

OTDK-dolgozat

2023.

Budapesti Gazdasági Egyetem
Külkereskedelmi Kar

**A NAGY NYELVI MODELLEK KORLÁTAI: MIÉRT NEM
HELYETTESÍTHETŐK AZ EMBEREK AZ LLM-EKKEL?**

**THE LIMITATIONS OF LARGE LANGUAGE MODELS: WHY LLMS
ARE NOT HUMAN REPLACEMENTS**

Madlena Eszter

Dr. Bánhalmi Árpád

Kézirat lezárásának dátuma: 2023. október 29.

TARTALOMJEGYZÉK

1.	BEVEZETÉS	1
2.	NEURÁLIS HÁLÓZATOK, MINT A NAGY NYELVI MODELLEK ELŐFUTÁRAI	2
2.1.	A LEGELSŐ NEURONHÁLÓK.....	2
2.2.	MÉLY NEURÁLIS HÁLÓZATOK.....	3
3.	TRANSZFORMER MODELLEK ÉS GENERATÍV MODELLEK	5
3.1.	A TRANSZFORMER MODELLEK ALAPJAI	5
3.2.	TRANSZFORMÁTOROK.....	5
3.3.	GENERATÍV MODELLEK	7
3.4.	A TRANSZFORMER MODELLEK ÉS A GENERATÍV MODELLEK ÖTVÖZÉSE: AZ ELSŐ LÉPÉSEK A MAI ISMERT LLM-EK FELÉ	8
4.	NAGY NYELVI MODELLEK	9
4.1.	ÁTTEKINTÉS	9
4.2.	AZ ISMERTEBB MODELLEK.....	10
4.3.	A NAGY NYELVI MODELLEK LIMITÁCIÓI.....	11
5.	RNN, MINT SZEMLÉLTETŐ PÉLDA.....	15
6.	ÖSSZEGZÉS	20
	IRODALOMJEGYZÉK.....	23
	MELLÉKLETEK.....	I
	I. melléklet.....	II

ÁBRAJEGYZÉK

1. ábra Neurális hálózat és mély neurális hálózatok (Sheikh, 2019).	3
2. ábra Az RNN (felső ábra) és transzformer modell (alsó ábra) közötti különbség vizualizálva (Joshi, 2020).....	6
3. ábra Loss probléma bemutatása	17
4. ábra Loss vizualizációja az 1. futtatás során a datawrapper.de segítségével	19

1. BEVEZETÉS

Napjainkban elengedhetlenné vált, hogy megtanuljunk olyan technológiákkal együtt élni, azokat használni, amik számunkra idegennek tűnhetnek. Ezeknek a lehetőségeit szükséges felismerni még a korai formájukban is, és nem szabad úgy kezelni, mint egy potenciális veszélyforrás, hanem sokkal inkább a hasznunkra kellene ezeket fordítani. A mesterséges intelligencia nagyobb térnyerésével bizonyossá vált, hogy szükségünk van megtanulni azokat használni, kiaknázni a bennük rejlő lehetőségeket. Olyan gyors és hatalmas léptékű fejlődés mutatnak, amit a mai világunkban érdemes, sőt ajánlott megismerni. Ezzel a nagy térnyeréssel már sokan éltek is, s a nagy nyelvi modellek (Large Language Model – LLM) megjelenése sokak képzeletét megragadta, néhányunkét felül is múlta. Ezek a modellek, mint pl. a ChatGPT (OpenAi) vagy a Llama 2 (Meta) elképesztő képességeket mutattak be a nagyközönség számára is. Koherens, válaszai kontextusukban relevánsak, tehát a kérdésekre megfelelő válasz jön létre és néha-néha kísértetiesen jó természetes nyelvi szövegek generálására képesek. Rövidebb esszék, versek, történetek megírásától a bonyolult, területspecifikus kérdések megválaszolásáig, több éves tanulást és olvasást kikerülve tudunk mi is fejlődni. Jelenleg ezek a modellek tűnnek a mesterséges intelligencia élvonalának, olyan tekintetben, hogy a széleskörben elterjedt és használata nagyrészt mindenki számára elérhető kategóriában vezető szerepet töltenek be ezek az architektúrák. Mindezek ellenére a nagy nyelvi modellek nem képesek teljesen új ötletek megalkotására, vagy a valóban emberszerű gondolkodásra. Ez kérdéseket vethet fel azzal kapcsolatban, hogy mi is a kreativitás és a megismerés, észlelés természete. Intelligenciájuk ezáltal látszólagos, valójában a hatalmas adatmennyiség, a vezérelt előképzés és az emberi visszajelzés adja ezeknek a valódi „csodáját”. Jelen kutatásom során azt szeretném megvizsgálni, hogy hogyan épül fel valójában egy nagy nyelvi modell, az alapvető koncepcióktól, mint a neuronhálók, egészen a mostani, számunkra is könnyen hozzáférhető, óriás modellekig. Vizsgálatuk közben szeretnék fényt deríteni arra, hogy miért nem képesek az ember egyedi kreativitását és találékonyságát megismételni, hogy merítenek meglévő adatkészletekből, és hogy az ezekre való támaszkodás miért korlátozza ezeket a modelleket a ténylegesen innovatív ötletek, gondolatok kialakításában.

Alapvető célom a dolgozattal, hogy a közismert, általánosan elterjedt, felszínesen megfogalmazott kételyeket, melyek a nagy nyelvi modelleket övezik megbízhatóság szempontjából, enyhítsem és deduktív módon támasszam alá, hogy az LLM-ek azokból a szövegekből tanulnak, amelyeket a készítőik gondosan összeválogattak, és ez teszi lehetővé számukra, hogy emberszerek által jól érhető nyelvi elemek kreáljanak, alkalmazkodjanak

különböző írási stílusokhoz, területekhez, de nem rendelkeznek valódi megértéssel, tudatossággal vagy teljesen eredeti ötletek kialakításának a képességével, vagy amennyiben igen, az általában nem lesz értelmes, és számunkra nem feldolgozható formában jelenik meg. Ilyen kirívó eset lehet, hogyha valami olyan összetett szót tanul meg a szövegkorpuszból, ami valójában az emberi nyelvek szempontjából ismeretlen és értelmetlen. Kutatásom során korábbi kutatások és kísérletek anyagaira támaszkodom, azonban a megalapozottabb bizonyítás érdekében a következőkben kifejtésre kerülő modellek, folyamatok közül elkészítettem egy visszacsatolt neurális hálózatot (Recurrent Neural Network - RNN), ami a dolgozat végén kerül kifejtésre.

2. NEURÁLIS HÁLÓZATOK, MINT A NAGY NYELVI MODELLEK ELŐFUTÁRAI

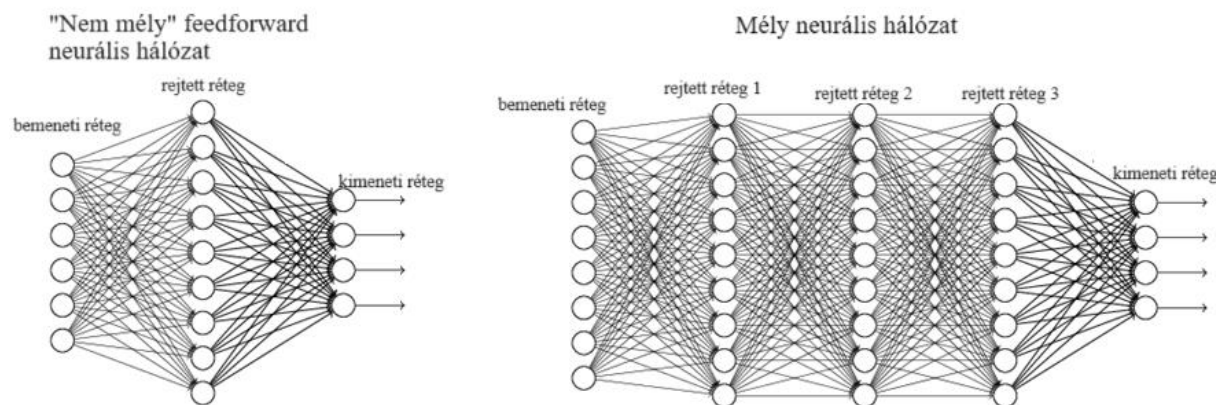
Napjainkban a hatalmas adatmennyiség egyre növekedő üteme természetszerűen hozta magával az ezekhez kapcsolódó technológiai fejlődést is. A gépi tanulás, a mesterséges intelligencia és a természetes nyelvi feldolgozás robbanásszerű növekedése a mély neurális hálózatok új eredményeinek köszönhetően ment végbe, ami a mélytanuló folyamatok modellezésének egyik leghatékonyabb eszköze lett (Gyires-Tóth, 2019).

2.1. A LEGELSŐ NEURONHÁLÓK

A neurális hálózatok, a mély neurális hálózatokhoz képest kisebb teljesítményre voltak képesek, egyrészt a kevesebb neuron miatt és másrészt a technikai háttér sem volt még biztosított. Alapvető koncepció az, hogy az emberi agy biológiai folyamatait utánozzák mesterséges környezetben. Olyan sajátosságokat vettek alapul, mint a nagy számú, de egyenként kicsi egységekből való felépítés, az ezek közötti kapcsolat stb. (Fazekas, 2013). Ezek a korai hálózatok figyelemreméltó képességgel rendelkeztek az adatok feldolgozására és a minták felismerésére, hiszen matematikai jellegű módszerekkel írták le a biológiai folyamatokat, de lehetőségeiket az egyszerűségük korlátozta. Az összetett és réteges minták kezelésére való képességük korlátozott volt (Fazekas, 2013). Ugyan ez a biológiai vonal csak a kezdetekben volt fontos és releváns számunkra, az alapvető ötlet forradalmi volt, hiszen a mesterséges neuronok a megadott adatokból tanulhattak és ezek alapján változtathatták meg az egymás közötti kapcsolataikat. Ezek a korai neurális hálózatok még leginkább a karakterfelismerésben, jelfelismerésben voltak használatosak, de ezen lehetőségeket a mély neurális hálózatok tovább mélyítették.

2.2. MÉLY NEURÁLIS HÁLÓZATOK

A neurális hálózatok és mély neurális hálózatok közti fő különbséget talán leginkább a rétegek számában lehetne megfogni. A rétegek száma relatíve „sok” (ez bizonyos esetekben már 2-3 rétegtől is soknak számíthat), akkor már mély neurális hálózatról beszélhetünk (1. ábra).



1. ábra Neurális hálózat és mély neurális hálózatok (Sheikh, 2019).

Tehát az alapvető ötlet itt az volt, hogy több rejtett réteget helyezzenek a bemeneti és kimeneti rétegek közé, ami egészen új irányokat nyitott meg. Ez a megoldás lehetővé tette a strukturálatlan adatok mintáinak azonosítását (Gilley et al., 2023). Előnyük abban rejlik, hogy a folyamatokból rögzített adatokat egyszerre tanulja meg a célnak leginkább megfelelő reprezentációra alakítani és modellezni (Gyires-Tóth, 2019). Mindegyik réteg egy felügyelt tanulási folyamaton keresztül hozzájárul a magasabb szintű jellemzők és reprezentációk kinyeréséhez, és az alakítástól függően így tanítjuk a modellünket tovább. A mély neurális hálózatok széleskörű elterjedése a gépi tanulás és a mesterséges intelligencia egy új korszakát hozta magával. Ez természetes különböző iparágakban is megjelenik, legyen az az egészségügy, pénzügy vagy az energetika (Gilley et al., 2023).

A mély neurális hálózatok közül szeretnék kiemelten beszélni az visszacsatolt neurális hálózatokról, vagyis a RNN-ekről. A dolgozat későbbi részében egy karakterszintű RNN modellen keresztül kerül szemléltetésre a mély neurális hálók legalapvetőbb képességei és limitációi. Az visszacsatolt neurális hálózatok széleskörben elterjedt és használt mély neuronhálók. Ez a hálózat egyenként dolgozza fel a bejövő adatokat, miközben megtartja azt az állapotot is, ami összegzi a korábbi inputok előzményét (Mehra, 2023). Különböző hosszúságú inputok és outputok kezelésére alkalmas, ami leginkább a természetes nyelvi feldolgozás területén lehet hasznos, hiszen fel tudja ismerni a különböző nyelvi szintéziseket és a beszédet (Mehra, 2023). Mivel az input outputját is képes visszaküldeni bemenetként a

modellbe, ez lehetővé teszi számukra, hogy a memóriáját használja korábbi bemenetek nyomon követésére (Mehra, 2023). Ezek a képességek alkalmasak arra, hogy remekül lehessen tanítani az RNN modelleket, és alapot adhassanak nagyobb modellek számára is.

Az ilyen algoritmusok fejlesztése egy folyamatos út volt, és habár az RNN-ek áttörő sikerek számítottak, így is szembesülni kellett a korlátokkal. A hatékony generáló feladatok elvégzéséhez szükséges számítási- és memóriaigények sokkal magasabbak, mint amire önmagában egy RNN képes (Bhaskar, 2023). Továbbá problémát jelent a gradiens eltűnő és növekvő mivolta, amiktől instabillá válik a modell. Előbbi esetében a gradiensek olyan kicsivé válnak, hogy nem lehet a modellt megfelelően tanítani. A második esetében pedig annyira nagyok lesznek, hogy a súlyokat instabil módon frissíti az iterációk alkalmával (Mehra, 2023).

Ennek a kiküszöbölésére érkezett meg a köztudatba az LSTM (Long Short-Term Memory) hálózat. A Long Short-Term Memory hálózatok egyfajta visszacsatolt neurális hálózatok, ami a sima RNN-ek eltűnő gradiens problémáját hivatottak megoldani, specializált cellák segítségével (Phi, 2019). Ezek bizonyos memóriacellák, amikben 3 kapu van: a bemeneti-kapu, a felejtés-kapu, és a kimeneti-kapu. Az első az új adatáramlást szabályozza, míg a felejtés-kapu az aktuális adatok megőrzéséről dönt, a kimeneti-kapu pedig a hálózat kimenete felé szabályozza az információáramlást a memóriacellától (Mehra, 2023). Ezzel megoldást kínáltak az eltűnő gradiens problémára, azonban ez sem tökéletes. Emellett itt szintén lineárisan következnek egymás után a rétegek, tehát párhuzamosan nehéz a tanítást bevezetni. A sima RNN-ekhez képest pedig szintén még nagyon magas a memóriaigénye.

Másik probléma ezeknél a modelleknél, az maga az előrejelzés pontossága, amikor a szöveggenerálásról van szó (Bhaskar, 2023). Röviden például a következő karakter jóslása egy ilyen modell számára továbbra is kihívást jelenthet pontosság tekintetében, hiszen, ha megadom a megelőző karakterek számát, amit figyeljen a tanulás során, akkor sem garantált az értelmes output, hiába volt az input egy értelmes szó. Ez adódhat abból is, hogy a karaktereket figyeli, és nem a szavakat, de sokkal inkább ez annak a hiánya, hogy nem tudja értelmezni a jelentést és nem tud hozzá kapcsolni kontextust.

A fentebb kifejtett működési elvek alapján érdemes rávilágítanunk az alapvető korlátokra, ami a mély neurális hálózatokban rejlik. Ugyanis a hálózatok és a képzési adatok között ezek alapján elválaszthatatlan kapcsolat rejlik. Képesek felismerni és reprodukálni a meglévő információkat és mintákat, valamint kiválóan használják a tanítási sémákat, de valóban újszerű, értelmes ötletek generálására nem képesek. A megismerés mélységének hiánya látszik, hiszen

értelmes, újszerű szöveget nem képesek generálni. Működésük, ahogy majd később látni is fogjuk, a tanult adatkészletre korlátozódik, ami valóban nagyon hasznos lehet előrejelzési és osztályozási feladatoknál, de eredeti tartalom létrehozásához, vagy önálló ötletgeneráláshoz kevés a mély neurális hálózatok képessége.

Tehát, ahogy láttuk a neurális hálózatoktól a mély neurális hálózatokig vezető út „csak” egy adatok által vezérelt finomítás, tanítás volt, nem pedig egy hatalmas ugrás a kreativitás, az észlelés vagy az önálló gondolkodás felé. Ez az alapvető koncepció fog minket végig kísérni a dolgozat további részében, hiszen ez a folyamat, séma rejlik a nagy nyelvi modellek elképesztő képességei mögött is. Továbbá ez fog minket elvezetni majd a dolgozat fő, kritikus kérdéséhez is, hogy a jelenlegi tudásunkkal ismerni vélt lehetőségek, korlátok tekintetében a nagy modellek valóban képesek-e az emberi kreativitást megismételni, megismerni az emberi nyelv valódi mélységeit, és az észlelés, megismerés képességét elsajátítani.

3. TRANSZFORMER MODELLEK ÉS GENERATÍV MODELLEK

A transzformátorokra való átállás sarkalatos pillanatot jelentett a mesterséges intelligencia fejlődésében, különösen a természetes nyelvi feldolgozás és a nagy nyelvi modellek tekintetében. Ahhoz, hogy megértsük a transzformátorok működését, még egy kicsit vissza kell kanyarodnunk a neurális hálózatok architektúrájára.

3.1. A TRANSZFORMER MODELLEK ALAPJAI

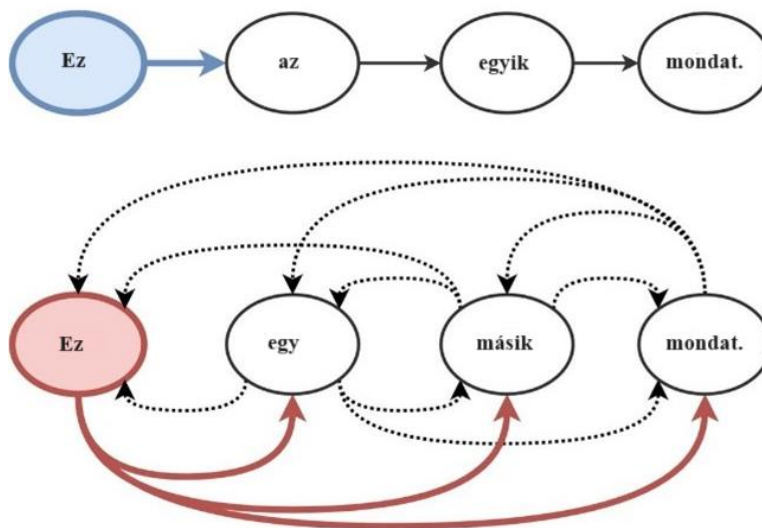
A kódoló-dekódoló architektúrákat Mehra (2023) a neurális hálózati modellek egyik formájaként írja le, amiket leginkább szekvenciális feladatokhoz használnak, mint a nyelvi fordítás. Az enkóder hálózat feldolgozza a bemeneti szekvenciákat, a dekóder hálózat pedig létrehozza a kimeneti szekvenciákat. Egy szabványos modellben a bemeneti szekvenciát egy rögzített hosszúságú vektoros reprezentációban kódoljuk, amit a dekóder felvesz és ebből előállítja a kimeneti szekvenciát. A szerző ugyanakkor felhívja a figyelmünket arra is, hogy amikor ezek a bemeneti adatok túl hosszúak a rögzített hosszúságú kódolás információvesztéssel járhat.

3.2. TRANSZFORMÁTOROK

A transzformer modellek szintén ezt a sémát követve működnek, de ugyanakkor forradalmasították a természetes nyelvi feladatokat végző modelleket, és új lehetőségeket nyitott meg a nagy nyelvi modellek létrejöttéhez (Bhaskar, 2023). Tehát a transzformer modellek létrejöttéhez nagyban hozzájárult a felismerés, hogy a létező modellek, mint például az RNN-ek és az LSTM-ek, küzdöttek az adatok hosszú távú függőségének megértésével,

például a hosszú mondatok kontextusával. Így szükségessé vált egy olyan modell létrehozása, ami képes hatékonyan rögzíteni ezeket a függőségeket. Ezek az új architektúrák egyedi figyelemmechanizmusaikkal újradefiniálták azt a módot, hogy a gépi tanulásban a modellek hogyan értik és generálják az emberi nyelvet.

A transzformátorok jelentősen eltérnek elődeiktől az adatszekvenciák kezelésének megközelítésében. Így, például a hagyományos rekurrens neurális hálózatokkal (RNN) ellentétben, amelyek szekvenciálisan dolgozzák fel az adatokat, a transzformátorokkal önfigyelő mechanizmust mutattak be. Vaswani et al. (2017) „A figyelem minden, amire szükség van” c. művében először tárgyalt figyelemmechanizmus lehetővé teszi a modell számára, hogy mérlegelje a bemeneti adatok különböző részeinek fontosságát, amikor előrejelzéseket készít, lehetővé téve az adatokon belüli összetett, hosszú távú összefüggések és függőségek felismerését és rögzítését. Ezzel annak a látszatát tudja kelteni a modell, hogy megérti a szavak relevanciáját és kontextusát (Bhaskar, 2023). Ezt azzal éri el, hogy a szavak közötti összefüggésekhez figyelem súlyokat rendel függetlenül azok helyzetétől (Bhaskar, 2023). Tehát mintha a szavakhoz is súlyokat rendelnénk, mindezek függetlenné válnak a helyzetüktől, és az outputban azokra a szavakra fog koncentrálni, amik kontextuálisan is jelentősek (2. ábra). Így a teljes bemeneti szekvenciára képes figyelni, felismeri a szöveggörnyezetet, így olyan hatást kelt, mintha valóban megértené a nyelvet, és úgy generálja azt.



2. ábra Az RNN (felső ábra) és transzformer modell (alsó ábra) közötti különbség vizualizálva (Joshi, 2020).

Ami igazán fontos itt, hogy ez a transzformer modell, miben és hogyan hatékonyabb az előző modellekhez képest. Az önfigyelő mechanizmussal ugyanis gyorsabban tudunk rövidebb szekvencia hosszúságokat feldolgozni, és le is lehet korlátozni, olyan tekintetben, hogy csak az

input adat közvetlen környezetét vegye figyelembe egészen hosszú sorozatok esetén (Cristina, 2021). Ugyanakkor arra is be lehet állítani, hogy a teljes input sorozatot vegye figyelembe, ami például a gépi fordításban lehet hasznos (Mehra, 2023). Ebből következtethető, hogy a korábban megemlített problémák a mesterséges neurális hálózatokhoz kapcsolódóan ezzel valamelyest kiküszöbölhetőek lettek. Az output ezzel a módszerrel pontosabb eredményt tud nekünk generálni a fentebb említett okok miatt.

A transzformer architektúrák leginkább alapként szolgáltak már a nagy nyelvi modellek létrehozásában. Ezek a mechanizmusok kritikus szerepet játszottak a szöveggenerálásra és fordításra képes modellek fejlesztésében (Dai et al., 2019). Így a valódi használatban leginkább ilyen formában lehet vele találkozni, és a szélesebb körben elterjedt modellek esetében nem látjuk a mögöttes működést, így a transzformer modellek formabontó működésével alapvetőleg nem találkozunk önmagában. Széleskörben elterjedt használata ismerős lehet sokunk számára, hiszen a Google BERT és a GPT-széria is ezt a módszert alkalmazta a nyelvi modellei létrehozásához. A BERT (Bidirectional Encoder Representations from Transformers) egy felügyelet nélkül képzett nyelvi modell, ami kizárólag egyszerű, jelöletlen szövegen volt tanítva. Előtanítja ezeket a mély, kétirányú reprezentációkat ezekből a jelöletlen szövegekből úgy, hogy a bal és jobboldali kontextusban is minden rétegen kondicionál (Devlin et al., 2018). Ezzel könnyebbé téve a későbbi finomhangolását a modellnek. Számunkra itt a lényeges, hogy ezt a működési mechanizmust a transzformer modellek bevezetésével tudták elérni, és ugrásszerű növekedést elérni a nagy nyelvi modellek területén.

3.3. GENERATÍV MODELLEK

A másik fontos téma a nagy nyelvi modellek teljes egészében történő megértéséhez a generatív modellek ismerete. A generatív modell a gépi tanulás egyik fajtája, aminek célja egy előre megadott adatkészlethez hasonló adatok generálása. Megtanulják a mögöttes struktúrákat, mintákat, majd ezeket felhasználva hozzák létre az új adatokat (Kingma és Welling, 2013). Ezeket a modelleket azóta széleskörben használják, leginkább képgenerálás, szöveggenerálás, zene készítés területein.

A legismertebb modellek, amik az utóbbi időben nagy hírnévre tettek szert, azok a GAN (Generative Adversarial Network) modellek. A GAN-ok egy generátor hálózatból és egy megkülönböztető hálózatból állnak, amiket „ellentétes” módon képeznek (Huang és Le, 2021). A generátor hálózat megtanulja a betanítási adatokhoz hasonló információt generálni, míg a megkülönböztető vagy diszkriminátor hálózat megtanulja megkülönböztetni a valós és a

képzett adatokat (Huang és Le, 2021). A GAN-okat is szintén több területen is alkalmazták sikeresen, de ez leginkább a képgenerálás fejlődéséhez járult hozzá (Huang és Le, 2021).

A transzformer modellek és a generatív modellek közötti kapcsolat a neurális hálózati architektúrákban használatában rejlik. Mindemellett a két modell más-más célt szolgál. A transzformátorokat leginkább sorozatmodellezésre, szekvenciális feldolgozásra és generálásra használják, különösen a természetes nyelvi feldolgozási feladatokban (Wang et al., 2020). Ahogy fentebb már kifejtésre került, sikeresen rögzítik a hosszú távú függőségeket (Cristina, 2021), és a gépi fordítási feladatokban és a szöveggenerálásban teljesedtek leginkább ki (Mehra, 2023). Ezzel párhuzamosan a GAN-okat leginkább generatív feladatokra képezik, és sikeresen képesek valósághű képek, videók és másfajta adatok létrehozásában (Huang és Le, 2021).

3.4. A TRANSZFORMER MODELLEK ÉS A GENERATÍV MODELLEK

ÖTVÖZÉSE: AZ ELSŐ LÉPÉSEK A MAI ISMERT LLM-EK FELÉ

A két koncepció megismerésével eljuthatunk oda, hogy megismerjük a nagy nyelvi modellek mögött rejlő struktúrákat. Az LLM-ek, vagyis a nagy nyelvi modellek, mint például a GPT-3 olyan generatív modellek, amelyeket hatalmas mennyiségű szöveges adaton tanítottak. Képesek koherens és kontextuálisan releváns szöveget generálni egy adott prompt alapján. Az LLM-ek sikere annak tudható be, hogy képesek megragadni a mögöttes mintákat és struktúrákat a képzési adatokban, lehetővé téve számukra, hogy olyan szöveget hozzanak létre, amely hasonlít az ember által írt tartalomhoz (Rothe et al., 2020). Ezen kívül pedig, az LLM-ekben használt transzformátorok lehetővé teszik számukra a hosszú távú függőségek modellezését a hatalmas szövegtörzsekben, amivel még koherensebb és releváns válaszokat tudnak létrehozni. Az előképzési szakaszban a fentebb is kifejtett jelöletlen szöveges adaton való képzés lehetővé teszi a modell számára a nyelv önmaga általi megismerését és annak statisztikai összefüggéseinek megtanulását (Rothe et al., 2020).

Lényegében a transzformátorokra való átállás jelentős előrelépést jelent a gépi tanulás és a szöveggenerálás területén. Ez kiegészülve a generatív modellek képességeivel valóban figyelemre méltó teljesítmény, ami felteszi számunkra a kérdést, hogy valójában a végső limitjei ezeknek az architektúráknak miben rejlenek. Meddig lehet őket fejleszteni annak érdekében, hogy még jobb eredményeket lehessen elérni? Hiszen már a transzformer modellek is kiválóan teljesítettek a komplex nyelvi feladatok elvégzésében (Wolf et al., 2020). Ugyanakkor a fentebb kifejtett működési elvek alapján továbbra is láthatjuk, hogy előre megadott adatok nélkül alapvetőleg a működés is kétséges. Pusztán abból tudnak tovább építkezni, amiket a fejlesztőik elérhetővé, tanulhatóvá tesznek a modellek számára, mint szövegtörzs, vagy a dolgozatban

jelenleg nem tárgyalt kép- és hanggenerálással kapcsolatos adatok. Amennyiben ezek hiányoznak ezen modellek számára nehéz létrehozni ezekből olyan dolgokat, amik nem feltétlenül léteznek, vagy értelmi és értelmezési problémákba is ütközhetünk. Chien és Kalita (2020) a nyelvi következtetési rendszerek limitációit vizsgálják, amik gyakran az említett transzformereken alapulnak, a nyelvi valódi megértésének tekintetében, valamint a hamis adatokra való sebezhetőségük tekintetében. Ugyanakkor munkájukban megjegyzik, hogy ebben a tekintetben szükséges a további kutatások elvégzése. Képességeik az adatvezérelt tanulás eredménye, és a neurális hálózatokhoz hasonlóan ők is elmaradnak a valódi kreativitástól, hiszen a meglévő adatokból kreálják az újat.

4. NAGY NYELVI MODELLEK

4.1. ÁTTEKINTÉS

A nagy nyelvi modellek megjelenése egy új fejezetet nyitott a természetes nyelvi feldolgozás, a gépi tanulás és a mesterséges intelligencia területein. A fenti fejezetekben tárgyalt működési elvek és mérföldkövek vezettek el oda, hogy a ma, már számunkra is elérhető, nagy nyelvi modellek elérhessék jelenlegi formájukat. A könnyebb átláthatóság érdekében egy rövid áttekintést nyújtok a feljebb tárgyalt fejlődési útról.

Az LLM-ek eredete a neurális hálózatok és az adatok szekvenciális feldolgozására tervezett visszacsatolt neurális hálózatok korai idejére vezethető vissza (Pascanu, 2012). Ezek a korai struktúrák ígéretesnek bizonyultak a természetes nyelvi feladatokban, de korlátokkal szembesültek, mint például a hosszú távú kontextusbeli függőségek megragadásával a szövegben, és szembesültek az eltűnő gradiens problémával (Pascanu, 2012). További fejlesztések és kutatások azonban találtak megoldást a korábban már kifejtett problémákra. Ilyen volt például az LSTM-ek megjelenése (Chung et al., 2014). Ezek a fejlesztések, valamint a nyelvi modellezési korlátok feltárása (Jozefowicz et al., 2016) hozzájárultak olyan nagy nyelvi modellek kifejlesztéséhez, amelyek forradalmasították a természetes nyelvi feldolgozást.

A nagy nyelvi modellek mai formájának kialakításához vezető úton döntő változás a transzformátorok bevezetése volt, egy innovatív neurális hálózati architektúra, amelyet hatékony párhuzamosításra és szekvenciális modellezésre terveztek (Wolf et al., 2020). A transzformátorok forradalmasították a természetes nyelvi feldolgozás területét azáltal, hogy orvosolták a korábbi modellek korlátait, például az RNN-ekét. A transzformátor modellek bemutatása és mások általi átvétele megnyitotta a kaput az LLM-ek ugrásszerű fejlődése előtt, amelyek a szöveggenerálásnak egy új korszakát nyitották meg (Devlin et al., 2018). A nagy

nyelvi modellek egyik korábban bemutatott példája a BERT (Devlin et al., 2018). Ahogy korábban kifejtésre került, ez a modell a transzformer architektúrákat és generatív modellekre jellemző mechanizmust is együttesen alkalmazta. A transzformátorok használatával az LLM-ek bebizonyították, hogy a tanulási képességeikkel még jobb teljesítményt tudnak nyújtani a természetes nyelvi feldolgozásban. Egy másik fontos előre lépési pont az előképzési és finomhangolási technikák alkalmazása az LLM-ekben. Ezeket a modelleket hatalmas mennyiségű szöveges adatra képezték ki. Az előképzést finomhangolás követi, ahol a modelleket konkrét nyelvértési- és generálási feladatokhoz igazítják (Devlin et al., 2018). Ezenkívül az LLM-ek létrehozása magába foglalja a modell méretének növelését és a kiterjedt képzési adatokhoz való hozzáférést. A transzformer modelleknél tárgyalt figyelemmechanizmus a nagy nyelvi modellek kulcsfontosságú építőköve (Vaswani et al., 2017). Az LLM-ek nyelvi modellezési feladatokat is végeznek, amelyek magukban foglalják a következő szó vagy szószorozat előrejelzését egy adott kontextusban. A különféle szöveges adatokra vonatkozó kiterjedt előképzés révén ezek a modellek jártasakká válnak a nyelvi minták és kapcsolatok széles spektrumának rögzítésében (Devlin et al., 2018). Ezek a működési tényezők a mély tanulási elvek és a neurális hálózatok használatával együtt hozzájárulnak a nagy nyelvi modellek hatékonyságához és sokoldalúságához.

4.2. AZ ISMERTEBB MODELLEK

A nagy nyelvi modellek sokféleségére pár példát szeretnék bemutatni, a különböző modellek rövid bemutatásán keresztül.

Az OpenAI által fejlesztett GPT-3 (Generative Pretrained Transformer 3) az egyik legnagyobb és legjelentősebb nagynyelvi modell (Raffel et al., 2019). A maga 175 milliárd paraméterével a GPT-3 rendkívüli szöveggenerálási képességekről tesz bizonyosságot (Raffel et al., 2019). Sokoldalúságát rengeteg feladatban bizonyította már, beleértve a tartalomgenerálást, a GPT-3 alapú chatbotokat és sok mást (Sezgin et al., 2022). A GPT-3-at számos ágazatban használták már, például közegészségügyi kérdésekben, egészségügyi ellátásban, forgatókönyvek generálásában (Jungwirth és Haluza, 2023; Nath et al., 2022). Valóság-hű, emberszerű szövegeket generáló képességét olyan területeken kamatoztatták, mint a játékok, a zene készítés világa és a vizualizáció (Vartinen et al., 2022; James et al., 2022; Paula és Susnjak, 2023). A GPT-3-at a ezen felül például címkézési költségek csökkentésében, az adatok összegzésében és a tartománytranszfer tanulásában rejlő potenciálja miatt is használták (Wang et al., 2021; Pang et al., 2022; Zhang et al., 2021). A GPT-3-at már a mérete miatt is, de

leginkább a benne rejlő potenciál miatt már rengetegen használták, több kutatás tárgyává is vált, és a nagyközönség számára is hatalmas előrelépést jelentett a megjelenése.

A másik nagyobb LLM, a korábban már sokat emlegetett BERT, amit a Google AI fejlesztett. A BERT-et széles körben alkalmazzák különféle területeken és feladatokban, beleértve a vélemény- vagy szentimentelemzést (Gao et al., 2019), a koreferencia-feloldást (Joshi et al., 2019), a kérdés-válasz eseteket (Qiao et al., 2019), és a rangsorolást (Qiao et al., 2019). Kiváló minőségű kontextualizált szóreprzentációkat generáló képessége a természetes nyelvi feldolgozás hatékony eszközévé tette (Zhou et al., 2021). A BERT jelentős hatást gyakorolt az NLP területére, és széles körben elismerték a nyelvi megértés és nyelvgenerálás javításához való hozzájárulását.

Számunkra érdekes lehet még a Google Research által fejlesztett T5 (Text-to-Text Transfer Transformer). Ez a modell minden természetes nyelvi feldolgozási feladatot szövegről-szövegre generálandó problémaként kezel (Raffel et al., 2019). Ez a megközelítés felhívta a figyelmet a T5 képességeire különféle nyelvi, szöveggeneráló feladatokban, ideértve a fordítást, a szövegösszegzést, a kérdések megválaszolását. (Raffel et al., 2019). Azzal, hogy a különböző feladatokat szövegtranszformációként kezeli, a T5 bemutatja az alkalmazkodóképességét a különböző nyelvi feldolgozási feladatokban (Raffel et al., 2019).

4.3. A NAGY NYELVI MODELLEK LIMITÁCIÓI

Így tehát láthatjuk, hogy a nagy modellek fejlesztéseik során hosszú úton mentek keresztül, és ez nem volt akadályok nélküli. Ma már széleskörűen alkalmazzák mindenféle iparágban, és pont emiatt rengeteg kérdést felvet az, hogy valójában hol vannak a határai az LLM-eknek, és rengeteg aggodalom övezi a jelenlétüket. A nagy nyelvi modellekkel kapcsolatos aggályok sokrétűek, és különböző kutatások, cikkek, publikációk tárgyalják őket. Az egyik legtöbbet tárgyalt ilyen jellegű aggodalom az elfogult és káros tartalomgenerálás lehetősége, mivel az LLM-ek akaratlanul is fenntarthatják a képzési adataikban jelenlévő sztereotípiákat és elfogultságokat, etikai kérdéseket vetve fel (Raffel et al., 2019). Ezenkívül a megtévesztésre irányuló támadásokra (megtévesztő input bevitele, például ellentétes példákkal) való fogékonyságuk és a félrevezető, hamis információk generálásának kockázata kihívást jelent a felelősségteljes használatukban (Hisamoto et al., 2020). Az LLM-ek képzéséhez és működtetéséhez szükséges hatalmas energiafogyasztás is már látható méreteket öltött, ami a fenntarthatóság szempontjából kérdőjelezhető meg (Chavannes et al., 2021), ugyanakkor ezt az aspektust jelen dolgozatomban nem tárgyalom. Ezen túlmenően aggályok merültek fel az adatvédelemmel és e modellek rosszindulatú felhasználásával kapcsolatban, dezinformáció

vagy hamisított tartalom generálásának tekintetében (Srinath et al., 2021). Az etikai kérdések megfontolása, akár az előképzés során használt adatok tekintetében, a felelősségteljes használat és az LLM-ek lehetséges negatív következményeinek kezelése a legfontosabbak bevezetésük során (Santy et al., 2021). Ezek az aggályok rávilágítanak a folyamatos kutatás és fejlesztés szükségességére a nagy nyelvi modellekkel kapcsolatos kockázatok mérséklése érdekében, ugyanakkor azt is megmutatják számunkra, hogy a fejlesztés és használat során fennáll annak a lehetősége, hogy ezeket valóban kiküszöböljük. Hiszen a lenyűgöző teljesítmények és a kockázatok ellenére is számos olyan dimenzióban alulmaradnak, ami valódi helyettesítőként szolgálhatna az emberi intelligenciára (Wolf et al., 2020). Ezen aggályokon megyek most végig, megkísérelve annak alátámasztását, hogy miért nem olyan aggasztó a helyzet, az általánosan elterjedt aspektussal ellentétben.

Az LLM-ek alapvetőleg adatvezérelt modellek, és méltán elismert teljesítményük ellenére az LLM-eket továbbra is korlátozza, hogy nagy mennyiségű szöveges adat előképzésére és finomhangolására támaszkodnak (Devlin et al., 2018). Kiválóak a nyelvi modellezési feladatokban, de gyenge fordításokkal vagy helytelen információkkal küszködhetnek, ha korlátozott adatokkal, ami jelenthet kevés adatot, vagy nem megfelelő minőségű adatot is, vagy speciális nyelvi kontextusokkal kell szembenéznük (Chronopoulou et al., 2020). Ezenkívül a nagy nyelvi modelleknek a további kérdései az értelmezhetőség tekintetében merülhetnek fel, mivel összetett architektúrájuk miatt gyakran fekete dobozoknak¹ tekintik őket (Azaria, 2022). Továbbá előfordulhat, hogy nem képesek teljesen megragadni az emberi nyelv árnyalatait, és hiányozhatnak a józan ész érvelési képességei (Schweidtmann et al., 2021). Ez leginkább olyan esetekben érzékelhető, saját személyes tapasztalat alapján, amikor az adott nyelv, egy olyan nyelv, amit az előképzési szakasz során nem tudott annyira megismerni, sok az átvitt értelem, és egy szónak több jelentése is létezik. Ilyenkor a megkülönböztetés a kontextustól függ, ami kevés vagy nem megfelelő minőségű adat esetében akadályokat szülhet.

A kreativitás és az innováció hiánya az LLM-ek egyik jelentős hátránya, ami szintén eloszlatja a kételyeket, ami a nagy nyelvi modelleket övezi. Míg az LLM-ek képesek szöveget generálni és megoldásokat kínálni a képzési adataik mintái alapján, nem képesek kreatívan gondolkodni vagy valóban eredeti ötleteket generálni (Du, 2023). Az LLM-ek a képzési adataikból elsajátított tudásra és mintákra korlátozódnak, és nem gondolkodhatnak e

¹ Belső működése láthatatlan az átlagos felhasználó számára.

meglévő tudás határain kívül (Du, 2023). Ez a korlát annak a ténynek köszönhető, hogy nem képesek a vágy megértésére és a kreativitás fogalmának megfogalmazására, ahogyan az emberek teszik (Du, 2023). Szöveggenerálási képességeik ellenére az LLM-eket végső soron a képzési adataik korlátozzák, és nem tudnak olyan ötleteket generálni, amelyek túlmutatnak azon, aminek ki vannak téve (Du, 2023). Ugyanakkor amikor ki is vannak téve ezeknek az adatoknak, és saját ötleteket generálnak, nem biztos, hogy megfelelő eredmény születik. Emellett láthatjuk, hogy a ChatGPT is teljes mértékben hihetőnek hangzó válaszokat adhat, de hiányzik a tartalom mélyebb megértése, ami potenciális ténybeli pontatlanságokhoz vagy logikai buktatókhoz vezethet (Leaver és Srdarov, 2023). Wang et al. (2023) elvégeztek egy kísérletet, ahol a Minecraft nevű nyílt világú videójátékban létrehoztak egy karaktert, amit LLM irányított. A karakter a modell képességeivel remekül tanult, és gyorsan fejlődött, azonban akadályai közt szerepelt, amikor már megismert elég tárgyat a játékban, ami legyen Tárgy 1 és Tárgy 2. Tárgy 1-ben megtalálható Összetevő 1 és Összetevő 2. Tárgy 2-ben Összetevő 1 és Összetevő 3. A játékban fennáll egy olyan szituáció, hogy olyan tárgyat nem lehet létrehozni, amiben Összetevő 2 és Összetevő 3 található meg, mert ilyen nem létezik a játékban. A szerzők pedig említik, hogy a karakter mégis ezt próbálta meg megalkotni, amivel megakadt a játékban. Ezzel láthatjuk, hogy ugyan képes volt valami újat alkotni a megadott kereteken belül, ugyanakkor ez kontextuálisan nem volt értelmezhető és releváns a játékban.

A nagy nyelvi modellből hiányoznak a valódi, emberszerű érvelések, a kritikus gondolkodás és az intellektuális ítélőképesség. Nem képesek az információkat értékelni, a bizonyítékokat mérlegelni, racionális következtetéseket levonni vagy árnyalt érveket alkotni, ami leginkább a nyelv és az emberi gondolkodás, az észlelés mélységéből fakad, ami hiányzik az LLM-ekből. Az emberektől eltérően az LLM-ek kizárólag a képzési adataik alapján adnak determinisztikus válaszokat, így alkalmatlanok a magasabb rendű kognitív képességeket igénylő feladatokra (Collins et al., 2022). Míg ezek a modellek sikereket értek el a néhány lépésből álló tanulási feladatokban, képességeik a képzési adataikban megjelenő mintákra és információkra korlátozódnak (Kojima et al., 2022), amiket már korábban is tárgyaltunk a fejezetek során. Hiányzik belőlük az érvelés képessége, ami abból a tényből fakad, hogy valójában a szövegtörzsből lekövetkeztethető statisztikai valószínűséget veszi alapul az érvelésében, legyen az matematika, fizika vagy más terület (Collins et al., 2022). Mivel az LLM-eket úgy tervezték, és tervezik, hogy a nyelv formáját és összefüggéseit tanulja meg és nem annak jelentését ragadják meg, így olyan feladatokkal küzdenek, amelyek szekvenciális aritmetikát, többlepcsős kérdés-válaszolást, mondatfelismerést és összehasonlítást, valamint

elágazó logikát igényelnek (Wu et al., 2021). Ez felvetheti annak a kérdését is, hogy a memóriájuk meddig és milyen hosszon képes kezelni az adat lekérdezést és az adatbevitelt. Az itt felsorolt korlátok kiemelik az LLM-ek és az emberi kognitív képességek közötti alapvető különbségeket, hangsúlyozva az óvatosság szükségességét, amikor a nagy nyelvi modellekre hagyatkozunk olyan feladatokban, amelyek kritikus gondolkodást és emberi ítélőképességet igényelnek.

Emellett öröklök a képzési adataikban jelenlévő torzításokat, amelyek elfogult vagy diszkriminatív tartalom generálásához vezethetnek (Nangia et al., 2020). Ez etikai aggályokat vet fel, különösen azokban az alkalmazásokban, ahol a méltányosság, valamint a kulturális és társadalmi különbségekre való érzékenység kulcsfontosságú (Frissen et al., 2022). Az LLM-ekben jelenlévő torzítások állandósíthatják a társadalmi elfogultságokat és megerősíthetik a sztereotípiákat (Guo & Çalışkan, 2021). A kutatások kimutatták, hogy az LLM-ek képesek megragadni és reprodukálni a nemhez, fajhoz, etnikai hovatartozáshoz és más tulajdonságokhoz kapcsolódó torzításokat (Abid et al., 2021). Ezek az elfogultságok különféle módokon nyilvánulhatnak meg, például negatív szóasszociációkban, elfogult mondatkiegészítésekben és szélsőséges érzelmek elemzésében (Sheng et al., 2021). A torzítások jelenléte az LLM-ekben rávilágít a modellek felelős használatának szükségességére, mivel valós ezen torzítások megjelenése súlyos következményekkel járhat (Mehrabi et al., 2019). Jelenleg is dolgoznak olyan módszerek kidolgozásán, amelyek kimutathatják és enyhíthetik az LLM-ek torzításait, de ez továbbra is folyamatos kihívást jelent (Cao et al., 2022). Az LLM-ek etikus használatának biztosítása megköveteli a képzési adataikban jelenlévő torzítások alapos mérlegelését, valamint kimeneteiknek az egyénekre és a társadalom egészére gyakorolt lehetséges hatását, amennyiben az nem megfelelően van felhasználva, vagy nincs utólag értékelve (Weidinger et al., 2021). Ezek az érvelések is alátámasztják miszerint a nagy nyelvi modellek ezeket nem önmaguktól generálják, hanem pusztán az adathalmaz, amire tanítják ezeket az architektúrákat, rendelkezik ezekkel a torzításokkal, sztereotípiákkal és nem ismeri a szociális érzékenységgel kapcsolatos emberi ismereteket és az empátia fogalmát. Pontosabban ismeri ezeknek a fogalmát, de nem tudja mélységében értelmezni azt és jelentőséget tulajdonítani neki. Ez megköveteli az emberi felügyeletet ilyen kérdések tárgyalásakor, hiszen az előképzés során szükséges ezek jelenlétét szűrni és szabályozni. Ezekből az elfogultságokból és szélsőségekből megállapítható, hogy ténylegesen az adata és az azok közötti statisztikai kapcsolatokra alapszik a működésük, hiszen ezek a negatív megítélések pusztán sztereotípiák és nem feltétlenül fedik a valóságot. Mivel a valóságtól eltérő dolgokat állít, és azokra alapozza

érvelését így feltételezhető, hogy távol állnak az emberszerű gondolkodástól az LLM-ek, nem képesek megérteni az általuk generált szöveg jelentését vagy jelentőségét (Clark et al., 2019).

Összefoglalva, láthatjuk, hogy bár a nagy nyelvi modellek hosszú utat tettek meg, még nem igazán érezhető annak a megalapozottsága, hogyan ki tudná váltani az embert teljes mértékben. Alapvetően olyan koncepciókon alapul a létezésük, amelyek a fentebb is tárgyalt adatvezéreltség, a finomhangolás. Az emberek előnyére válhat ilyen szempontból a kreativitás hiánya, a valódi megértés és észlelés hiánya. Ezek nélkül nem tud teljes mértékben egy emberpótlék lenni. Szignifikáns előrelépést mutattak a gépi tanulás és mesterséges intelligencia területein, amik elhanyagolhatatlanok, de véleményem szerint ez sokkal inkább azt követeli meg, hogy ne arra fókuszáljunk ezzel kapcsolatban, hogy mi az amit elvehet tőlünk, hanem, hogy mi tud hozzánk adni, hogyan tudjuk előnyünkre formálni a nagy nyelvi modellek kínálta lehetőségeket.

Azonban mielőtt a legvégső következtetéseket vonnám le, szeretném egy kísérlettel bemutatni, az LLM-ek alapjaiul is sok esetben szolgáló RNN-ek képességeit. Mindezt annak érdekében, hogy bizonyosságot nyerjen az adatvezéreltség és a finomhangolás szerepe, és annak a bizonyítása, hogy az új dolgok létrehozása a meglévők nélkül nem lehetséges.

5. RNN, MINT SZEMLÉLTETŐ PÉLDA

Most, hogy áttekintettem a nagy nyelvi modellek felépítését az alapoktól kezdve, szeretném bemutatni a saját példámat, amin keresztül a visszacsatolt neurális hálózat képességeit és korlátait szeretném bemutatni. Mint már korábban kifejtettem a visszacsatolt neurális hálózatok tökéletesen használhatóak például nyelvfelismerésre, idősor előrejelzésre, hiszen ezek remek tanulási képességgel rendelkeznek (Gilley, 2023). A továbbiakban leginkább szemléltetésre szeretném használni, de be is mutatnám a működését, hogy több értelmet nyerjen a működése. Az elkészült kód a saját, és az internet adta lehetőségek munkája, amiket a magyarázat során hivatkozom, valamint Gong 2016-os GitHub-on elérhető oktatóanyaga. Megépítésében segítségemre volt a ChatGPT és a Blackbox AI is. A teljes kód az I. mellékletben található meg, ebben a fejezetben csupán részleteire fogok hivatkozni.

A kód első részletét két részre lehet osztani. Először a könyvtárat importálja, ami esetünkben a Numpy, ami nagyban megkönnyíti a dolgunkat: `import numpy as np`. Ezt a numerikus számításokhoz használják. A Numpy sokféle függvényt tartalmaz, például logikai összehasonlításokhoz is. Mivel a Numpy gyorsabb, mint maga a Python, így érdemes már alapvetőleg úgy írni a kódot, hogy vagy vektorizálunk, vagy Numpy kompatibilis függvényeket

használnuk. Így fel tud gyorsulni a kódunk, és maga a kódolás is (Raj, 2022). A másik része a `# Minta szöveg`, ami a szöveg minta, amiből tanítom a modellt. Ez a fő adatunk, ami alapján a kimenetet meg fogja határozni. Jelenleg ez 3 rövid „mondatból” áll.

A `# Szöveg előfeldolgozás` rész a szöveget, a fentebb említett 3 „mondatot” dolgozza fel, karakterszintű leképezések létrehozásához. A `chars` a szövegben megtalálható egyedi karakterek listája. A `char_to_idx` egy szótár, ami a karaktereket egész indexekre képezi le, az `idx_to_char` pedig az a szótár, ami ezeket az indexeket karakterekre képezi vissza. A `vocab_size` az adatkészletben lévő karakterek teljes száma, „nagysága”.

A következő lépés a modell `# Hiperparaméterek` meghatározása volt. Ezek határozzák meg a modell architektúráját és a tanulási rátát, a betanulási folyamatot. A `hidden_size` a rejtett réteg mérete, a `sequence_length` a bemeneti szekvenciák hossza, a `learning_rate` pedig a betanulás során a gradiens süllyedés lépéseinek méretét határozza meg. A matematikában ez egy optimalizáló algoritmus egy differenciálható függvény lokális minimumának megtalálására. Jelenleg számunkra ebben a modellben a gépi tanulási megközelítésben értelmezhető. Tehát a paraméterek optimális értékeit találja meg a meghatározott tanulási ráta és kezdeti paraméterértékek segítségével (Lu, 2022). Ezen hiperparaméterek finomhangolása bizonyult az egyik legidőigényesebb feladatnak, hiszen megtalálni a megfelelő kombinációt egy ilyen kicsi adatkészleten kihívás, főleg, mivel egy viszonylag kisebb modellről van szó, ami alapvetőleg is karakterszintű. Ezen felül, az eredmény értékelése is ezektől a paraméterektől függött. Az output értékelése szubjektív ebben az esetben, mert annyi különböző beállítástól függ a végeredmény, de a végeredménnyel kapcsolatos elképzelések is befolyásolhatták a beállításokat, attól függően, hogy mennyire szeretnék egy teljesen egyező szöveget, mennyire akarom diverzifikálni az outputot az RNN adta keretek között.

A következő szekció a `# Modell paramétereinek inicializálása`. A `Wxh, Whh, Why` a modell súlymátrixai, a `bh, by` pedig a torzításvektorok. A súlymátrixok a bemenetek súlyait határozzák meg annak érdekében, hogy kimenetet tudjanak produkálni. A pontosság és a teljesítmény meghatározásában játszanak szerepet (Klein, 2023). Az első az input adatok és rejtett rétegek közötti súlyokat tartalmazza, a második magukat a rejtett rétegek közöttit, a harmadik pedig az output és a rejtett rétegek közötti súlyokat foglalja magában. A torzításvektorok a mesterséges neurális hálózatokban pedig úgy definiálhatók, mint a jellemzők és a súlyok szorzatához hozzáadott konstans. Ez az eredmény ellensúlyozására szolgál (Ganesh, 2020). Nem függenek a bementektől, csak a kimeneti eredményeket befolyásolják. Érdekes ezeket kicsi és random számokra állítani, ugyanis ezzel lehet biztosítani, hogy a kimeneti eredményben ne jelenjen

meg túlillesztés² és javíthatja a hálózat teljesítményét (Brownlee, 2018). Tehát ahogy itt is látszik a rejtett rétegek és az output közötti torzításvektorok vannak beállítva. Jelenleg itt a kezdeti érzékek vannak beállítva a modellnek, amiket a képzés során fog frissíteni, hogy képes legyen megfelelően reprezentálni és előrejelzést készíteni a sorozatos adatokból.

A következő szekció maga `# Képzés`. Végighalad a szövegtészlet minden egyes karakterén és kiszámítja a veszteséget és frissíti a modell paramétereit a beállított epoch-ok alkalmával. Itt, amit fontos lehet megemlíteni, hogy `loss` egy változó, amely a tanítás során a veszteséget kumulatíván számolja a tanítási szekvencia minden karakterére. Ennek a bizonyos keretek között tartása bizonyult a másik kihívásnak, ugyanis a hiperparaméterek állítása nagyban befolyásolta a `loss`-t. A túl magas `loss` teljesen inkoherens adathalmazhoz vezetett, ami korántsem hasonlított a meglévő adatkészletre. Például amikor a egy korábban általam használt adatkészlet a következő volt: „Gyűlölök mindent. Utálom az embereket. Utálom magam.” a 3. ábrán látható eredmény született, relatíve a képzési adathoz képest magas `loss` értékkel.

```
Iteration 100, Loss: 163.19829265163293
Iteration 200, Loss: 179.07772481454077
Iteration 300, Loss: 173.56587937803425
Iteration 400, Loss: 176.46378759067315
Iteration 500, Loss: 172.19741613176225
Iteration 600, Loss: 165.89960188963383
Iteration 700, Loss: 128.84697033967848
Iteration 800, Loss: 151.85401036567745
Iteration 900, Loss: 171.8552788281043
Gyűlölök mindent. Utálom az embereket. Utálom magam.
m eön. mő e nrlo nom lz m m m a mml em r m g ezmbm lmm et e mē m lon l e ez lo e emmāem.m.l l miezlolilim.nzl m m lőē fzm. eem.
. m.mn n t ō e fz m m moltl e m m ez ezmmmm lrkzt.melm zē lomt mnte z m lom.lzmi .mim m lo lomt.n ēvm eomtr m lknbl mōmāmmm knbe f em. e
zkēbt létmb tdlēl ezm.lzm.m ezm.erm.lzlo mē eznz t.
```

3. ábra Loss probléma bemutatása

A képzés fő része a tanítási szekvencia karakterein megy végig, és számolja az előbb kifejtett veszteséget, `loss`-t. A bemenetet (`x`) one-hot kódolással (minden egyedi egész értékhez egy bináris változót ad) reprezentálja, majd kiszámolja a rejtett rétegeket (`h`) és a kimenetet (`y`). Az aktivációs függvény a rejtett rétegekhez a tangens hiperbolicus függvény (`np.tanh`) használatával kapcsolódik. Az `loss` kiszámolása a célsorozat és a modell által előrejelzett valószínűségek alapján történik. Ebben a részben fontos még kiemelni, hogy a gradiens normalizálása is itt történik meg a súlyok túlzott változásának elkerülése érdekében. Az `np.clip` függvény segítségével a gradiens korlátozva vannak egy adott érték tartományon belül.

A fentebb betáplált képzést a `# Képzési loop` szekcióban meghatározott számú iterációig folytatja a modell. Minden egyes ismétlésnél frissíti a paramétereit annak érdekében, hogy jobb output-ot tudjon létrehozni. Az epoch-ok számát jelenleg 1000-re állítottam, ez bizonyult a

² Pontos előrejelzési adatokat ad a modell a már meglévő adatokra, de nem új adatokra.

legjövödelmezőbbnek annak tekintetében, hogy a loss-t minimalizálni szerettem volna. A tanulást a hiperparaméterekben lefektetett tanulási ráta is befolyásolja, ami leginkább a gyorsaságban látszik meg. Ez egyfajta stabilitást is ad a modellnek, mert a túl gyors ráta nem megfelelő outputot fog eredményezni, nem fogja tudni a modell rendesen megtanulni az adatkészletet. Míg a nagyon lassú pedig az 1000 iterációra nem megfelelő, mert ideje nincs megtanulni. Ennél hosszabb iteráció pedig túltanuláshoz vezethet.

Az utolsó részben pedig már a tanult adatokból való `# Szövegenerálás` rész látható, előre meghatározott hosszban és karakterrel, majd ezt a képernyőre is kiírja.

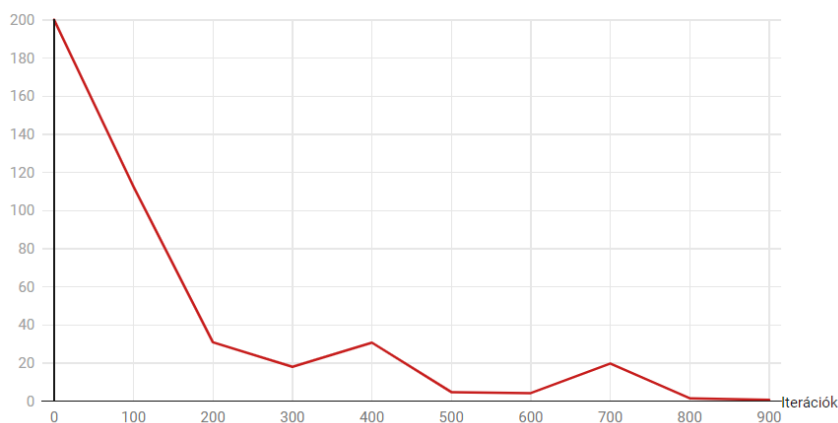
A kód tesztelése és futtatása Visual Studio Code-ban történt meg. Az első futás után az I. mellékletben található képzési adat alapján a következő szöveg generálására volt képes:

```
A mai zh angolul
Angol nem jó. PALI
AnALjt. PAngolul nem óA nem tudok.
Angol nem jó. PALI
AnAolt. nAm jó. PALI
Anem tuk nem tudol.
Angol nem jó. PALI
AnAngol nem jó. PALI
AnAngolul nem tudok.
Angol nem jó. PALI
Angol nem jó. PALI
deAngolul nem tudok.
Angol nem jó. PALI
AnALjó
nAok. PAnejolul nem t nAm jó. PALI
Aneolt. nAngol nem jó. PALI
Angolul nem tudok.
Angol nem jó. PALI
AnA jó. PAngolul nem tudok.
Angol nem jó. PALI
AnALjó
nem jó. PALI
I.
Angolul nem tud
```

Ahogy fentebb is látható, találhatóak közte értelmes mondatok, de azok olyanok, amik már előtte is léteztek a mondatban, és csak akkor lett koherens, amikor ugyanezeket használta fel. Ezen kívül pusztán azokból a karakterekből dolgozott, amiket már korábban beadtam a modellbe. Például az I betűt csak nagybetűként ismerte, így a szövegben is csak nagybetűként szerepel. Korábbi fejezet során említettem, hogy a következő karakter jóslásának problémája gondot jelenthet az ilyen modellekben. A kódban meg van adva, azon karakterek száma, amiket figyelnie kell a tanulás során, amihez társíthatja a következő karakterre vonatkozó előrejelzést. Mivel az input is egészen rövid, így nem sok karaktert tud figyelni, ennek következménye, hogy az output sok helyen inkoherensnek tűnik. Emellé társul még az a koncepció is, hogy

alapvetőleg nem társul megértés, pusztán statisztikai összefüggéseken alapul a modellnek a felépítése és működése. Továbbá nem változatos, és nem szolgál új értelemmel, ideával a létrehozott szöveg. Ebből következtethetünk és alátámaszthatjuk azt az állítást, hogy ezeknél a modelleknel nem a valódi észlelésről és megértésről van szó, hanem csupán a finomhangolási módszereket jól implementált architektúrákról van szó. Ugyanakkor fontos megjegyezni itt is, hogy itt csupán egy karakterszintű RNN modellről van szó, ami korántsem egyenlő az általunk is jól ismert nagy nyelvi modellekkel, mint például a ChatGPT, hiszen hiányoznak belőle a korábban kifejtett transzformerek, a figyelemmechanizmus, a generatív jelleg. Így nem lehet egy fogalom és következtetés alatt megfogni őket, azonban az alapja az ilyen nagy modelleknek is a kisebb neurális hálózatokra épül, és azt nem lehet tagadni, hogy azok is a meglévő adatkészleteikre hagyatkoznak.

1. futtatás közbeni loss értékek



4. ábra Loss vizualizációja az 1. futtatás során a datawrapper.de segítségével

A 4. ábrán látható megjelenítve a loss egyre csökkenő mivolta. Ez esetünkben jót jelent, hiszen folyamatosan tanul a modell. További kutatást és kísérletezést igényel ennek a stabilizálása, ugyanis itt is látszik, hogy néhol kiugrik ez az érték, tehát hiába, hogy tanul, néhol instabillá válik. Fontos megjegyezni, hogy ezen modellel a célom az adatfüggőség bemutatása volt, így a loss nagyon alacsony értékének elérése volt a cél, annak érdekében, hogy látványos legyen a bemeneti adatok karaktereitől eltérő karakterek hiánya. Ugyanakkor az ideális loss értéket mi határozzuk meg, ahhoz, hogy valódi, emberszerű szöveget kapjunk vissza, hogy a gondolkodás látszatát fenntartsuk, szükséges némi veszteség elérése, hiszen nem teljesen ugyanazt a szöveget szeretnénk visszakapni.

Fontos ismét kiemelni, hogy mindezek mellett ez egy nagyon egyszerű, karakterszinten működő modell, ami nem teljes mértékben tükrözi a nagy nyelvi modellek teljes képességének

spektrumát, valamint hiányoznak belőle azok az összetevők, amik az LLM-eket LLM-é teszi. Ezzel a kísérlettel pusztán a legalapvetőbb mechanikával működő modellek képességeit szerettem volna megmutatni, ami alátámasztja, hogy már a legkisebb elemében is, mindene attól függ milyen és mennyi adatot kap, és később ezeket az adatokat milyen módon használhatja fel. Más szavakkal milyen tanulási korlátozások és szabadságok elérhetőek számára, ami például jelentheti a különböző paramétereket, tanulási sebességet.

6. ÖSSZEGZÉS

Láthattuk tehát, hogy a nagy nyelvi modellek milyen utat is jártak be egészen addig, amíg elérték a ma is ismert formájukat, amiből már a nagyközönség is kaphatott ízelítőt, leginkább a ChatGPT-nek köszönhetően. Annak széleskörű megjelenése sokakban aggodalmakat keltett a felhasználását illetően és további lehetséges képességei alapján. A legtöbb aggály továbbra is a jövőbeli formáját illetően fogalmazódott meg, olyan tekintetben, hogy mennyire tud kiváltani egy hús-vér embert.

Ezen aggályra reagálva mutattam be a nyelvi modellek fejlődésének történetét és ezáltal a mechanikát is, ami alapján működnek ezek. Láthattuk a dolgozat elején, hogy a mai legszélesebb körben ismert nyelvi modellek alapjai és elődei az ún. neurális hálózatok. Működésüket, még ha csak a felszínt kapargatva is, de megismertük, ezzel kaptunk egy képet arról, hogy meddig és milyen minőségben tudjuk őket önmagukban hasznosítani. A neurális hálózatokat bemutató fejezet végén kifejtésre került, hogy ezek az architektúrák esszenciálisan adatok által vezérelt modellek tanítása, nem pedig egy önálló entitás létrejötte, ami képes lett volna a kreativitás, az észlelés vagy az önálló gondolatok létrehozására. Később kifejtésre kerültek a transzformátorok és a generatív modellek, majd ezeknek az ötvöztetésével és lehetőségeiknek kiaknázásával a nagy nyelvi modellek kifejlesztése. Itt is felszínesen láthattuk a működési elveiket, és azt, hogy továbbra sem saját gondolatokból táplálkoztak, hanem az ember által készített adathalmazokkal dolgoztak különböző matematikai módszerek segítségével. A figyelemmechanizmus bevezetése ezekbe a modellekbe egy sarkalatos pont volt a fejlődésüknek útján, ami méginkább emberszerűvé tette ezeket, és felszínre került általa a kérdés, hogy miképpen is működnek ezek és mennyire áll közel az emberi agy működéséhez. Ahogyan pedig az LLM-ek kialakultak méginkább sokak számára aggasztó lehetett az általuk generált szövegek minősége, mennyisége. Azonban a valódi, mögöttes működés és architektúra alapjaiban változatlan maradt. Így kijelenthető, hogy továbbra is adatvezérelt modellekről van szó, ami segít alátámasztani, miszerint olyan dolgokat nem tud létrehozni, aminek sosem volt kitéve, amivel sosem találkozott. A kreativitás és az innováció hiánya a nagy nyelvi modellek

másik olyan tulajdonsága, ami segíthet a kételyek eloszlatásában. Képesek új szöveget generálni és megoldásokat kínálni a képzési adatokban megtalálható minták alapján, ez nem nevezhető kreativitásnak és innovációnak. A képzési adataik és a fejlesztők által megadott paraméteres korlátok keretei között képesek működni, így ezek valódi képességeinek mi szabjuk meg a határait. Ezenkívül pedig tárgyaltuk az etikával és szociális érzékenységgel kapcsolatos kérdéskört, amivel véleményem szerint továbbra is alá tudjuk támasztani azt az állítást, hogy nem képesek az emberszerű, kognitív gondolkodásra, a megértésre, a nyelvi mélységek megismerésére. Ez egy óriási hátrány, ami miatt nem teljes mértékben képes az ember helyett minden feladatot ellátni.

A dolgozat végén bemutatásra került RNN pedig annak érdekében készült el, hogy szemléltethessem az adatvezéreltséget, az ember által megszabott korlátokat, aminek berkein belül képes működni. Fontos ismét kiemelni, hogy egy karakterszintű, nagyon egyszerű rekurrens neurális hálózatról van szó, ami nem egyenlő az LLM-ekkel, így nem erős evidencia minden pont alátámasztására, de az esszenciáját képes bemutatni számunkra. Ezzel reményeim szerint szélesebb körben és közérthetőbb módon rá tudtam világítani a limitációkra, amikkel ezek a modellek sokkal inkább egy emberi kiegészítő lehet, mintsem helyettesítő.

Véleményem szerint, és azon vélemények és állítások alapján, amiket a dolgozat folyamán tárgyaltam a nagy nyelvi modelleknek vannak hátrányai az emberre nézve, de ezek sokkal inkább olyan architektúrák, amiket nem fenyegetésként kellene kezelni, hanem mint egy lehetőség az emberi képességek hatékonyságának növelésére. Folyamatok gyorsítására tökéletesen alkalmas egy ilyen modell beépítése a mindennapjainkba, akár a munkahelyekre. Monoton ellenőrzési feladatokban kiválóan tudnak szűrni, és előrejelzési adatokat is kreálni, amikkel a hatékonyságunkat tudjuk növelni. Természetesen jelenlegi formájukban még vannak hibáik, és további kutatásokat igényel a teljeskörű korlátaik megismerése. Ugyanakkor a dolgozat során tárgyalt érvekkel összhangban az embereket sokkal inkább a használatukra lehet buzdítani, mintsem az aggodalmak számának növelésére. Az LLM-ek esetében mi szabunk határt: milyen mennyiségben, milyen minőségű adatnak tesszük ki őket, mihez engedjük hozzáférni, milyen paramétereken belül engedjük szabadjárá az öntanulás útján. Alkalmazkodási képességeiket már most láthatjuk, hiszen rengeteg ágazatban hasznosították már a tudásukat, ami a feladatok hatékonyságát, gyorsaságát növelte, egyszerűbbé tette számunkra is a folyamatokat. Távlatos analógiákban a folyamatosan fejlődő világgal lépést kell tartani számunkra is, és véleményem szerint a helyes hozzáállás az alkalmazkodás és a

folyamatos tanulás az emberek számára is. A világ folyamatos változása magával hozza a szükséges rugalmasságot, ami elengedhetetlen a mai világban.

További kutatás szükséges a dolgozat során bemutatott RNN finomhangolásának és mélyebb megismerésének érdekében, valamint későbbi lehetséges továbbfejlesztésének érdekében is. Továbbá az újabbnál újabb kutatások és fejlesztések során egyre több lehetőséget tárnak fel, és az LLM-ek korlátait feszegetve keresik azok további limitációit, valamint a még emberszerűbb mechanikák elérését. Ezeket a továbbiakban érdemes lehet figyelemmel kísérni, hiszen ezek fognak további alapokat adni a dolgozatban tárgyalt kérdéskörre is. A későbbiekben célom folytatni a kísérletezést a különböző modellek működésével kapcsolatban, és azok képességeit mélyebben megismerni.

Összességében a limitációk bemutatásával, és maga a működési elvek bemutatásával rá tudtam világítani arra, hogy miért is kellene nagyobb erőfeszítést fordítani az aggodalmak eloszlatására, valamint az azokkal kapcsolatos technikai megoldások fejlesztésére és javítására koncentrálni, annak érdekében, hogy ne ember-helyettesítő legyenek az LLM-ek, hanem az embert segítő 21. századi mechanizmusok.

IRODALOMJEGYZÉK

Abid, A., Farooqi, M. és Zou, J. (2021). Persistent Anti-Muslim Bias in Large Language Models. *arxiv.org*. [online] Letölthető: <https://arxiv.org/abs/2101.05783>.

Azaria, A. (2022) ChatGPT Usage and Limitations. OSF Preprints. doi: 10.31219/osf.io/5ue7n.

Bhaskar, Y. (2023). *Introduction to LLMs and the generative AI : Part 1- LLM Architecture, Prompt Engineering and LLM Configuration*. [online] Medium. Letölthető: <https://medium.com/@yash9439/introduction-to-llms-és-the-generative-ai-part-1-a946350936fd> [Utolsó letöltés: 2023. október 26.].

Brownlee, J. (2018). *Why Initialize a Neural Network with Random Weights?* [online] Machine Learning Mastery. Letölthető: <https://machinelearningmastery.com/why-initialize-a-neural-network-with-random-weights/>.

Cao, Y.T., Pruksachatkun, Y., Chang, K.-W., Gupta, R., Kumar, V., Dhamala, J. és Galstyan, A. (2022). *On the Intrinsic and Extrinsic Fairness Evaluation Metrics for Contextualized Language Representations*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2203.13928>.

Chien, T. és Kalita, J. (2020). Adversarial analysis of natural language inference systems. pp.1–8. doi:<https://doi.org/10.1109/ICSC.2020.00008>.

Chronopoulou, A., Stojanovski, D. és Fraser, A. (2020). *Reusing a Pretrained Language Model on Languages with Limited Corpora for Unsupervised NMT*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2009.07610>.

Chung, J., Gulcehre, C., Cho, K. és Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. [online] arXiv.org. Letölthető: <https://arxiv.org/abs/1412.3555>.

Clark, K., Khandelwal, U., Levy, O. és Manning, C.D. (2019). What does BERT look at? An analysis of BERT's attention. [online] Association for Computational Linguistics, pp.276–286. doi:<https://doi.org/10.18653/v1/W19-4828>.

Collins, K.M., Wong, C., Feng, J., Wei, M. és Tenenbaum, J.B. (2022). *Structured, flexible,*

and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. [online] arXiv.org.

doi:<https://doi.org/10.48550/arXiv.2205.05718>.

Cristina, S. (2021). *The Transformer Model*. [online] Machine Learning Mastery. Letölthető: <https://machinelearningmastery.com/the-transformer-model/>.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. és Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

doi:<https://doi.org/10.18653/v1/p19-1285>.

Devlin, J., Chang, M.-W., Lee, K. és Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. [online] arXiv.org. Letölthető: <https://arxiv.org/abs/1810.04805>.

Du, M. (2023) How large language model and Computer Vision demonstrate Creativity in an evolutionary perspective: Can AI Understand Desire?. PsyArXiv. doi: 10.31234/osf.io/ejrvk.

Fazekas, I. (2013). *Neurális hálózatok*. [online] Debrecen: Debreceni Egyetem Informatikai Kar. Letölthető: https://gyires.inf.unideb.hu/GyBITT/19/Neuralis_halozatok_v8.pdf [Utolsó letöltés: 2023. október 26.].

Frissen, R., Adebayo, K.J. és Nésa, R. (2023). A machine learning approach to recognize bias and discrimination in job advertisements. *AI & SOCIETY*, [online] 38(2), pp.1025–1038. doi:<https://doi.org/10.1007/s00146022015740>.

Ganesh, S. (2020). *What's The Role Of Weights And Bias In a Neural Network?* [online] Medium. Letölthető: <https://towardsdatascience.com/whats-the-role-of-weights-és-bias-in-a-neural-network-4cf7e9888a0f>.

Gao, Z., Feng, A., Song, X. és Wu, X. (2019). Target-Dependent Sentiment Classification With BERT. *IEEE Access*, 7, pp.154290–154299. doi:<https://doi.org/10.1109/access.2019.2946594>.

Gilley, S., Franks, L., Brizi, M., Gayhardt, L., Gronlund, C.J., Quintanilla, L., Withee, K., Coulter, D. és Lu, P. (2023). *Mély tanulás és gépi tanulás - Azure Machine Learning*. [online]

learn.microsoft.com. Letölthető: <https://learn.microsoft.com/hu-hu/azure/machine-learning/concept-deep-learning-vs-machine-learning?view=azureml-api-2> [Utolsó letöltés: 2023. október 12.].

Gong, Y. (2016). *Implementing Recurrent Neural Network from Scratch*. [online] GitHub. Letölthető: <https://github.com/gy910210/rnn-from-scratch> [Utolsó letöltés: 2023. október 28.].

Guo, W. és Caliskan, A. (2021). Detecting emergent intersectional biases: Contextualized word embeddings contain a distribution of human-like biases. [online] Virtual Event, USA: Association for Computing Machinery, pp.122–133. doi:<https://doi.org/10.1145/3461702.3462536>.

He, K., Zhang, X., Ren, S. és Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770–778. doi:<https://doi.org/10.1109/cvpr.2016.90>.

Hisamoto, S., Post, M. és Duh, K. (2020). Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Transactions of the Association for Computational Linguistics*, [online] 8, pp.49–63. doi:https://doi.org/10.1162/tac1_a_00299.

Huang, S.-C. és Le, T.-H. (2021). Chapter 11 - Generative adversarial network. In: S.-C. Huang and T.-H. Le, eds. [online] Academic Press, pp.255–281. doi:<https://doi.org/10.1016/B978-0-323-90198-7.00011-2>.

Joshi, C. (2020). *Transformers are Graph Neural Networks*. [online] NTU Graph Deep Learning Lab. Letölthető: <https://graphdeeplearning.github.io/post/transformers-are-gnns/>.

Joshi, M., Levy, O., Zettlemoyer, L. és Weld, D. (2019). *BERT for Coreference Resolution: Baselines and Analysis*. [online] ACLWeb. doi:<https://doi.org/10.18653/v1/D19-1588>.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N. és Wu, Y. (2016). *Exploring the Limits of Language Modeling*. [online] arXiv.org. Letölthető: <https://arxiv.org/abs/1602.02410>.

Jungwirth, D. és Haluza, D. (2023). Artificial Intelligence and Public Health: An Exploratory Study. *International Journal of Environmental Research és Public Health*, 20(5), p.4541.

doi:<https://doi.org/10.3390/ijerph20054541>.

Kingma, D.P. és Welling, M. (2013). *Auto-Encoding Variational Bayes*. [online] arXiv.org. Letölthető: <https://arxiv.org/abs/1312.6114>.

Klein, B. (2023). *14. Neural Networks, Structure, Weights and Matrices*. [online] python-course.eu. Letölthető: <https://python-course.eu/machine-learning/neural-networks-structure-weights-és-matrices.php>.

Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. és Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. *arXiv:2205.11916 [cs]*. [online] Letölthető: <https://arxiv.org/abs/2205.11916>.

Leaver, T. és Srdarov, S. (2023) “ChatGPT Isn’t Magic : The Hype and Hypocrisy of Generative Artificial Intelligence (AI) Rhetoric”, *M/C Journal*, 26(5). doi: 10.5204/mcj.3004.

Lu, J. (2022). *Gradient Descent, Stochastic Optimization, and Other Tales*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2205.00832>.

Maddigan, P. és Susnjak, T. (2023). Chat2VIS: Generating Data Visualisations via Natural Language using ChatGPT, Codex and GPT-3 Large Language Models. *arXiv:2302.02094 [cs]*. [online] Letölthető: <https://arxiv.org/abs/2302.02094>.

Mehra, A. (2023). *Evolution of Neural Networks to Large Language Models*. [online] Labellerr. Letölthető: https://www.labellerr.com/blog/evolution-of-neural-networks-to-large-language-models/?fbclid=IwAR1T65MCX7WRMWicHujLgy_F1qduM7H95Bacmf5UeBMIM0k8FrouRj7Y_fs [Utolsó letöltés: 2023. október 26.].

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K. és Galstyan, A. (2019). A Survey on Bias and Fairness in Machine Learning. *arXiv:1908.09635 [cs]*. [online] Letölthető: <https://arxiv.org/abs/1908.09635>.

Nangia, N., Vania, C., Bhalerao, R. és Bowman, S.R. (2020). CrowS-Pairs: A challenge dataset for measuring social biases in masked language models. [online] Association for Computational Linguistics, pp.1953–1967. doi:<https://doi.org/10.18653/v1/2020.emnlp-main.154>.

- Nath, S., Marie, A., Ellershaw, S., Korot, E. és Keane, P.A. (2022). New meaning for NLP: the trials and tribulations of natural language processing with GPT-3 in ophthalmology. *British Journal of Ophthalmology*, [online] 106(7), pp.889–892. doi:<https://doi.org/10.1136/bjophthalmol-2022-321141>.
- Pang, B., Nijkamp, E., Kryściński, W., Savarese, S., Zhou, Y. és Xiong, C. (2022). *Long Document Summarization with Top-down and Bottom-up Inference*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2203.07586>.
- Pascanu, R., Mikolov, T. és Bengio, Y. (2012). On the difficulty of training recurrent neural networks. doi:<https://doi.org/10.48550/arxiv.1211.5063>.
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. és Zettlemoyer, L. (2018). Deep contextualized word representations. [online] Association for Computational Linguistics, pp.2227–2237. doi:<https://doi.org/10.18653/v1/N18-1202>.
- Phi, M. (2019). *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. [online] Medium. Letölthető: <https://towardsdatascience.com/illustrated-guide-to-lstms-és-gru-s-a-step-by-step-explanation-44e9eb85bf21>.
- Puvis de Chavannes, L.H., Kongsbak, M.G.K., Rantza, T. és Derczynski, L. (2021). *Hyperparameter Power Impact in Transformer Language Model Training*. [online] ACLWeb. doi:<https://doi.org/10.18653/v1/2021.sustainlp-1.12>.
- Qiao, Y., Xiong, C., Liu, Z. és Liu, Z. (2019). Understanding the Behaviors of BERT in Ranking. doi:<https://doi.org/10.48550/arxiv.1904.07531>.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. és Sutskever, I. (2019). *Language models are unsupervised multitask learners*.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. és Liu, P.J. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. [online] arXiv.org. Letölthető: <https://arxiv.org/abs/1910.10683>.
- Raj, R. (2022). *Numpy explained- Language of Numpy*. [online] Thrive in AI. Letölthető: <https://medium.com/thrive-in-ai/numpy-explained-language-of-numpy-972cc7c99042> [Utolsó letöltés: 2023. október 12.].

- Rothe, S., Narayan, S. és Severyn, A. (2020). Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, [online] 8, pp.264–280. doi:https://doi.org/10.1162/tacl_a_00313.
- Santy, S., Rani, A. és Choudhury, M. (2021). Use of formal ethical reviews in NLP literature: Historical trends and current practices. [online] Association for Computational Linguistics, pp.4704–4710. doi:<https://doi.org/10.18653/v1/2021.findings-acl.414>.
- Schweidtmann, A.M., Bongartz, D., Grothe, D., Kerkenhoff, T., Lin, X., Najman, J. és Mitsos, A. (2021). Deterministic global optimization with Gaussian processes embedded. *Mathematical Programming Computation*, 13(3), pp.553–581. doi:<https://doi.org/10.1007/s12532-021-00204-y>.
- Sezgin, E., Sirrianni, J. és Linwood, S.L. (2022). Operationalizing and Implementing Pretrained, Large Artificial Intelligence Linguistic Models in the US Health Care System: Outlook of Generative Pretrained Transformer 3 (GPT-3) as a Service Model. *JMIR Medical Informatics*, 10(2), p.e32875. doi:<https://doi.org/10.2196/32875>.
- Sheikh, S. (2019). *ARTIFICIAL INTELLIGENCE BASED CHATBOT FOR HUMAN RESOURCE USING DEEP LEARNING A DISSERTATION*. doi:<https://doi.org/10.13140/RG.2.2.25465.52322>.
- Sheng, E., Chang, K.-W., Natarajan, P. és Peng, N. (2021). *Societal Biases in Language Generation: Progress and Challenges*. [online] arXiv.org. Letölthető: <https://arxiv.org/abs/2105.04054>.
- Srinath, M., Wilson, S. és Lee, G.C. (2021). Privacy at scale: Introducing the PrivaSeer corpus of web privacy policies. [online] Association for Computational Linguistics, pp.6829–6839. doi:<https://doi.org/10.18653/v1/2021.acl-long.532>.
- Stephen James Krol, Maria Teresa Llano és McCormack, J. (2022). Towards the Generation of Musical Explanations with GPT-3. *Lecture Notes in Computer Science*, pp.131–147. doi:https://doi.org/10.1007/978-3-031-03789-4_9.
- Vartinen, S., Hamalainen, P. és Guckelsberger, C. (2022). Generating Role-Playing Game Quests With GPT Language Models. *IEEE Transactions on Games*, pp.1–12. doi:<https://doi.org/10.1109/tg.2022.3228480>.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. és Polosukhin, I. (2017). *Attention Is All You Need*. [online] arXiv.org. Letölthető: <https://arxiv.org/abs/1706.03762>.
- Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L. és Anandkumar, A. (2023). *Voyager: An Open-Ended Embodied Agent with Large Language Models*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2305.16291>.
- Wang, H., Wu, Z., Liu, Z., Cai, H., Zhu, L., Gan, C. és Han, S. (2020). *HAT: Hardware-Aware Transformers for Efficient Natural Language Processing*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2005.14187>.
- Wang, S., Liu, Y., Xu, Y., Zhu, C. és Zeng, M. (2021). Want To Reduce Labeling Cost? GPT-3 Can Help. *arXiv:2108.13487 [cs]*. [online] Letölthető: <https://arxiv.org/abs/2108.13487>.
- Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., Kenton, Z., Brown, S., Hawkins, W., Stepleton, T., Biles, C., Birhane, A., Haas, J., Rimell, L., Hendricks, L.A. és Isaac, W. (2021). Ethical and social risks of harm from Language Models. *arXiv:2112.04359 [cs]*. [online] Letölthető: <https://arxiv.org/abs/2112.04359>.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S. és Drame, M. (2020). Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. doi:<https://doi.org/10.18653/v1/2020.emnlp-demos.6>.
- Wu, T., Terry, M. és Cai, C.J. (2022). *AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2110.01691>.
- Zhang, X., Gu, S.S., Matsuo, Y. és Iwasawa, Y. (2022). *Domain Prompt Learning for Efficiently Adapting CLIP to Unseen Domains*. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2111.12853>.
- Zhou, Y., Song, C., Li, J., Wu, Z., Bian, Y., Su, D. és Meng, H. (2022). *Enhancing Word-*

Level Semantic Representation via Dependency Structure for Expressive Text-to-Speech Synthesis. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2104.06835>.

MELLÉKLETEK

I. melléklet

```
1. import numpy as np
2. #numpyrandom seed
3. # Minta szöveg
4. text_data = ""A mai zh angolul volt.
5. Angolul nem tudok.
6. Angol nem jó. PALI
7. ""
8. text_data_first=text_data[0]
9.
10. # Szöveg előfeld1.
11. olgozás
12. chars = list(set(text_data))
13. char_to_idx = {ch: i for i, ch in enumerate(chars)}
14. idx_to_char = {i: ch for i, ch in enumerate(chars)}
15. vocab_size = len(chars)
16.
17. # Hiperparaméterek
18. hidden_size = 100
19. sequence_length = 30
20. learning_rate = 0.01
21.
22. # Modell paraméterek inicializálása
23. Wxh = np.random.randn(hidden_size, vocab_size) * 0.01
24. Whh = np.random.randn(hidden_size, hidden_size) * 0.01
25. Why = np.random.randn(vocab_size, hidden_size) * 0.01
26. bh = np.zeros((hidden_size, 1))
27. by = np.zeros((vocab_size, 1))
28.
29. # Képzés
30. def train():
31.     h_prev = np.zeros((hidden_size, 1))
32.     inputs = [char_to_idx[ch] for ch in text_data]
33.     targets = inputs[1:] + [char_to_idx['\n']] # Következő karakter jóslása
34.
35.     loss = 0
36.     dWxh, dWhh, dWhy = np.zeros_like(Wxh), np.zeros_like(Whh), np.zeros_like(Why)
37.     dbh, dby = np.zeros_like(bh), np.zeros_like(by)
38.
39.     dhraw = np.zeros_like(h_prev) # dhraw inicializálása
40.
41.     for t in range(len(inputs)):
42.         x = np.zeros((vocab_size, 1))
43.         x[inputs[t]] = 1
44.         h = np.tanh(np.dot(Wxh, x) + np.dot(Whh, h_prev) + bh)
45.         y = np.dot(Why, h) + by
46.         p = np.exp(y) / np.sum(np.exp(y))
47.         loss += -np.log(p[targets[t], 0])
48.
49.         dy = np.copy(p)
50.         dy[targets[t]] -= 1
51.         dWhy += np.dot(dy, h.T)
52.         dby += dy
53.         dh = np.dot(Why.T, dy) + np.dot(Whh.T, dhraw)
54.         dhraw = (1 - h * h) * dh
55.         dbh += dhraw
56.         dWxh += np.dot(dhraw, x.T)
57.         dWhh += np.dot(dhraw, h_prev.T)
58.         h_prev = h
59.
60.     for dparam in [dWxh, dWhh, dWhy, dbh, dby]:
61.         np.clip(dparam, -2, 2, out=dparam)
62.
63.     return loss, dWxh, dWhh, dWhy, dbh, dby
64.
65. # Képzési loop
```

```

66. for iteration in range(1000):
67.     loss, dWxh, dWhh, dWhy, dbh, dby = train()
68.     for param, dparam in zip([Wxh, Whh, Why, bh, by], [dWxh, dWhh, dWhy, dbh, dby]):
69.         param -= learning_rate * dparam
70.
71.     if iteration % 100 == 0:
72.         print(f"Iteration {iteration}, Loss: {loss}")
73.
74. # Szövegenerálás
75. def generate(seed, length=100, temperature=0.1):
76.     h = np.zeros((hidden_size, 1))
77.     x = np.zeros((vocab_size, 1))
78.     x[char_to_idx[seed]] = 1
79.     generated_text = seed
80.
81.     for _ in range(length):
82.         h = np.tanh(np.dot(Wxh, x) + np.dot(Whh, h) + bh)
83.         y = np.dot(Why, h) + by
84.         p = np.exp(y) / np.sum(np.exp(y))
85.         next_char_idx = np.random.choice(range(vocab_size), p=p.ravel())
86.         next_char = idx_to_char[next_char_idx]
87.         generated_text += next_char
88.         x = np.zeros((vocab_size, 1))
89.         x[next_char_idx] = 1
90.
91.     return generated_text
92.
93. # Szövegenerálás
94. generated_text = generate(text_data_first, length=500)
95. print(generated_text)
96.

```