

Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

Theorem 1

Pro libovolnou matici $A \in \mathbb{C}^{n,n}$ existuje unitární matice U taková, že

$$A = U^*RU,$$

kde R je horní trojúhelníková matice.

Theorem 2

Pro libovolnou normální matici A existuje unitární matice U taková, že

$$A = U^*RU,$$

kde R je diagonální matice. Je-li A hermitovská, pak R má na diagonále reálná čísla.

SVD rozklad

Ukážeme si určité zobecnění této věty, tzv. **singulární rozklad (SVD, singular value decomposition)**, který říká, že

$$\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^*,$$

kde $\mathbb{X} \in \mathbb{C}^{n,m}$ je libovolná, $\mathbb{U} \in \mathbb{C}^{n,n}$ a $\mathbb{V} \in \mathbb{C}^{m,m}$ jsou unitární a $\Sigma \in \mathbb{R}^{n,m}$ je diagonální s nezápornými prvky.

Nejprve si odvodíme SVD pro čtvercové matice a následně zobecníme pro obdélníkové.

Lemma 3

Bud' $\mathbb{X} \in \mathbb{C}^{n,n}$, pak platí:

- Pokud \vec{v} je jednotkový vlastní vektor matice $\mathbb{X}^* \mathbb{X}$ příslušný k nenulovému vlastnímu číslu λ , potom $\mathbb{X} \vec{v}$ je vlastní vektor matice $\mathbb{X} \mathbb{X}^*$ se stejným příslušným vlastním číslem λ . Navíc platí $\|\mathbb{X} \vec{v}\|_2 = \sqrt{\lambda}$.
- Pokud \vec{u} je jednotkový vlastní vektor matice $\mathbb{X} \mathbb{X}^*$ příslušný k nenulovému vlastnímu číslu λ , potom $\mathbb{X}^* \vec{u}$ je vlastní vektor matice $\mathbb{X}^* \mathbb{X}$ se stejným příslušným vlastním číslem λ . Navíc platí $\|\mathbb{X}^* \vec{u}\|_2 = \sqrt{\lambda}$.

Proof.

Nechť \vec{v} je jednotkový vlastní vektor matice $\mathbb{X}^* \mathbb{X}$, pak platí

$$\mathbb{X}^* \mathbb{X} \vec{v} = \lambda \vec{v} \Rightarrow \mathbb{X} \mathbb{X}^* (\mathbb{X} \vec{v}) = \mathbb{X} (\mathbb{X}^* \mathbb{X} \vec{v}) = \mathbb{X} \lambda \vec{v} = \lambda (\mathbb{X} \vec{v}).$$

Pro normu platí

$$\|\mathbb{X} \vec{v}\|_2^2 = (\mathbb{X} \vec{v}, \mathbb{X} \vec{v}) = \vec{v}^* \underbrace{\mathbb{X}^* \mathbb{X} \vec{v}}_{=\lambda \vec{v}} = \vec{v}^* \lambda \vec{v} = \lambda \|\vec{v}\|_2^2 = \lambda.$$

Druhé tvrzení se dokáže stejně. □

Lemma 4

Necht' $\mathbb{X} \in \mathbb{C}^{n,n}$, potom matice $\mathbb{X}^\mathbb{X}$ a $\mathbb{X}\mathbb{X}^*$ mají stejná vlastní čísla $\lambda_1, \dots, \lambda_n$.
Necht' $\vec{v}_1, \dots, \vec{v}_n$ jsou příslušné ortonormální vlastní vektory hermitovské matice $\mathbb{X}^*\mathbb{X}$, pak lze najít ortonormální vlastní vektory $\vec{u}_1, \dots, \vec{u}_n$ hermitovské matice $\mathbb{X}\mathbb{X}^*$, které splňují*

$$\mathbb{X}\vec{v}_i = \sqrt{\lambda_i}\vec{u}_i \text{ pro } i = 1, \dots, n.$$

Proof.

- Pokud \vec{v}_i je vlastní vektor matice $\mathbb{X}^*\mathbb{X}$ příslušející nenulovému vlastnímu číslu λ_i , pak definujeme

$$\vec{u}_i = \frac{\mathbb{X}\vec{v}_i}{\sqrt{\lambda_i}}$$

a podle předchozího lematu je \vec{u}_i jednotkový vlastní vektor matice $\mathbb{X}\mathbb{X}^*$. Dále platí

$$\vec{u}_i^* \vec{u}_j = (\mathbb{X}\vec{v}_i)^* (\mathbb{X}\vec{v}_j) = \frac{1}{\sqrt{\lambda_i\lambda_j}} \vec{v}_i^* \underbrace{\mathbb{X}^*\mathbb{X}\vec{v}_j}_{=\lambda_j\vec{v}_j} = \sqrt{\frac{\lambda_i}{\lambda_j}} \vec{v}_i^* \vec{v}_j = 0.$$



Proof.

- Pokud má matice $\mathbb{X}^*\mathbb{X}$ r nenulových vlastních čísel, pak má stejný počet nenulových vlastních čísel i matice $\mathbb{X}\mathbb{X}^*$. U obou tedy zbývá $n - r$ nulových vlastních čísel a příslušných vlastních vektorů. Pak pro libovolný z těchto vlastních vektorů platí

$$\sqrt{\lambda_i} \vec{u}_i = \mathbb{X} \vec{v}_i = \vec{0},$$

neboť je-li \vec{v}_i nulový vlastní vektor matice $\mathbb{X}^*\mathbb{X}$, pak platí

$$\mathbb{X}^*\mathbb{X} \vec{v}_i = \vec{0} \Leftrightarrow (\mathbb{X}^*\mathbb{X} \vec{v}_i, \vec{v}_i) = 0 \Leftrightarrow (\mathbb{X} \vec{v}_i, \mathbb{X} \vec{v}_i) = 0 \Leftrightarrow \mathbb{X} \vec{v}_i = \vec{0}.$$



Theorem 5

Pro libovolnou matici $\mathbb{X} \in \mathbb{C}^{n,n}$ existují unitární matice $\mathbb{U} \in \mathbb{C}^{n,n}$ a $\mathbb{V} \in \mathbb{C}^{n,n}$ a diagonální matice $\Sigma \equiv \text{diag}\{\sigma_1, \dots, \sigma_n\} \in \mathbb{R}^{n,n}$, kde $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ takové, že platí

$$\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^*.$$

Sloupce matice \mathbb{U} se nazývají **levé singulární vektory** a řádky matice \mathbb{V}^* se nazývají **pravé singulární vektory**. Hodnoty σ_i se nazývají **singulární hodnoty** matice \mathbb{X} . Je-li r nejmenší takové, že $\sigma_r > 0$, pak matice \mathbb{X} má hodnost r .

Proof.

Z předchozího víme, že pro vlastní čísla $\lambda_1, \dots, \lambda_n$, vlastní vektory $\vec{u}_1, \dots, \vec{u}_n$ matice $\mathbb{X}\mathbb{X}^*$ a vlastní vektory $\vec{v}_1, \dots, \vec{v}_n$ matice $\mathbb{X}^*\mathbb{X}$ platí

$$\sqrt{\lambda_i} \vec{u}_i = \mathbb{X} \vec{v}_i,$$

což lze psát jako

$$(\vec{u}_1, \dots, \vec{u}_n) \Sigma = \mathbb{X} (\vec{v}_1, \dots, \vec{v}_n),$$

kde $\Sigma = \text{diag} \{ \sqrt{\lambda_1}, \dots, \sqrt{\lambda_n} \} = \text{diag} \{ \sigma_1, \dots, \sigma_n \}$. Maticově pak lze psát $\mathbb{U}\Sigma = \mathbb{X}\mathbb{V}$ a přenásobením maticí \mathbb{V}^* zprava dostáváme

$$\mathbb{U}\Sigma\mathbb{V}^* = \mathbb{X}.$$



SVD rozklad

Pokud matice X není čtvercová, můžeme jí na čtvercovou doplnit přidáním nulových řádků nebo sloupců.

SVD rozklad

Lemma 6

Bud' $\mathbb{X} \in \mathbb{C}^{r,c}$ obdélníková matice a necht' $\mathbb{B} \in \mathbb{C}^{n,n}$ je čtvercová matice, která vznikla z matice \mathbb{X} doplněním nulových sloupců a řádků tak, že $n = \max\{r, c\}$. Pak existuje rozklad

$$\mathbb{B} = (\mathbb{X}, \mathbf{0}) = \underbrace{\mathbb{U} \begin{pmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}}_{\Sigma} \underbrace{\begin{pmatrix} \mathbb{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbb{V}_2 \end{pmatrix}}_{\mathbb{V}^*} \text{ pokud } c < n,$$

a

$$\mathbb{B} = \begin{pmatrix} \mathbb{X} \\ \mathbf{0} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbb{U}_1 & \mathbf{0} \\ \mathbf{0} & \mathbb{U}_2 \end{pmatrix}}_{\mathbb{U}} \underbrace{\begin{pmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}}_{\Sigma} \mathbb{V}^* \text{ pokud } r < n,$$

kde $\mathbb{U}, \mathbb{V} \in \mathbb{C}^{n,n}$, $\Sigma \in \mathbb{R}^{n,n}$, $\mathbb{U}_1 \in \mathbb{C}^{r,r}$, $\mathbb{V}_1 \in \mathbb{C}^{c,c}$, $\mathbb{U}_2 \in \mathbb{C}^{n-r,n-r}$, $\mathbb{V}_2 \in \mathbb{C}^{n-c,n-c}$ a $\Sigma_1 \in \mathbb{R}^{\min\{r,c\}, \min\{r,c\}}$.

Proof.

Ukážeme pro případ $\mathbb{B} = (\mathbb{X}, \mathbf{0})$. Pak $\mathbb{B}^*\mathbb{B} = \begin{pmatrix} \mathbb{X}^* \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbb{X} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbb{X}^*\mathbb{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ a tato matice má maximálně c nenulových vlastních čísel. Ty tvoří matici $\Sigma \in \mathbb{R}^{c,c}$ a příslušné vlastní vektory tvoří sloupce matice $\mathbb{V}_1 \in \mathbb{C}^{c,c}$. Matici $\mathbb{V}_2 \in \mathbb{C}^{n-c,n-c}$ tvoří libovolné ortonormální vektory tvořící bázi \mathbb{C}^{n-c} . Je-li pak \vec{v}_i i -tý sloupec matice \mathbb{V} , pak je

$$\mathbb{B}^*\mathbb{B} \begin{pmatrix} \vec{v}_{i,1} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbb{X}^*\mathbb{X}\vec{v}_{i,1} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \lambda_i\vec{v}_{i,1} \\ \mathbf{0} \end{pmatrix} = \lambda_i\vec{v} \text{ pro } i \leq c,$$

$$\mathbb{B}^*\mathbb{B} \begin{pmatrix} \mathbf{0} \\ \vec{v}_{i,2} \end{pmatrix} = \begin{pmatrix} \mathbb{X}^*\mathbb{X}\mathbf{0} \\ \mathbf{0}\vec{v}_{i,2} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} = 0\vec{v} \text{ pro } i > c,$$

tj. vektory $\vec{v}_1, \dots, \vec{v}_n$ jsou vlastní vektory matice $\mathbb{B}^*\mathbb{B}$ a $\sigma_1, \dots, \sigma_c, 0, \dots, 0$ příslušná vlastní čísla. Matici \mathbb{U} pak lze zkonstruovat stejně jako v důkazu SVD rozkladu pro čtvercové matice.

Theorem 7

Plný singulární rozklad (Full SVD): Pro libovolnou matici $\mathbb{X} \in \mathbb{C}^{n,m}$ existují unitární matice $\mathbb{U} \in \mathbb{C}^{n,n}$ a $\mathbb{V} \in \mathbb{C}^{m,m}$ a diagonální matice $\Sigma \equiv \text{diag}\{\sigma_1, \dots, \sigma_d\} \in \mathbb{R}^{d,d}$ pro $d = \min\{n, m\}$, kde $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d \geq 0$ takové, že platí

$$\mathbb{X} = \mathbb{U} \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} \mathbb{V}^* \text{ pro } m < n,$$

$$\mathbb{X} = \mathbb{U} (\Sigma \ 0) \mathbb{V}^* \text{ pro } m > n,$$

Sloupce matice \mathbb{U} se nazývají **levé singulární vektory** a řádky matice \mathbb{V}^* se nazývají **pravé singulární vektory**. Hodnoty σ_i se nazývají **singulární hodnoty** matice \mathbb{X} . Je-li r nejmenší takové, že $\sigma_r > 0$, pak matice \mathbb{X} má hodnost r .

SVD rozklad

- Tento tzv. plný rozklad obsahuje uměle přidané sloupce matic \mathbb{U} a \mathbb{V} , které se násobí s nulovým blokem přidaným k matici Σ .
- Pokud je odstraníme, dostáváme tzv. ekonomický rozklad.

Theorem 8

Ekonomický singulární rozklad (Economy SVD): Pro libovolnou matici $\mathbb{X} \in \mathbb{C}^{n,m}$ existují unitární matice $\mathbb{U} \in \mathbb{C}^{n,d}$ a $\mathbb{V} \in \mathbb{C}^{m,d}$ a diagonální matice $\Sigma \equiv \text{diag}\{\sigma_1, \dots, \sigma_d\} \in \mathbb{R}^{d,d}$ pro $d = \min\{n, m\}$, kde $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d \geq 0$ takové, že platí

$$\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^*.$$

SVD rozklad

Full SVD

$$\begin{bmatrix} \mathbf{X} \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{\mathbf{U}} & \hat{\mathbf{U}}^\perp \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \hat{\Sigma} \\ \mathbf{0} \end{bmatrix}}_{\Sigma} \begin{bmatrix} \mathbf{V}^* \end{bmatrix}$$

Economy SVD

$$= \begin{bmatrix} \hat{\mathbf{U}} \end{bmatrix} \begin{bmatrix} \hat{\Sigma} \end{bmatrix} \begin{bmatrix} \mathbf{V}^* \end{bmatrix}$$

Zdroj: S.L. Brunton, J.N. Kutz, Data-driven science and engineering, Cambridge university press, 2022.

SVD rozklad

- Dále lze ještě eliminovat nulové singulární hodnoty σ_j .
- Tím dostáváme tzv. kompaktní singulární rozklad.

Theorem 9

Kompaktní singulární rozklad (Compact SVD): Pro libovolnou matici $\mathbb{X} \in \mathbb{C}^{n,m}$ existují unitární matice $\mathbb{U} \in \mathbb{C}^{n,r}$ a $\mathbb{V} \in \mathbb{C}^{m,r}$ a diagonální matice $\Sigma \equiv \text{diag}\{\sigma_1, \dots, \sigma_r\} \in \mathbb{R}^{r,r}$ pro $\text{rank } \mathbb{X} = r \leq \min\{n, m\}$, kde $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ takové, že platí

$$\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^*.$$

- V různých aplikacích nás často zajímá nejlepší aproximace matice.
- Platí

$$\mathbb{X} = \sum_{i=1}^{\min\{n,m\}} \sigma_i \vec{u}_i \vec{v}_i^*.$$

- Nabízí se tedy zahodit členy, kde je σ_i hodně malé, tj.

$$\mathbb{X} \approx \mathbb{X}_k = \sum_{i=1}^{k < \min\{n,m\}} \sigma_i \vec{u}_i \vec{v}_i^*.$$

- Tím dostáváme tzv. zkrácený singulární rozklad.

Theorem 10

Zkrácený singulární rozklad (Truncated SVD): Pro libovolnou matici $\mathbb{X} \in \mathbb{C}^{n,m}$ existují unitární matice $\mathbb{U} \in \mathbb{C}^{n,k}$ a $\mathbb{V} \in \mathbb{C}^{m,k}$ a diagonální matice $\Sigma \equiv \text{diag}\{\sigma_1, \dots, \sigma_k\} \in \mathbb{R}^{k,k}$ pro $k < \text{rank}\mathbb{X} \leq \min\{n, m\}$, kde $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \geq 0$ takové, že platí

$$\mathbb{X} \approx \mathbb{U}\Sigma\mathbb{V}^*,$$

SVD rozklad

Full SVD

$$\mathbf{X} = \underbrace{\begin{bmatrix} \tilde{\mathbf{U}} & \hat{\mathbf{U}}_{\text{rem}} & \hat{\mathbf{U}}^{\perp} \end{bmatrix}}_{\mathbf{U}} \begin{bmatrix} \tilde{\Sigma} & & \\ & \hat{\Sigma}_{\text{rem}} & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{V}}^* \\ \mathbf{V}_{\text{rem}} \end{bmatrix}$$

Truncated SVD

$$\approx \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\Sigma} \\ \tilde{\mathbf{V}}^* \end{bmatrix}$$

Zdroj: S.L. Brunton, J.N. Kutz, Data-driven science and engineering, Cambridge university press, 2022.

Theorem 11

Je-li $\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^*$ SVD rozklad matice \mathbb{X} , pak platí

$$\|\mathbb{X}\|_2 = \sigma_1 \quad \|\mathbb{X}\|_F = \sqrt{\sum_{i=1}^m \sigma_i^2},$$

kde $\|\cdot\|_F$ je Frobeniova norma

$$\|\mathbb{X}\|_F = \sqrt{\sum_{i,j=1}^{n,m} |a_{ij}|^2}.$$

Principle component analysis, PCA

- půjde nám o nalezení hlavních komponent podle variance dat
- předpokládáme, že data $\vec{x}_1, \dots, \vec{x}_n$ tvoří řádky matice \mathbb{X}
- napočítáme střední hodnotu

$$\vec{\mu} = \frac{1}{n} \sum_{i=1}^n \vec{x}_i$$

- data vycentrujeme

$$\mathbb{M} = \mathbb{X} - \vec{1}\vec{\mu}^T$$

- napočítáme a proximujeme kovarianční matici

$$\mathbb{C} = \frac{1}{n} \mathbb{M}^* \mathbb{M} \approx \mathbb{V} \Delta \mathbb{V}^* \text{ a } \mathbb{U} = \mathbb{M} \mathbb{V}$$

SVD rozklad

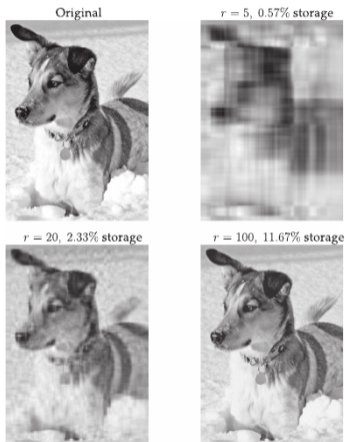


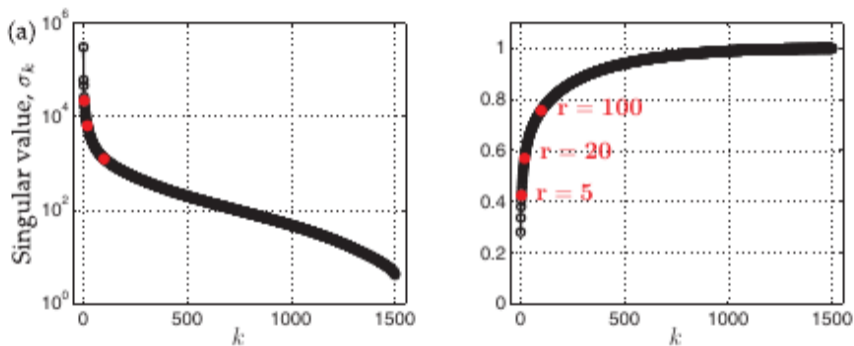
Figure 1.3 Image compression of Mordecai the snow dog, truncating the SVD at various ranks r . Original image resolution is 2000×1500 .

Aplikace v datové kompresi –
obrázek lze chápat jako matici a
tu aproximovat pomocí SVD

- původní obrázek
- $k = 5, 0.57\%$ místa
- $k = 20, 2.33\%$ místa
- $k = 100, 11.67\%$ místa

Zdroj: S.L. Brunton, J.N. Kutz, Data-driven science and engineering, Cambridge university press, 2022.

SVD rozklad



Zdroj: S.L. Brunton, J.N. Kutz, Data-driven science and engineering, Cambridge university press, 2022.

SVD rozklad

X



XX^*



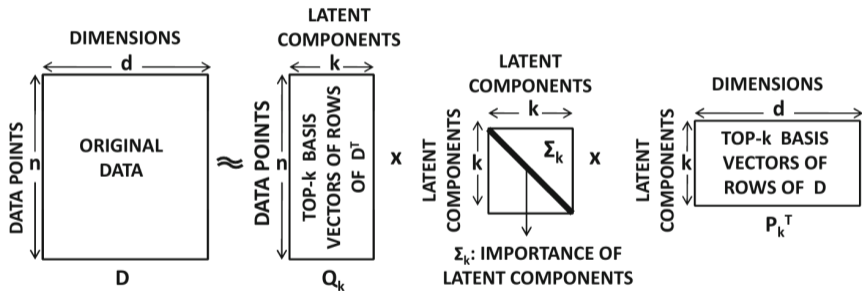
X^*X



Zdroj: S.L. Brunton, J.N. Kutz, Data-driven science and engineering, Cambridge university press, 2022.

SVD rozklad

Ze vztahu $\mathbb{X} \approx \mathbb{X}_k = \sum_{i=1}^k \sigma_i \vec{u}_i \vec{v}_i^*$ vidíme, že SVD generuje bázi jak pro řádky tak i pro sloupce matice \mathbb{X} .



Zdroj: Ch. C. Aggarwal, Linear Algebra and Optimization for Machine Learning, Springer, 2020.

SVD rozklad

- SVD rozklad se také někdy zapisuje ve zkrácené formě.

$$\mathbb{X} \approx \underbrace{(\mathbb{U}\Sigma)}_{=\mathbb{U}_\Sigma} \mathbb{V}^* = \mathbb{U}_\Sigma \mathbb{V}^*$$

- Přenásobením maticí \mathbb{V} zprava, pak dostáváme vztah

$$\mathbb{X}\mathbb{V} = \mathbb{U}_\Sigma.$$

- Výpočet SVD rozkladu pak lze také formulovat jako optimalizační úlohu:

$$\max_{\mathbb{V}} \|\mathbb{X}\mathbb{V}\|_F^2 \text{ za podmínky } \mathbb{V}^*\mathbb{V} = \mathbb{I}_k \in \mathbb{R}^{k,k},$$

neboť

$$\max_{\mathbb{V}} \|\mathbb{X}\mathbb{V}\|_F^2 = \max_{\vec{v}_1, \dots, \vec{v}_k} \sum_{i=1}^k \|\mathbb{X}\vec{v}_i\|^2 = \max_{\vec{v}_1, \dots, \vec{v}_k} \sum_{i=1}^k \vec{v}_i^* \mathbb{X}^* \mathbb{X} \vec{v}_i = \sum_{i=1}^k \sigma_i^2.$$

SVD rozklad

Výpočet SVD rozkladu lze provést také následující minimalizační úlohou:

$$\min_{\mathbf{U}_\Sigma, \mathbf{V}} \|\mathbf{X} - \mathbf{U}_\Sigma \mathbf{V}^*\|_F^2.$$

- Zde ani nemusíme udávat podmínku $\mathbf{V}^* \mathbf{V} = \mathbf{I}_k$.
- Každé optimální řešení $(\mathbf{U}_\Sigma^{(0)}, \mathbf{V}^{(0)})$ lze převést takto

$$\mathbf{U}_\Sigma^{(0)} \mathbf{V}^{(0)*} = \underbrace{\mathbf{U}_\Sigma^{(0)} \mathbf{R}^*}_{\mathbf{U}} \underbrace{\mathbf{Q}^*}_{\mathbf{V}^*}, \text{ kde } \mathbf{V}^{(0)} = \mathbf{Q} \mathbf{R}.$$

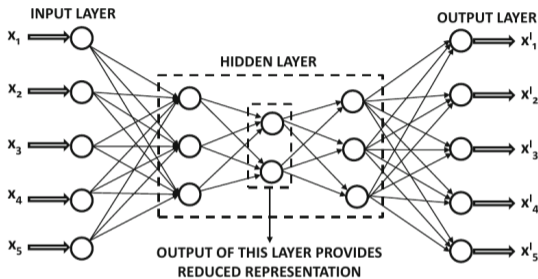
- Podmínky optimality jsou

$$\mathbf{X} \mathbf{V} - \mathbf{U}_\Sigma \underbrace{\mathbf{V}^* \mathbf{V}}_{=\mathbf{I}} = \mathbf{0} \Leftrightarrow \mathbf{X} = \mathbf{U}_\Sigma \mathbf{V}^*.$$

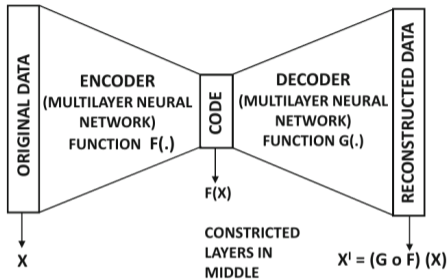
Autoenkodéry

- autoenkodér je neuronová síť, která má vstupní a výstupní vrstvu o stejné dimenzi
- úkolem autoenkodéru je replikovat data ze vstupu na výstup
- jako ztrátová funkce se tedy volí l_2 norma rozdílu mezi vstupními a výstupními daty
- autoenkodér by se tedy měl naučit identické zobrazení
- to není vůbec jednoduché i v případě, kdy budou mít i všechny vnitřní vrstvy stejnou dimenzi jako vstup a výstup
- problematické jsou nelinearity na jednotlivých neuronech
 - například ReLU zahazuje informace o záporných vstupech
 - sigmoid a tanh jsou sice teoreticky invertovatelné, ale numericky velmi špatně

Autoenkodéry



(a) Three hidden layers



(b) General schematic

Ch.C.Aggarwal, Neural Networks and Deep Learning.

- hlavní podstata autoenkodérů ale spočívá v tom, že vnitřní vrstvy lze udělat výrazně menší než vstupní a výstupní

Autoenkodéry

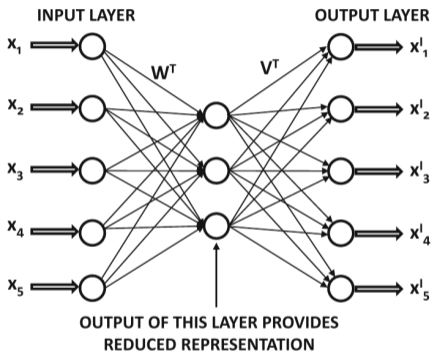
- autoenkodér je symetrický podle prostřední nejužší vrstvy
- první polovina se nazývá *encoder*
 - cílem enkoderu je provést kompresi vstupních dat na dimenzi prostřední vrstvy
- druhá polovina se nazývá *dekodér*
 - cílem dekodéru je provést rekonstrukci vstupních dat z informace na prostřední vrstvě
- enkodér by měl extrahovat nejpodstatnější informace ze vstupních dat tak, aby z nich dekodér mohl co nejlépe zrekonstruovat vstupní data

Autoenkodéry

- autoenkodér lze využít pro výpočet maticového rozkladu
- uvažujeme síť s pouze jednou vnitřní vrstvou a bez nelinearit
- efekt takové sítě pak lze zapsat jako součin

$$V^T W^T X$$

- ukážeme si souvislost s SVD rozkladem



Ch.C.Aggarwal, Neural Networks and Deep Learning.

Autoenkodéry

- na vstup autoenkodéru budeme zadávat řádky matice \mathbb{X} a na výstupu budeme požadovat to samé, tj.

$$\mathbb{V}^T \mathbb{W}^T \mathbb{X}^T = \mathbb{X}^T \Leftrightarrow \mathbb{X} \mathbb{W} \mathbb{V} = \mathbb{X}$$

- při trénování autoenkodéru budeme řešit úlohu

$$\min_{\mathbb{V}, \mathbb{W}} \|\mathbb{X} \mathbb{W} \mathbb{V} - \mathbb{X}\|^2$$

- řešením je $\mathbb{W} \mathbb{V} = \mathbb{I}$, tj. \mathbb{W} musí být pseudo-inverze matice \mathbb{V}

$$\mathbb{W} = \left(\mathbb{V}^T \mathbb{V} \right)^{-1} \mathbb{V}^T$$

- SVD rozklad matice \mathbb{X} lze psát jako

$$\mathbb{X} = \mathbb{U}\Sigma\mathbb{V}^T$$

- přenásobením maticí \mathbb{W}^T zprava dostáváme

$$\mathbb{X}\mathbb{W}^T = \mathbb{U}\Sigma\mathbb{V}^T\mathbb{W}^T = \mathbb{U}\Sigma(\underbrace{\mathbb{W}\mathbb{V}}_{=\mathbb{I}})^T = \mathbb{U}\Sigma$$

- tj. na vstupu prostřední vrstvy tedy dostáváme matici $\mathbb{U}\Sigma$

Autoenkodéry

- výhodou tohoto přístupu je, že lze snadno rozšířit o nelinearity
- SVD rozklad hledá lineární transformaci, která "odhaluje podstatu" dat
- autoenkodér dokáže hledat i nelineární transformace



2-D VISUALIZATION WITH
NONLINEAR AUTOENCODER



2-D VISUALIZATION WITH
PCA

Ch.C.Agarwal, Neural Networks and Deep Learning.

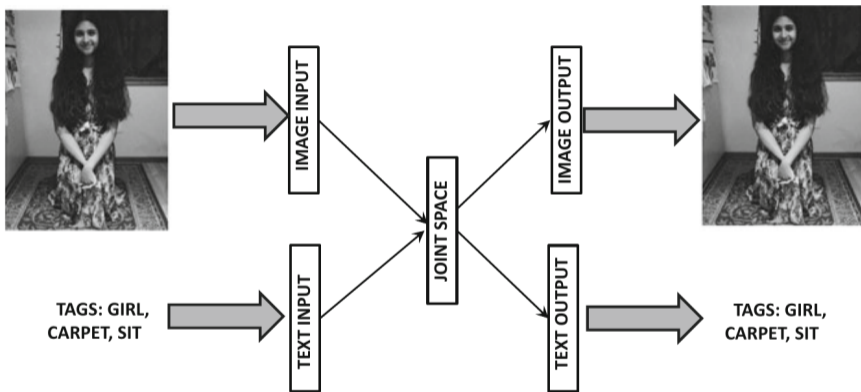
Autoenkodéry - odstranění šumu a detekce outlierů

- nelineární autoenkodéry lze použít např. pro odstraňování šumu nebo detekci outlierů
- obojí se špatně kóduje pomocí jednoduchých transformací
- za účelem odstranění šumu můžeme AE trénovat tak, že na vstup dáváme obrázek s přidaným šumem a na výstupu požadujeme původní obrázek bez šumu

Autoenkodéry - generování obrázků

- AE můžeme natrénovat na zpracování obrázků
- potom lze vzít pouze část odpovídající dekodéru
- pokud budeme dekodéru na vstup dávat různé vektory, bude generovat odpovídající obrázky
- k tomu se často používají tzv. variační AE (*variational autoencoders*)
- u nich nás zajímá hlavně prostřední skrytá vrstva, která může mít předem dané náhodné rozdělení, např. Gaussovské
- enkodér pak provádí transformaci vstupních dat na data s tímto rozdělením
- pokud pak generujeme náhodné vektory s Gaussovským rozdělením, dekodér k nim vytváří příslušné originály

Autoenkodéry - spojování multimodálních dat



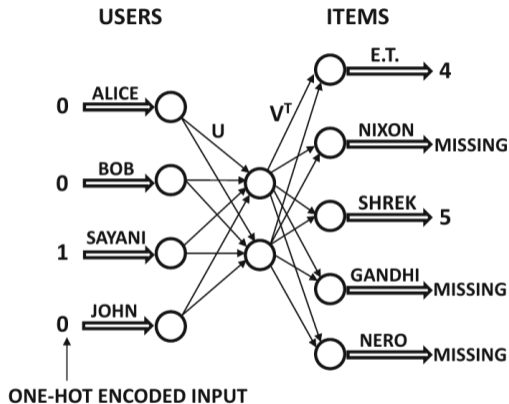
Ch.C.Agarwal, Neural Networks and Deep Learning.

Recommender systems

- uvažujeme úlohu, kdy máme matici $A \in R^{n,d}$
- pro n uživatelů a d produktů říká prvek a_{ij} , jak hodnotí uživatel i produkt j
- v praxi ale každý uživatel hodnotí jen několik málo produktů a proto je většina prvků nedefinovaných
- data tedy popisujeme formou trojic (*userId*, *itemId*, *rating*)
- autoenkodér dostává na vstupu *userId* ve formě *one-hot-encoding* vektoru
- výstupem je vektor o d složkách s hodnocením jednotlivých produktů

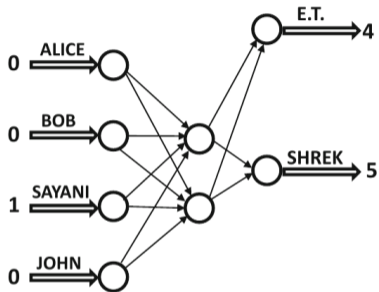
Recommender systems

- ztrátová funkce je definována jako l_2 norma chyby na výstupu
- komplikace je v tom, že pro některé výstupy není chyba definována
- proto děláme update pouze pomocí výstupů, které známe
 - při výpočtu gradientu ignorujeme výstupy, pro které chyba není definována
 - váhy, pro které není gradient definován, ignorujeme

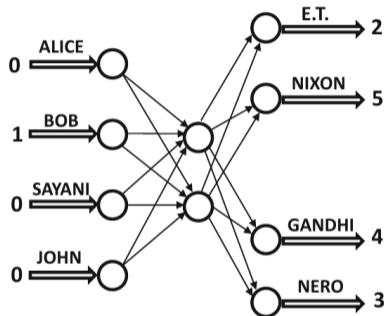


Ch.C.Aggarwal, Neural Networks and Deep Learning.

Recommender systems



OBSERVED RATINGS (SAYANI): E.T., SHREK



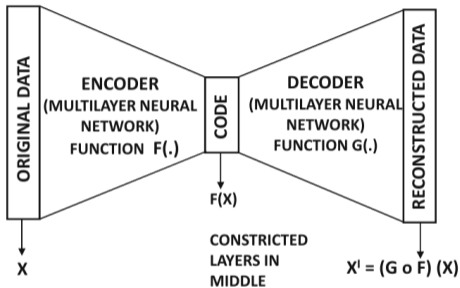
OBSERVED RATINGS (BOB): E.T., NIXON, GANDHI, NERO

Ch.C.Agarwal, Neural Networks and Deep Learning.

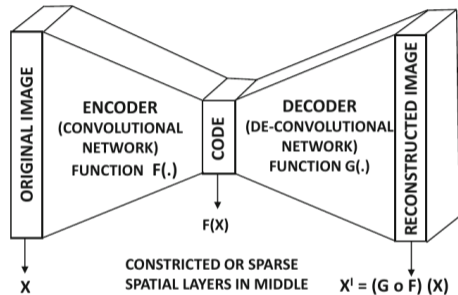
Recommender systems

- v každé iteraci tak vlastně trénujeme jen určitou část sítě
- to se někdy používá i při učení za pomoci kompletních dat pouze za účelem urychlení trénování
- díky tomuto postupu se tak dá napočítat maticový rozklad matice s nedefinovanými prvky

Konvoluční autoenkodéry



(a) A traditional autoencoder architecture

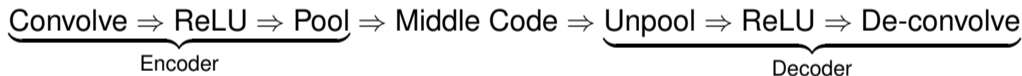


(b) Convolutional autoencoder

Ch.C.Aggarwal, Neural Networks and Deep Learning.

Konvoluční autoenkodéry

- jde o AE, kde obě části jsou konvoluční sítě
- prostřední vrstva pak může být buď plně propojená nebo také konvoluční
- architektura AE by měla být symetrická, tj. mělo by docházet k symetrickému střídání konvolučních vrstev, nelinearit a poolingů



- např. ReLU není možné přímo invertovat, tj. decoder není přesným opakem encoderu

Konvoluční autoenkodéry - komprese obrazových daty

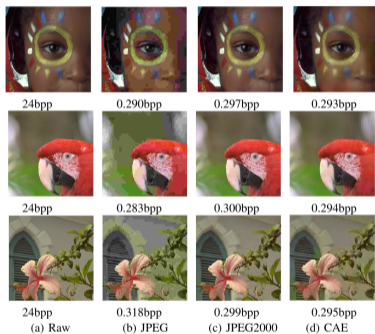
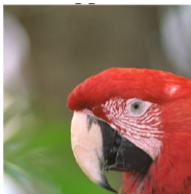
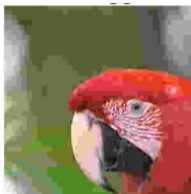


Fig. 6: Examples of raw image (a) and reconstructed images (300×300) cropped from Kodak images using (b)JPEG, (c)JPEG2000 and (d)CAE.

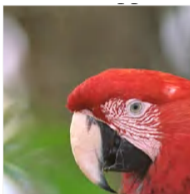
Konvoluční autoenkodéry - komprese obrazových daty



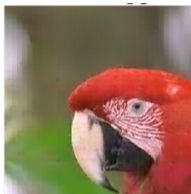
24bpp



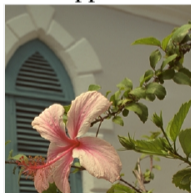
0.283bpp



0.300bpp

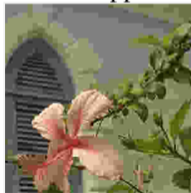


0.294bpp



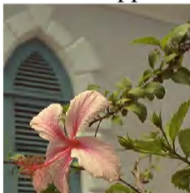
24bpp

(a) Raw



0.318bpp

(b) JPEG



0.299bpp

(c) JPEG2000



0.295bpp

(d) CAE

Konvoluční autoenkodéry - předtrénování

- hlavní potíží při aplikování CNN může být nedostatek otagovaných dat
- lze si pomoci předtrénováním sítě pomocí AE
- síť, kterou chceme trénovat si představíme jako enkodér a doplníme k ní symetrický dekodér
- vzniklý AE mohou trénovat pouze na samotných obrázcích, tj. bez supervize
- tímto přístupem si ale mohou velice dobře natrénovat počáteční vrstvy sítě, které se učí nižší úroveň abstrakce
- následně si vezmu jen enkodér a jako klasickou CNN ho dotrénuji na otagovaných snímcích, kterých už ale nyní potřebují výrazně méně