

SUI - 2022/2023 otazky zo skusky

Na zaciatku netusime v akom mieste sa nachadzame teda:

mozne pozicie agenta $[0,1,2,3,4]$ - potrebujeme vybrat nejaku vhodnu akciu ktorou zacneme, z grafu nas moze lakovat vybrat akciu X keďže z 1,4 vedie priamo do 0 avšak vidime ze medzi 2,3 budeme cyklist preto si radsej najprv zvolime naprieklak akciu B

$[1,2,3,4] \rightarrow$ zvolili sme akciu B $\rightarrow [4,1,3]$

mozme zopakovat vyber akcie B

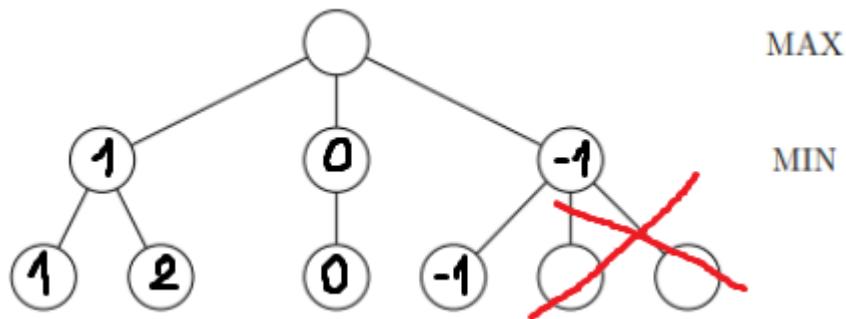
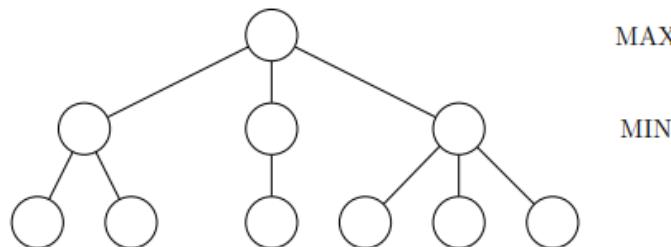
$[4,1,3] \rightarrow$ akcia B $\rightarrow [3, 4]$

z grafu vidime ak znova vyberieme akciu B tak nas aktualny stav ostane len 3 co nam vyhovuje

$[3,4] \rightarrow$ akcia B $\rightarrow [3]$, teraz vieme v akom sme stave a jednoduchou sekvenciou akcii sa dostaneme do ciela

$[3] \rightarrow$ akcia A $\rightarrow [4] \rightarrow$ akcia X $\rightarrow [0]$

2. Uvažujte sekvenční deterministickou hru na obrázku. Jaké maximální úspory (v uzlech, které nemusíte prohledávat) lze dosáhnout, když pro hledání nejlepšího tahu namísto MiniMaxu použijete Alpha – Beta prořezávání? Vepište do obrázku ohodnocení listových uzelů, které k tomu povede, a vyznačte, které uzly se nebudou prohledávat.



Vďaka Alfa-beta prorezavani možeme ušetriť maximalne 2 uzly ktore nemusíme prehľadávať, nakoľko na základe hodnoty -1 je jasné že v danom podstrome nam protivník vie vybrať horšiu hodnotu ako je v ostatných 2 podstromoch teda zvýšne hodnoty nas nezaujímajú keďže zisk z daného podstromu by bol horší ako pro zvyšných 2

Greedy search : g: 0 , h: h^* - kedze pri greedy search nas nezaujima cena urazenej

3. Máte k dispozici odladěnou implementaci A* $a^*(problem, g, h)$. Na vstupu očekává `problem` (umí rozbalovat uzly a ověřovat, zda jsou cílem), `g` (popisuje cenu uražené cesty, kde cesta je seznam uzlů) a `h` (popisuje odhad zbývající ceny do cíle); původní funkce jsou `g*` a `h*`. Jaké `g` a `h` funkci `a*` předáte, aby simulovala greedy search a depth-first search (DFS)?

cesty ale len odhadovaná cena do ciela preto g bude rovno 0

DFS : g: 1/cas pridania uzlu , h: 0 // check pls +0

4. Máme CSP úlohu s proměnnými X_1, X_2, X_3 a odpovídajícími doménami $D_1 = \{1, 2, 3, 4, 5\}, D_2 = D_3 = \{1, 2, \dots, 9, 10\}$. Proveďte inferenci tak, abyste zajistili konzistenci hran pro omezení $3X_2 < X_3, X_3 < X_1$.

Po inferencii zostanu v domenach:

`D_1={5}`

`D_2={1}`

`D_3={4}`

// check pls +1

5. Mějme dvourozměrná data pro tři třídy. Na těchto datech chceme natrénovat gaussovský klasifikátor. Vyjmějte všechny parametry, kterými je takový klasifikátor popsán a které je potřeba na datech odhadnout. Jakou mají tyto parametry podobu? Kolika reálnými čísly jsou všechny tyto parametry reprezentovány? Jak tyto parametry na datech odhadneme? Jak tyto parametry využijeme k výpočtu pravděpodobnosti, že nově příchozí vzor patří do jedné, druhé či třetí třídy?

kedze sa jedna o 3 triedy bude klasifikator obsahovat 3 gaussovky

pre kazdu gaussovku potrebujeme:

vektor strednych hodnot - 2 realne cisla (2D data)

kovariacnu maticu - kedze su to 2 rozmerne data tak bude 2×2 - 4 realne cisla

apriory pravdepodobnost danej triedy - 1 realne cislo

celkovo $3 * (2 + 4 + 1) = 21$

tieto hodnoty odhadneme z trenovacich dat nasledovne:

prve rovnica je pre odhad apriory pravdepodobnosti tried

dalsie 2 rovnice su pre odhad vektoru strednych hodnot a kovariacnej matice

p osti ze novo prichodzi vzor patri do jednej, druhej alebo tretej triedy

Class priors can be ML estimated as the proportions of the example counts

$$P(c) = \frac{N_c}{\sum_k N_k} \quad P(blue) \frac{400}{400 + 600}$$

Probability density function for each class is assumed to be 2D Gaussian

$$p(\mathbf{x}|c) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

and its parameters ML estimated as

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \mathbf{x}_{cn} \quad \boldsymbol{\Sigma}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} (\mathbf{x}_{cn} - \boldsymbol{\mu}_c)(\mathbf{x}_{cn} - \boldsymbol{\mu}_c)^T$$

Class posterior probability for new observations is obtained from the prior and class pdf-s using Bayes rule:

$$P(c|\mathbf{x}) = \frac{p(\mathbf{x}|c)P(c)}{\sum_k p(\mathbf{x}|k)P(k)}$$

6. Ke stavbě klasifikátoru můžeme využít *generativní model* nebo můžeme použít *model trénovaný diskriminativně*. V čem se tyto dva přístupy liší? Uveďte konkrétní příklady modelů/klasifikátorů pro oba přístupy. Jaké jsou jejich výhody a nevýhody?

Srovnání:

- generativní:
 - obvykle méně náchylné k přetrénování s malým množstvím tréninkových dat
 - modulární: složité modely vycházejí z jednoduchých
- diskriminativní:
 - dominantní přístup v dnešní době
 - méně plýtvání parametry
 - obvykle lepší výkon s dostatkem dat
 - umožňuje end-to-end řešení → neřeší dílčí kroky, modelování dat, atd.

For generative models we learn $p(\mathbf{x}|c)$ and $p(c)$ and use Bayes rule to derive $p(c|\mathbf{x})$

Logistic regression is discriminative approach, where we learn to estimate directly $p(c|\mathbf{x})$

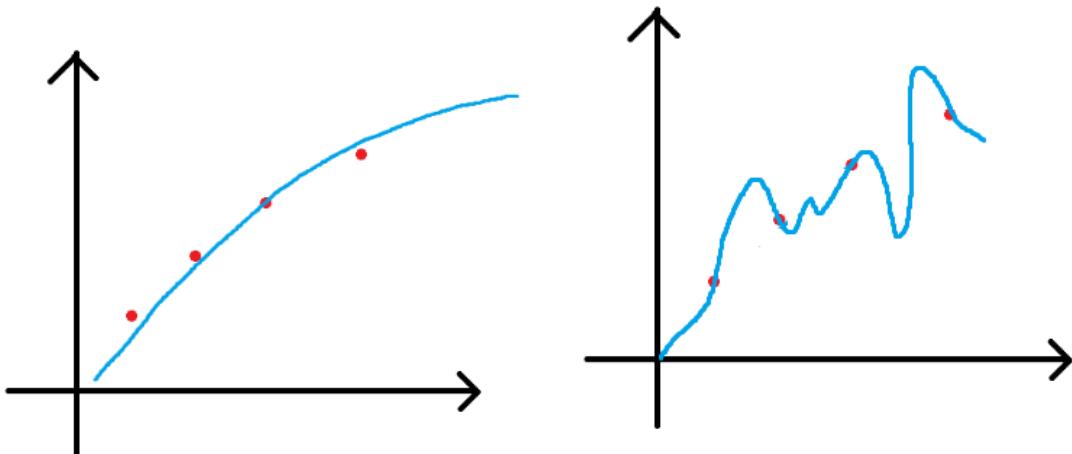
There are other discriminative training strategies that do not even try to estimate any probability

For example, Support Vector Machines directly finds decision boundary that maximizes margin between the classes

diskriminativny model - lineарне/логистичка регресия

generativne modely - гауссовский классификатор, категорический классификатор

7. Nakreslete obrázek s příkladem trénovacích dat pro regresní problém. Dále nakreslete příklad řešení, které se mohl na těchto datech naučit model polynomální regrese druhého řádu ($K = 2$), který generalizuje. Vysvětlete co a proč jste nakreslili a čím je způsobeno, že model generalizuje. Co by se muselo stát či udělat, aby NEgeneralizoval?



obrazok vľavo generalizuje , obrazok v pravo negeneralizuje, znamena to to ze polynomialna regresia ktora negeneralizuje presne opisuje trenovacie data avsak pri novych nevidenych datach by nefungoval najlepsie (nastane tak ked je priliz vysoky stupen polynomu pre malo trenovacich dat alebo ak sa v datach nachadzaju outliers, ktori sposobuju velku chybu), polynomialna regresia vľavo prave naopak generalizuje, teda nepopisuje trenovacie data presne ale na doposial nevidenych datach bude fungovať spravne

// check pls +0

8. 20 % populace se nakazilo určitým blíže nespecifikovaným virem. Pro tento virus máme k dispozici test, který je pozitivní pouze pro 90 % lidí nakažených tímto virem. Nicméně, pokud má člověk pozitivní test, je 10% šance, že nakažen není. Jestliže u osoby náhodně vybrané z celé populace provedeme test, jaká je pravděpodobnost, že výsledek testu bude pozitivní?

$$p(\text{nakazený}) = 0.2$$

$$p(\text{zdravý}) = 0.8$$

$$p(\text{pozitivní test} | \text{nakazený}) = 0.9$$

$$p(\text{negativní test} | \text{nakazený}) = 0.1$$

$$p(\text{zdravý} | \text{pozitivní test}) = 0.1$$

$$p(\text{nakazený} | \text{pozitivní test}) = 0.9$$

nasledne na zaklade Bayesovho pravidla:

$$p(\text{pozitivní test}) = (p(\text{pozitivní test} | \text{nakazený}) * p(\text{nakazený})) / p(\text{nakazený} | \text{pozitivní test})$$

$$p(\text{pozitivní test}) = (0.9 * 0.2) / 0.9 = 0,2$$

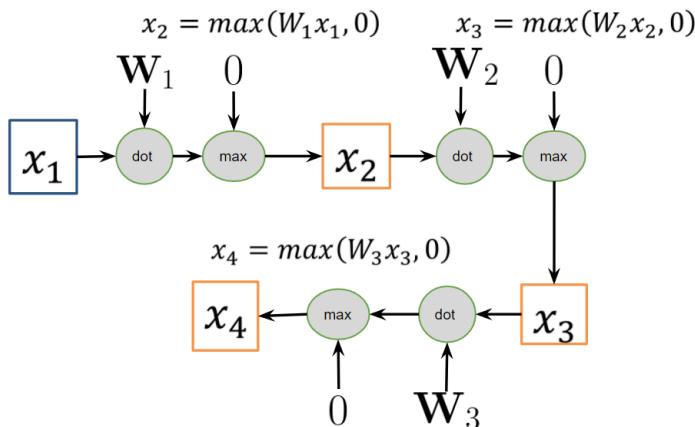
9. Jaké znáte podobory umělé inteligence? (jen vyjmenovat)

Strojove uceni, NN , evolucne alg, pocitacove videni, robotika, zpracovanie reči/ prirodzeneho jazyka, expertne systemy

10. Jaké podmínky musí splňovat výpočetní graf u dopředných neuronových sítí (úplně všechny ty, o kterých byly přednášky)? Nepište souvislý text, stačí hesla. Nakreslete příklad.

- musi byt acyklicky orientovany graf

Computational graph



nieco taketo nakreslit, kladne asi v zjednodusenej podobe kde bude napriklad len w1 a w2

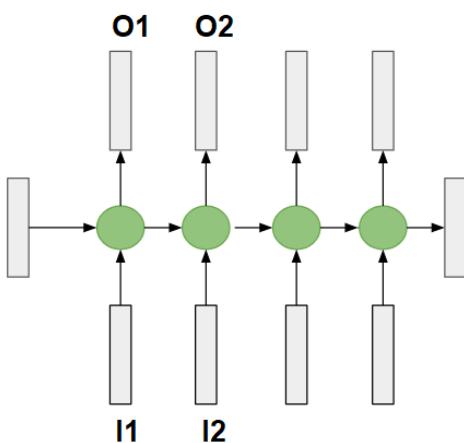
11. Co jsou „učitelné“ parametry konvoluční vrstvy? Odpovězte tak dvěma slovy. Napište obecný matematický výraz pro počet těchto parametrů u standardní konvoluční vrstvy. (parametr = jedno reálné číslo)

v konvolučnej vrstve sa učia konvolučné jadrá, ktorých je v danej vrstve niekoľko
konvolučné jadrá bývajú zvyčajne velmi male, obvykle o velkosti 3x3
cize matematicky výraz pre pocet tychto parametrov by som napisal ako

- $k * (x * x * c + 1)$
- k - počet konvolučných jadier v konvolučnej vrstve
 - x - výška/šírka konvolučného jadra
 - c - počet kanálov - obvykle 3 - RGB
 - $+1$ je pre bias filtro

// check i guess +0

12. Nakreslete jednu standardní rekurentní vrstvu nad sekvencí délky 4 jako výpočetní graf (obecně a nezávisle na typu rekurentní sítě). V diagramu pak pro dva libovolné výstupní vektory označte, na jakých všech vstupech dané vrstvy (vstupních vektorech) každý z nich závisí. V odpovědi nesmí být žádný text kromě symbolů „O1“, „I1“, „O2“, „I2“ označujících vámi zvolené výstupy (O1 a O2) a k nim všechny náležící vstupy (I1 a I2).



O1 - bude zavisiť len na I1

O2 - bude zavisiť na I1 a I2

asi este bude obe zavisiet na Input state vector (vstup uplne nalavo danej rekurentnej vrstvy) ale na to sa nepyta v otazke ig cize to asi netreba

13. Jakým postupem generují výstupní sekvence (např. věty) autoregresivní sequence-to-sequence (Seq2-seq) modely, které se používají například pro překlad jazyka? Vysvětlete jen funkci autoregresivního dekodéru. Ideálně napište i to, jakou pravděpodobnostní funkci tyto neuronové sítě modelují. V ideálním případě vysvětlete algoritmus generování jen neformálním pseudokódem.

$$P(w_1, w_2, w_3, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\dots P(w_n|w_{n-1}, w_{n-2}, \dots, w_1)$$

$$P(w_1, w_2, w_3, \dots, w_n) = \prod P(w_i|w_{i-1}, w_{i-2}, \dots, w_1)$$

sekvencie sa generuju tak ze pravdepodobnosť celych viet je rozlozena na pravdepodobnosti vsetkych slov, ktore su vzdy podmienene predchadzajucimi slovami vo vete

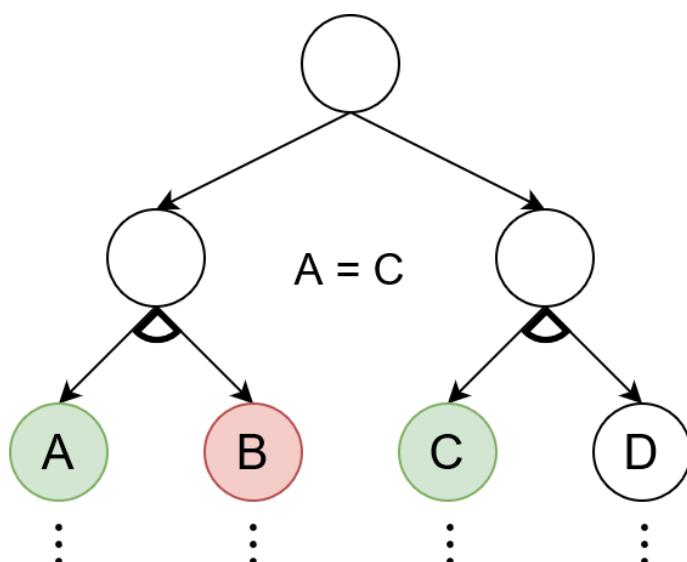
NN toto modeluju tak ze siet ma na vstupe slova prefixov a na vystupe je klasifikatorom priradený do triedy odpovedajucej danemu slovu v jazyku
je to možné aplikovať v rekurentnych sietach, ale v dnesnej dobe sa používajú hlavne siete s

Attention - takzvané Transformery

14. Je při posilovaném učení vhodné, aby agent dostal odměnu jen při splnění úlohy, nebo i průběžně podle toho, jak se blíží splnění úlohy? Proč? Může to naopak způsobit nějaké problémy?

//TODO - ak niekto viete kludne to vypracujte

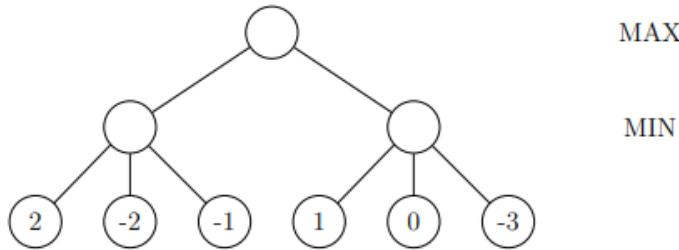
1. Při každém prohledávání je možné urychlit výpočet, pokud si zapamatujeme stavů, které nevedou k řešení. Při prohledávání s nejistotou je možné dosáhnout zrychlení i při zapamatování stavů, které k řešení vedou. Nakreslete (pokud možno malý) AND-OR strom, ve kterém k tomu dojde, a stručně popište, jak to proběhne.



Zelený uzol = řešiteľný, červený uzol = neriešiteľný.

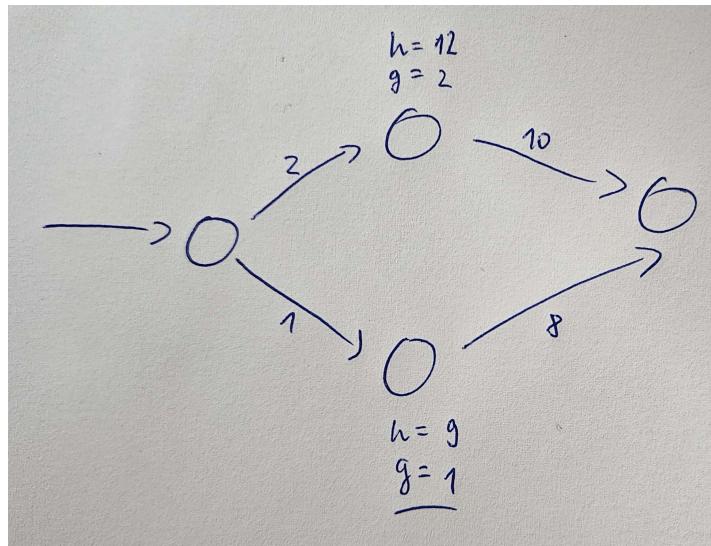
Uzol A by bol rovnaký ako uzol C (mali by rovnaký hash).

- Zistí, že uzol A je riešiteľný, zapamäta si ho.
- Zistí, že uzol B nie je riešiteľný, preto musí hľadať ďalej.
- Zistí, že uzol C je rovnaký ako A takže ho nemusí riešiť, lebo si zapamätal, že je riešiteľný -- ušetril robota
- Algoritmus Alfa-Beta zrychluje riešení dvouhráčových her s nulovým součtem tím, že umožňuje nevyhodnocovať niektoré stavy, resp. podstromy. Dosažené zrychlenie je ovšem závislé na pořadí, ve kterém se stavy výběc berou v potaz. Navrhnete uspořádání uzlů v zobrazené hře, které povede k maximálnímu prořezávání a znázorněte, které uzly se nebudou vyhodnocovat. **Dejte pozor, aby ste nezměnili hru jako takovou.**



- najprv prejdeme lavy podstrom , jeho uzly (2, -2, -1) tu nemusime menit poradie, je to jedno kedze nemame momentalne co odrezat musime ho prejst cely
 - nasledne prejdeme do praveho podstromu , kde ked prehodime hodnoty -3 a 1 tak prva hodnota ktoru skontrolujeme bude -3 , v tomto bode zistime ze nam super vie dat horsiu hodnotu ako je akakolvek v lavom podstrome, teda hodnoty 0 a 1 nemusime uz kontrolovat (mozme ich zarezat) cim usetrime vyhodnotenie 2 uzlov
3. U heuristiky v A* požadujeme, aby dávala optimistický odhad zbývající ceny do cíle. Demonstrujte na vhodném příkladu s jediným cílovým uzlem, že i při porušení této podmínky může dojít k nalezení optimálního řešení. Do prohledávacího grafu vepište jak ceny hran, tak hodnoty heuristiky.

(je to trochu zmeneny priklad z democvika)



5. Mějme třídimenzionální data pro dvě třídy. Na těchto datech chceme natrénovat gaussovský klasifikátor. Vyjmenujte všechny parametry, kterými je takový klasifikátor popsán a které je potřeba na datech odhadnout. Jakou mají tyto parametry podobu? Kolika reálnými čísly jsou všechny tyto parametry reprezentovány? Jak tyto parametry na datech odhadneme? Jak tyto parametry využijeme k výpočtu pravděpodobnosti, že nově příchozí vzor patří do jedné či druhé třídy?

kedze sa jedna o 2 triedy bude klasifikator obsahovať 2 gaussovky

pre kazdu gaussovku potrebujeme:

vektor strednych hodnot - 3 realne cisla (3D data)

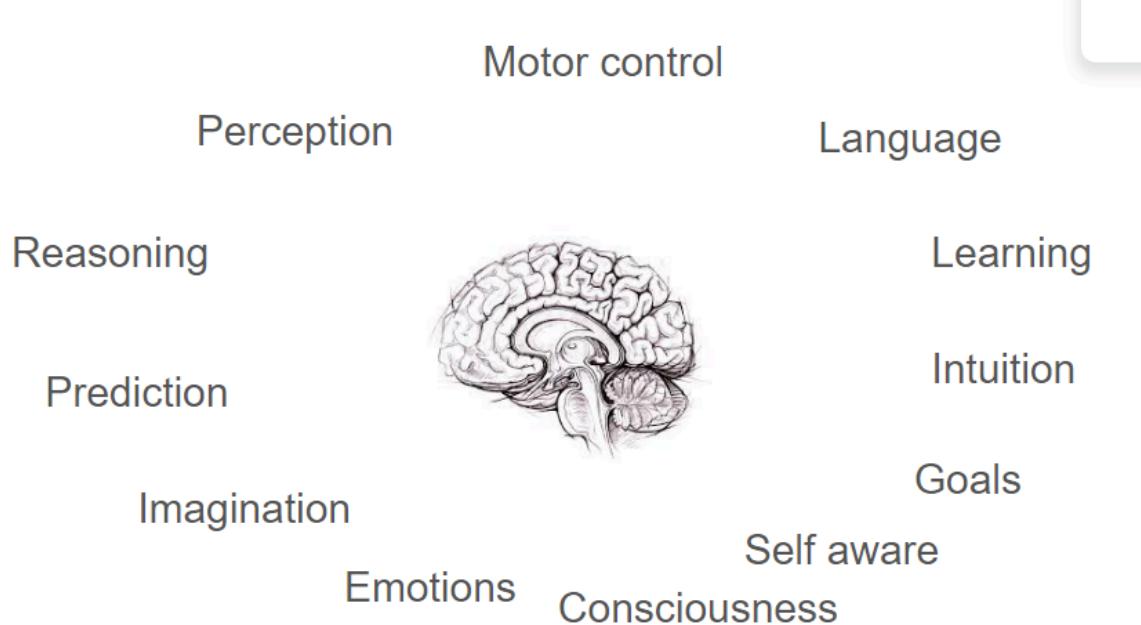
kovariacnu maticu - kedze su to 3 rozmerne data tak bude $3 \times 3 = 9$ realnych cisel

apriory pravdepodobnosť danej triedy - 1 realne cislo

celkovo $2 * (3 + 9 + 1) = 26$

tieto hodnoty odhadneme z trenovacich dat rovnako ako bolo ukazane v priklade 5 na 2 strane dokumentu

9. Jaké jsou různé aspekty inteligence? (jen vyjmenovat) Musí agent vykazovat všechny tyto aspekty, abyste ho považovali za inteligenčního?

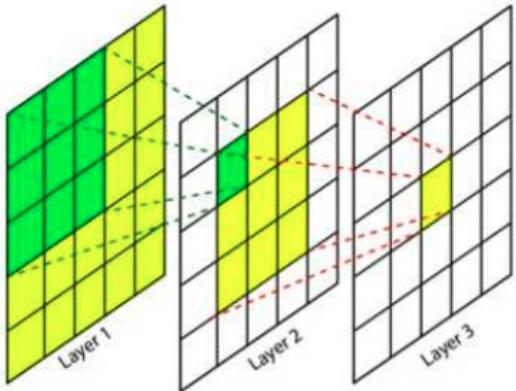


povedal by som ze nemusi kedze ani mnozstvo ludi vsetky nema a ludi povazujeme za inteligentnych (prepokladam ze takato jebnuta odpoved by sa mu pacila)

10. Co je „gradient“ v kontextu učení neuronových sítí? Jak je při učení využit?

Gradient sa v kontexte učenia NN využíva pri takzvanom Gradientnom zostupe ktorým sa optimalizuju vahy NN v iteraciach. V každej iterácii je spocítaný gradient (ako parcialna derivácia objektívnej funkcie voči všetkym optimalizovaným parametrom NN). Uprava parametrov prebieha tak že je od nich odčítaná hodnota gradientu nasobená learning rate (učicí konštantu), ktorá ovláda rýchlosť optimalizácie.

11. 2D konvoluční síť začína dvěma konvolučními vrstvami. První vrstva má 16 jader o velikosti 3×3 a druhá 32 jader o velikosti 5×5 . Velikost vstupního obrázku je 64×64 a má 3 kanály. Napište velikost percepčného pole aktivací druhé vrstvy (tedy na aké oblasti závisí hodnoty v jednom „pixelu“ výstupu druhé vrstvy). Velikost percepčného pole napište jak v pixelech vstupního obrázku, tak výstupní „aktivitační mapy“ první vrstvy. Odpověď nesmí obsahovat žádný souvislý text delší než dvě slova.



perceptive field = 7×7

aktivacna mapa prvej vrstvy = 5×5

12. Co je to obousměrná (bidirectional) rekurentní vrstva? Jak se liší od „normální“ rekurentní vrstvy? Odpovězte krátce a jen vysvětlete rozdíl, výhody a nevýhody/omezení. Nesmíte vysvětlovat obecně rekurentní vrstvy.

Obojsmerne rekurentna vrstva kombinuje 2 "klasicke" rekurentne vrstvy, jedna spracovava sekvenciu jednym smerom a druhia opacne, 2 vystupy v kazdej pozicii sa konkatenuju a davaju finalny vystup. Takato struktura ma zmysel len ak siet nemusi byt kauzalna (pri real time detekcii slov v reci sa nemozeme pozerať do buducna) avšak pri spracovani hotovyh nahravok to je možné.

13. Jaká je základní myšlenka attention mechanizmu (pozornosti) v neuronových sítích? Jaké má výhody oproti rekurentním vrstvám? V jakých úlohách se například používají?

Je to mechanizmus (vrstva siete) ktorá umožnuje aby sa siet naucila extra hrať potrebnú informáciu zo zdrojových dat. Jedna sa o pevné spojenie s celymi zdrojovými datami avšak siet si na zaklade svojho vnútorného stavu dokáže nastavovať vahu jednotlivych časti zdrojových dat – uči sa priradzovať vysoko vahu dôležitým datam.

Attention sa využíva pri rozpoznávaní písaného textu alebo pri preklade kde potrebujeme určity kontext

2. Minimax je populární algoritmus pro řešení dvouhráčových her, kde předpokládáme, že jsou oba hráči racionální, tedy že maximalizují svůj užitek. Přesto polovina algoritmu spočívá v hledání *minimálních* hodnot. Vysvětlete tento rozpor a pojmenujte odpovídající vlastnost her, pro něž minimax lze použít.

Vlastnosť her - hry s nulovym suctom

Pri tomto type hier nam ide o maximalizovanie nasho zisku avšak superovi ide o jeho minimalizovanie (preto MinMax) dane algoritmy spocivaju v hľadaní minimalnych hodnot nakolko nam ako zacinajucemu hracovi ide o to zistit ako nam v jednotlivych uzloch dokáže super "uskodit" teda hľadame aky minimalny zisk vieme v ktorej vetve ziskat, kedze ratame s tym ze super hra racionalne a bude vzdy pre nas vyberat ten najmensi zisk.

4. Uvažujte agenta, ktorý je založen na hľadovém lokálním prohľadávaní ve formě hill-climbingu s restarty. Takový agent umí *například* řešit problém N dam nebo najít ideální pozici nové budovy v diskretizovaném světě. Uvedte, co takový agent *obecně* potřebuje od prostředí (problému) samotného a následně pak klasifikujte agentovu reprezentaci stavu (atomická / faktorizovaná / strukturovaná). Odpověď strukturujte do odrážek.

//TODO

5. Na dvoudimenzionálních datech pro tři třídy jsme natrénovali gaussovský klasifikátor. Pro nově příchozí testovaný vzor \mathbf{x} nám tento generativní model umožní určit, s jakou aposteriorní pravděpodobností tento vzor patří do každé třídy. Jak tyto pravděpodobnosti vypočteme? Uveďte přesný postup, co a jak konkrétně počítat. Předpokládejte, že již máte k dispozici funkci pro vyhodnocení hustoty pravděpodobnosti vícerozměrného gaussovského rozložení $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Vymenujte všechny parametry, kterými je tento gaussovský klasifikátor popsán, a napište, kde se tyto parametry využijí při vypočtu aposteriorních pravděpodobností tříd. Jakou mají tyto parametry podobu? Kolika reálnými čísly jsou všechny tyto parametry reprezentovány? Jak jsme tyto parametry na trénovacích datech odhadli?

otazka 5 strana 2

6. Uveďte dva příklady diskriminativně trénovaných modelů pro klasifikaci, které umožňují vyhodnocení aposteriorních pravděpodobností tříd pro vstupní (testovaný) vzor. Alespoň pro jeden z těchto dvou příkladů uveďte, jak se tyto aposteriorní pravděpodobnosti konkrétně spočítají. V čem se jejich výpočet liší ve srovnání s generativními klasifikačními modely?

Rozdiel s generativnymi:

For generative models we learn $p(\mathbf{x}|c)$ and $p(c)$ and use Bayes rule to derive $p(c|\mathbf{x})$

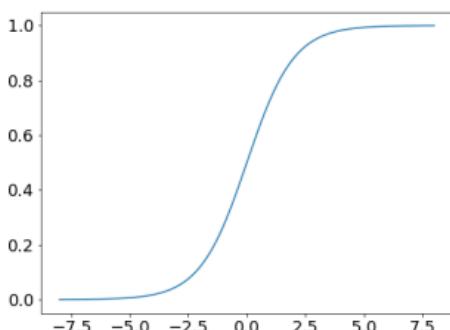
Logistic regression is discriminative approach, where we learn to estimate directly $p(c|\mathbf{x})$

2 priklady : linearna a logisticka regrese

Vypocet aposterior pre linearnu:

Let's concentrate directly at modeling the posterior $p(c|\mathbf{x})$

- For now, let's consider only binary classifier (i.e. $c \in \{0,1\}$), where the posterior $p(c|\mathbf{x})$ is just a function, with output constrained to be in the range of 0 and 1
- We were able to learn functions from data using linear regression.
- Let's use linear regression-like model with the output mapped to the 0 – 1 range using logistic sigmoid function $\sigma(a)$:



$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

$$P(c = 1|\mathbf{x}) = \sigma(\hat{\mathbf{x}}^T \mathbf{w})$$

Vypocet aposterior pre logisticku:

2-class linear logistic regression models directly posterior probability of class $c = 1$ as

$$P(c = 1|\mathbf{x}) = \sigma(\hat{\mathbf{x}}^T \mathbf{w})$$

Probability of the other class $c = 0$ is

$$P(c = 0|\mathbf{x}) = 1 - P(c = 1|\mathbf{x})$$

10. Jaká je role loss funkce v učení s učitelem? Co je jejím vstupem a výstupem?

Pri uceni s učiteľom nam ide o to mapovať vstup na pozadovaný výstup

Do loss funkcie vstupuju výsledky pre celý dataset a ground truth, násim cieľom je aby výsledna loss funkcia (suma chyb na vsetkych datach) bola co najmenej, nakolko nam udava chybovost nasho modelu - teda ako veľmi sa lisia výstupy nasej siete od pozadovanych výstupov.

//check pls +0

11. Je možné, že pri niejakých hodnotách učící konstanty (learning rate) není možné získať dobré naučený model neuronové sítě pri použití algoritmu typu GD a SGD? Vysvetlite, proč to tak není, nebo uved'te, k čemu může dojít.

Ano je to možné, nakolko pri veľkej hodnote learning rate budeme pri GD/SGD robiť veľke kroky pri zmenach parametrov v jednotlivych iteraciach a možno nastanú ze preskocime do oblasti kde bude chyba vyrazne vyšia a možno dosť k divergencii učenia - teda chyba bude nekontrolovatelné rast k nekonečnu. Vo všeobecnosti je ale tazké určiť čo je veľký learning rate, záleží na zložitosti a iných vlastnostiach optimalizovanej funkcie, teda optimálna konštantá LR možno byť pre rozne úlohy vyrazne ina , musí sa to vyskúsať.

12. Jaký počet operací násobení je potřeba pro výpočet jednoho výstupního pixelu jednoduché 2D konvoluční výstvy z přednášek? Napište pouze matematický výraz a heslovitě význam symbolů, které v daném výrazu použijete.

pre 1 pixel = $k * k * c * M$

k = výška/sírka konvolučného jadra

c = počet kanalov konvolučného jadra

M = počet konvolučných jadier v danej konvolučnej vrstve

Ak by to chcel pre celý obrazok tak = $H * W * k * k * c * M$ kde H, W sú výška a sírka vstupného obrazku

Na prednáške sa pytal aj na pamäťovú zložitosť takejto výstvy

miesto na disku = $k * k * c * M$ - rovnako ako pri počte operácií násobenia pre 1 pixel

1. Výsledkom prohľadávání v známém, deterministickém a plne pozorovateľnom prostredí (pomocí, napr., IDS) je agentový plán v forme lineárnej posloupnosti akcií. Co je výsledkom prohľadávania v známém, nedeterministickém a plne pozorovateľnom prostredí? (Návod: není to nic Markovovo.) Jak se v tom uplatní agentovy senzory?

//TODO

2. Váš agent hraje deterministickou sekvenční hru pro dva hráče pomocí algoritmu Minimax. Při hraní proti sadě referenčních soupeřů na evaluační sadě her dosáhl průměrného zisku na hru 42 a na referenčním stroji mu trvala volba tahů v průměru 13.57 sekund. Uveďte, jak očekáváte, že se tyto metriky změní (zvětší/zmenší), když Minimax nahradíte algoritmem Alfa-Beta. Odhadněte, jak moc se změní, příp. stručně popište, jaké informace Vám k takovému odhadu chybí.

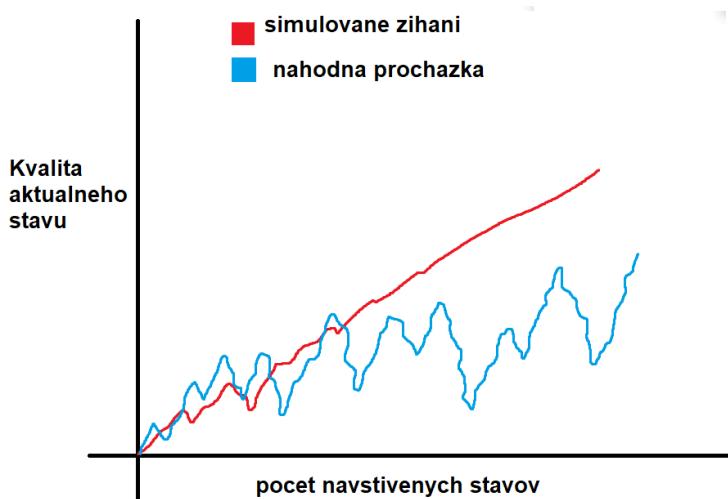
Priemerny zisk sa nijako nezmeni nakolko jediny rozdiel medzi minmaxom a alfabetou je to ze alfabeta je rychlejsia ak dokaze nejake uzly zarezat.

Priemerna volba tahu sa urcite nezvacsi (v pripade ze alfabeta nedokaze ziadne uzly zarezat tak by mal byt priemerny cas rovnaky ako pri minmaxe) avsak moze sa potencionalne znizit ak alfabeta dokaze zarezat nejake uzly a tym padom ich nekontrolovat avsak takyto odhad bez blizsich informacii o danej hre a ohodnoteni koncovych uzlov nedokazeme urcit.

3. Jakou neinformovanou metodu je nejlepší použít (nejméně paměti, nejkratší čas), pokud víme, že nejlevnější řešení prohledávací úlohy je v 8. úrovni prohledávacího stromu? Metodu pojmenujte nebo stručně popište. Odůvodněte, proč je metoda optimální z hlediska nalezeného řešení i výpočetních nároků.

Ak predpokladame ze cely prehľadavaci strom ma vacsiu hlbku ako 8 tak si zvolime DLS (ak by mal hlbku 8 tak je tato volba ekvivalentna s DFS) , DLS bude prehľadavat dany strom do vzdy do maximalnej hlbky 8 teda co sa tyka pamatovych narokov tak bude mat maximalne v pamati 8 uzlov, kde DFS bude mat maximalne N uzlov (N hlbka celeho strom) a BFS v najhorsom pripade bude mat 2^9 uzlov +2 , kedze bude mat uz v pamati vsetkych nasledovnikov uzlov 8 urovne (az na 2 z nasho cieloveho uzlu) , takze po pamatovej narocnosti bude DLS najlepsie, rovnako je to optimalna metoda ktora vzdy najde riesenie v pozadovanej hlbke. Riesenie by naslo aj DFS (v horsom case a hornej pamatovej narocnosti ak teda ratame s worst case polohou cieloveho uzla v 8 urovni , teda ze riesenie nebude hned na prvej ceste ktorou sa pusti aj DFS aj DLS) a rovnako by ho naslo aj BFS avsak s ovela horsimi pamatovymi narokami.

4. Náhodná procházka je postup lokálního prohledávání, při němž se algoritmus opakováně slepě (náhodně) přesouvá z aktuálního stavu do sousedního a pouze si pamatuje, který dosud navštívený stav byl nejlepší. Nakreslete graf vývoje kvality aktuálního stavu jako funkci počtu navštívených stavů. Do stejného grafu přidejte průběh kvality aktuálního stavu pro simulované žíhání. Grafy volte tak, aby ilustrovaly hlavní rysy obou algoritmů, a stručně okomentujte, jak je to v nich vidět.



Myslim si ze toto by mohlo byt fajn, nakolko pri simulovanom zihani bude dochadzat k vyberu horsich susedov len zo zaciatku prehladavania nakolko casom sa bude znizovat teplota a tym padom aj pravdepodobnosť vyberu horsieho suseda cize po urcitom case bude ta kvalita len rast , kde naopak pri nahodnej prochazke k tomu moze dochadzat cely cas prehladavania nakolko nezalezi ci susedny uzel je lepsi/horsi vzdy len nahodne nejaký vyberieme a pamatame si najlepsi aky sme zatial nasli.

// cakam na odpoved od dedobeda na dc uvidime

8. Jaký je rozdíl mezi klasifikačním a regresním problémem? Pro každý tento problém uveďte alespoň dva různé modely, které lze trénovat pro jejich řešení. Jakou podobu mají trénovací data pro tyto modely/problémy? Jaké objektivní funkce se používají při trénovaní těchto modelů?

2023/2024 riadny termin

1. V jistém, tedy plně pozorovatelném a deterministickém, prostředí hledá agent cestu do cílového stavu pomocí prohledávání stavového prostoru, kde každý uzel odpovídá stavu prostředí. Jakému prostředí odpovídá prohledávání AND-OR stromu? Jakým akcím/jevům odpovídají hrany vedoucí z uzlu AND, resp. uzlu OR?

Akemu prostredi odpovida prohledavani AND-OR stromu ->

2. Máme úlohu CSP s proměnnými X_1, X_2 a X_3 a odpovídajícími doményami $D_1 = D_2 = D_3 = \{1, 2, \dots, 10\}$. Pro omezení $x_1^2 < x_2, 3x_2 < x_1 + x_2$ a $x_3 < 2x_1$ proveděte inferenci uzlů, hran a cest a zapište výsledek.

	1	2	3	4	5	6	7	8	9	10
x_1	X	X	X	X	X	X	X	X	X	X
x_2	X	X	X	X	X	X	X	X	X	X
x_3	X	X	X	X	X	X	X	X	X	X

$$D_1 = D_2 = D_3 = \{1, 2, \dots, 10\}$$

// check pls +1

3. Mějme prohledávací problém, kde každý stav má právě dva potomky a nejdělsší cesta od kořene (hloubka 0) k uzlu měří 20 kroků. V prostoru jsou tři řešení, v hloubce 5, 10 a 15, a žádné z nich není následníkem jiného. Pro prohledávání do šířky (BFS) a do hloubky (DFS) uveďte, jak kvalitní řešení naleznou (v krocích) a kolik paměti spotřebují (v uzlech) v nejlepším a nejhorším případě. U BFS předpokládejte udržování množiny CLOSED, u DFS nikoliv.

BFS -> najde riesenie v hlbke 5, takze riesenie bude mat 5 krokov (dlzka cesty)

-> v najlepsom pripade bude riesenie prvy skumany uzol v 5 urovni
 teda v closed bude korenovy uzol , 2 uzly 1 urovne, 4 uzly 2 urovne, 8 uzlov 3 urovne
 16 uzlov 4 urovne a v open bude 32 uzlov 5 urovne (kde 1 je prave ten cielovy) teda celkovo = $1 + 2 + 4 + 8 + 16 + 32 = 63$ uzlov bude celkovo v pamati

-> v najhorsom pripade bude bude cielovy uzol nas posledny preskumavany uzol{5} v hlbke 5:

teda v closed bude 62 uzlov -> vsetky uzly 0,1,2,3,4,5 urovne az na nas hladany a v open bude 62 uzlov v hlbke 6 co su potomkovia vsetkych uzlov urovne 5 az na nas cielovy stav celkovo teda $62+62 = 124$

DFS -> best case najde riesenie v hlbke 5 , cize v 5 krokoch

-> v OPEN budu susedia ktorych sme cestou nadol stromu nepreskumali teda -> 1 z 1 urovne, 1 z 2 urovne, 1 z 3 urovne, 1 z 4 urovne a 1 z 5 urovne celkovo 5 uzlov

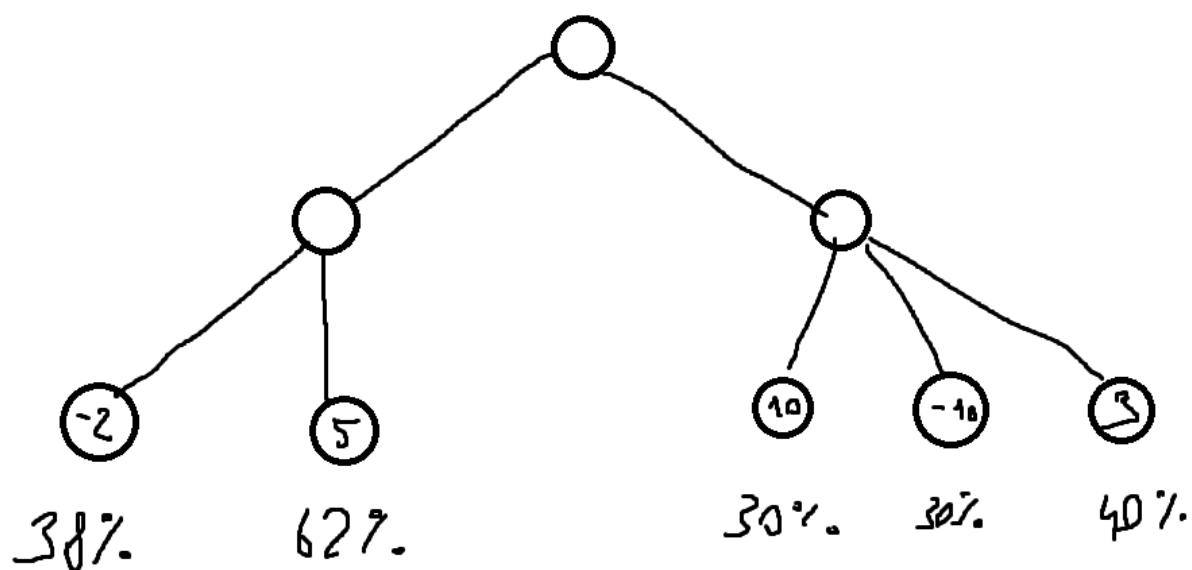
DFS -> worst case najde riesenie v hlbke 15, teda riesenie bude v 15 krokoch

-> v OPEN rovnako ako pri best case budu vsetci susedia ktorych sme cestou nepreskumali teda 15 (tu je avsak mozne povedat ze najhorsi case bude ked sice najde riesenie v hlbke 15 ale uz predtym preskumal nejaku cestu upne na spodok stromu teda o dlzke 20 -> cize pamatova narocnost 20 uzlov)

4. Jak byste upravili minimax, pokud hráte velký počet her proti soupeři, který hraje náhodně? Potom předpokládejte, že jste navíc schopni během těchto her trénovat model, který odhaduje, jak bude soupeř hrát (má na výstupu kategorické rozložení pravděpodobnosti, že si soupeř v daném stavu vybere danou akci). Jak dále upravíte minimax, aby výstupy z tohoto modelu zužitkoval?

Uprava ak hraje nahodne -> budem vyberat vetvu kde je najvaci priemerny zisk a nie ten kde mam najmensiu stratu ako to je pri klasickom minmaxe

Vysledky z modelu ->



Uvazujem ze mam nasledujuci hru , vysledky modelu mi hovoria pravdepodobnosti vyberu listovych uzlov v jednotlivych podstromoch

- v kazdom podstrome si urcim % zisku a % straty, rovnako ako velka je dana strata/zisk
- nasledne vyberiem podstrom s najvacsiou % zisku , ak su % rovnake tak vyberiem ten podstrom kde je vacsi priemerny zisk pre danu pravdepodobnosť

5. Na trojdimenzionálních datech pro dvě třídy jsme natrénovali gaussovský klasifikátor. Pomocí Bayesova vzorce můžeme pro nově příchozí testovaný vzor \mathbf{x} spočítat, s jakou aposteriorní pravděpodobností tento vzor patří do které třídy. Napište tento vzorec a vysvětlete, jak přesně vyhodnotíme jednotlivé výrazy v tomto vzorci (tak, aby podle vašeho popisu bylo možné tento výpočet implementovat). Předpokládejte, že již máte k dispozici funkci pro vyhodnocení hustoty pravděpodobnosti vícerozměrného gaussovského rozložení $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Vyjmějte všechny parametry, kterými je tento natrénovaný gaussovský klasifikátor popsán, a napište, jak se tyto parametry využijí (do kterých výrazů se dosadí) při vypočtu aposteriorních pravděpodobností tříd. Jakou mají tyto parametry podobu? Kolika reálnými čísly jsou všechny tyto parametry reprezentovány? Jak jsme tyto parametry na trénovacích datech odhadli?

Prva cast otazky: asi ?

Bayes rule:

$$P(\text{grenade}|\text{heavy}) = \frac{P(\text{heavy}|\text{grenade})P(\text{grenade})}{P(\text{heavy})}$$

Druha cast : vsetky parametre

mame 2 triedy -> 2 gaussovky -> pre kazdu potrebujeme:

vektor strednych hodnot -> 3 realne cisla (kedze su to 3D data)

kovariacnu maticu -> 9 realne cisla

apriory pravdepodobnost -> 1 realne cislo

celkovo $2 * (3 + 9 + 1) = 26$

Tieto hodnoty sme odhadli nasledovne:

$$P(c) = \frac{N_c}{\sum_k N_k} \quad P(\text{blue}) \frac{400}{400 + 600}$$

Aprior pravdepodobnost

Kovariacna matica a vektor strednych hodnot:

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} \mathbf{x}_{cn} \quad \boldsymbol{\Sigma}_c = \frac{1}{N_c} \sum_{n=1}^{N_c} (\mathbf{x}_{cn} - \boldsymbol{\mu}_c)(\mathbf{x}_{cn} - \boldsymbol{\mu}_c)^T$$

Jak sa tieto parametry využijú (do ktorých výrazov sa dosadia) pri vypočte aposteriornich pravdepodobností tried ? //TODO

Evidence -> sa spocita ako suma cez vsetky triedy

$$\text{sum}(P(\text{trieda_i}) * P(\text{pozorovanie} | \text{trieda_i}))$$

6. Jak odhadnu podmíněnou pravděpodobnost, že dostanu infarkt, když budu tlustý $P(\text{infarkt}|\text{tlustý})$, pokud mám statistiky pro vzorek populace, kde je u každého člověka záznam o tom, zda byl tlustý a zda prodělal infarkt? Tuto pravděpodobnost přitom chceme odhadnout pomocí *metody maximální věrohodnosti* (Maximum Likelihood estimation).

$P(\text{infarkt} | \text{tlusty}) = \text{pocet tlstych s infarktom} / \text{pocet tlstych}$ //check

7. Pokud u metody k-nejbližších sousedů (K-nearest neighbours) zvolíme K=1, nebude pravděpodobně výsledný klasifikátor dobře generalizovat pro nová data. Co to znamená? Proč k tomu v tomto případě pravděpodobně dojde?

To ze klasifikator nebude dobře generalizovat znamena ze na novych doposial nevidenyh datach (inych ako trenovacich) nebude model vraciť spravne vysledky, rozhodovacia hranica pri K=1 bude priliz komplexna

8. Uveďte dva příklady diskriminativně trénovaných modelů pro klasifikaci, které umožňují vyhodnocení aposteriorních pravděpodobností tříd pro vstupní (testovaný) vzor. Alespoň pro jeden z těchto dvou příkladů uveďte, jak se tyto aposteriorní pravděpodobnosti konkrétně spočítají pro příchozí testovaný vzor \mathbf{x} . Na základě uvedeného výpočtu aposteriorních pravděpodobností vysvětlete, proč se jedná o diskriminativně trénovaný klasifikátor.

2 priklady diskriminativne trenovanych modelov pre klasifikaci -> logisticka regresia, NN s softmaxom/log sigmoidou , inac to moze znamenat aj regresni model, SVM
vzorec vypoctu aposteriornej pravdepodobnosti pre logisticku regresiu (binarny klasifikator)

2-class linear logistic regression models directly posterior probability of class $c = 1$ as

$$P(c = 1|\mathbf{x}) = \sigma(\hat{\mathbf{x}}^T \mathbf{w})$$

Probability of the other class $c = 0$ is

$$P(c = 0|\mathbf{x}) = 1 - P(c = 1|\mathbf{x})$$

$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$

Jedna sa o diskriminativne trenovany klasifikator nakolko priamo modelujeme vypocet aposteriornej pravdepodobnosti triedy naroziel od generativnych klasifikatorov kde sme sa museli naucit rozdelenie pozorovanych dat pre kazdu triedu, rovnako vypocitat aprior pravdepodobnost pre kazdu triedu a nasledne pomocou Bayesovho vzorca sme pocitali aposterorne pravdepodobnosti kazdej triedy.

9. Jaké znáte vědní podobory umělé inteligence? (jen vyjmenovat)

Vedne podobory = strojove uceni, NN, evolucne algo, computer vision, robotika, zpracovanie reci, zpracovanie prirodzeneho jazyka, expertne systemy

Obory s ktorymi suvisi = ekonomia, statistika, psychology, neurovedy, logika, filozofia ...

10. Co znamená, že nějaká metoda (model) strojového učení řeší úlohu end-to-end? Heslovitě vypište možné výhody a nevýhody.

End-to-end:

je pristup kedy zložitu úlohu učíme v 1 kroku

vstupom su teda raw data

chyba je vypocitana na cielovej ulohe

jeden model je celý optimalizovaný pre dosiahnutie celkovo minimalnej chyby

Vyhody:

- potrebuje GT(ground truth) len pre cielovu ulohu ktorá je ta co nas skutočne zaujíma
 - celkovy system je jednodussi, je to len jedna NN namiesto kolekcie samostatnych systemov
 - Je mozne dosiahnut lepsich vysledkov, cely model je optimalizovany pre cielove pouzitie, obecne sa nedeje ze by sa v pociatocných fazach zpracovania stratila podstatna informacia pre riesenie ulohy
 - v niekorych pripadoch je takyto system vypocetne efektívnejši, nakolko sa uci pocitat len dolezite veci pre danu ulohu

Nevyhody:

- vacsinou potrebujeme vacsie mnozstvo trenovacich dat
 - vacsinou sa take modely dlhsie trenuju
 - moze byt zlozitejšie interpretovať chovanie daneho modelu

11. Uveďte aspoň jednu chybovou (loss) funkci, kterou můžete použít pro binární klasifikaci. Jaká bude v tom případě aktivační funkce na výstupu neuronové sítě a jak můžeme interpretovat jednotlivé výstupy sítě?

Priklad loss funkcie -> cross entropy

aktivácia funkcia -> sigmoid ?? //check

Jak muzeme interpretovat jednotlive vystupy siete ?

- pravdepodobnosť že nase vstupne data patria do triedy X a pravdepodobnosť že patria do triedy Y ako komplement teda $1-p(X | \text{data})$

12. Co přesně modelují autoregresivní jazykové modely ve smyslu „pravděpodobnosti“? Ideálně napište matematický výraz s krátkým komentářem.

$$P(w_1, w_2, w_3, \dots, w_n) = \prod P(w_i | w_{i-1}, w_{i-2}, \dots, w_1)$$

Autoregresivne jazykove modely v zmysle "pravdepodobnosti" modeluju pravdepodobnosť nasledujuceho slova na zaklade predchadzajucej sekvencie slov. Teda miesto pravdepodobnosti celych viet je ta pravdepodobnosť rozdelena na súčin pravdepodobností všetkých slov, podmienených predchadzajúcimi slovami vo vete.

13. Vytvořte aplikaci strojového učení, ke které neexistují veřejné datasety. Jaké kroky budete muset podniknout pro přípravu dat pro experimenty? Jaké budete potřebovat nástroje? Na co si případně musíte dát pozor? (Napište stručné odrážky.)

pridem na Fiver a kupim si za 30 eur inda co mi to spravi

Kroky - najst vhodne data na internete a stiahnut si ich

- ak nie su anotovane tak nejakym SW ich anotovat
- prefiltrovat
- odstranit chybne data
- normalizovat
- pri obrazkovych datach mozme urobit data augmentation

Naco si dam pozor

- licencia dat
- nevhodnost dat pre moju ulohu
- nevyvazenosť dat
- atď sracky pisat

14. Co je to alignment v kontextu velkých jazykových modelů? Vyjmenujte i cíle, kterých tímto procesem můžeme chtít dosáhnout nebo co chceme potlačit (odrážkový seznam). Pokud možno, pojmenujte jednu techniku, která se pro alignment používá.

Model Alignment

- We (sometimes) want the models to be useful to humans
 - (and NOT to provide bad press for the companies)
 - (and NOT to cause lawsuits for the companies)
- Some people may be:
 - offended by the models for many reasons
 - offended that the model talks to someone else about some topics
 - offended that models are biased
 - angry that models “harm children”
 - angry that models provide some “harmful” information
- If we get AGI? How will we assure that it acts in the “best interest” of humanity?

uprava modelu tak aby sa choval primerane a neoffendol nasich cielovych uživatelov v ziadnom ohlade, cim predchadzame sudnym sporom a podobne voci nasej spoločnosti

1. V jistém, tedy plně pozorovatelném a deterministickém, prostředí hledá agent cestu do cílového stavu pomocí prohledávání stavového prostoru, kde každý uzel odpovídá stavu prostředí. Jakému prostředí odpovídá prohledávání prostoru belief stavů? Čemu odpovídá jeden belief state?

1 belief state odpoveda mnozine stavov v ktorych sa momentalne agent moze nachadzat
Jakemu prostredi odpovida prohledavani prostoru belief stavu - ??????? //TODO pls

4. Uvažujte sekvenční hru dvou hráčů s nulovým součtem, stavový prostor hry je úplný binární strom o hloubce D . Jakou časovou složitost má algoritmus minimax při hledání nejlepšího prvního tahu (začíná hráč, který maximalizuje)? Potom navrhnete, jak minimax zrychlit pomocí paralelizace, a uvedete, jak se ona časová složitost změní v ideálních podmírkách (nekonečno procesorů, nulová komunikační latence atp.).

rebel129 Today at 6:42 PM
ja som tam písal tuším niečo ako $2^{\alpha}(D+1) - 1$... teda že sa prehľadajú všetky uzly a pri paralelizácii je možné tie listové uzly vždy vyhodnocovať zároveň takže by zložitosť bola v podstate D ... mal som za to 3/4b
 1

7. K čemu je parametr K u metody k-nejbližších sousedov (K-nearest neighbours)? Jak tento parametr zvolíme a jak jeho nastavení ovlivní úspěšnost klasifikace?

Parameter K udava pocet najblizsich susedov z ktorych budeme urcovat do ktorej triedy zaradime vstupne dato. Chceme tento parameter nastaviti na neparne cislo aby nam nemohla nastat "remiza" medzi triedami, vo vacsine volime $K = 9$.

9. Jaké jsou různé aspekty inteligence? (jen vyjmenovat) Musí podle vás agent vykazovat všechny tyto aspekty nebo jejich určitou podmnožinu, abyste ho považovali za inteligentního?
10. Jaký je význam pojmu „Deep learning“? Kam v tomto případě vkládáme naši znalost o řešeném problému?
 - znalosti o problému ukládáme v rámci NS → v nastavení vah u jednotlivých přechodů mezi neurony
 - extrakci příznaků (předzpracování vstupních dat) nedefinujeme ručně, ale použijeme hlubokou NS, která se extrakci příznaků naučí sama (dáme jí víc dat pro trénování)
 - složitější a výpočetně náročnejší, ale výsledky jsou lepší a není potřeba ručně vymýšlet předzpracování dat
 - extrém - end-to-end přístup → jedna neuronová síť (uvnitř může být složitá), která jako vstup bere naše data a vrátí nám očekávaný výstup → neobsahuje mezikroky
11. Jak nejfektívnejši a obecně pomocí jedné matematické operace (nebo dvou) můžete zapsat plně propojenou (lineární) vrstvu neuronových sítí? Napište matematicky danou funkci. Označte vstup, výstup, učené parametry.

Linear layer

$$\begin{bmatrix} 1 & 0 & 3 \end{bmatrix} \times \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = -1, 0, -1, -1$$



$$\begin{bmatrix} 1 & 0 & 3 \end{bmatrix} \times \begin{bmatrix} 2 & 0 & -1 & 1 \\ -1 & 2 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 0 & -1 & -1 \end{bmatrix}$$

$$f(\mathbf{x}; \mathbf{W}) = \mathbf{W}\mathbf{x}$$

vstup je \mathbf{x} , ucene parametri su vahy \mathbf{W} , a vystup je $f(\mathbf{x}; \mathbf{W}) = \mathbf{W}\mathbf{x}$, vektor realnych cisel je vystup

ak si dobre pamatam, tak som tam dal nieco ako $[1, x_1, \dots, x_n] * [w_0, w_1, \dots, w_n] = [\text{class_A_P}, \text{class_B_P}, \dots]$ a mam za to fullku.
Podla mna toto je ta jedna matematicka operacia a ked to iste das klasicky do $f(x) = ax + b$ su 2 operacie

12. Vysvetlete funkci attention vrstvy. Stačí vysvetlit základní myšlenku bez matematiky. Jak se přenáší informace, co se učí? Můžete ilustrovat diagramem.

Attention je mechanizmus (vrstva sítí), která umožňuje, aby se síť naučila "vytahovat" potřebnou informaci ze zdrojových dat. Je to pevné spojení s celými zdrojovými daty, ale síť si dynamicky podle svého vnitřního stavu (aktivací) dokáže nastavovat váhu jednotlivých částí zdrojových dat --- naučí se přiřazovat vysokou váhu užitečným datům. Attention se nepřenese informaci z jednoho zdrojového "místa", ale může přiřadit libovolný počet nenulových vah. Třeba u překladu jazyka v ukázce je to jednoznačně potřeba.,

13. Jak zajistíte při provádění většího počtu ML experimentů v týmu, aby tento proces byl dlouhodobě udržitelný a reprodukovatelný a aby členové týmu mohli spolupracovat a mít přehled o výsledcích experimentů celého týmu? Uveďte vhodné SW nástroje. (Napište stručné odrážky.)

teams

google sheets

discord
hocijake komunikacie kanaly
github alebo ine verzovacie systemy
cloud/db pre dataset
jira - issue tracking

git, jira, ms teams/zoom/slack, robit zalohy a ukladat veci na cloude, pouzivat industry standard toolky na pracu (pytorch, tensorflow)

14. Jakým nejjednoduším postupem můžete použít velký jazykový model pro řešení konkrétní úlohy zpracování textu (například sumarizace, detekce klíčových slov, hodnocení odpovědí v testu ze SUI)?

tak tu som napisal, ze by som sa registroval na openAI API, nastavil mu kontext na ulohu, ktoru potrebujem poslat to do chatgpt, sparsoval a mam vysledok za 4b