Vážená pani profesorka, milí spolužiaci,

**Rád by som Vám predstavil moju tému** „Využitie neurónových sietí v moderných aplikáciách.“

**Cieľom mojej práce je informovať ľudí o Umelej inteligencii,** čo je v súčasnosti jednou z najrýchlejšie sa rozvíjajúcich oblastí technológií. A jej podstatou je vytváranie strojov ktoré sa dokážu naučiť ako dosiahnuť požadovaný cieľ

**Nástup** **umelej** **inteligencie** so sebou prináša aj množstvo výziev a etických problémov. Napríklad, keďže stroje sú čoraz schopnejšie vykonávať úlohy, ktoré vykonávajú ľudia, mohlo by to znamenať že vlastne začnú vytláčať ľudí z pracovných miest. Existuje aj riziko zaujatosti týchto systémov, ktoré môžu udržiavať diskrimináciu a posilňovať sociálne nerovnosti.

**Základom umelej inteligencie je takáto neurónová** **sieť.** Skladá sa vlastne z troch alebo viac vrstiev. Na obrázku vidíme neurónovú sieť s tromi vrstvami predstavenými stĺpcami. prvá je vstupná vrstva, nasleduje skrytá vrstva a nakoniec výstupná vrstva. Vstupná vrstva je tá, ktorá prijíma vstup od používateľa vo forme výzvy, obrázku alebo niečoho takého. Ďalej sú skryté vrstvy, zložené z neurónov ktoré sú navzájom poprepájané a tvoria samotnú výpočtovú jednotku siete. Vo výstupnej vrstve sa vytvorí výsledok výpočtu vo forme slova, kategórie alebo čohokoľvek čo požadujeme ako odpoveď. A ten vlastne spôsob na základe ktorého sa to všetko posunie dopredu sú tie prepojenia medzi vrstvami. Každé z nich môže mať hodnotu, ktorá určuje silu spojenia medzi dvoma uzlami v sieti. Toto je uzol. *[ukáž]* Tieto hodnoty sa počas učenia menia. Ich zmena by ich nakoniec mala viesť k tomu, že pre každý vstup, ktorý zadáme, bude na výstupe požadovaný výsledok, čiže výsledok aký my chceme pre ten vstup.

**Najjednoduchšia neurónová sieť ktorá sa používa dodnes je Perceptron**. Je tvorená len dvoma vrstvami, vstupnou a výstupnou. Vďaka tomuto je len tzv. lineárnym kvalifikátorom, čo vlastne znamená, že ak sú dané dva vstupy, vypočíta výstup na základe priamky niekde v grafe týchto dvoch vstupov. 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 novým príkladom.

**V súčasnosti najpoužívanejšou neurónovou sieťou je Viacvrstvový** **perceptrón** Vlastne, podobá sa Perceptronu, ale pozostáva z viacerých skrytých vrstiev, ktoré jej dovolia sa naučiť aj vzťahy ktoré nie sú dané len priamkou, kvôli tomu sa používa hlavne na trénovanie zložitejších systémov. Je základom pre ďalšie typy neurónových sietí, ako je napríklad konvolučná neurónová sieť.

**Táto sieť** dokáže analyzovať obrázky alebo videá, klasifikuje ich a nachádza v nich vzory. Bežne sa používa aj v počítačovom videní, rozpoznávaní tváre alebo spracovaní reči.

Medzi zaujímavé prípady ako bola využitá je napríklad aj klasifikácia rakovinových buniek podľa genetickej výbavy pomocou ich snímok. Vytvorenie falošných videí, ktoré pôsobia dojmom, že človek hovorí niekoľkými jazykmi alebo aj rozpoznávanie vzorov ciest pre samojazdiace autá.

Jednou z aplikácií ktorý využíva tento model je **DALL-E 2** od spoločnosti OpenAI . Účelom tohto algoritmu je generovať počítačovo vytvorené realistické obrázky na základe textovej výzvy

Tu môžete vidieť 2 takéto obrázky.

Nevytvára však len takéto snímky. Dokáže rozšíriť aj existujúce obrázky, napríklad maľby, pričom sa drží štýlu originálu. Dokáže dokonca vyplniť aj vymazané časti obrázkov, keď je modelu daný popis toho, čo má doplniť.

Môže aj vygenerovať variácie rôznych obrázkov ako napríklad táto, maľby Hviezdnej noci od Van Gogha

Algoritmus ChatGPT je tiež vlastne algoritmus vyvinutý spoločnosťou OpenAI

Tento algoritmus je vlastne v súčasnosti najpokročilejším vytvoreným modelom prirodzeného jazyka. Je schopný vytvárať konverzácie podobné ľudským v takmer každom jazyku. Možno ho použiť aj na napísanie románov, ktoré obsahujú zdanlivo reálne ľudské správanie, aj keď sme zadali len detaily nejakej zápletky a opisy postáv príbehu. Aj keď tento algoritmu má nejaké menšie chyby, v súčasnosti je schopný udržiavať konverzáciu, ktorá sa najviac podobá človeku a je aj verejne dostupný. No onedlho plánuje aj Google prekonať tento model vlastným. Model ChatGPT si v poslednom čase získal veľkú popularitu na internete, kde mu ľudia kladú otázky a vedú s ním filozofické konverzácie.

**Nedávno však vyšlo najavo že ľudia začali túto technológiu používať aj na nekalé účely.** Využívali ju na podvody s phishingom, tvorenie falošných spravodajských článkov alebo na písanie falošných recenzií, ako je napríklad táto. Tieto zistenia spôsobili, že mnohí ľudia žiadajú zvýšenú reguláciu tejto technológie. Domnievam že jediný spôsob ako môžeme dúfať v takúto regulácie je ak nastane kontrola zo strany spoločností, ktoré takéto modely vyvíjajú a používajú.

Ako praktickú časť mojej práce SOČ som si vybral návod ako vytvoriť konvolučnú neurónovú sieť na rozpoznanie nejakých obrázkov. Pred začatím je dôležité mať už základne znalosti programovacieho jazyka Python.

Najprv potrebujete nainštalovať samozrejme knižnicu TensorFlow.

Následne je potrebné získať dáta na ktorých budeme náš model trénovať.

Tieto dáta sa dajú získať z viacerých zdrojov na internete a to aj odporúčam, keďže obstarávanie takýchto dát je veľmi zdĺhavý proces.

Načítame potrebné knižnice, importujeme sadu údajov a načítame údaje do 4 premenných.

Keďže farba obrázkov je v rozsahu 0 - 255 a Keras ich akceptuje v rozsahu 0 - 1, budeme musieť obrázky pred trénovaním našej siete premeniť na takéto hodnoty.

Ďalej, určíme kategórie, na ktoré budeme rozdeľovať naše údaje. To, dosiahneme spustením funkcie .to\_categorical(). Počet tried alebo kategórií si taktiež uložíme do premennej class\_num.

Teraz musíme zvážiť model našej konvolučnej neurónovej siete. Najčastejšie sa používa Sekvenčný a je to aj ten ktorý použijeme teraz.

Prejdime k hlavnému stavebnému prvku všetkých neurónových sietí, k vrstvám. Prvou vrstvou, ktorú určíme, bude konvolučná vrstva, ktorá na vstupnom obrázku spustí špecifické filtre. V tejto vrstve musíme špecifikovať niekoľko parametrov.

* Počet filtrov, ktoré chceme aby vrstva mala, v tomto prípade 32.
* Veľkosť každého filtra (v tomto prípade 3 x 3).
* Tvar vstupu, ktorý bude rovnaký ako tvar našich dát.
* Aktivačnú funkcia, pre nás to bude Rectified Linear Unit.
* A nakoniec musíme určiť padding, ktorý by sme použili ak by sa naše obrázky líšili veľkosťou, no keďže sa nelíšia, nastavíme ho na "same", čiže rovnaký.

Vrstva Dropout spôsobuje to, že odstráni niektoré spojenia medzi uzlami (neurónmi) našej siete, aby sa zabránilo nadmernému prispôsobeniu sa zadaným údajom.

Vrstva Batch Normalization alebo teda dávková normalizácia, nám uľahčí prácu s dátami a tiež pomôže zvýšiť výkon.

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 ich dôležité prvky.

"Sploštíme" naše údaje znížením počtu odovzdaných argumentov vrstvou Flatten a pridáme ďalšiu Dropout vrstvu

Pridáme Dense vrstvu s 32 neurónmi, zodpovednú 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.

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). Jej aktivačná funkcia bude "softmax", ktorá vyberie neurón s najvyššou hodnotou ako výstup.

A to je všetko. Teraz už len musíme uzavrieť náš model. Spustíme funkciu model.compile() s niekoľkými parametrami

* Stratovou funkciou, na tento typ kategorického výberu sa výborne hodí kategorická krížová entropia,
* Optimalizátor, my použijeme "Adam", čo je predvolený pre Keras a je aj veľmi rýchly.
* Metrika, podľa ktorej budeme hodnotiť výkonnosť nášho modelu, my použijeme "accuracy", čiže presnosť.

Teraz sme úspešne definovali náš model, už ho len musíme natrénovať. Urobíme to tak, že spustíme funkciu model.fit() s parametrami našich údajov a odpovedí ale aj s niekoľkými novými parametrami ako sú epochy a veľkosti dávky.

Epocha predstavuje jedno prebehnutie každej vzorky v súbore údajov.

Veľkosť dávky (batch size) je počet vzoriek, ktoré bude model spracovávať naraz v pamäti počítača.

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.

Dúfam, že si z tejto práce odnesiete niečo, čo vám v budúcnosti pomôže

Ďakujem za pozornosť