Vážená pani profesorka, milí spolužiaci,

**Rád by som Vám predstavil moju tému** „Využitie neurónových sietí v moderných aplikáciách.“ Zaoberá sa tým, ako sa náš svet môže posúvať dopredu využitím týchto technológii ku zlepšeniu našej životnej úrovne.

**Cieľom mojej práce je informovať ľudí o Umelej inteligencii.** V súčasnosti je jednou z najzaujímavejších a najrýchlejšie sa rozvíjajúcich oblastí technológií. Podstatou umelej inteligencie je vytváranie strojov, ktoré sa dokážu učiť z údajov a uskutočniť rozhodnutia na základe tohto učenia.

**Nástup** **umelej** **inteligencie** so sebou prináša aj množstvo výziev a etických problémov. Napríklad, keďže stroje sú čoraz schopnejšie vykonávať úlohy, ktoré tradične vykonávajú ľudia, mohli by vytlačiť pracovníkov v niektorých odvetviach. Existuje aj riziko zaujatosti systémov umelej inteligencie, ktoré môžu udržiavať diskrimináciu a posilňovať sociálne nerovnosti.

**Základom umelej inteligencie je neurónová** **sieť, ktorá** sa skladá z troch alebo viacerých vrstiev. V jednoduchej neurónovej sieti zobrazenej na obrazovke sú tri vrstvy: vstupná vrstva, skrytá vrstva a výstupná vrstva. Vstupná vrstva prijíma vstup od používateľa vo forme výzvy, obrázku alebo iných. Ďalej sú skryté vrstvy, zložené z neurónov, ktoré sú prepojené a tvoria výpočtovú jednotku. Výstupná vrstva generuje výstup vo forme slova, kategórie alebo čohokoľvek zobrazujúc požadovanú odpoveď. Neurónové siete fungujú vďaka prepojeniam medzi jednotlivými vrstvami, takzvanými synapsiami. Každá z nich môže mať hodnotu, ktorá určuje silu spojenia medzi dvoma uzlami v sieti. Tieto hodnoty sa počas učenia menia. Komplexné váhy neurónovej siete by nakoniec mali viesť k tomu, že pre každý vstup, ktorý je zadaný, bude na výstupe požadovaný výsledok.

**Perceptron** je najjednoduchšia neurónová sieť, ktorá sa používa dodnes. Táto sieť je typom lineárneho kvalifikátora, čo 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 našim novým príkladom, keď sa do siete privádza čoraz viac údajov.

**Viacvrstvový** **perceptrón** je v súčasnosti najpoužívanejšou neurónovou sieťou. Podobá sa originálnej sieti Perceptron, ale pozostáva z viacerých skrytých vrstiev. Používa sa na trénovanie zložitejších systémov, pretože je schopná spracovať viac informácií a dokáže sa naučiť nelineárne vzťahy medzi rôznymi vstupmi a výstupmi. Je aj základom pre ďalšie typy neurónových sietí, ako je napríklad konvolučná neurónová sieť.

**Táto sieť** dokáže analyzovať údaje, ako sú obrázky alebo videá, klasifikovať ich a nachádzať v nich vzory. Bežne sa používa v úlohách, ako je počítačové videnie, rozpoznávanie tváre alebo spracovanie reči.

Medzi zaujímavé prípady jej využitia patrí klasifikácia rakovinových buniek na základe ich genetickej výbavy pomocou snímok buniek. Vytvorenie falošných videí, ktoré pôsobia dojmom, že človek hovorí niekoľkými jazykmi alebo aj rozpoznávanie vzorov ciest pre samojazdiace autá.

Jednou z aplikácií využívajúcich tieto modely je **DALL-E 2** od spoločnosti OpenAI . Účelom algoritmu je demonštrovať možnosti strojového učenia na generovanie obrázkov, na základe textovej výzvy. Tieto vyzerajú realisticky napriek tomu, že sú celé vytvorené počítačom.

Tu sú 2 takéto obrázky.

DALL-E 2 však nevytvára len jednotlivé snímky. Dokáže tiež rozšíriť existujúce obrazy, napríklad rozšíriť maľby, pričom dodržuje štýl 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 táto, svetoznámej maľby Hviezdnej noci od umelca Van Gogha

**No táto spoločnosť vyvinula aj ďalšie algoritmy ktoré si užívajú svetlo slávy na internete, napríklad ChatGPT 3.0**. V súčasnosti je tento algoritmus 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 hodnoverné ľudské správanie, aj keď stroj dostal len detaily zápletky a opisy postáv príbehu. 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 ostať. Model ChatGPT si v poslednom čase získal veľkú popularitu na internete, kde mu ľudia kladú otázky a vedú s ním filozofické konverzácie.

**V poslednej dobe však 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, aby sa zabránilo zneužívaniu týchto systémov. Domnievam sa, že jediným spôsobom, ako môžeme dúfať v jej kontrolu je samoregulácia zo strany spoločností, ktoré ju vyvíjajú a používajú.

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

Všetko začne stiahnutím a nainštalovaním knižnice TensorFlow.

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

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 pretransformovať.

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

Teraz musíme zvážiť model našej konvolučnej neurónovej siete. Keras ich ponúka niekoľko, ale najčastejšie sa používa sekvenčný a je to aj model, ktorý budeme používať.

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