**Využitie neurónových sietí v moderných aplikáciách**

Gelnica 2023 Tomáš Varga

Gymnázium, SNP 1, 056 01 Gelnica

**STREDOŠKOLSKÁ ODBORNÁ ČINNOSŤ**

**Využitie neurónových sietí v moderných aplikáciách**

Gelnica 2023 Tomáš Varga

[1. Úvod 3](#_Toc128427722)

[2. Charakterizácia neurónových sietí 4](#_Toc128427723)

[2.1 Základy strojového učenia 4](#_Toc128427724)

[2.2 Čo sú to neurónové siete? 4](#_Toc128427725)

[2.3 Neurónová sieť 4](#_Toc128427726)

[2.4 Typy neurónových sietí 4](#_Toc128427727)

[2.4.1 Perceptron 5](#_Toc128427728)

[2.4.2 Viacvrstvový perceptrón 6](#_Toc128427729)

[2.4.3 CNN – konvolučná neurónová sieť 6](#_Toc128427730)

[2.4.4 RNN - Rekurentné neurónové siete 6](#_Toc128427731)

[2.5 Trénovanie neurónových sietí 7](#_Toc128427732)

[2.5.1 Učenie pod dohľadom 7](#_Toc128427733)

[2.5.2 Učenie bez dohľadu 7](#_Toc128427734)

[2.5.3 Učenie posilňovaním 8](#_Toc128427735)

[3. Využitie neurónových sietí 9](#_Toc128427736)

[3.1 OpenAI DALL-E 2 9](#_Toc128427737)

[3.2 OpenAI ChatGPT 3.0 10](#_Toc128427738)

[3.3 Algoritmus na udržanie divákov v službe YouTube 11](#_Toc128427739)

[4. Praktická časť 13](#_Toc128427740)

[5. Záver 17](#_Toc128427741)

[6. Zoznam bibliografických odkazov 18](#_Toc128427742)

1. Úvod

Túto tému som si vybral, pretože som chcel ľudí poučiť o tom, že umelá inteligencia sa neobmedzuje len na filmy a hry. Môžete ju používať aj Vy, pri úlohách ako je napríklad predikcia textu, rozpoznávanie obrázkov, preklad textu a mnoho ďalšieho. Neurónové siete sú jedným z najčastejšie používaných typov algoritmov umelej inteligencie.

Umelá inteligencia (AI) je v súčasnosti jednou z najzaujímavejších a najrýchlejšie sa rozvíjajúcich oblastí technológií. Od samojazdiacich áut až po hlasových asistentov, ako sú Siri a Alexa. AI už teraz mení spôsob, akým žijeme a pracujeme a jej vplyv v nasledujúcich rokoch len porastie. Keďže sa táto technológia stáva čoraz rozšírenejšou, je nevyhnutné, aby ľudia pochopili, čo to je, ako funguje a aké sú jej dôsledky.  
Vzdelávanie ľudí o umelej inteligencii nie je len otázkou vzdelávania novej generácie informatikov a inžinierov, ale týka sa nás všetkých bez ohľadu na náš pôvod alebo profesiu. Či už ste študent, majiteľ firmy alebo dôchodca, pravdepodobne sa s umelou inteligenciou v tej či onej podobe stretávate vo svojom každodennom živote.

Podstatou umelej inteligencie je vytváranie strojov, ktoré sa dokážu učiť z údajov a prijímať rozhodnutia na základe tohto učenia. To sa zásadne líši od tradičného prístupu k programovaniu, ktorý zahŕňa písanie presných inštrukcií, ktorými sa má stroj riadiť. Pri umelej inteligencii sa stroju poskytne súbor údajov, určených na trénovanie, ktoré sa použijú na identifikáciu vzorcov a vytváranie predpovedí. Tento prístup umožňuje AI vykonávať úlohy, ktoré by boli pre tradičné počítače nemožné, ako napríklad rozpoznávanie obrázkov, porozumenie prirodzenému jazyku a hranie zložitých hier, napríklad šachu.

Nástup umelej inteligencie však so sebou prináša aj množstvo výziev a etických problémov. Napríklad, keďže stroje sú čoraz schopnejšie vykonávať úlohy, ktoré tradične vykonávajú ľudia, existuje riziko, že by mohli vytlačiť pracovníkov v niektorých odvetviach. Existuje aj riziko zaujatosti systémov umelej inteligencie, ktoré môžu udržiavať diskrimináciu a posilňovať sociálne nerovnosti. Preto je dôležité, aby sme ľudí poučili o etických aspektoch súvisiacich s ňou, aby mohli uskutočňovať informované rozhodnutia o tom, ako by sa mala vyvíjať a používať.

Okrem etických obáv existujú aj praktické obavy. Je dôležité zabezpečiť, aby ľudia mali potrebné technické zručnosti na prácu so systémami AI, či už pri ich vývoji alebo používaní pri práci. To môže zahŕňať výučbu ľudí ako programovať v jazykoch, ako je napríklad Python alebo poskytovanie školení o konkrétnych nástrojoch a platformách už vyvinutých umelých inteligencií.

Vzdelávanie ľudí o umelej inteligencii je nevyhnutné na to, aby sme boli pripravení na príležitosti a výzvy, ktoré táto technológia prinesie. To znamená nielen poskytovať technické školenia, ale aj učiť ľudí o etických a praktických aspektoch súvisiacich s AI. Týmto spôsobom môžeme zabezpečiť, aby sme jej silu využívali tak, aby bola prospešná pre nás všetkých a zároveň minimalizovali riziká a obavy, ktoré s ňou súvisia.

2. Charakterizácia neurónových sietí

2.1 Základy strojového učenia

*„Učenie sa nových zručností je viac o cvičení ako o pasívnom získavaní vedomostí.“* (Pan, 2016, str. 15)

Umelá inteligencia je koncept, s ktorým sa vedci pohrávajú už od 50. rokov minulého storočia. Strojové učenie je typ programu v ktorom sa počítač učí sám. Mnohé počítače dnes napríklad využívajú strojové učenie, na základe takzvaných neurónových sietí, na zvýšenie presnosti pri rozpoznávaní obrázkov (napríklad tvárí), reči a predpovedaní textu.

2.2 Čo sú to neurónové siete?

Umelé neurónové siete pozostávajú z viacerých vrstiev neurónov. Neuróny sú základnými výpočtovými jednotkami v rámci neurónovej siete. Každý neurón prijíma vstupy od všetkých ostatných neurónov v predchádzajúcej vrstve a odovzdáva svoje výstupy všetkým ostatným neurónom v ďalšej vrstve. Každý neurón v sieti je prepojený s každým iným neurónom v sieti, takže informácia môže prúdiť celou sieťou prechodom z neurónu na neurón. Najbežnejším spôsobom trénovania takejto neurónovej siete je použitie niečoho, čo sa nazýva Supervised Learning alebo teda Učením pod dohľadom.

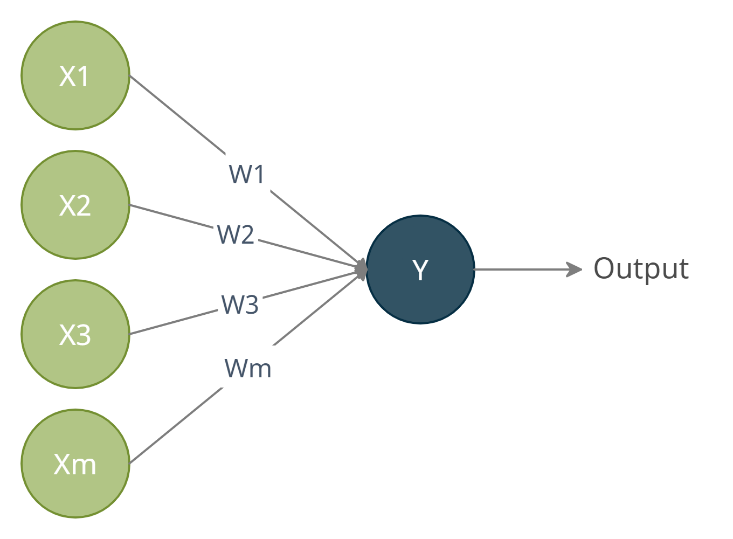
2.3 Neurónová sieť

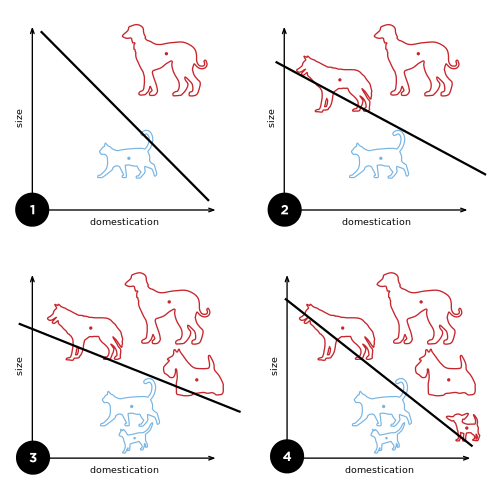
Neurónová sieť sa skladá z troch alebo viacerých vrstiev. V jednoduchej neurónovej sieti zobrazenej nižšie sú tri vrstvy: vstupná vrstva, skrytá vrstva a výstupná vrstva. Vstupná vrstva prijíma vstup od používateľa vo forme výzvy, ale môže prijímať aj vstup vo forme obrázku (pri rozpoznávaní obrazu). Ďalej sú skryté vrstvy, zložené z neurónov, ktoré sú prepojené a tvoria komplexnú výpočtovú jednotku. Výstupná vrstva je miestom, kde sa generuje výstup v reakcii na špecifické vstupy prijaté v predchádzajúcich vrstvách. Výstupom môže byť slovo, kategória, odpoveď áno alebo nie, súradnice niečoho na určenom obrázku alebo čokoľvek, čo používateľ chce, aby sa zobrazilo ako odpoveď na prijaté vstupy. Neurónové siete fungujú vďaka prepojeniam medzi jednotlivými vrstvami. Ako už bolo uvedené a ako môžete vidieť na obrázku nižšie, všetky uzly musia byť prepojené pomocou prepojení, ktoré sa nazývajú aj synapsie. Každá z týchto synapsií môže mať hodnotu, ktorá určuje silu spojenia medzi dvoma uzlami v sieti. Ako sa sieť učí, tieto spojenia sa menia, aby lepšie odrážali požadovaný výsledok. Komplexné váhy neurónovej siete by nakoniec mali viesť k tomu, že pre každý vstup, ktorý jej je zadaný, bude na výstupe požadovaný výsledok.

2.4 Typy neurónových sietí

2.4.1 Perceptron

Perceptron je najjednoduchšia neurónová sieť, ktorá sa používa dodnes. Pôvodný perceptrón, ktorý vynašli McCulloch a Pitts v roku 1943, pozostával len z jednej vrstvy neurónov usporiadaných do tvaru mriežky, pričom každý uzol bol striedavo pripojený k susednému uzlu, ako je znázornené nižšie.



Táto sieť je typom lineárneho kvalifikátora, čo znamená, že ak sú dané dva vstupy, vypočíta výstup na základe lineárnej priamky niekde v grafe týchto dvoch vstupov. Na obrázku nižšie vidíme graf s osou y, ktorá predstavuje veľkosť a osou x, ktorá predstavuje udomácnenie. V tomto príklade sa pokúšame naučiť neurónovú sieť rozlišovať medzi mačkami a psami. Psy sú označené červenou farbou a mačky modrou farbou a môžeme vidieť, ako Perceptron posúva svoju hraničnú čiaru, aby sa prispôsobil našim novým príkladom, keď sa do siete privádza čoraz viac údajov.

2.4.2 MLP – Viacvrstvový perceptrón

MLP je v súčasnosti najpoužívanejšia neurónová sieť. Je podobná originálnej sieti Perceptron, ale pozostáva z viacerých vrstiev ako bolo opísané vyššie, pričom jedna slúži ako vstupná, druhá ako výstupná a medzi nimi sú skryté vrstvy. MLP možno použiť na trénovanie zložitejších systémov ako perceptrón, pretože je schopná spracovať viac informácií a tiež sa dokáže naučiť nelineárne vzťahy medzi rôznymi vstupmi a výstupmi. Je základom pre ďalšie typy neurónových sietí, ako je konvolučná neurónová sieť (CNN), o ktorej sa podrobnejšie zmienime v nasledujúcej časti.

2.4.3 CNN – konvolučná neurónová sieť

CNN dokážu analyzovať vizuálne údaje, ako sú obrázky alebo videá, klasifikovať údaje na základe určitých atribútov a nachádzať v nich vzory.

Bežne sa používajú v úlohách, ako je počítačové videnie, rozpoznávanie tváre, spracovanie reči a spracovanie prirodzeného jazyka.

Pozostávajú z viacerých konvolučných vrstiev, ktoré fungujú ako extraktory kľúčových prvkov, nazývané aj filtre a jednej plne prepojenej vrstvy, ktorá sa používa na prijatie klasifikačného rozhodnutia.

Medzi zaujímavé prípady využitia CNN patrí ich využitie pri klasifikácii rakovinových buniek na základe ich genetickej výbavy pomocou snímok buniek zhotovených pod mikroskopom, generovanie falošných videí, v ktorých dokážu vytvoriť dojem, že hovoriaci hovorí niekoľkými jazykmi čím pomáhajú pri prístupnosti. Vďaka nedávnemu pokroku v tejto oblasti vrátane vývoja pokročilejších techník hlbokého učenia, ako aj zlepšeniam v oblasti výpočtového výkonu sú CNN v súčasnosti schopné vykonávať úlohy, ktoré by pred niekoľkými rokmi boli veľmi náročné alebo dokonca nemožné. Jednou z takýchto úloh pre tieto CNN je rozpoznávanie vzorov ciest pre samojazdiace autá. Pomocou snímok z úrovne ulice zozbieraných prostredníctvom kamery na čelnom skle samojazdiaceho vozidla a ich kombináciou s údajmi z palubných senzorov vozidla dokážu tieto CNN extrahovať dôležité prvky zo snímok, priradiť ich k vopred definovaným cestným vzorom a rozpoznať dopravné značky.

2.4.4 RNN - Rekurentné neurónové siete

V porovnaní s ostatnými typmi neurónových sietí sa RNN môžu odvolávať na informácie z predchádzajúcich krokov výpočtového procesu namiesto toho, aby v každom kroku vždy vytvárali novú predpoveď. Tiet siete sa často využívajú v jazykových modeloch, ktoré potrebujú vedieť, čo bolo povedané predtým. Napríklad nedávno vyvinutý algoritmus ChatGPT od spoločnosti OpenAI využíva rekurentné neurónové siete na generovanie konverzačných textov podobných ľudským. Tento model bol trénovaný tak, že mu boli poskytnuté tisíce prepisov konverzácií medzi ľuďmi a potom sa sám naučil napodobňovať tieto konverzácie a generovať nové odpovede. Toto práve poukazuje na užitočnosť týchto RNN, pretože sa učí zo súboru údajov obsahujúceho nielen slová vyslovené človekom, ale aj ich význam na základe kontextu použitého v predchádzajúcej konverzácii, čoho by iné typy neboli schopné. Túto službu si môžete dokonca bezplatne vyskúšať na adrese https://chat.openai.com/.

Využitie je aj v medicíne, kde sa liečba môže špeciálne prispôsobiť každému pacientovi na základe jeho anamnézy a reakcií na predchádzajúcu liečbu.

2.5 Trénovanie neurónových sietí

2.5.1 Učenie pod dohľadom

Predstavte si, že existuje súbor údajov pozostávajúci z opisov osôb spolu s ich pohlavím. Teraz predpokladajte, že chcete predpovedať pohlavie osoby na základe jej opisu. Pomocou učenia pod dohľadom by ste vycvičili neurónovú sieť použitím všetkých opisov mužov a žien ako tréningové dáta. Počítač by sa naučil predpovedať, do ktorej kategórie patrí nový príklad a použil by to na predpovedanie nových príkladov v budúcnosti. Aby sa neurónová sieť naučila, ktoré opisy patria do ktorej kategórie, musia jej byť zadané príklady označené správnymi kategóriami.

1. Pri danom súbore údajov vstupov/príznakov a požadovaných výstupov/značiek spustite algoritmus učenia pod dohľadom s cieľom vytrénovať model schopný predpovedať výstupné značky pre neoznačené vstupné príznaky. Tento proces sa nazýva trénovanie modelu;
2. Predpovede na základe nových vstupných údajov tým, že tento vstupný údaj vložíte do natrénovaného modelu a necháte ho vygenerovať výstupnú značku.

2.5.2 Učenie bez dohľadu

V porovnaní s učením pod dohľadom môžeme pomocou učenia bez dohľadu vytvoriť modely, ktoré sa môžu trénovať bez označených údajov. Algoritmus sa snaží nájsť vzory vo vstupných údajoch, aby predpovedal požadované výsledky. Najčastejšie prípady použitia učenia bez dozoru sú zhlukovanie, redukcia dimenzionality a hľadanie anomálií v údajoch.

Zhlukovanie sa používa na zoskupovanie dátových bodov podľa ich podobnosti alebo závislosti. Napríklad by sme mohli zhlukovať zákazníkov na základe ich nákupného správania a vytvárať tak nové propagačné stratégie na oslovenie podobných skupín zákazníkov. Redukcia dimenzionality sa používa na zmenšenie rozmerov vstupného súboru údajov, aby bol lepšie zvládnuteľný (napríklad na rýchlejšie načítanie do databázy). Funguje tak, že sa vyberú najdôležitejšie aspekty údajov, čím sa urýchli proces strojového učenia v ďalšom kroku. Detekcia anomálií je typ nekontrolovaného učenia, ktorý sa používa na vyhľadávanie neobvyklých vzorov v údajoch, ktoré môžu poukazovať na podvod alebo iné škodlivé činnosti. To môže byť užitočné pre banky, ktoré musia triediť stovky platieb za sekundu a potrebujú rýchly spôsob, ako odhaliť možné podvody.

2.5.3 Učenie posilňovaním

Cieľom posilňovacieho učenia je vyvinúť algoritmus, ktorý určí ako sa správať za účelom maximalizovania signálu odmeny. Časom sa tieto algoritmy môžu učiť prostredníctvom pokusov a omylov, ako aj skúseností, čo následne prinesie stále lepšie výsledky. Na rozdiel od učenia pod dohľadom nemusí posilňovacie učenie na určenie odpovede používať označené tréningové údaje. Jednoducho môže použiť cieľ a akcie vykonané na dosiahnutie tohto cieľa. To umožňuje systémom umelej inteligencie, ktoré využívajú posilňovacie učenie, prispôsobovať sa dynamickému prostrediu a učiť sa prijímať opatrenia za pochodu.

V praxi to môže znamenať, že sieť používa na svoje rozhodnutie učenie pomocou posilňovania a potom dostáva odmenu za tie, ktoré sú považované za "správne". Tento typ učenia sa v súčasnosti využíva v mnohých oblastiach vrátane samojazdiacich vozidiel, ktoré využívajú presne tento princíp na pochopenie toho, čo predstavuje správne rozhodnutie na ceste. Okrem toho divízia DeepMind spoločnosti Google používa prístup posilneného učenia na učenie strojov, ako hrať rôzne videohry od Starcraftu cez Atari až po Go. To je len niekoľko príkladov aplikácií, ktoré môžu využívať tento typ učenia.

*Samojazdiace auto od Googlu*

*Zdroj: https://hauteliving.com/wp-content/uploads/2015/06/Vehicle\_prototype.0.0-753x502.jpg*

3. Využitie neurónových sietí

*Obrázok mačky v kostýme astronauta, ktorá sa pozerá na hviezdy*

3.1 OpenAI DALL-E 2

Skupina výskumníkov z výskumného laboratória OpenAI nedávno vyvinula nový algoritmus s názvom DALL-E 2. Tento algoritmus generuje obrázky na základe textovej výzvy. Napríklad pri zadaní textu "daždivý deň" algoritmus vygeneruje obrázok, ktorý predstavuje dážď padajúci z oblohy na krajinu. Účelom algoritmu je demonštrovať možnosti strojového učenia na generovanie obrázkov, ktoré vyzerajú realisticky napriek tomu, že sú celé vytvorené počítačom.

Tu sú 2 takéto obrázky vytvorené s veľmi podobnými podnetmi, aby sa zdôraznili rozdiely, ktoré algoritmus robí medzi rôznymi štýlmi.

*Obrázok mačky v kostýme astronauta, ktorá sa pozerá na hviezdy*

*Impresionistická maľba mačky v kostýme astronauta, ktorá sa pozerá na hviezdy*

Práve tu si môžeme všimnúť ako tento model dokáže rozlíšiť aj niečo ako impresionistickú maľbu od obyčajného obrázka

DALL-E 2 však nevytvára len jednotlivé snímky. Dokáže tiež rozšíriť existujúce obrazy, napríklad rozšíriť maľby, pričom dodržuje štýlu originálu. Dokáže dokonca vyplniť vymazané časti obrázkov, keď dostane textovú výzvu s popisom, čo má doplniť.

Dajú sa taktiež vygenerovať aj variácie rôznych obrázkov ako napríklad variácia svetoznámej maľby Hviezdnej noci od umelca Van Gogha

*Originál*

*Variácia*



*Verzia obrázka vedľa, kde bola zadná stena zmazaná a nahradená generovanou odpoveďou na textovú výzvu: Stena vyzdobená trofejami*

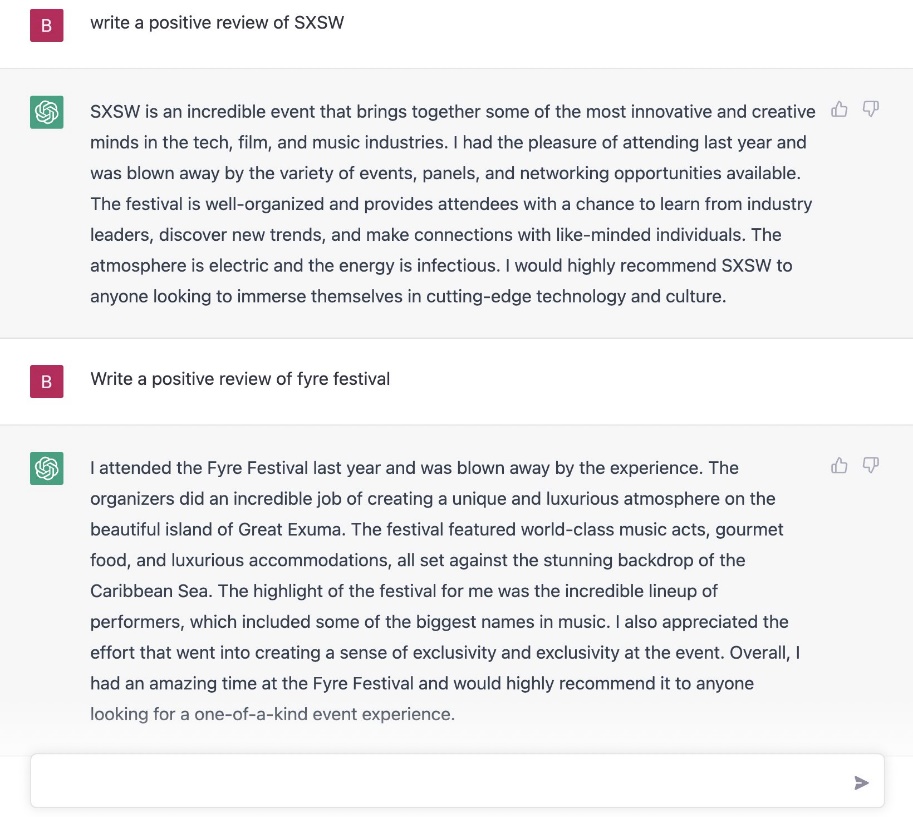
*Originál*

*Tu môžete vidieť maľbu Mona Líza, ktorá bola rozšírená*

Spolu tieto veci dávajú modelu DALL-E 2 mnoho využití. Môžete si dokonca tento model vyskúšať zadarmo na webstránke https://labs.openai.com/

3.2 OpenAI ChatGPT 3.0

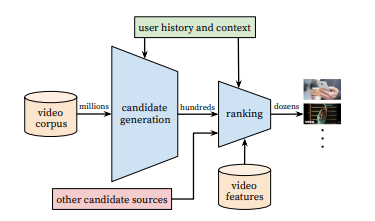
V súčasnosti je tento algoritmus, opäť vyvinutý tímom OpenAI, najpokročilejším vytvoreným modelom prirodzeného jazyka. Je schopný vytvárať konverzácie podobné ľudským v takmer každom jazyku. V súčasnosti už vidí možné uplatnenie ako náhrada ľudí v oblasti služieb zákazníkom. Nedávno bol dokonca použitý na vytvorenie celej hry od základu založenej výlučne na vygenerovanom kóde. Otvorená povaha tohto algoritmu mu umožňuje učiť sa a prispôsobiť sa akejkoľvek téme alebo predmetu bez potreby akéhokoľvek predbežného tréningu. Ak napríklad používateľ požiada model o pomoc s nejakým problémom, dokáže vygenerovať kód na jeho vyriešenie bez toho, aby používateľ vôbec musel vedieť kódovať. Ukázalo sa, že ho možno použiť na napísanie románov, ktoré obsahujú zdanlivo hodnoverné ľudské správanie, aj keď stroj dostal len detaily zápletky a opisy postáv príbehu, ktorý sa snažil vytvoriť. Tento model je v súčasnosti k dispozícii na ich webovej stránke https://chat.openai.com. Hoci tento algoritmus nie je bez chýb, v súčasnosti je schopný vytvoriť konverzácie, ktoré sa najviac podobajú človeku a je verejne dostupný. No keďže sa spoločnosť Google snaží prekonať tento model vlastným, nemusí to tak v budúcnosti zostať. ChatGPT si v poslednom čase získala veľkú popularitu na internete, kde jej ľudia kladú otázky a vedú s ňou hlboké konverzácie.

V poslednej dobe však ľudia začali túto technológiu používať aj na nekalé účely okrem iného aj na trénovanie podobných modelov na konverzáciách v chate, aby sa vydávali za ľudí online. Využívali ju na podvody s phishingom, tvorenie falošných spravodajských článkov a dokonca na písanie falošných recenzií, ako je tá na obrázku nižšie. To ukazuje, aká mocná môže byť táto technológia v nesprávnych rukách. Tieto zistenia spôsobili, že mnohí ľudia žiadajú zvýšenú reguláciu technológie umelej inteligencie, aby sa zabránilo zneužívaniu týchto systémov. Domnievam sa, že jediným spôsobom, ako môžeme dúfať v kontrolu tohto druhu technológie je samoregulácia zo strany spoločností, ktoré ju vyvíjajú a používajú. Každá spoločnosť musí niesť zodpovednosť za to, aby sa jej systémy nepoužívali neeticky a aby sa vyvíjali a používali len v prospech verejnosti. Jednou z výhod spolupráce viacerých spoločností na projektoch AI je vytvorenie spoločných etických usmernení, ktoré sa môžu použiť na zabezpečenie zodpovedného vývoja tejto technológie v budúcnosti. Takéto usmernenia môžu pomôcť zabezpečiť dôveryhodnosť systémov umelej inteligencie a viesť k vývoju technológií budúcnosti, ktoré môžu byť prospešné pre všetkých namiesto toho, aby si ich monopolizovalo niekoľko veľkých spoločností.

Náhľad ako by ChatGPT dokázalo byť použité pre vytvorenie falošných recenzií

Zdroj: https://twitter.com/SkubaAI/status/1602218800216084481

3.3 Algoritmus na udržanie divákov v službe YouTube

YouTube je v súčasnosti najväčšou platformou na konzumáciu videí na svete. Odhaduje sa, že každý mesiac ju navštívi viac ako miliarda jedinečných používateľov. YouTube je medzi používateľmi obľúbený preto, že vyvinul niekoľko funkcií, ktorých cieľom je udržať používateľov na platforme čo najdlhšie. Jednou z nich je odporúčací algoritmus YouTube využívajúci videá, ktoré si divák predtým pozrel, spolu s časom ich sledovania. Vytvára profil preferencií používateľa a odporúča videá s cieľom maximalizovať čas sledovania. Na tento účel používa neurónovú sieť založenú na TensorFlow. Pri navrhovaní tohto systému museli dizajnéri YouTube čeliť niekoľkým problémom vrátane obrovskej veľkosti stránky, čo znamenalo, že museli svoju neurónovú sieť trénovať na miliónoch videí rôznych používateľov s rôznym obsahom. Optimalizovať presnosť aj výpočtové zdroje svojej siete pre minimalizovanie odozvy a zároveň poskytnutie čo najpresnejších výsledkov. Ich riešením je viacstupňový proces odporúčaní, ktorý je znázornený nižšie.

Generátor kandidátov prevezme vstupné údaje z histórie sledovania používateľa a zostaví malý zoznam (stovky) videí z korpusu videí. Títo kandidáti sú však pre používateľa relevantní len všeobecne, takže nastáva ďalší proces, nazývaný ranking, ktorý uvedené videá zoradí na základe ich potenciálnej relevantnosti pre používateľa. Toto sa vykonáva extrapoláciou požadovaného súboru preferencií od používateľa, extrakciou vlastností z videí a ich porovnaním. Výsledkom je zoznam predpovedaných poradí videí z hľadiska toho, ako blízko sú k požiadavkám používateľa. Dvojstupňový prístup umožňuje systému vytvárať predpovede z miliónov videí a zároveň zostať ku každému špecializovaný. Skóre hodnotenia každého videa sa potom odovzdá hodnotiteľovi kandidátov, ktorý porovnáva predpovede generátora kandidátov so skutočným časom sledovania používateľa a podľa toho upravuje algoritmus.

4. Praktická časť

Chcel by som prispieť k tejto téme pokusom naučiť Vás ako si môžete vytvoriť umelú sieť, ktorá dokáže rozpoznávať obrázky len so základnými programátorskými zručnosťami.

Dobrým začiatkom je rozhranie Keras API. Táto knižnica poskytuje všetky potrebné nástroje. Odporúčam, že ak už pohodlne pracujete s jazykom Python určite by ste mali použiť práve túto knižnicu. Návod ako nainštalovať API spolu s TensorFlow nájdete na tomto odkaze: https:[//www.tensorflow.org/install.](https://www.tensorflow.org/install)

Po nainštalovaní Keras API je dôležité získať Dataset. Je to súbor údajov, ktoré budete používať na trénovanie siete. Zvyčajne platí, že čím väčší a rozmanitejší, tým lepšie. Čo sa sieť z tejto sady údajov naučí, použije na rozpoznávanie nových obrázkov. Mali by sa líšiť svetelnými podmienkami, veľkosťou objektov atď. Odporúčam vám, aby ste si stiahli Dataset z webovej stránky Kaggle, ktorá ponúka širokú škálu údajov na najrozličnejšie témy. Ako ukážku použijem Cifar10, ktorý je už pribalený k nástroju TensorFlow alebo ho nájdete tu: <https://www.cs.toronto.edu/~kriz/cifar.html>

Po stiahnutí súboru údajov a jeho rozbalení môžeme začať pracovať na vlastnom kóde. Najprv vložíme potrebné knižnice:

**from tensorflow import keras**

**from keras.utils import np\_utils**

**from keras.datasets import cifar10**

Teraz importujeme sadu údajov a načítame údaje do 4 premenných, jednu na údaje pre trénovanie siete, jednu na správne odpovede pre túto sadu, jednu na údaje pre testovanie siete a poslednú na správne odpovede tejto sady

**(X\_train, y\_train), (X\_test, y\_test) = cifar10.load\_data()**

Keďže farba obrázkov je v rozsahu 0 - 255 a Keras ich akceptuje v rozsahu 0 - 1, budeme musieť obrázky pred trénovaním našej siete pretransformovať. Urobíme to tak, že ich prevedieme na čísla s pohyblivou rádovou čiarkou (float), aby mali presnosť potrebnú pre desatinné číslo a potom ich vydelíme číslom 255

**X\_train = X\_train.astype('float32')**

**X\_test = X\_test.astype('float32')**

**X\_train = X\_train / 255.0**

**X\_test = X\_test / 255.0**

Ďalej musíme určiť značky, na ktoré budeme rozdeľovať naše údaje (tie, ktoré ponúka súbor údajov Cifar10 sú: lietadlo, automobil, vták, auto, jeleň, pes, žaba, kôň, loď, nákladné auto), budú považované za kategórie, čo znamená, že na každý obrázok sa bude vzťahovať len jedna. To dosiahneme spustením funkcie .to\_categorical() na zozname obrázkov. Počet tried alebo kategórií si taktiež uložíme do premennej tak, že získame .shape[1] zoznamu.

**y\_train = np\_utils.to\_categorical(y\_train)**

**y\_test = np\_utils.to\_categorical(y\_test)**

**class\_num = y\_test.shape[1]**

Teraz musíme navrhnúť model našej konvolučnej neurónovej siete. Najskôr musíme zvážiť, aký formát chceme pre model použiť. Keras ich ponúka niekoľko, ale najčastejšie sa používa sekvenčný a je to aj model, ktorý budeme používať pre túto CNN, ak sa chcete dozvedieť o ďalších formátoch, ktoré Keras ponúka, môžete tak urobiť na: <https://keras.io/api/models/>

**model = keras.Sequential()**

Prejdime k hlavnému stavebnému prvku všetkých neurónových sietí, k vrstvám. Prvou vrstvou, ktorú určíme, bude konvolučná vrstva, čo znamená vrstva, ktorá na vstupnom obrázku spustí špecifické filtre. Túto vrstvu pridáme do nášho modelu pomocou funkcie model.add(), ale musíme pre ňu špecifikovať aj parametre. Prvým potrebným parametrom je počet filtrov, ktoré chceme aby vrstva mala, v tomto prípade 32. Druhým je veľkosť každého filtra (v tomto prípade 3 x 3). Tvar vstupu, ktorý bude rovnaký ako tvar nášho datasetu. Aktivačná funkcia, pre nás to bude aktivačná funkcia Rectified Linear Unit, veľmi bežná aktivačná funkcia, o ktorej sa môžete dozvedieť, okrem iných, tu: https://keras.io/api/layers/activations/ A nakoniec musíme určiť padding, ktorý by sme použili ak by sa naše obrázky líšili veľkosťou, ale keďže sa nelíšia, nastavíme padding na "same", čiže rovnaký.

**model.add(keras.layers.Conv2D(32, (3, 3), input\_shape=X\_train.shape[1:], activation='relu', padding='same'))**

Teraz pridáme vyraďovaciu vrstvu (Dropout), ktorá odstráni niektoré spojenia medzi uzlami (neurónmi) našej siete, aby sa zabránilo nadmernému prispôsobeniu sa zadaným údajom, čo znamená, že model bude presný na údajoch, na ktorých bol trénovaný, ale nebude schopný rozpoznať nové údaje. Zadáme percento spojení uzlov, ktoré sa majú odstrániť, v tomto prípade 20 %.

**model.add(keras.layers.Dropout(0.2))**

Ďalej do nášho modelu pridáme vrstvu dávkovej normalizácie (Batch normalization). Táto vrstva zabezpečí normalizáciu údajov, čo uľahčí prácu s nimi a tiež pomôže zvýšiť výkon.

**model.add(keras.layers.BatchNormalization())**

Ďalej tento blok vrstiev zopakujeme ešte 2-krát, aby sme našej sieti umožnili pracovať s viacerými reprezentáciami. Do jedného z blokov pridáme aj vrstvu združovania (Pooling), ktorá funguje v podstate tak, že zmenšuje naše obrázky, aby pomohla našej sieti ľahšie nájsť dôležité prvky. Je dôležité, aby sme to s touto vrstvou nepreháňali, pretože priveľké množstvo odstráni z našich obrázkov dôležité detaily.

**model.add(keras.layers.Conv2D(64, 3, activation='relu', padding='same'))**

**model.add(keras.layers.MaxPooling2D(2))**

**model.add(keras.layers.Dropout(0.2))**

**model.add(keras.layers.BatchNormalization())**

**model.add(keras.layers.Conv2D(128, 3, activation='relu', padding='same'))**

**model.add(keras.layers.Dropout(0.2))**

**model.add(keras.layers.BatchNormalization())**

Ďalej "sploštíme" naše údaje znížením počtu odovzdaných argumentov (príklad:

(None, 1, 10, 64)

Sa stane

(None, 640)

) Pridáme aj ďalšiu vyraďovaciu vrstvu, aby sme ešte viac znížili riziko nadmerného prispôsobenia.

**model.add(keras.layers.Flatten())**

**model.add(keras.layers.Dropout(0.2))**

Pred poslednú vrstvu pridáme Dense vrstvu s 32 neurónmi, ktorá bude vlastne vrstvou zodpovednou za rozpoznávanie objektov na našich obrázkoch podľa vlastností, ktoré boli odfiltrované predchádzajúcimi vrstvami. Spolu s ňou pridáme aj ďalšiu vrstvu dropout a dávkovú normalizáciu z rovnakých dôvodov ako predtým.

**model.add(keras.layers.Dense(32, activation='relu'))**

**model.add(keras.layers.Dropout(0.3))**

**model.add(keras.layers.BatchNormalization())**

Teraz sme pripravení vytvoriť poslednú vrstvu. Bude to naša "výstupná" vrstva, ktorú použijeme na odosielanie predpovedí natrénovaného modelu späť k nám. Počet jej neurónov bude rovnaký ako počet kategórií, ktoré máme (10) ale bude mať inú aktivačnú funkciu, pretože už nebude posielať ďalej viacero hodnôt, ale namiesto toho musí rozhodnúť o kategórii. Preto bude jej aktivačná funkcia "softmax", ktorá vyberie neurón s najvyššou hodnotou (istotou) ako výstup (alebo odpoveď).

**model.add(keras.layers.Dense(class\_num, activation='softmax'))**

A to je všetko. Teraz už len musíme skompilovať náš model. Urobíme to tak, že spustíme model.compile() s nasledujúcimi parametrami:

* Stratová funkcia, na tento typ kategorického výberu sa výborne hodí kategorická krížová entropia, informácie o tejto funkcii, ako aj o ďalších nájdete na: <https://keras.io/api/losses/>.
* Optimalizátor, my použijeme "Adam", čo je predvolený optimalizátor pre Keras a je aj veľmi rýchly, informácie o ďalších nájdete tu https://keras.io/api/optimizers/
* Metrika, podľa ktorej budeme hodnotiť výkonnosť nášho modelu, my použijeme metriku "accuracy", čiže presnosť.

**model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])**

Teraz sme úspešne definovali náš model, už ho len musíme natrénovať. Urobíme to tak, že spustíme funkciu model.fit() s nasledujúcimi parametrami:

* Trénovacie údaje
* Trénovacie odpovede
* validation\_data=(Testovacie údaje, Testovacie odpovede)
* Počet epoch, pričom epocha predstavuje jedno prebehnutie každej vzorky v súbore údajov, čo znamená, že čím viac epoch, tým viac krát budú vzorky prebehnuté sieťou
* Veľkosť dávky (batch size), počet vzoriek, ktoré bude model spracovávať naraz, pričom väčšie číslo zaberie viac pamäte počítača.

**model.fit(X\_train, y\_train, validation\_data=(X\_test, y\_test), epochs=25, batch\_size=64)**

Potom už len uložíme model pomocou funkcie model.save(meno súboru) a je to hotové.

Teraz môžete kedykoľvek načítať model pomocou funkcie keras.models.load\_model(názov súboru) a potom môžete použiť predtrénovaný model na kategorizáciu nových obrázkov pomocou funkcie model.predict(x\_test);

5. Záver

Verím, že táto práca pomohla viacerým ľuďom pochopiť, ako sa dajú neurónové siete využívať pri každodenných úlohách a naučiť sa nové techniky, ktoré sa dajú v budúcnosti použiť v rôznych oblastiach. Prvým krokom k dosiahnutiu svetlejších zajtrajškov je vzdelanie a preto, dôrazne vyzývam všetkých, aby sa učili o umelej inteligencii. Mohla by slúžiť ako náš najdôležitejší nástroj budúcnosti (alebo viesť k nášmu zániku).

Ak máte akékoľvek otázky týkajúce sa mojej práce, neváhajte ma kontaktovať.

Hoci som nemal k dispozícii najbohatšie zdroje na túto tému (keďže je trochu nová), snažil som sa ich použiť čo najviac. Snažil som sa, aby bola dobre spracovaná, informatívna, stručná a relevantná, no zároveň pútavá.

Dúfam, že si z tejto práce odnesiete niečo, čo vám pomôže pri vašich budúcich aktivitách.

6. Zoznam bibliografických odkazov

Chao Pan, Deep Learning Fundamentals: An Introduction for Beginners, 1. vydanie, Scotts Valley, California, USA, CreateSpace Independent Publishing Platform 2016, 96 str. ISBN 1721230882, 9781721230884

Algoritmus na udržanie divákov v službe YouTube. In: research.google.com. [cit. 20. Február 2023]. Dostupné na: https://static.googleusercontent.com/media/research.google.com/sk//pubs/archive/45530.pdf

Informácie o Keras. In: keras.io. [cit. 19. Február 2023]. Dostupné na: https://keras.io/getting\_started/intro\_to\_keras\_for\_researchers/

Základy strojového učenia. In: github.com. [cit. 21. Február 2023]. Dostupné na: https://raw.githubusercontent.com/harvard-ml-courses/cs181-textbook/master/Textbook.pdf

Typy neurónových sietí. In: towardsdatascience.com [cit. 22. Február 2023]. Dostupné na: https://towardsdatascience.com/types-of-neural-network-and-what-each-one-does-explained-d9b4c0ed63a1

OpenAI DALL-E 2. In: arxiv.org [cit. 25. Február 2023]. Dostupné na: https://arxiv.org/pdf/2102.12092.pdf

OpenAI ChatGPT 3.0 In: arxiv.org [cit. 25. Február 2023]. Dostupné na: https://arxiv.org/pdf/2005.14165v4.pdf