18NES1 Neuronové sítě 1 - Samoorganizace

18NES1 - 20. hodina, LS 2024/25

Zuzana Petříčková

28. dubna 2025

Co jsme probírali minule

Konvoluční neuronové sítě - dokončení

- Přenesené učení (transfer learning), fine-tuning
- Známé architektury konvolučních neuronových sítí
- Aplikace konvolučních neuronových sítí pro řešení různých typů úloh
- **Lehký úvod do dalších modelů hlubokého učení - zbyl na dnešek (viz minulé slidy)**

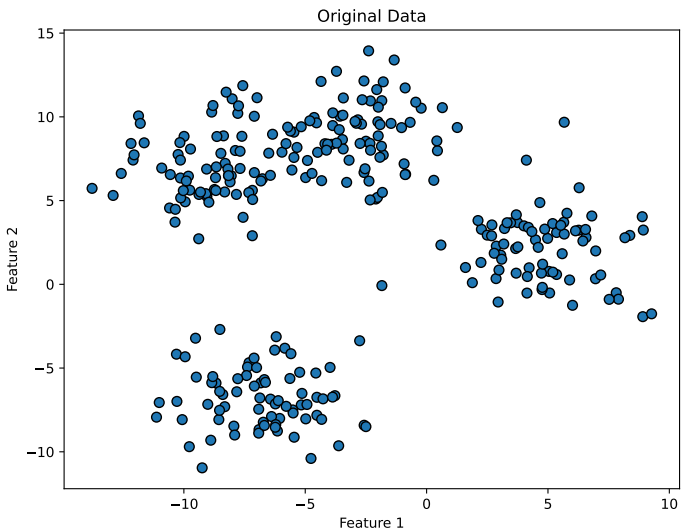
Tento týden

Učení bez učitele (samoorganizace, unsupervised learning)

- Obecně
- Klasifikace a Algoritmus k nejbližších sousedů
- Shlukování a Algoritmus k středů (k-means clustering)
- Ukázky a příklady

Učení bez učitele, unsupervised learning, samoorganizace

- trénovací množina T tvaru $T = \{\vec{x}_1, ..., \vec{x}_N\}$ (pouze vstupy)
- $\vec{x}_i \in R^n$ je (i-tý) trénovací vstupní vzor, požadovaný výstup neznáme
- **Myšlenka:** model sám rozhodne, jaká odezva je pro daný vzor nejlepší a podle toho nastaví své váhy $\rightarrow$ samoorganizace (self-organisation)

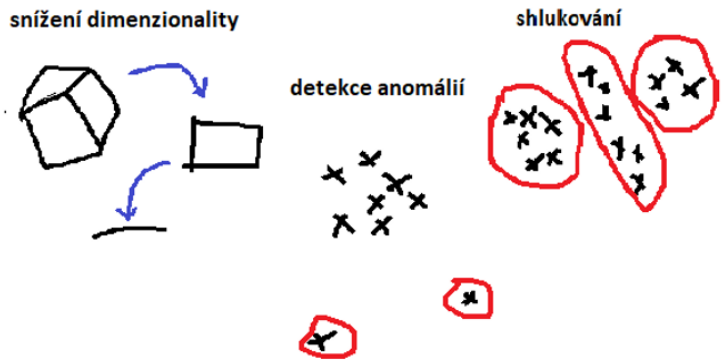


- máme data a neznáme jejich strukturu
- snažíme se strukturu a vlastnosti dat odhalit, najít v nich vzory, popř. strukturu

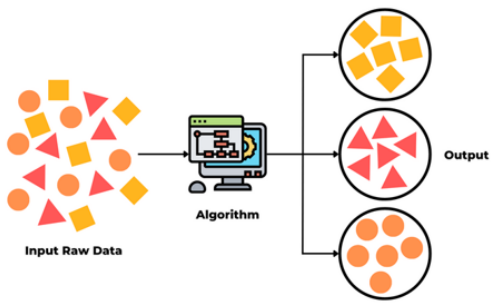
Učení bez učitele, unsupervised learning, samoorganizace

- **Cíl učení:** najít strukturu nebo vzory v datech
- **Aplikace: snížení dimenzionality** (komprese dat, vizualizace),
  detekce anomálií (např. v bankovních transakcích),
  shlukování (např. zákazníků podle chování, detekce pagiátů)
  e-komerce (doporučovací systémy)

Typy úloh:



Inspirováno: https://towardsdatascience.com/unsupervised-learning-algorithms-cheat-sheet-d391a39de44a



https://eastgate-software.com/what-is-unsupervised-learning/#ftoc-heading-7

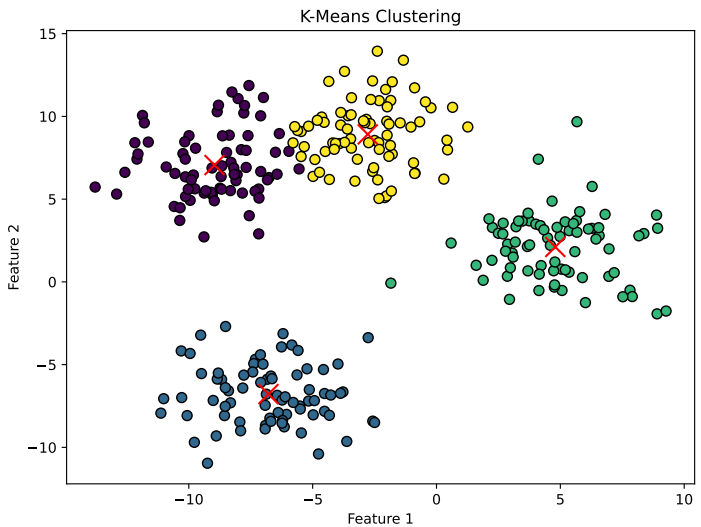
Učení bez učitele, unsupervised learning, samoorganizace

Shluk (klastr)

- Skupina vzorů s **velkou podobností mezi sebou** a **malou podobností se vzory s ostatních shluků**
- Zjednodušeně: **podobnost = vzdálenost**

Shlukování (klastrování)

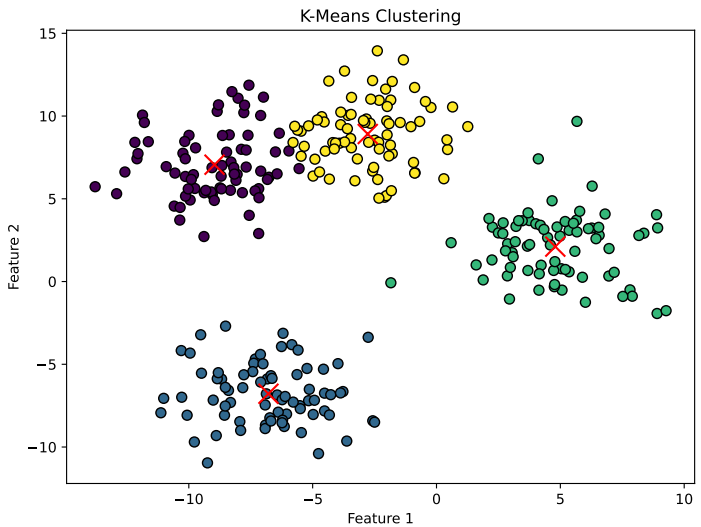
- Disjunktní rozdělení dat na shluky



Shlukování

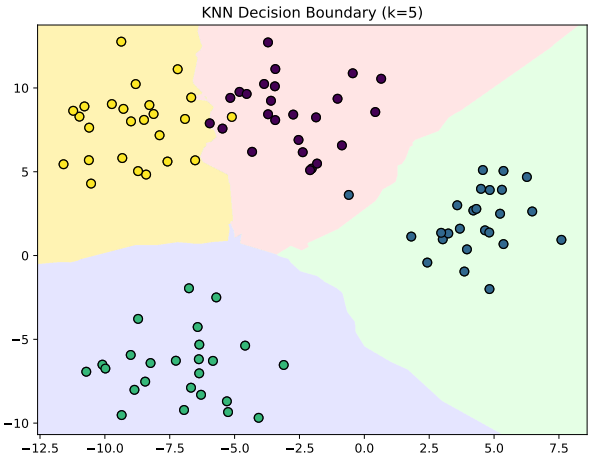
Problémy:

- Jak určit počet a rozložení shluků v příznakovém prostoru?
- Jak vybrat reprezentanta/y shluku?
  - Vhodně vybrané / všechny trénovací vzory patřící do shluku
  - např. střed shluku (*těžiště*, *centroid*)



Odbočka: Algoritmus k nejbližších sousedů

- Klasifikační metoda, učení s učitelem:
  Vzory z trénovací množiny jsou uloženy a klasifikovány do jedné z l různých tříd
- Neznámý vstupní vektor je zařazen do té třídy, ke které patří většina z k nejbližších vektorů z uložené množiny



- Nejjednodušší varianta: 1 nejbližší soused,
  klasifikační model = uložená data

Odbočka: Algoritmus k nejbližších sousedů

Jak počítat vzdálenost (podobnost) číselných vektorů?

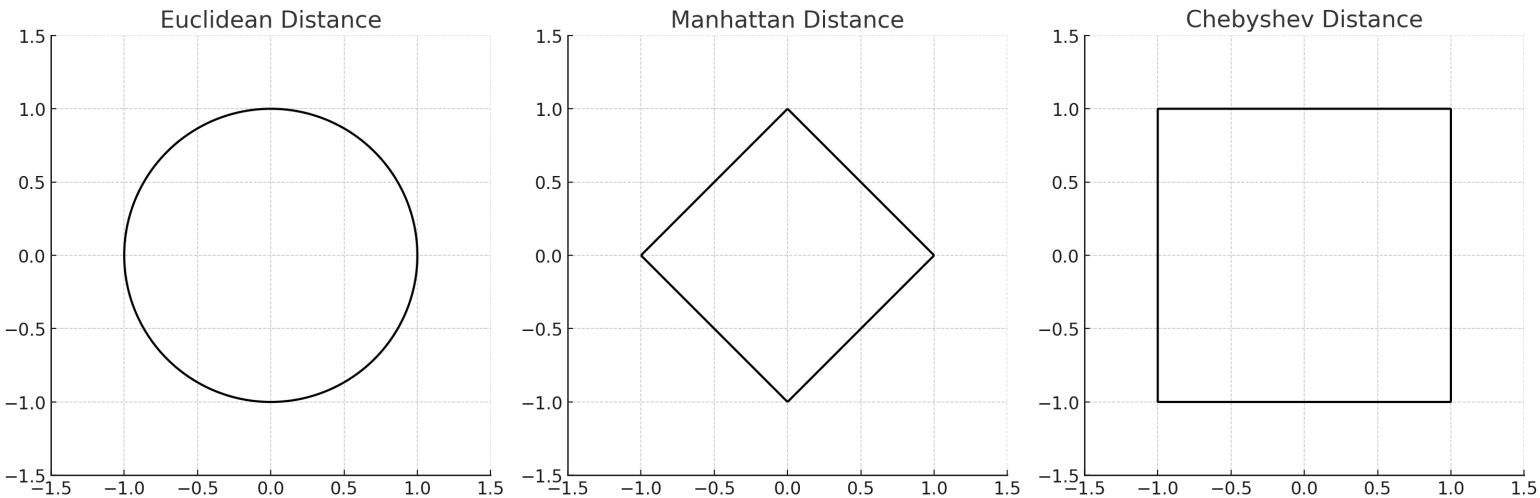
- **Euklidovská vzdálenost:** $d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$
- pokud jen porovnáváme vzdálenosti: $d(\vec{p}, \vec{q}) = \sum_{i=1}^{n}(p_i - q_i)^2$

Ale existují i alternativní metriky, např.:

- **Manhattan (městská) metrika**: $d(\vec{p}, \vec{q}) = \sum_{i=1}^{n} |p_i - q_i|$
- **Čebyševova metrika** $d(\vec{p}, \vec{q}) = \max_i |p_i - q_i|$
  ... **„Co je největší problém?"**
- **Minkewského metrika:** $d(\vec{p}, \vec{q}) = (\sum_{i=1}^{n} |p_i - q_i|^r)^{\frac{1}{r}}$
  ... zobecnění předchozích metrik $(r = 2, 1, \to \infty)$
- **Kosínová podobnost:** $\cos(\vec{p}, \vec{q}) = \frac{\vec{p} \cdot \vec{q}}{\|\vec{p}\| \|\vec{q}\|}$
  ... nezajímá nás velikost vektorů, ale jejich směr (např. zpracování textu)
- (a další)

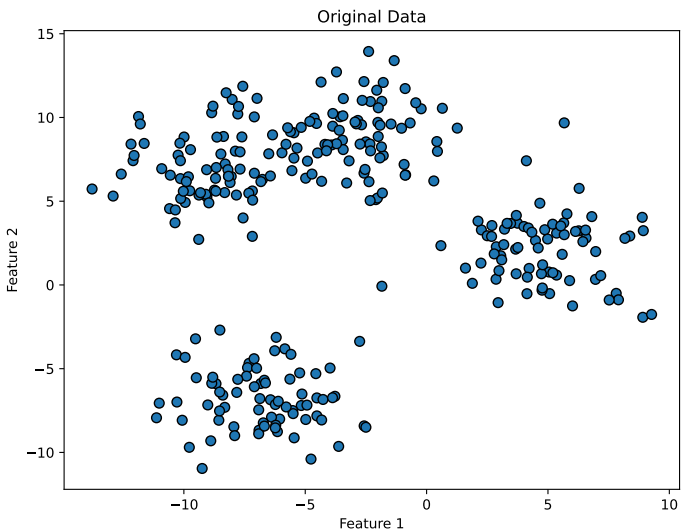
Odbočka: Algoritmus k nejbližších sousedů

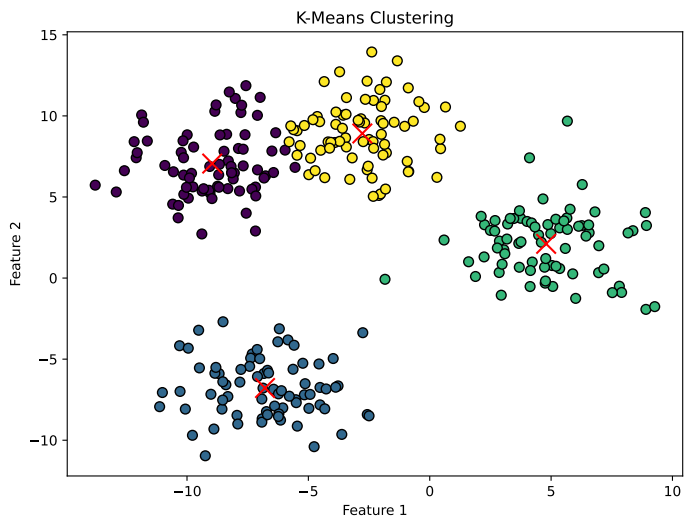
Jak počítat vzdálenost (podobnost)?



Algoritmus k středů (k-means clustering)

- Učení bez učitele
- Vstupní vzory jsou klasifikovány do k různých shluků, každý shluk I je reprezentován svým centroidem (středem, těžištěm) $\vec{c_I}$
- Nový vektor $\vec{x}$ je zařazen k tomu shluku i, jehož centroid $\vec{c_I}$ je mu nejblíže





Algoritmus k středů (k-means clustering)

1. Je dána trénovací množina $T = \{\vec{x}_1, ..., \vec{x}_N\}, \vec{x}_i \in R^n$
2. Zvolíme k náhodných vektorů $\vec{c}_l, l = 1, ..., k$ (z R^n nebo z T) za středy (centroidy) shluků
3. Opakujeme:
   - Přiřadíme každý vektor z T k nejbližšímu středu shluku
   - Přepočítáme středy shluků na základě přiřazených vzorů:

$$\vec{c}_l = \frac{1}{n_l} \sum_{l_i=1}^{n_l} (\vec{x}_{l_i})$$

   n_l ... počet vektorů přiřazených k l-tému shluku
   l_i ... indexuje vektory přiřazené k l-tému shluku
   - Předchozí dva kroky opakujeme, dokud se mění příslušnost trénovacích vzorů ke shlukům

Parametry algoritmu k-means

- Počet shluků k
- Metrika vzdálenosti - jak počítat vzdálenost (podobnost) vektorů?
- Inicializační metoda - jak inicializovat centroidy?
- Kritéria ukončení - kdy algoritmus ukončit?

Parametry algoritmu k-means

Inicializace centroidů

- Výsledek k-means závisí na počáteční volbě centroidů.
- Špatná inicializace $\rightarrow$ špatné shluky, pomalá konvergence, uvíznutí v lokálním minimu.
- Možnosti inicializace:
  - Náhodné vektory v prostoru R^n nebo z rozsahu vzorů
  - Náhodný výběr bodů z trénovací množiny T
  - **k-means++**:
    - první centroid se volí náhodně,
    - další centroidy se vybírají s pravděpodobností úměrnou čtverci vzdálenosti od nejbližšího již zvoleného centroidu.
  - Vícenásobné spuštění algoritmu a výběr nejlepšího řešení (nejnižší suma čtverců vzdáleností)

Parametry algoritmu k-means - shrnutí

- Počet shluků k (obvykle zadán uživatelem)
- Metrika vzdálenosti (standardně Euklidovská)
- Inicializační metoda (random, k-means++, vlastní volba)
- Kritéria ukončení:
  - dokud se mění příslušnost vzorů
  - dokud se mění centroidy
  - dosažení maximálního počtu iterací
- Další vylepšení: Pokud k nějakému centroidu není přiřazen žádný vzor $\rightarrow$ inicializujeme ho znova

Příklad (ukázka)

kmeans_clustering.ipynb,

- Ukázka vlastní implementace k-means algoritmu včetně vizualizace průběhu učení
- Několik datových sad, různé možnosti inicializace vah

Otázky:

- Jaký má na učení vliv inicializace centroidů?
- Jak dlouho trvá učení pro větší data?
- Jak najít optimální počet shluků?
- Jak zhodnotit, zda jsou vytvořené shluky kvalitní?

Algoritmus k středů (k-means clustering)

Výhody

- Rychlý algoritmus, jednoduchý na implementaci
- Vhodný pro rychlý náhled na strukturu dat

Nevýhody

- Nutnost zvolit počet shluků předem
- Dávkové zpracování (problém pro velká data nebo online učení)
- Vysoká citlivost k počáteční volbě centroidů *(... obrázek)*
- Citlivost k odlehlým vzorům
- Pro složitá data nemusí být úspěšný: vyhledává sférické shluky *(... obrázek)*
- Problém, pokud je vysoká dimenze vstupních dat (*prokletí dimenzionality*), popř. navzájem silně korelované příznaky