

STU - vypracované otázky

Obsah

Seznam otázek	2
1. Prokletí dimenzionality (Curse of dimensionality) + metody pro řešení	3
2. Bootstrap, použití, princip a vzoreček + Co je to resubstituční chyba	4
3. Bootstrap a Crossvalidation - rozdíly, vzorce, kdy se která metoda používá	5
4. Meta učení: princip + bagging, stacking	7
5. K-means (co to je, jestli je hierarchická a deterministická, proč/proč ne?) + vzorce	9
6. Vzorec pro entropie, informační zisk + výpočet, ID3	11
7. Dopředná a zpětná selekce, rozdíly a k čemu jsou.	14
8. Základní 3 operátory genetických algoritmů a popsat je	15
9. GAN bloková schéma + hlavní (ten mrtě dlouhý) vzorec + popsat, jak se probíhá učení	16
10. Co je kovariance, popsat matematicky, vybrat kovar. matice z uvedených 4 a oduvodnit, proc nejsou kovarianci, ktere nejsou.	18
11. Učení s ucitelem: scéma + popsat + lossfunction, chybová funkce, nákladová matice	20
12. Popsat normalizaci. K čemu se používá + vzorec pro normalizaci se střední hodnotou.	24
13. 3 aktivační(přenosové?) funkce + vzorec. ReLU a její úprava (Leaky ReLu)	24
14. Vzorec pro accuracy, specificity, senzitivity a F1 míru. Vypočítat pro 2 dvojpolové tabulky. Popsat který model je lepší na co. (ten příklad z přednášky)	27
15. Rozhodovací stromy, popsat strukturu RS (kořenový uzel, uzel, větev, list, hloubka stromu), vysvětlit pojem atribut, hodnota atributu, klasifikační atribut, záznam, napsat vzorec pro entropie a napsat algoritmus učení RS	29

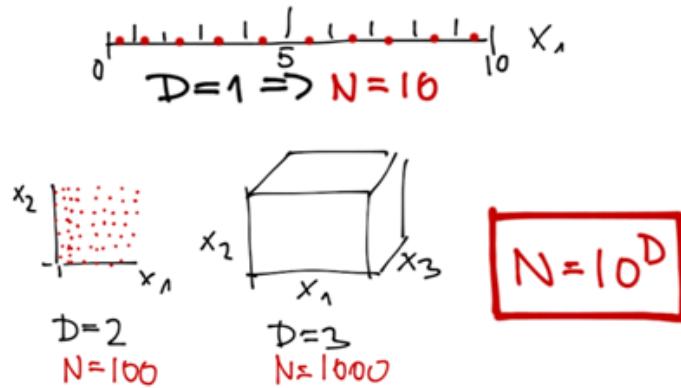
1. Prokletí dimenzionality (Curse of dimensionality) + metody pro řešení

Obtížnost učení roste exponenciálně s počtem dimenzí. Příklad viz slajdy. Možnosti řešení: selekce příznaků na ty, které jsou relevantní. Případně přímo selektovat příznaky při extrakci.

Učení založené na instancích snímky 18 a 19

a)

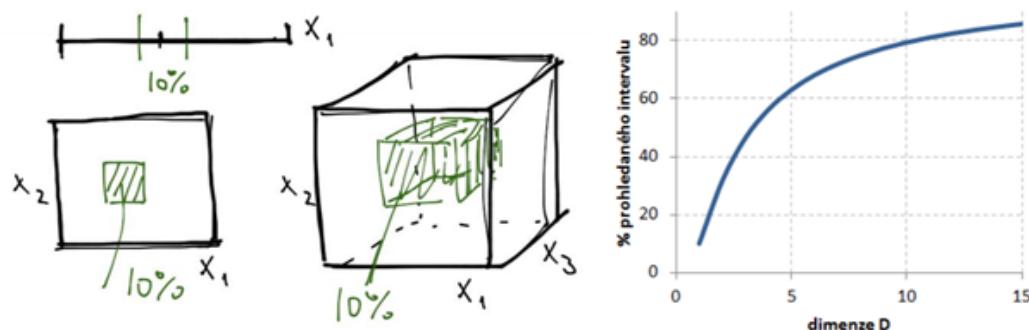
- Vstupní veličina x_1 je definována na intervalu $<0;10>$.
- Chci mít uloženo N prvků, které budou vstupní prostor dostatečně charakterizovat – na jejich základě budu klasifikovat nové prvky.
- Rozhodl jsem se, že mi bude stačit 10 prvků rovnoměrně rozložených na veličině x_i (v hodnotách 0,5; 1,5; ...; 9,5).
- Kolik prvků potřebuju při $D=1$ (D je počet vstupních veličin), $D=2$, $D=10$?



18

b)

- Chci klasifikovat nový prvek tak, že **prohledám 10%** definičního oboru definovaného M -rozměrnou krychlí
- Kolik procent z intervalu na každé vstupní veličině x_i budu prohledávat při $D=1$, $D=2$, $D=3$, $D=10$?



Při $D=1$ je třeba prohledat **10%** intervalu každé veličiny.

Při $D=2$ je třeba prohledat **32%** intervalu každé veličiny.

Při $D=3$ je třeba prohledat **46%** intervalu každé veličiny.

19

2. Bootstrap, použití, princip a vzoreček + Co je to resubstituční chyba

Použití: zejména v případech nedostatku dostupných dat

Princip: metoda rozdělí data na trénovací a testovací. Je vygenerován velký počet trénovacích souborů B_i o četnosti N prvků **výběrem s opakováním** ze základního souboru všech N dostupných dat.

Odhad přesnosti modelu snímky 8 a 9.

Výpočet (tradiční bootstrap):

- **tradiční bootstrap:**

$$\hat{Err}_{boot} = \frac{1}{N} \frac{1}{|B|} \sum_{j=1}^{|B|} \sum_{i=1}^N LF(y_i, \tilde{f}^{B_j}(x_i))$$

kde \tilde{f}^{B_j} je model naučený na B_j -tý výběr, testuje se na původním souboru

Výpočet (0,632 bootstrap; přesnější):

$$\hat{Err}_{Btest} = \frac{1}{|B|} \sum_{j=1}^{|B|} \frac{1}{|C_j|} \sum_{i=1}^{|C_j|} LF(y_i, \tilde{f}^{B_j}(x_{C_j}))$$

kde C_j je množina všech prvků neobsažených ve výběru B_j

- odhad celkové chyby modelu metodou 0,632 bootstrap je:

$$Err = 0,632 \cdot Err_{Btest} + 0,368 \cdot Err_{boot}$$

- **Resubstituční chyba** – chyba zjištěná na datech použitých na trénování – vede k podhodnocení skutečné chyby

3. Bootstrap a Crossvalidation - rozdíly, vzorce, kdy se která metoda používá

Tabulka hlavních rozdílů mezi metodami:

	Cross Validation	Bootstrap
Pozitivní	<ul style="list-style-type: none">• nepřekrývání testovacích dat• jednoduché• $K=10$ – snížená výpočetní náročnost	<ul style="list-style-type: none">• lze použít při malém počtu dat• statisticky zajímavé pro intervalové odhadы charakteristik datových souborů
Negativní	<ul style="list-style-type: none">• nejednoznačné stanovení velikosti K• požaduje více dat	<ul style="list-style-type: none">• chyba překrytím trénovacích a testovacích dat (resubstituční chyba)• výpočetně náročné

Vlastnosti Bootstrap viz otázka 2.

Odhad přesnosti modelu snímky 11 až 22

Princip CV:

- CV je metoda sloužící k odhadu skutečné chyby modelu, tedy k posouzení hypotézy, do jaké míry data odpovídají danému modelu.
- Princip spočívá v tom, že je datový soubor rozdělen na určitý počet pokud možno stejně velkých disjunktních množin K . Na základě tohoto dělení je K -krát nastaven a vyhodnocen model tak, že je postupně vždy jedna množina použita jako testovací a sjednocení ostatních množin jako trénovací soubor dat. Je tak získáno K různě nastavených modelů.
- Součet všech vypočtených odchylek slouží k určení skutečné chyby modelu vytvořeného na základě použitých dat.

Vzorec pro K=10 nebo K=5

- Celková chyba modelu metodou Err_{CV} je dána průměrem chyb Err všech dílčích modelů:

$$\text{Err}_{\text{CV}} = \frac{1}{K} \sum_{i=1}^K \text{Err}\left(y_{\kappa(i)}, \tilde{f}^{-\kappa(i)}(x_{\kappa(i)})\right)$$

kde K je počet podmnožin vytvořených z úplného datového souboru, $\kappa(i)$ je i -tá podmnožina, $y_{\kappa(i)}$ a $x_{\kappa(i)}$ jsou výstupní a vstupní data obsažená v podmnožině $\kappa(i)$ a $\tilde{f}^{-\kappa(i)}$ je model nastavený bez použití podmnožiny $\kappa(i)$.

Vlastnosti Cross Validation:

- použití
 - srovnání více přístupů, výběr nejlepšího (různé typy modelů, různá nastavení jednoho typu modelu)
 - stanovení předpokládané přesnosti modelu (průměrem parametrů jednotlivých modelů, použití stejné metodiky pro všechna dostupná data)
- vlastnosti
 - výhody: vyšší přesnost, kompromis k -fold má výhody přesnosti (one-leave-out) a zároveň rozumné výpočetní náročnosti
 - nadhodnocení chyby, u one-leave-out velký rozptyl odhadu, časová náročnost

Příklad CV vs. Bootstrap

	$\text{Err}_{\text{train}}$	$\text{Err}_{\text{Boots}}$	Err_{632}	Err_{REAL}	Err_{CV}	$\text{Err}_{\text{Btest}}$
gen1	0,210	0,237	0,284	0,287	0,312	0,311
gen2	0,100	0,112	0,156	0,343	0,133	0,174
Titanic	0,300	0,349	0,408	0,302	0,433	0,442

Z uvedených experimentů je zřejmé, že k odhadům chyby je třeba přistupovat obezřetně, jejich **přesnost je velice citlivá** na kvalitu výběru (z chybných dat korektním postupem dobrý model nikdy nezískáte!). V příkladech je pracováno s výběrem 30 záznamů, takže zkreslení je velké. Použití CV není s tímto výběrem smysluplné (použito pro ukázku růstu odhadu chyby).

21

4. Meta učení: princip + bagging, stacking

Cílem je k predikci použít více modelů

Meta učení snímky 1 až 9

Základní parametry k určení: použití trénovacích dat, typ modelů, jak budeme klasifikovat

Bagging – princip:

Vytvořit M množin z trénovacích dat metodou bootstrap. Z těchto dat naučit M modelů stejného typu. Při klasifikaci použít třídu, na které se shodlo nejvíce modelů (v případě regrese použít průměrnou hodnotu). Modely mají stejnou váhu. Princip založen na teorii bias-variance dekompozice.

- použití v případě **nedostatku dat**
- zvyšuje **stabilitu a přesnost** (snižuje **varianci** a riziko **přeucení**)
- typická aplikace na nestabilní modely (např. rozhodovací stromy, NN)
- předností RS je jejich interpretovatelnost, která se však použitím baggingu ztrácí
- nepoužívá se na lineární (velký bias, silná generalizace) a robustní (IBL) modely – nemá efekt
- existuje i váhová varianta MetaCost (pokud výstupy modelu mají pravděpodobnostní charakter – Bayes, NN)

8

Tvorba modelu:

Z N trénovacích dat vybrat náhodně N instancí výběrem s opakováním (bootstrap). Zopakovat M -krát; pokaždé jinou N -tici prvků.

Učení probíhá tak, že se vybere náhodně s opakováním N prvků, naučí se model a uloží jeho nastavení. Opakovat M -krát. Na konci je M modelů stejného typu s rozdílným nastavením.

Stacking:

- učíme model, který kombinuje **libovolné typy modelů** nižší úrovně
- na vyšší úrovni učíme zpravidla jednoduchý (např. lineární) model
- je výhodné, když model nižší úrovně kromě predikce udává také míru důvěryhodnosti své predikce (logitový model, NN, Bayes, ...)
- trénovacích data určena **výběrem typu Cross-Validation**, což je výpočetně náročné
- variantou je označení dílčích modelů **váhou w_i** , kterou je **násobena predikce každého modelu nižší úrovně**
- modely nižší úrovně, **úroveň 0**, jsou učeny vždy na základě stejné množiny trénovacích dat, jejich predikcí na testovacích datech získáme vstupní hodnoty do modelu **úrovně 1** (metalearner). Jeho struktura je jednoduchá (lineární, RS). Počet vstupů do meta-modelu odpovídá počtu modelů úrovně 0, nebo jeho násobku počtem tříd (když je výstupem nižšího modelů pravděpodobnost klasifikace do každé ze tříd)
₉
- cílem je nastavit váhy w_i pro M modelů f_i nižší úrovně tak, aby byla minimalizována chyba LF od požadovaného výsledku
- *test* představuje množinu testovacích dat. Zápis f^{-t} znamená model nastavený na základě trénovacích dat (bez t testovacích), x^{+t} testovací data, $|test|$ počet dělení pomocí Cross-Validation

$$w = \arg \min_w \sum_{t=1}^{|test|} LF\left(y^{+t}, \sum_{j=1}^M w_j \cdot f_j^{-t}(x^{+t})\right)$$

- ideálně jsou váhy nastaveny tak, aby jejich součet byl roven 1

5. K-means (co to je, jestli je hierarchická a deterministická, proč/proč ne?) + vzorce

Je nehierarchická, nedeterministická metoda, protože pokaždě vede k jinému výsledku. To je způsobeno tím, že na začátku algoritmu jsou centroidy zpravidla zvoleny náhodně.

Shluková analýza snímky 20 až 23

- při vytváření malého počtu shluků z velkého počtu dat
- pro data kvantitativní bez odlehlých hodnot (je třeba normalizovat jednotlivé veličiny)
- pokaždě vede k jinému řešení
- rychle iteruje i při velkém počtu dat, nalezení zpravidla lokálního optima
- vhodné pro větší počet dat

K-means – princip

- dánou: data X , počet shluků K (cíleně, náhodně, heuristiky)
- cíl: nalezení K shluků tak, že mezishluková suma čtverců bude minimalizována
- princip
 - určení počátečních K charakteristických vektorů μ
 - v cyklu opakuj:
 - podle charakteristického vektoru μ přiřaď všem bodům jejich třídu
 - spočítej z bodů jednoho shluku jejich nový charakteristický vektor μ podle těžiště nebo průměru
 - ukončovací podmínka:
 - konec, pokud 1 (nebo 2) nová iterace nezpůsobí změnu klasifikace ani jednoho z prvků

K-means – algoritmus

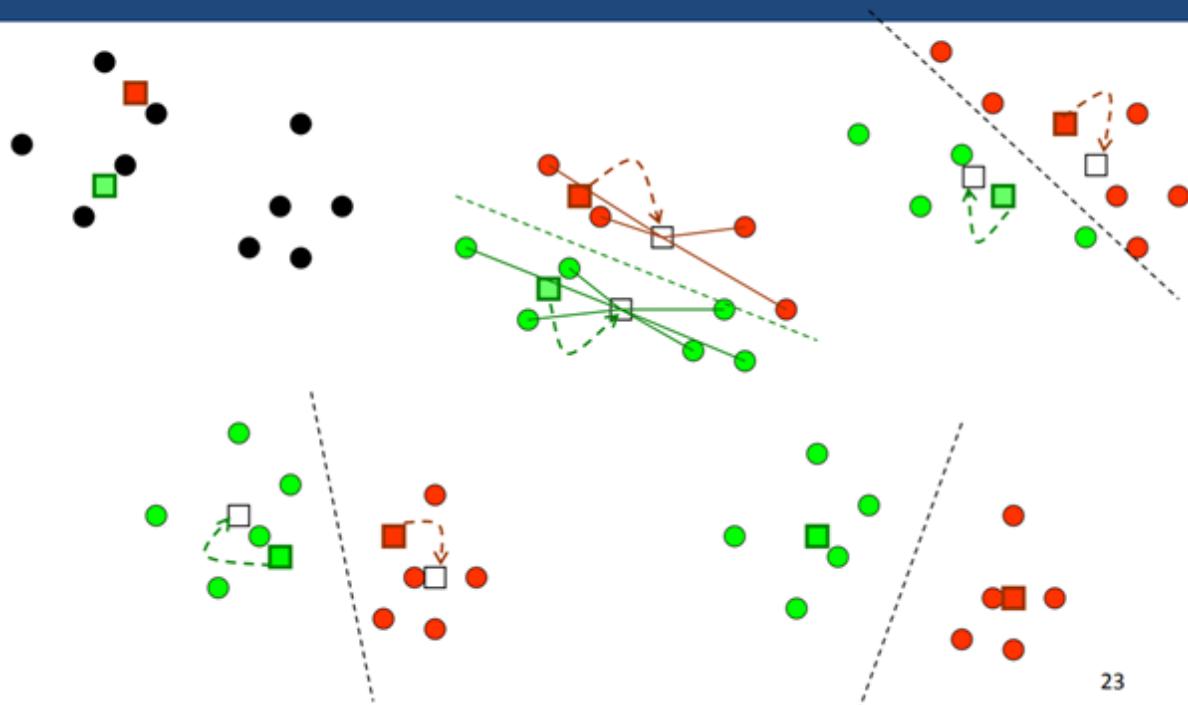
- K shluků, vstupní data x_1, \dots, x_N
- 1. Počáteční určení K center (náhodné, apriorní znalost, těžiště všech bodů a extrémy, atd.) – μ_j
- 2. Klasifikuj do tříd C_1, \dots, C_K podle minima euklidovské vzdálenosti od vektoru μ_j
- 3. Opakuj v cyklu
 - každému prvku x_i je přiřazena klasifikace g_i

$$g_i = \arg \min_{j=1, \dots, K} \|x_i - \mu_j\|$$

- přeypočítej vektory μ_j

$$\mu_j = \frac{1}{|C_j|} \sum_{i=1}^N \begin{cases} x_i, & g_i = C_j \\ 0, & g_i \neq C_j \end{cases}$$

K-means – příklad



23

6. Vzorec pro entropie, informační zisk + výpočet, ID3

Teorie informace
Rozhodovací stromy

Entropie

- **střední hodnota informace** $I(X)$ získaná zjištěním konkrétní hodnoty X

$$H(X) = E[I(X)] = - \sum_{\forall x \in \mathcal{X}} p(x)I(x) = - \sum_{\forall x \in \mathcal{X}} p(x) \log p(x)$$

- Je-li $x = [a, b, b, a, c, a]$, pak

$$\begin{aligned} H(x) &= E[I(x)] = - \sum_{\forall x \in \mathcal{X}} p(x) \log p(x) = \\ &= -\frac{3}{6} \log \frac{3}{6} - \frac{2}{6} \log \frac{2}{6} - \frac{1}{6} \log \frac{1}{6} = \frac{3}{6} 1 + \frac{2}{6} 1,58 + \frac{1}{6} 2,58 = 1,46 \end{aligned}$$

Sdružená a podmíněná entropie

- sdružená entropie je definována přímo z matice sdružené pravděpodobnosti:

$$H(X, Y) = - \sum_{\forall x \in X} \sum_{\forall y \in Y} p(x, y) \log p(x, y)$$

- podmíněná pravděpodobnost je definována jako:

$$p(x|y) = \frac{p(x, y)}{p(y)}$$

- z čehož lze dosazením získat podmíněnou entropii:

$$H(X|Y) = - \sum_{\forall x \in X} \sum_{\forall y \in Y} p(x, y) \log p(x|y) = - \sum_{\forall x \in X} \sum_{\forall y \in Y} p(x, y) \log \frac{p(x, y)}{p(y)}$$

Vzájemná informace

- vzájemná informace

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- je odvozena z kombinací předešlého jako:

$$-\sum_{\forall x \in X} \sum_{\forall y \in Y} p(x, y) \log p(x) + \sum_{\forall x \in X} \sum_{\forall y \in Y} p(x, y) \log \frac{p(x, y)}{p(y)} =$$

$$\sum_{\forall x \in X} \sum_{\forall y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

- Poznámka: jsou-li veličiny X a Y nezávislé, potom platí, že $p(x, y) = p(x)p(y)$. Podíl těchto pravděpodobností je roven 1 a $\log 1 = 0$. Tedy pro dvě nezávislé veličiny bude vzájemná informace rovna 0.

ID3 – základní údaje

- Klasifikace - **nominální** vstupy a výstupy
- Větvení podle **výčtu dělicího atributu**
- Kvalita dělení posouzena **entropií** – ML
- Předností je schopnost vybrat z velkého množství atributů ty vhodnější, **vždy** vytvořen **stejný strom** (deterministický)
- **Není garance** vygenerování optimálního stromu

Tvorba stromu pomocí ID3

1. Informace o uzlu: $H(S)$, počet prvků
2. Rozhodni, zda dělit
3. Vypočítej průměrnou neuspořádanost dosud nepoužitých atributů – $H(S|A_i)$
4. Vyber atribut s nejnižším $H(S|A_i)$

Příklad ze slajdů spálí se / nespálí se:

The image shows a handwritten derivation of the formula for information entropy $H(S/\text{barva})$. The formula is given as:

$$H(S/\text{barva}) = \frac{\text{počet řádků}}{N} \left(-\frac{\text{blond}}{\text{počet řádků}} \log_2 \frac{\text{blond}}{\text{počet řádků}} - \frac{\text{blond \& spálí}}{\text{počet řádků}} \log_2 \frac{\text{blond \& spálí}}{\text{počet řádků}} \right) + \frac{\text{počet hnědých řádků}}{N} \left(\dots \right)$$

Below the formula, there is handwritten text explaining the calculation:

Proč nuly: 0

$$\frac{3}{8} \left(-\frac{3}{3} \log_2 \frac{3}{3} - \frac{0}{3} \cdot \log_2 \frac{0}{3} \right)$$

To samý se zrzavýma vlasama

N - počet řádků / dat

7. Dopředná a zpětná selekce, rozdíly a k čemu jsou.

(PŘEDZPRACOVÁNÍ DAT)

Slouží k výběru vícerozměrných veličin, je to typ Wrapper (používají během selekce skutečný model, přesnější, přesnost pomocí accuracy, pořadí výběru veličiny odpovídá pořadí její relevance, výpočetně náročný)

Dopředná selekce:

- je dáno $|A|$ atributů a kritérium kvality predikce $C(A_1, \dots, A_k)$
- kritériem kvality může být např. nějaký typ modelu společně s metodou odhadu chyby (např. 10fold-Cross-validation)
- Výběr k veličin:
 - vypočti kritérium C pro všechny jednotlivé atributy $|A|$ a vyber jeden nejlepší
 - vypočti kritérium C pro všechny dvojice tvořené nejlepším atributem z předešlého kroku a jedním dalším, vyber pář s největším C
 - pokračuj s přidáváním atributů až bude vybráno k veličin
- máme-li např. 40 veličin a chceme vybrat nejlepších 10, je zapotřebí provést pouhých 3.550 výpočtů
- nalezení optimální kombinace není garantováno

Zpětná selekce:

- je dáno $|A|$ atributů a kritérium kvality predikce $C(A_1, \dots, A_k)$
- kritériem kvality může být např. nějaký typ modelu společně s metodou odhadu chyby (např. 10fold-Cross-validation)
- Výběr k veličin:
 - vypočti kritérium C pro všechny atributy $|A|$
 - vypočti kritérium pro všechny kombinace s $|A|-1$ veličinami a vyber tu s největší hodnotou
 - pokračuj odebíráním až zbude k veličin
- máme-li např. 40 veličin a chceme vybrat nejlepších 10, je zapotřebí provést pouhých 7.760 výpočtů
- nalezení optimální kombinace není garantováno

8. Základní 3 operátory genetických algoritmů a popsat je

Genetické algoritmy slouží k prohledávání prostoru hypotéz (např. nastavení modelu) napodobováním přirozeného vývoje v přírodě (evoluce) - přežívá přizpůsobenější jedinec

Přednosti:

- lze řešit libovolně složitý problém bez apriorní znalosti

Omezení:

- pokaždé nalezeno jiné řešení, nemožnost rozpoznat optimum, ne vždy dostatečně přesné, řada parametrů algoritmu - stupňů volnosti

• Operátory

- kódování
- fitness (kvalita)
- selekce
- křížení
- mutace
- elitismus
- ukončovací podmínka

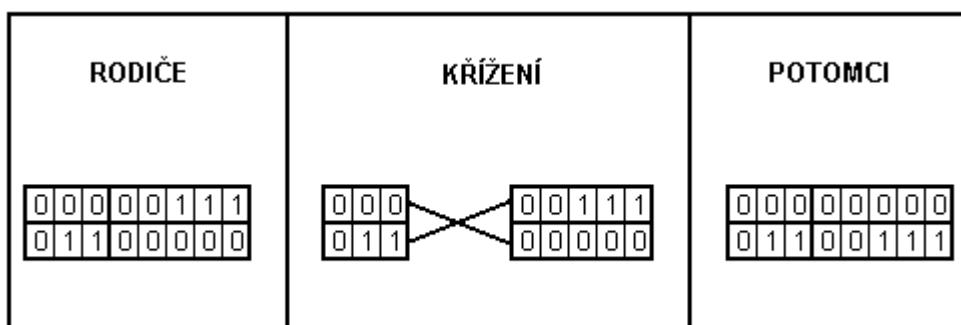
Operátorů je tam popsaných víc, ale tipuji, že ty nejdůležitější by mohly být selekce, křížení a mutace?

Křížení

- jedinec = křížení (rodič1, rodič2);
- základní "evoluční" operátor GA
- na základě křížení může ze dvou slabých rodičů vzniknout silný potomek
- různé typy křížení podle typu kódování chromozomu (bin., real.)

Binární kódování

8 genů, 7 pozic pro křížení, 3. pozice křížení



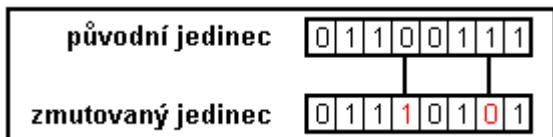
Reálné kódování

- Simple crossover, Convex crossover, Blend crossover- α (BLX- α)

Mutace

- jedinec = mutace(jedinec);
- dochází k ní zřídka
- změna genu "ladící" konečnou kvalitu
- dělení na 2 typy podle typu kódování chromozomu (bin., real.)

Binární kódování



Reálné kódování

- změna o předem danou hodnotu, o náhodnou hodnotu
- nahrazení náhodným číslem z daného intervalu
- dynamická mutace (Michalewicz), mění svou velikost v čase

Selekce

- rodiče_generace_N = selekce(generace[N-1]);
- na základě selekce jsou vybrání jedinci, kteří se stanou rodiči
- 3 základní typy selekce - vážená ruleta, poziční selekce, metoda TURNAJ

Vážená ruleta - lokální extrém

Poziční selekce - modifikace rulety, práce s pořadím (ordinalita)

Metoda TURNAJ - Vyber náhodně k jedinců s pravděpod. p vyber nejlepšího, s pravděpod. p(1-p) druhého nejlepšího až po k-1 (poslední z k-tice má p rovnou doplňku součtu předešlých psí do 1).

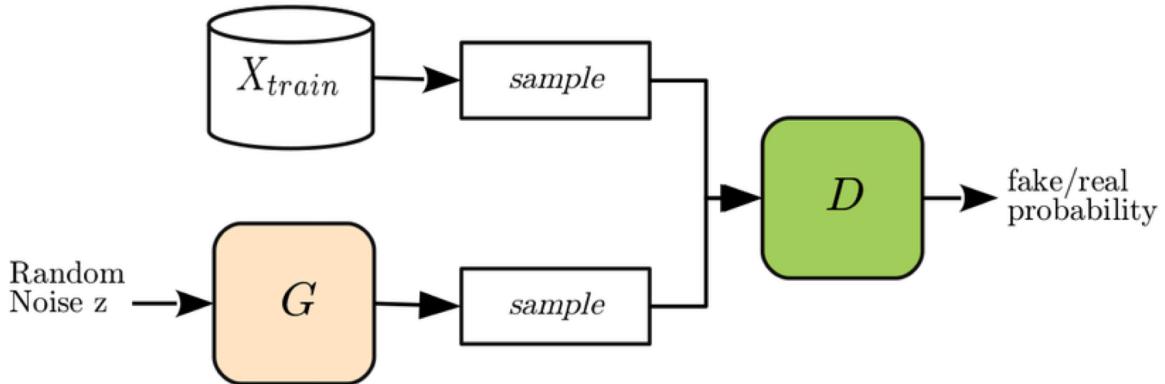
9. GAN bloková schéma + hlavní (ten mrtě dlouhý) vzorec + popsat, jak se probíhá učení

GAN framework používá soupeření dvou hlubokých sítí (deep network), kdy se každá síť snaží porazit tu druhou.

Diskriminátor sleduje reálná data z trénovací množiny a syntetická data z generátoru. Jeho úkolem je klasifikovat každou příchozí instanci dat jako real nebo fake.

Generátor se snaží zmást diskriminátor, aby si myslel, že data, která generuje jsou reálná (real).

Když jsou generátor a diskriminátor nakonfigurovány správně, dorazí k Nash equilibrium (Nashově rovnováze), kde ani jeden z nich není schopen najít žádnou výhodu nad tím druhým.



GANs – Nitty-gritty

Cost function $V(D, G)$ is given by the cross entropy sums (see Kullback–Leibler divergence):

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

Value of $\min_G \max_D V(D, G)$ Expectation prob. of $D(\text{real})$ prob. of $D(\text{fake})$
 Minimize G Maximize D x is sampled from real data z is sampled from $N(0, I)$ fake

For better understanding:

$$V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

Entropy that the data x from real distribution $p_{\text{data}}(x)$ passes through the discriminator (best case scenario). "D" tries to maximize this to 1.

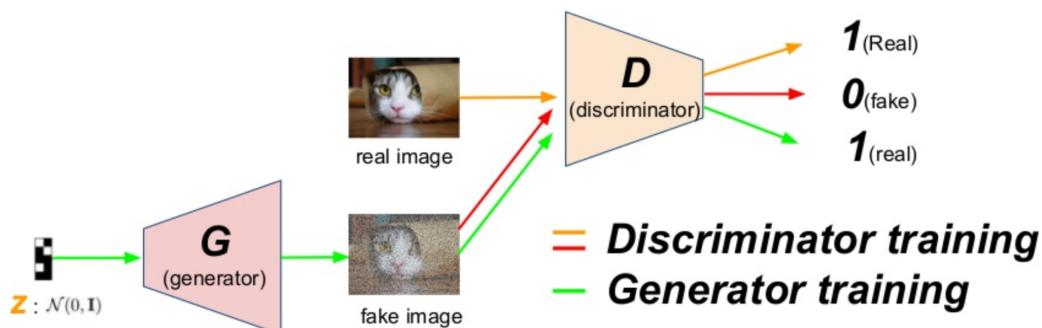
Entropy that the data z from random input $p_z(z)$ passes through „G“, which then generates a fake sample subsequently passed through „D“ to identify the fakeness (worst case scenario). „D“ tries to maximize it to 0.

Učení GAN (training GAN)

Trénovací fáze má 2 hlavní podčásti, které jsou prováděny sekvenčně (pomocí minibatch):

Průchod 1: Učení diskriminátoru a zmražení generátoru (síť G pouze přeposílá a nepoužívá žádnou zpětnou propagaci (no backpropagation is applied))

Průchod 2: Učení generátoru a zmražení diskriminátoru (znovupoužívání zpětné propagace)



10. Co je kovariance, popsat matematicky, vybrat kovar. matice z uvedenych 4 a odvodnit, proc nejsou kovarianci, ktere nejsou.

Kovariance

je to Ko a variance

- variance je rozptyl. Charakteristika která určuje jak se v daném souboru hodnot tyto hodnoty odlišují od své průměrně hodnoty.

Výběrová variance

$$s_{xx} = s^2 = \frac{\sum (x - \bar{x})^2}{N-1}$$

- pro kovarianci se jedna závorka nahradí $(y - (\bar{y}))$ (průměrné)
- pojem kovariance říká, jak se shodují 2 veličiny (x a y) v odchylkách od své střední hodnoty.

Korelace - r

- je korelační koeficient
- je vážený výběrovou směrodatnou odchylkou obou veličin (x a y)
- může nabývat hodnoty od -1 do 1

Regresy rx

- je zde děleno jenom směrodatnou odchylkou jedné z těch veličin na druhou
- **Kovariance** – „jak se shodují 2 veličiny v odchylkách od své střední hodnoty“. Může nabývat záporných, nulových i kladných hodnot. Čím větší absolutní hodnota (pro daný příklad), tím větší lineární závislost.

$$s_{xy} = \frac{\sum (x - \bar{x})(y - \bar{y})}{n-1}$$

- **Korelace** – normovaná kovariance (oběma směrodat. odch.), míra lineární závislosti

$$r = \frac{s_{xy}}{s_x s_y} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2} \cdot \sqrt{\sum (y - \bar{y})^2}}$$

- **Regresy** – normovaná kovariance, zohledňuje závislost proměnných.

$$r_x = \frac{s_{xy}}{s_x s_x} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

Kovarianční matice pro 2 veličiny:

$$\begin{bmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{bmatrix}$$

$$s_{xx} = \frac{\sum (x - \bar{x})^2}{n-1} \quad \longrightarrow \quad \text{rozptyl v } x$$

$$s_{yy} = \frac{\sum (y - \bar{y})^2}{n-1} \quad \longrightarrow \quad \text{rozptyl v } y$$

$$s_{xy} = s_{yx} = \frac{\sum (x - \bar{x})(y - \bar{y})}{n-1} \quad \longrightarrow \quad \text{kovariance}$$

- S_{xx} a S_{yy} nikdy nemůžou být záporně (pokud tomu tak je nejedná se o kovarianční matici)
- S_{xy} a S_{yx} musí být vždy stejné (pokud tomu tak není, nejedná se o kovarianční matici)
- Korelační koeficient může nabývat hodnot od -1 do 1
- Teoretická možnost jak určit, že daná matice je správná je pokud je její determinant roven 0 (učitel ale řekl, že je otázka jestli je to **oprávněná** nebo **neoprávněná generalice**.)

Které z kovariančních matic nemohly z dat vzniknout a proč?

A)

$$\begin{bmatrix} -4 & 2 \\ 2 & 1 \end{bmatrix}$$

NE

Rozptyl
nemůže být
záporný

B)

$$\begin{bmatrix} 4 & 1 \\ -1 & 1 \end{bmatrix}$$

NE

Matice není
symetrická
(kovariance
 s_{xy} a s_{yx}
musejí být
stejné)

C)

$$\begin{bmatrix} 9 & -3 \\ -3 & 1 \end{bmatrix}$$

ANO

Toto může
být
kovarianční
matice

D)

$$\begin{bmatrix} 9 & 5 \\ 5 & 1 \end{bmatrix}$$

NE

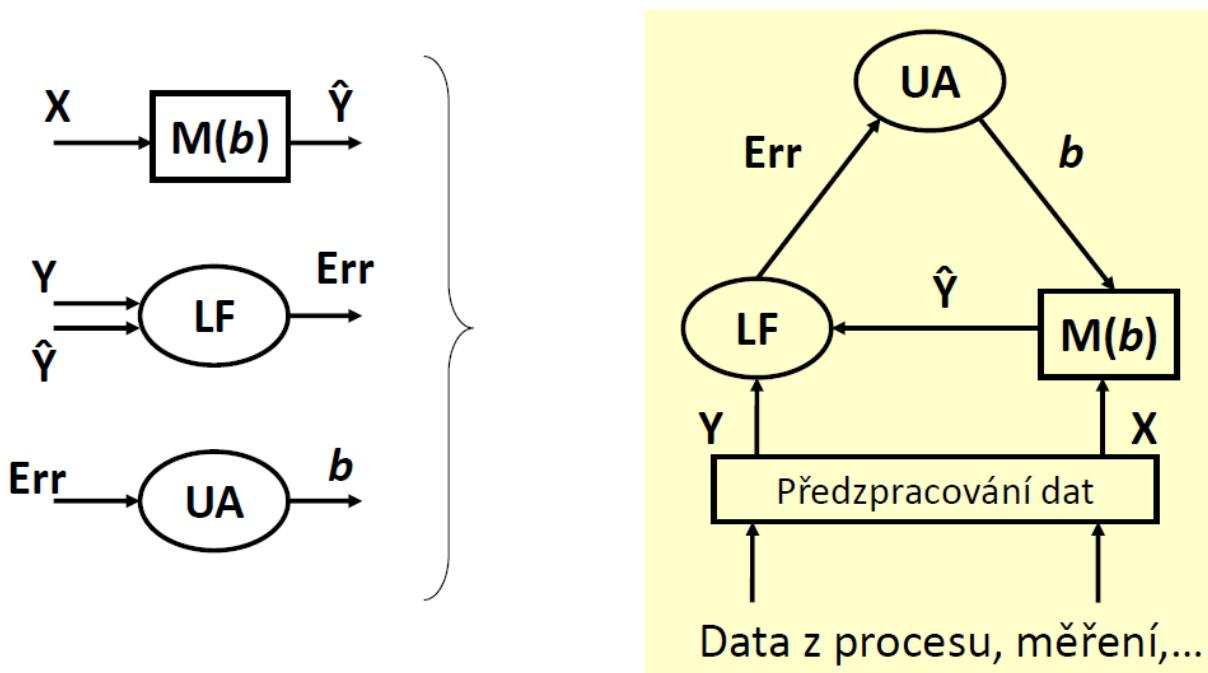
Nedefinuje elipsu
ale hyperbolu,
nemohlo vzniknout
z naměřených dat
(korelace = 5/3 > 1)

Jak poznat, která matice je nebo není kovarianční matice bylo popsáno.

11. Učení s učitelem: scéma + popsat + lossfunction, chybová funkce, nákladová matice

- supervised learning
- vstupní veličina X
- výstupní veličina y (nebo \mathbf{g})
- chceme najít model, který bude dobrě "fitovat" ty vstupy s výstupy
- model, který je příliš přesný na trénovacích datech není zpravidla moc dobrý do reálného provozu (pravděpodobně je přeучen, tedy příliš dobře naučen na trénovací data)

Schéma



VIS: ? nakreslete schéma „procesu modelování s učitelem“

$M(b)$ je model

- má nějaké parametry b (v případě neuronové sítě jsou parametry b váhy jednotlivých neuronů, typy přechodových funkcí (sigmouda, hyperbolická tangenta) u rozhodovacího stromu je parametrem ve kterém uzlu je použit pro následující dělení jaký parametr. (jaký je tam práh))
- vstupují do něj data X (obrázek, data z dotazníku atd.. (vektor dat))
- výstupem je Y se stříškou (je to odhad)

LF (lossfunction)

- udává penalizaci za špatné rozhodnutí
- pokud se tyhle 2 hodnoty budou odlišovat bude generovat chybu (lineárně nelineárně řečeno v některé z dalších přednášce prý)

UA (učící algoritmus)

- Na vstupu vygenerovanou chybu (na základě velikosti/přítomnosti chyby upravuje parametry modelu)
 - následující iterace se děje další sekvence učení s opravenými parametry
 - tímto způsobem se iterativně postupně ladí tento model do jakého optimálního řešení
- Po seskládání do daného trojúhelníku těchto 3 prvků vypadne schéma které znázorňuje základní proces modelování s učitelem

LF lossfunction (chybová funkce)

- $(x, y, f(x)) \in X \times Y \times Y$ je uspořádaná trojice, kde
 - x je vstupní hodnota
 - y je výstupní hodnota
 - $f(x)$ je predikovaná výstupní veličina
- funkce $LF: X \times Y \times Y \rightarrow [0; \infty)$, pro kterou platí, že pro $\forall x \in X$ a $\forall y \in Y$ je $LF(x, y, y) = 0$, je označována jako **chybová funkce**
- chybová funkce (*Loss Function*) \Rightarrow **LF**
 - Nejlépe aby pro kterékoliv X bylo y a $f(x)$ stejné
 - LF nabývá hodnot od 0 (včetně) do nekonečna
 - pokud $LF=0$, model vyhodnotil správně všechny záznamy, které měl. Dá se říct, že model je naučený dokonale. Pokud byly použity nepřesné nebo nesprávné trénovací data (nebo zanesli nějakou chybu) potom to, že je tam 0 nemusí nutně znamenat, že v praxi to bude fungovat jak má.

LF – kvantitativní

- Nejmenší čtverec vzdálenosti (MNČ)

$$LF = [y_i - f(x_i, \mathbf{b})]^2, \quad Err = \frac{1}{N} \sum_{i=1}^N [y_i - f(x_i, \mathbf{b})]^2$$

- Váhová MNČ

$$LF = [y_i - f(x_i, \mathbf{b})]^2 \cdot f(x_i), \quad Err = \frac{1}{N} \sum_{i=1}^N [y_i - f(x_i, \mathbf{b})]^2 \cdot f(x_i)$$

$$LF = [y_i - f(x_i, \mathbf{b})]^2 \cdot \frac{1}{x_i^2}, \quad Err = \frac{1}{N} \sum_{i=1}^N [y_i - f(x_i, \mathbf{b})]^2 \cdot \frac{1}{x_i^2}$$

- Absolutní chyba

$$LF = |y_i - f(x_i, \mathbf{b})|, \quad Err = \frac{1}{N} \sum_{i=1}^N |y_i - f(x_i, \mathbf{b})|$$

- Polynomická chyba

$$LF = \frac{1}{d} |y_i - f(x_i, \mathbf{b})|^d$$

LF kvantitativní

1. Absolutní chyba
 - y_i je jedna hodnota ground truth která tam měla být
 - $f(x_i, b)$ je jedna odezva toho modelu pro vstupní prvek X
 - celková chyba se stanovuje až na konci. Tedy sečtou se všechny LF a vydělí počtem. Projeví se tahle změna až po dávce daných záznamů a následně se všechno opakuje znovu.
2. Metoda nejmenších čtverců (MNČ)
 - místo absolutní hodnoty je použit kvadrát rozdílu
3. Váhová MNČ
 - a. je to metoda MNČ ale přibýde k ní výstupní hodnota z modelu x_i
 - b. váhování rozdílu převrácenou hodnotou kvadrátu vstupu
4. Polynomická chyba
 - rozšíření MNČ kde se nepoužívá kvadrát ale nějaká obecná d-tá mocnina rozdílu

LF – kvantitativní – robustní

- absolutní chyba + pásmo necitlivosti ϵ

$$LF(x_i, y_i, f(x_i, \mathbf{b})) = \max(|y_i - f(x_i, \mathbf{b})| - \epsilon, 0)$$

- Huberova robustní chyba

$$LF(x_i, y_i, f(x_i, \mathbf{b})) = \begin{cases} \frac{1}{2\sigma} [y_i - f(x_i, \mathbf{b})]^2 & , \text{pokud } |y_i - f(x_i, \mathbf{b})| \leq \sigma \\ |y_i - f(x_i, \mathbf{b})| - \frac{\sigma}{2} & , \text{jinak} \end{cases}$$

- **Vždy se jako vstupní informace objevuje** $[y_i - f(x_i, \mathbf{b})] = \Delta y_i$

LF kvantitativní - robustní

1. absolutní chyba + pásmo necitlivosti ϵ
 - základem je absolutní chyba předešlého příkladu, ve které hledáme maximum z nuly. Tato hodnota je snížena o ϵ , díky kterému zde vznikne necitlivost.
 - do určité hodnoty kterou bude tento model vykazovat se budeme tvářit jako by se nic nedělo.

- pásmo necitlivosti se zavádí protože by to mohlo kmitat to dolaďování parametrů. Avšak dané doladění by již nemělo zásadní vliv na klasifikační úspěšnost celkovou.

1. Huberova robustní chyba

- začátek MNČ ale vynásobeno nějakou hodnotou $1/(2\sigma)$ a platí jenom pokud je menší nebo rovno σ
- jinou hodnotu to má jinde
- je to po částech lomená čára

Nákladová matice (cost matrix)

LF – kvalitativní – vícerozměrné

- větší váha opakováním daných prvků
- nákladová matice (cost matrix)

model	realita	A	B	C
A	0	4	10	
B	1	0	8	
C	1	1	0	

- pokud nejde použít čtyřpolní tabulka (tedy buď jsme použili krém nebo nepoužili atd.)
- pokud máme vícerozměrné klasifikační třídy (není to jenom pozitivní nebo negativní)
- na diagonále chceme mít co nejnižší čísla
- hodnota která tam je je hodnota penalizace špatného zařazení
 - pokud klasifikuj A jako A potom nepenalizuji
 - Když případ který klasifikuj jako A ale reálně to je B potom penalizuji 4mi jednotkami
 - Když stejný případ klasifikuj jako C ale reálně to je B potom penalizuji jenom jedničkou

12. Popsat normalizaci. K čemu se používá + vzorec pro normalizaci se střední hodnotou.

Normalizace veličin

- aby byly vstupní veličiny souměřitelné, jsou normalizovány
- typickou normalizací je standardizované normální rozložení se střední hodnotou 0 a rozptylem 1 (ze z na x)

$$\bar{z}_j = \frac{1}{N} \sum_{i=1}^N z_{ij} \quad s_j = \left[\frac{1}{N} \sum_{i=1}^N (z_{ij} - \bar{z}_j)^2 \right]^{\frac{1}{2}} \quad x_{ij} = \frac{z_{ij} - \bar{z}_j}{s_j}$$

- použití u předzpracování (přípravě) dat

Typy normalizace

- **lineární** (typické intervaly $\langle 0;1 \rangle$, $\langle -1;1 \rangle$)
$$x_{norm} = \frac{x - X_{\min}}{X_{\max} - X_{\min}}$$
- **střední hodnotou a rozptylem** (při normálním rozložení 99% v intervalu $-3;3$)
$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad \bar{s} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$
$$x_{norm} = \frac{x_i - \bar{x}}{\bar{s}} \quad x_{norm} = \frac{x_i - \bar{x}}{r \cdot \bar{s}}$$
- **logitovou funkcí** (interval $(0;1)$ nebo $(-1;1)$)
$$x_{norm} = \frac{1}{1 + e^{-x}}$$

?v jakém intervalu bude proměnná normalizovaná střední hodnotou a rozptylem; proč?

V intervalu $(\underline{x} - \frac{s}{2}, \underline{x} + \frac{s}{2})$, obdobné jako lineární normalizace, pouze jsou stanoveny meze ze vstupních dat

13. 3 aktivační(přenosové?) funkce + vzorec. ReLU a její úprava (Leaky ReLu)

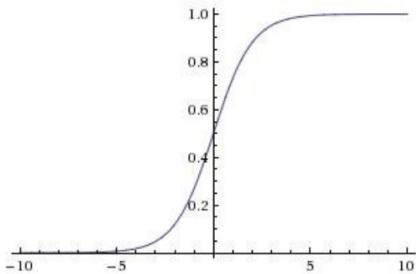
Sigmoid

- převede čísla do rozsahu $[0,1]$

- historicky populární, protože mají hezkou interpretaci jako saturující “firing rate” neuronu

Nevýhody:

1. Saturated neurons “zabijí” gradienty
2. Výstupy sigmoidu nejsou centrovány k nule (zero-centered)
3. $\exp()$ je trochu výpočetně náročný



Sigmoid

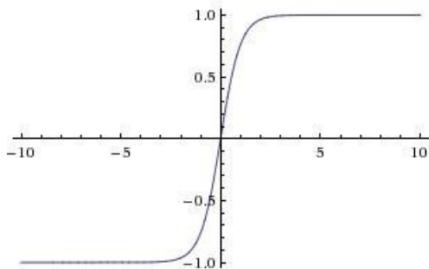
$$\sigma(x) = 1/(1 + e^{-x})$$

- Squashes numbers to range [0, 1]
- Historically popular since they have nice interpretation as a saturating “firing rate” of a neuron

1. Saturated neurons “kill” the gradients
2. Sigmoid outputs are not zero-centered
3. $\exp()$ is a bit compute expensive

tanh(x)

- převede čísla do rozsahu [-1, 1]
- **Výhoda:** centrovány na nulu (zero centered)
- **Nevýhoda:** stále zabijí gradient při saturaci



tanh(x)

- Squashes numbers to range [-1, 1]
- zero centered (nice)
- still kills gradients when saturated :(

ReLU (Rectified Linear Unit)

- počítá $f(x) = \max(0, x)$

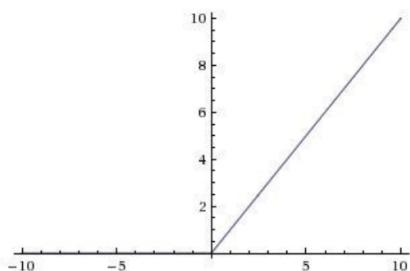
Výhody

- nesaturuje
- velmi výpočetně efektivní

- konverguje v praxi mnohem rychleji než sigmoid/tanh

Nevýhody

- výstup není centrován k nule
- jednotky ReLU mohou “umřít”



Computes $f(x) = \max(0, x)$

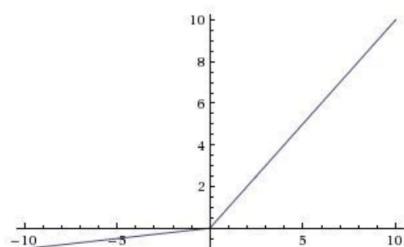
- Does not saturate (in +region)
- Very computationally efficient
- Converges much faster than sigmoid/tanh in practice (e.g. 6x)

ReLU (Rectified Linear Unit)

- Not zero-centered output
- ReLU units can “die”

Leaky ReLU

- nesaturují
- výpočetně efektivní
- konvergují v praxi mnohem rychleji než sigmoid/tanh
- “neumřou”



- Does not saturate
- Computationally efficient
- Converges much faster than sigmoid/tanh in practice! (e.g. 6x)
- will not “die”.

Leaky ReLU

$$f(x) = \max(0.01x, x)$$

[Mass et al., 2013] [He et al., 2015]

In practice

- Use ReLU. Be careful with your learning rates
- Try out Leaky ReLU / Maxout / ELU
- Try out tanh but don't expect much
- Don't use sigmoid

14. Vzorec pro accuracy, specificity, senzitivitu a F1 míru. Vypočítat pro 2 dvojpolové tabulky. Popsat který model je lepší na co. (ten příklad z přednášky)

Ze skript:

Tabulka A.1: Rozdělení výsledků binární klasifikace

		Skutečnost	
		positivní	negativní
Predikce	positivní	A (TP)	B (FP)
	negativní	C (FN)	D (TN)

Senzitivita

Senzitivita je číslo udávané v intervalu <0;1> nebo v procentech <0;100>, které udává poměr úspěšně zachycených pozitivních událostí vůči všem skutečným pozitivním událostem.

$$\text{senzitivita} = \frac{A}{A+C}$$

Specificita

Určuje počet úspěšně zachycených negativních událostí vůči všem skutečně negativním událostem.

$$\text{specificita} = \frac{D}{B+D}$$

Přesnost (accuracy)

Počet všech správně klasifikovaných / počet měření.

$$\text{presnost} = \frac{A+D}{A+B+C+D}$$

Z přednášek:

LF – kvalitativní – binární

model \ realita	$+, P$	$-, N$
$+, \hat{P}$	TP – true positive	FP – false positive
$-, \hat{N}$	FN – false negative	TN – true negative

- $P = TP + FN$ (skutečně pozitivních), $\hat{P} = TP + FP$ (predikovaných), N, \hat{N}
- $T = TP + TN$ (správně klasifikovaných), $T = FP + FN$ (chybně klas.)
- celková správnost (*accuracy*) $Acc = T / (T+F)$
- chyba (*error*) $Err = F / (T+F)$
- senzitivita (úplnost, *recall*) $Sens = TP / (TP+FN) = TP / P$
- specificita $Spec = TN / (TN+FP) = TN / N$
- negativní prediktivní hodnota $NPV = TN / \hat{N}$
- pozitivní prediktivní hodnota (přesnost, *precision*) $PPV = TP / \hat{P}$
- F míra $F = \frac{2 \cdot PPV \cdot Sens}{PPV + Sens} = \frac{2 \cdot TP}{\hat{P} + P}$

Příklad: accuracy vs ROC vs F-míra

Případ 1

m \ r	+	-
+	81	5
-	9	5

V obou případech

$$p(m=+ | r=+) = 0,9$$

$$p(m=- | r=-) = 0,5$$

Případ 2

m \ r	+	-
+	45	25
-	5	25

1. případ

- $Acc = T / (T+F)$ 0,86
- $Err = F / (T+F)$ 0,14
- $Sens = TP / (TP+FN)$ 0,9
- $Spec = TN / (TN+FP)$ 0,5
- $NPV = TN / N$ 0,36
- $PPV = TP / P$ 0,83
- F míra 0,92

2. případ

- 0,7
- 0,3
- 0,9
- 0,5
- 0,83
- 0,64
- 0,75

Často se používá **plocha pod ROC křivkou (AUC)**, která vzniká vynesením bodů senzitivity a specificity do 2D grafu. Více v přednášce o předzpracování dat.

15. Rozhodovací stromy, popsat strukturu RS (kořenový uzel, uzel, větev, list, hloubka stromu), vysvětlit pojem atribut, hodnota atributu, klasifikační atribut, záznam, napsat vzorec pro entropie a napsat algoritmus učení RS

Výhody

- rychlosť predikcie
- interpretovateľnosť
- minimálne citlivé na irrelevantné atributy
- nízké nároky na predzpracovanie dát

Nevýhody

- ortogonálne delenie priestoru - nemusí náležať ani jednoduchou funkčnou závislosť

- Rozhodovací strom – *hierarchický nelineární systém (model)*
 - Hloubka stromu
 - Uzel
 - Kořenový uzel
 - List
 - Větev
-

Atribut - vlastnost. Svou hodnotou popisuje nebo charakterizuje prvek nebo sledovaný objekt. (váha, výška hlasu, náušnice,...)

Hodnota atributu - číslo nebo symbol. Vyjadřuje konkrétní stav atributu. (velká, střední, malá,...)

Klasifikační atribut - je tvořen třídami. Hodnoty určují typ záznamu.

Záznam - soubor údajů o jednom objektu složený z vektoru hodnot vstupních atributů a výstupního atributu nebo klasifikátoru [Z₁, Z₂,...]

Entropie - míra neuspořádanosti

- **Entropie**

$$H = - \sum_{i=1}^C p(G_i) \log_2 p(G_i)$$

Základní princip tvorby RS

- V cyklu opakuj
 1. Získej informace o uzlu
 2. Rozhodni o uzlu, zda bude dál dělen (krok 3.) nebo z něj udělej list a rozhodni o jeho výstupní hodnotě
 3. Vyber nejlepší atribut na větvení
 4. Rozděl data do nových uzlů
- Prořezej strom