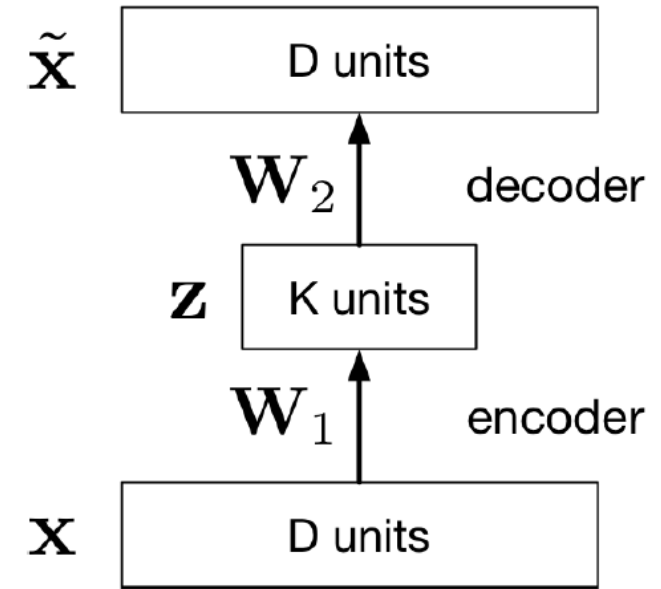


С автоенкодерите може да се наложат многомерни данни към двумерно пространство, с което да е възможна визуализация. Намират се абстрактни признаци и без използване на 'учител'.

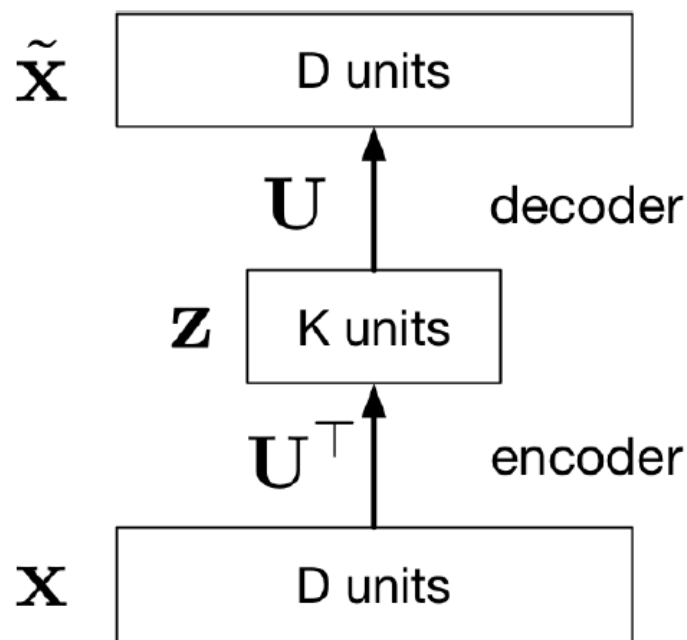
- Най-простият вид автоенкодери са с един скрит слой, линейни активационни функции и квадратична функция на грешката.

$$\mathcal{L}(\mathbf{x}, \tilde{\mathbf{x}}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$$

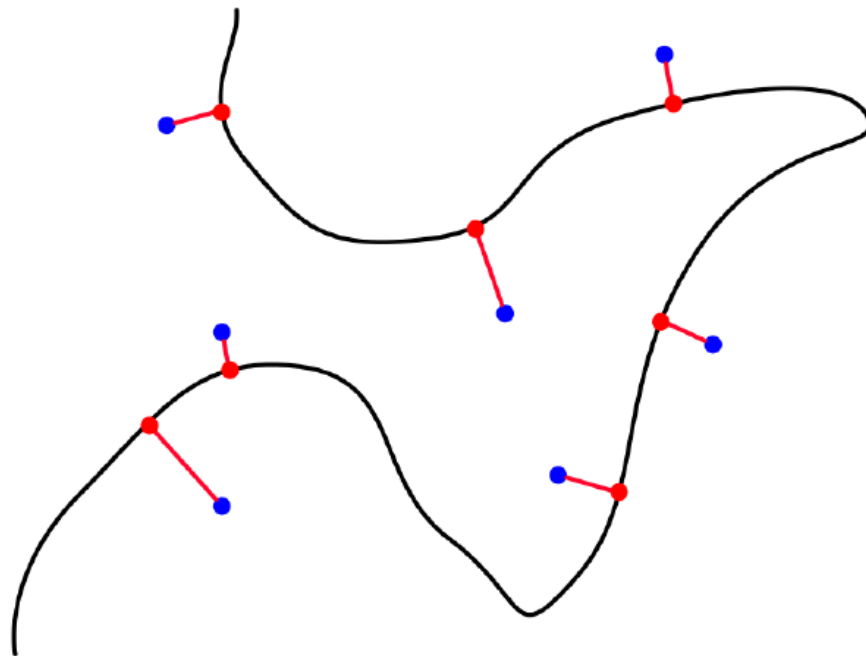
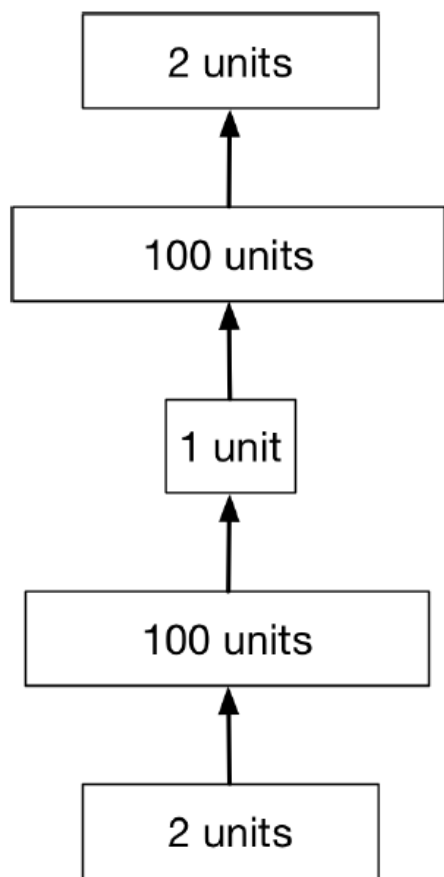
- Невронната мрежа изчислява $\tilde{\mathbf{x}} = \mathbf{W}_2 \mathbf{W}_1 \mathbf{x}$, което е линейна функция.
- Ако $K \geq D$, ние можем да избираме \mathbf{W}_2 и \mathbf{W}_1 така, че произведението $\mathbf{W}_2 \mathbf{W}_1$ да дава единичната матрица като резултат.
- Но нека $K < D$:
 - \mathbf{W}_1 проектира \mathbf{x} към K -мерно пространство, извършващо компресия на информацията.



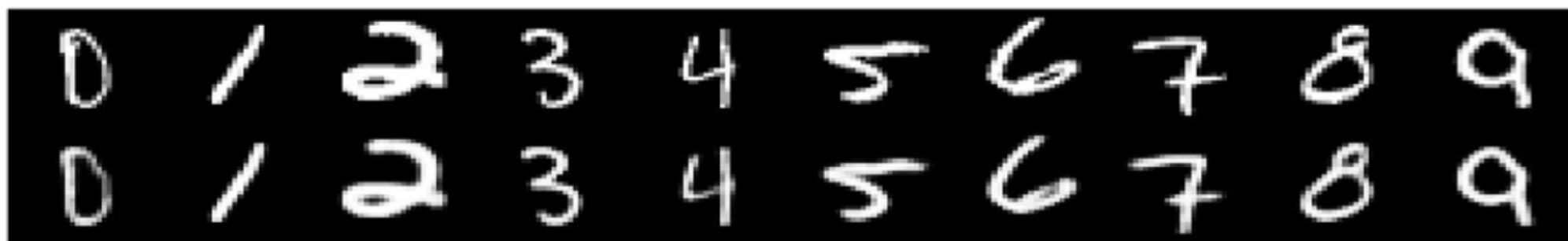
- Наблюдава се, че изходът на автоенкодера трябва да лежи в K -мерното подпространство разпределен между стълбовете на \mathbf{W}_2 .
- Наблюдава се, че най-доброто възможно K -мерно подпространство по отношение на грешката от реконструкция е PCA пространството
- Автоенкодера постига това със задаване на $\mathbf{W}_1 = \mathbf{U}^\top$ и $\mathbf{W}_2 = \mathbf{U}$ По
- Така, оптималните тегла за един линеен автоенкодер се явяват на практика главните компоненти!



- Дълбоките нелинейни автоенкодери проектират данните от входното пространство не в подпространство, а в **групи/колекция**.
- Тези групи представляват карта на декодера.
- Това представлява един вид **нелинейна редукция на размерността**.



- Нелинейните автоенкодери могат да съхранят и научат много по-силни кодирания за дадена размерност, в сравнение с линейните автоенкодери!



real
data

30-D
deep auto



30-D
PCA

Пример за 2-мерен автоенкодер представящ заглавията от вестниците. Класовете са оцветени в цвят според тематиката, но на алгоритъма не са били подавани имената/заглавията на класовете ('без учител').

