JUHÁSZ LEVENTE ZSOLT *

AI-innovációk – Nagy Nyelvi Modellek

Összefoglalás: A tanulmány a mesterségesintelligencia-kutatás legújabb trendjeit, a Nagy Nyelvi Modelleket tekinti át. A modelleket elhelyezi a gépi tanulás és a Deep Learning tágabb kontextusába. Röviden érinti a Transformer-alapú modelleket a technikai fejlődésben megelőző neurálisháló-architektúrákat. Végül áttekintést adunk a Transformer nyelvi modellről, és az ez alapján fejlesztett speciális generatív MI-hálózatokról, a Nagy Nyelvi Modellekről, amelyek az olyan emberszerűen kommunikáló dialógusrendszereknek az alapját is képezik, mint a ChatGPT.
Kulcsszavak: Mesterséges intelligencia, neurális hálózat, nagy nyelvi modellek.

Abstract: This paper reviews the latest trends in Artificial Intelligence research, focusing on the Large Language Models. It places these models in the broader context of machine learning and Deep Learning. It briefly touches on Transformer-based models, neural network architectures that preceded current technical advancements. Finally, we get an overview of the Transformer language model and those specialized generative AI networks developed based on it, known as Large Language Models. These models form the basis for dialogue systems that communicate in a human-like manner, such as ChatGPT.
Keywords: Artificial Intelligence, Neural Networks, Large Language Models.

2022 októberében jelentette be az OpenAI [1] mesterségesintelligencia-fejlesztéssel foglalkozó vállalat, hogy egy új chatbot-rendszert fejlesztett ki. Ez a ChatGPT, ami egy olyan GPT-architektúrájú nyelvi modellt foglal magában, amelyet alkalmassá tettek további tanítással a felhasználókkal folytatott kommunikációra. Az alkalmazás hatalmas sikert aratott, a megjelenést követő hónapokon belül a történelem legtöbb felhasználóval rendelkező alkalmazása lett. A mesterséges rendszert az interneten hozzáférhető hatalmas nyelvi

* *Dunaújvárosi Egyetem, Tanárképző Központ*
Email: juhaszle@uniduna.hu

[1] OpenAI. (2023): *GPT-4 Technical Report. arXiv:2303.08774 [cs.CL].* [Letöltés dátuma: 2023. szeptember 17.] Forrás: https://arxiv.org/abs/2303.08774

anyagon tanították, ami alapján meglepően helyes és „emberszerű” válaszokat generált a felhasználók kérdéseire és kereséseire. Kérésre történeteket, verseket, óravázlatokat, tartalmi összefoglalókat, hosszabb-rövidebb programkódokat generált, vagy akár életvezetési tanácsokat is adott. Mind a kutatói, fejlesztői és felhasználói közösséget elvarázsolták az új dialógusrendszer képességei. Jónéhányan már az emberi intellektuális képességek meghaladásáról beszéltek, és a mesterséges intelligencia szabályozására hívtak fel. A dolgozat a ChatGPT-t és társait lehetővé tevő technikai rendszereket tekinti át, hogy jobban megérthessük ezek működését, hogy kevésbé egy veszélyes fekete dobozként tekintsünk rá, hanem inkább egy sok lehetőséget magában rejtő technikai rendszerként.

Gépi tanulás

Az új évezred informatikai fejlődésének egyik legfontosabb ösztönzője az, hogy a mindent behálózó számítástechnikai eszközök (internet, IoT, mobilalkalmazások) egyre nagyobb mennyiségű információ rögzítését teszik lehetővé. A korábban nem jellemző, megnövekedett volumenű és sebességű adattömeg feldolgozásának igénye új adatfeldolgozási technológiák fejlesztésére és alkalmazására késztette a kutatás-fejlesztést és az üzleti szereplőket is. Ezeknek az új technológiáknak az összefoglaló neve a Big Data. A nagy adattömeg hozzáférhetősége a mesterséges intelligencia (MI) kutatásoknak és fejlesztéseknek is új lendületet adott. A kutatások egyik legkurrensebb részterülete a gépi tanulás (machine learning), amely a statisztikai tanulás módszereit alkalmazva, olyan szoftvereket épít, amelyek képesek adatokból tanulva, bizonyos előre be nem programozott teljesítményeket elsajátítani. A legtöbb ma is használatos tanulóalgoritmus valamilyen változatban már a ’80-as évektől ismert volt a téma kutatói számára, de ahhoz, hogy sok esetben az emberi teljesítményt is megközelítő, vagy esetleg meghaladó teljesítményt érjenek el, nagyon sok adatra van szükség a tanításukhoz. Ahhoz, hogy a nagy adathalmazból kivárható idő alatt az optimalizációs tanulás használható minőségű modelleket alakítson ki, a hardver- és szoftvertechnológiai eszközök fejlődése, új algoritmusok kifejlesztése is nagyban hozzájárult.

A gépi tanulásnak 3 nagy csoportja van, ezek a *felügyelt tanulás,* a *tanulás felügyelet* nélkül és *megerősítéses* tanulás. [2] A felügyelt tanulás esetén – ez általában

[2] Géron, A. (2022): *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* 3rd edition. Sebastopol: O’Reilly Media.

osztályozási vagy regressziós feladat – az algoritmus úgy tanul, hogy a tréningfázis alatt „címkézett adatokat" kap, és a végeredmény az lesz, hogy a program képes lesz erre a címkére, remélhetőleg elég pontos becsléseket tenni olyan adatok esetén is, amelyekkel korábban nem találkozott. A felügyelt tanulásra példa az arcfelismerés, a lineáris regresszió vagy a szentiment analízis. A felügyelet nélküli tanulás esetén az adatok nem címkézettek. A program az adatelemek közötti összefüggéseket, rejtett struktúrát fedezi fel és tanulja meg. Ebbe a típusba tartoznak a klaszterezési (csoportosítási) eljárások, anomália- és újdonság-detektálás, vizualizáció és dimenziócsökkentés, asszociációs szabálytanulás. A megerősítéses tanulás a Markov Döntési Folyamattal modellezhető: egy ágens egy állapottér felett tanulja meg a környezeti megerősítések hatására, hogy az egyes állapotokban mi az optimális választás (policy) egy összesített hasznossági függvény szempontjából. A megerősítéses tanuláson alapuló modellek kiemelt alkalmazásai az önvezető rendszerek irányítása.

A bonyolult stratégiai játékokban, mint például a go, ezen modellekkel megvalósított rendszerek teljesítménye már a legkiválóbb emberi (világbajnoki) eredményeket is meghaladja. [3]

Deep Learning

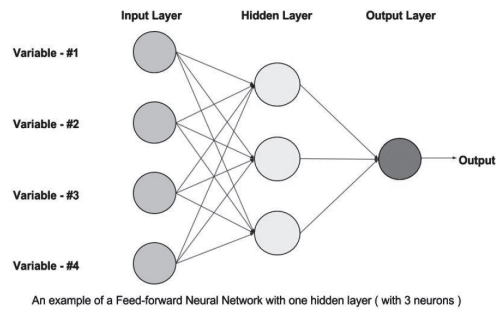
A gépi tanuláson belül is a legimponálóbb tanulási teljesítményeket a mesterséges neuronháló modellek adják. [4] Ezek minden említett gépi tanulási feladat megoldásának részei lehetnek. Mint a modell neve is utal rá, absztrakt feldolgozó egységek (ún. mesterséges neuronok) összeköttetései alkotják. A tanulás itt is optimalizációt jelent, amely során a kapcsolatok „súlyozása" változik. Az ún. feed-forward hálózatokban az információ egy irányban terjed, legegyszerűbb típusa a perceptron, amely egy rejtett (köztes) neuron réteget tartalmaz a bemeneti és a kimeneti réteg között *(1. ábra)*. A több rétegű neurális hálózatok márkaneve „deep learning" modellek. A mélyebb hálózatok általában jobb tanulási teljesítménnyel rendelkeznek. A mesterséges neuronhálózatok szoftveres megvalósítása tulajdonképpen mátrixműveletek segítségével történik. Mind a rendszerek bemenete, mind a rendszert alkotó neuronok kapcsolatai mátrix formában reprezentálódnak. A rendszer emlékezetét a neuronok közötti kapcsolatokat tartalmazó mátrixok értékei (a súlyok) jelentik. A tanulás/tanítás folyamán az elvárt eredményektől való eltérés alapján a back-propagation (hiba visszaterjesztés) algoritmus segítségével a kapcsolati súlyok olyan

[3] Russell, S.–Norvig, P. (2021): *Artificial Intelligence: A Modern Approach 4th Edition.* Hoboken: Pearson.

[4] Goodfellow, I.–Bengio, Y.–Courville, A. (2016): Deep Learning. *Adaptive Computation and Machine Learning series*. Cambridge: MIT Press.

változtatására törekedik a rendszer, hogy a továbbiakban az eltérés (a veszteség) minél kisebb legyen. Az algoritmus alapját képező iteratív numerikus matematikai eljárás a grádiens-csökkentés (gradient descent).

1. ábra. Feed-forward hálózat



Forrás: learnopencv.com, 2023.

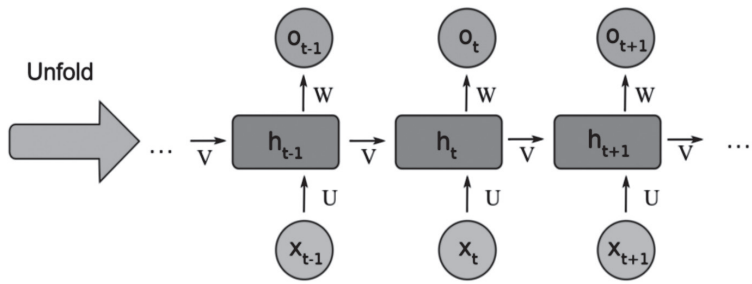
Az állati „korai látás" vonáskiemelő filtereit modellezik, illetve ezeket tanulják meg az úgynevezett konvolúciós neuronhálók (CNN). Ezeket a gépi látást megvalósító rendszerekben alkalmazzák.

Szekvenciális hálózatok

Szekvenciális adattípusok és idősor jellegű információk feldolgozására az ún. rekurrens neurális hálózatok alkalmasak. [5] A hálózatok ekkor ütemezve kapják a bemenetet. Például egy mondat esetén szavanként. Minden további item esetén a megelőző állapotokról kialakított reprezentációk is visszatáplálódnak a rendszerbe (rekurzió). A hálózat így belső állapottal (memóriával) rendelkezik, és képes az input anyagban esetleg egymástól időben távol eső dependenciákat is feldolgozni. A szekvenciális hálózatok így képesek például egy egész mondat reprezentációját kialakítani *(2. ábra).*

[5] Zhang, A.–Lipton, Z. C.–Li, M.–Smola, A. J. (2023): *Dive into Deep Learning.* Cambridge: Cambridge University Press. Forrás: https://d2l.ai/

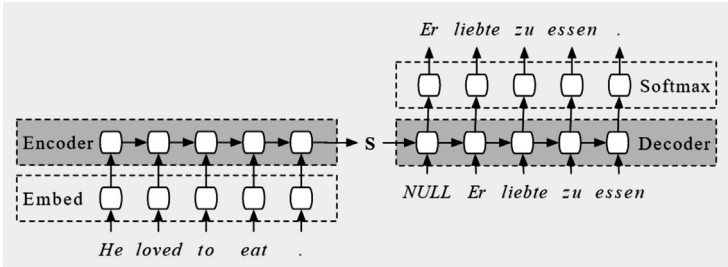
2. ábra. Rekurrens hálózat



Forrás: www.analyticsvidhya.com, 2023.

A rekurrens hálók leggyakrabban alkalmazott típusai a GRU (Gated Recurrent Unit) és a LSTM (Long Short-Term Memory), amelyekben a modellek maguk is több kapuzófunkciót ellátó neuronhálózatból épülnek fel. A nyelvi egységeknek, tokeneknek az ilyen rendszerekben kialakított vektor-reprezentációját nevezik beágyazásoknak (embeddings). Ezek a tokenek szavak vagy mondatok szintaktikai és szemantikai tulajdonságait tárolják. Bizonyos feladattípusok, amelyek szekvenciális kimenettel rendelkeznek, megoldására a sorrendi hálózatok speciális modelljei a legalkalmasabbak. Ezeket decoder-encoder (seq2seq) hálózatoknak nevezik, és tulajdonképpen két külön tréningelt modellből állnak. Az egyik a kódoló (encoder) rész, amely az input nyelvi anyag alapján a tokenek gépi reprezentációját (embeddings) alakítja ki, készít egy belső modellt a nyelvről. A másik dekódoló (decoder) rész ezeket a beágyazásvektorokat felhasználva képes az input sorozatokat kimeneti sorozatokra leképezni. Az ilyen architektúrák tipikus felhasználási területe a gépi fordítás, azaz mondatonkénti, vagy szövegrészenkénti automatikus nyelvi fordítás *(3. ábra)*.

3. ábra. Seq2seq hálózat



Forrás: Bahdanau–Cho–Bengio, 2014.

Natural Language Processing (NLP)

A gépi fordítás a nyelvtechnológia egyik legsikeresebb területe. [6] A mesterséges intelligencia emberi nyelv feldolgozásával és produkciójával foglalkozó területe a természetes nyelvi feldolgozás (natural language processing, NLP), avagy nyelvtechnológia. Számos más egyéb összefüggő részfeladat van az NLP-n belül: keresés szövegekben, szövegek kivonatolása, információ-visszakeresés, szentiment-analízis, szintaktikai-szemantikai elemzés, entitás-azonosítás, videó-/kép feliratozás, beszédfelismerés és beszédgenerálás. A dialógusrendszerek fejlesztése (konverzációs mesterséges intelligencia) szintetizáló feladat az NLP-n belül, hiszen a gépi nyelvmegértés és nyelvgenerálás modelljeit egyaránt felhasználja. Az NLP problémák megoldásában a mélyhálóalapú rendszerek általában rendkívül jól, sokszor az emberi eredményeket megközelítve, vagy azokat túlteljesítve működnek. [7]

[6] Jurafsky, D.–Martin, J. H. (2023): *Speech and Language Processing.* (3rd ed. draft). Forrás: https://web.stanford.edu/~jurafsky/slp3/

[7] Introduction to Generative AI. Course. (2023): *Google Cloud.* Letöltés dátuma: 2023. szeptember 17. Forrás: https://www.cloudskillsboost.google/course_templates/536

Generatív MI

Működésük, illetve a kimenetük jellege alapján a gépitanulás-modelleknek két csoportját különböztetik meg: generatív és diszkriminatív modellek.

A diszkriminatív modellek általában elemek osztályozását, vagy valamilyen információ előrejelzését tanulják meg. A felügyelt tanulás esetén az adatpontok tulajdonságai és a hozzákapcsolt célértékek vagy címkék kapcsolatát sajátítják el. A különböző klasszifikációs módszerek és a lineáris regresszió módszerei tartoznak ide. Jellemző kimenetük: számérték, diszkrét osztály, vagy valószínűség értékek.

A generatív modellek (GenAI) ezzel szemben olyan új adatokat képesek létrehozni, amelyek valószínűségi eloszlásukban hasonlítanak azokra az adathalmazokra, amelyeken tréningelték őket. A generatív modellek egy statisztikai modellt építenek a beérkező tanító adatokból. Ezen statisztikai modell alapján a különböző lehetséges kimenetek valószínűségét ki tudják számolni, illetve a modell alapján képesek új kimenetek generálására. A nagyobb valószínűségű mintázatok gyakrabban jelennek meg a létrehozott kimenetben. Tipikus kimenetek: szöveg, beszéd, zene, kép, videó és programkód. A generálás általában valamilyen input hatására indul el. A kiváltó bemeneti adatot a szaknyelvben promptnak (súgásnak) nevezik. A bemenetek (prompts) és kimenetek kombinációi alapján négyféle GenAI alaptípusról beszélnek:

– *Text-to-text.* A modell bemenete valamilyen szöveg, amely szövegtípusú kimenetet generál. A tréning folyamán általában szövegpárokat kapnak, és ezek közötti kapcsolatokat kell megtanulni (pl. egyik nyelvről a másik nyelvre történő fordítás). Alkalmazásuk: szöveggenerálás, szövegosztályozás, összefoglalók készítése szövegekből, fordítás, szöveges keresés, kivonatolás, csoportosítás, tartalomjavítás, újraírás.
– *Text-to-image.* A tanuló adatok nagy mennyiségű kép- és feliratozás-párosítást tartalmaznak. A feliratozás általában a képen ábrázoltak tömör leírása. A modellek kimenete a leírás, mint prompt alapján generált kép (pl. Diffusion-modell). Alkalmazásai: képgenerálás és képszerkesztés.
– *Text-to-video, text-to-3D:* Cél a szöveg promptból mozgóképreprezentációt létrehozni. A szövegbemenet akár egy rövid cím, néhány mondatos leírás, vagy egy komplett forgatókönyv is lehet. A text-to-3D modellek leírásból 3 dimenziós vizuális objektumokat hoznak létre. Alkalmazási területek: videógenerálás és szerkesztés, virtuális valóság, videójáték-elemek generálása, animációja.
– *Text-to-task:* Szövegesen megadott instrukciók végrehajtása a feladat. Mint például kérdések megválaszolása, kérésre keresések végrehajtása, előrejelzések kiszámítása és közlése, kódgenerálás leírás alapján. Gyakori alkalmazások: Szoftverágensek, virtuális asszisztensek, automatizáció.

Nyelvi modellek

A szöveg/beszéd alapú generatív modelleket általában nyelvi modelleknek nevezzük. Ezek olyan függvényeknek tekinthetők, amelyek valamilyen szó- vagy tokensorozathoz valószínűségi értéket tudnak hozzárendelni. Azaz, ha egy adott nyelvhez tartozó anyagon tanítunk egy ilyen modellt, akkor az képes lesz kiszámítani, hogy egy adott szósorozat milyen gyakori egy adott nyelvben. Ez a nyelvi valószínűségi tudás sokféle NLP-feladatban felhasználható, többek között szövegek generálásában is. Tisztán statisztikai alapú nyelvi modelleket próbáltak létrehozni legkorábban. Ezekben főleg a különböző hosszúságú szósorozatoknak a tanulómintában való előfordulások aránya alapján alakították ki a valószínűségi eloszlás modelleket (n-gramm modellek). Mivel egy adott szósorozat nem feltétlenül jelent meg a tanulóhalmazban, ezért a hiányzó kombinációk valószínűségének becslésére korrekciós eljárásokat kellett alkalmazni. Ezekkel a gyakoriságon alapuló módszerekkel a generált szövegek minősége csak korlátozott volt. [6] A gyakoriságon alapuló modelleket a neuronhálókra épülő modellek váltották. Az új évezredben, a Deep Learning-technológia folyamatos fejlődésével, egyre inkább ezek a modellek váltak jellemzővé. Kezdetben elsősorban a már korábban említett rekurrens hálóarchitektúrán alapuló modellekkel dolgoztak, amelyek a tisztán statisztikai modelleknél jobban teljesítettek, de a különböző, a modellek teljesítményét mérő tesztekben még messze az emberi értékeket produkáltak. A nyelvi modellek technológiában az ún. Transformer-architektúra hozta az áttörést 2017-ben. [8]

Figyelmi mechanizmus

A korábban ismertetett encoder–decoder-modellben az a kódoló (encoder) hálózat az egész bemenetről (mondatról) készített el egy közös vektorreprezentációt (rejtett állapot), amelyet aztán a dekódoló hálózatnak továbbítva lehetségessé vált egy új szöveg (pl. a szöveg idegennyelvi megfelelője) generálása. A dekódoló részben az új szöveg generálása autoregresszív módon történik, azaz az egyes fázisokban a hálózat a rejtett állapotot reprezentáló beágyazás-vektor mellett megkapja a maga által generált tokent is benementként. Az újabb elem generálása, ezen két bemenet információinak felhasználásával történik meg, egészen a mondatvége-token generálásáig.

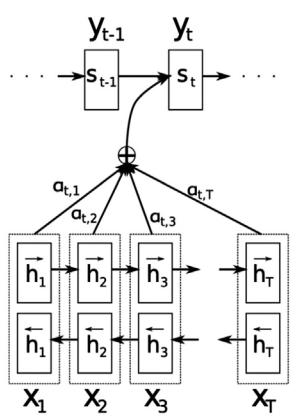
A közösállapot-vektor azonban a legszűkebb keresztmetszeti pontként viselkedik

[6] Jurafsky, D.–Martin, J. H. (2023): *Speech and Language Processing.* (3rd ed. draft). Forrás: https://web.stanford.edu/~jurafsky/slp3/

[8] Waswani, A.–Shazeer, N.–Parmar, N.–Uszkoreit, J.–Jones, L.–Gomez, A. N.–Polosukhin, I. (2017): *Attention Is All You Need.* (arXiv:1706.03762. kötet). Letöltés dátuma: 2023. szeptember 17. Forrás: https://arxiv.org/abs/1706.03762

ezekben a rendszerekben. Mivel a hossza véges, ezért információtárolási kapacitása is korlátozott. Egy mondat feldolgozásának folyamatában ez azt jelenti, hogy a mondat elején lévő tokenekkel kapcsolatos információk kevésbé maradtak hozzáférhetőek a dekódolás generatív szakaszában. Így a bemeneti és kimeneti mondat szavai közötti hosszabb dependenciákat kevésbé tudta felhasználni a rendszer. A seq2seq-modelleknek ezeken a korlátain igyekezett javítani az ún. figyelmi (attention) mechanizmus *(4. ábra)*. Ebben egyrészt a dekódoló rész nemcsak a végső rejtett állapotvektort kapja meg, hanem a kódolás (első rész) minden lépéséhez tartozó állapotvektorokat is. Így a dekódoláshoz sokkal több kontextuális információ áll rendelkezésre. A figyelmi mechanizmus azt teszi lehetővé, hogy a dekóder az egyes elemek generálásakor szabályozni tudja azt, hogy az egyes állapotvektorokat milyen súlyozással vegye figyelembe (azaz, mennyire „figyeljen" rájuk). A rendszer ezeket az értékeket is a tréning során tanulja meg. A figyelem mechanizmusa képes megragadni a fordítandó és a lefordított mondatok egyes elemei közötti interdependenciákat, és ezeket a szöveg generálásban is felhasználni, amely által a fordítások minősége jelentősen javult. [9]

4. ábra. Figyelmi mechanizmus



Forrás: Bahdanau–Cho–Bengio, 2014.

[9] Bahdanau, D.–Cho, K.–Bengio, Y. (2014): *Neural Machine Translation by Jointly Learning to Align and Translate.* (arXiv:1409.0473. kötet). Letöltés dátuma: 2023. szeptember 17. Forrás: https://arxiv.org/abs/1409.0473

Transformer-modellek

A Transformer-modell szintén a figyelmi mechanizmust használja. Azonban nem az input szövegek és a célszövegek közötti kapcsolatok megragadására fókuszál, hanem az ugyanazon bemeneti szövegek részei közötti szemantikai és szintaktikai összefüggések kiemelésére törekszik (self-attention). Olyan, az egyes tokenekhez tartozó állapot-reprezentációk létrehozása a célja, amelyek függnek attól, hogy az adott elem milyen szövegkörnyezetben helyezkedik el. Ez azt jelenti, hogy az ugyanazon szó esetén kapott beágyazott vektorok, az azok szövegbeli előfordulásától függően különbözőek lesznek. [10]

A Transformer-rendszerek, az ún. self-attention

A 2017-ben publikált, és áttörést jelentő Transformer-modell [8] szintén seq2seq-architektúrájú volt, azaz egy-egy encoder és decoder modulból épült fel *(5. ábra)*. Mindkét modul több feldolgozó rétegből állt. Ezek közül a legnagyobb technológiai újítást az úgynevezett „több fejes figyelmi réteg"-ek (multi-head attention) jelentették. A „több fejes" kifejezés arra utal, hogy a feldolgozások egymással párhuzamosan, lényegileg egymástól függetlenül mentek végbe, amelyek eredményét a rendszer egy későbbi fázisban egyesíti. Az eredeti Transzformer modell 8 fejjel dolgozott. Az újabb Transformer-alapú modellekben a párhuzamos szálak száma már ennek többszöröse. A párhuzamos működések lehetővé teszik, hogy a betáplált szövegek különböző aspektusaira koncentráljon a rendszer, és azt is megkönnyítik, hogy a modell rendkívül könnyen futtatható párhuzamos hardveregységeken (elsősorban a grafikus feldolgozásra szánt GPU-kon működjön), ezáltal lerövidítve a tanítási periódust. A figyelmi alrétegben a feldolgozandó tokeneknek többszörös reprezentációjával (vektorok) dolgozik a rendszer. A *query vektor* a feldolgozás alatt lévő token reprezentációja. A *key vektorok* a kontextus elemeinek reprezentációi, amelyekkel összehasonlítjuk a query vektort. A *value vektorok* pedig a tokenek információtartalmát képviselik. A query-, key- és value vektorok előállításához W^Q, W^K, és W^V súlymátrixokat használunk, a rendszer tanítása során ezeknek az értékei változnak. Az input szöveg elemeihez tartozó query-, key- és value-vektorok együttesen alkotják a Q-, K- és V-mátrixokat.

[8] Waswani, A.–Shazeer, N.–Parmar, N.–Uszkoreit, J.–Jones, L.–Gomez, A. N.–Polosukhin, I. (2017): *Attention Is All You Need.* (arXiv:1706.03762. kötet). Letöltés dátuma: 2023. szeptember 17. Forrás: https://arxiv.org/abs/1706.03762

[10] Yildirim, S.–Asgari-Chenaghlu, M. (2021): *Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques.* Birmingham: Packt Publishing.

Vektorok hasonlóságát skaláris szorzatuk segítségével fejezhetjük ki. Ennek alapján a tokenek közötti relációkat az úgynevezett Skálázott Skaláris Szorzat Figyelem (Scaled Dot-product Attention) segítségével számolja ki a rendszer:

$$Attention(Q,\ K,\ V) = \ Soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

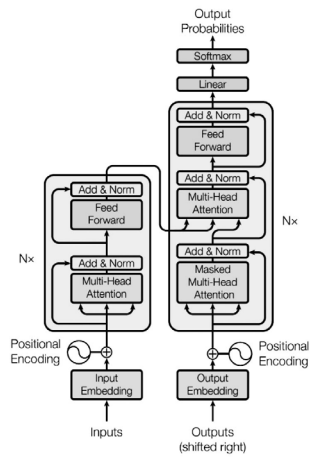
A képletben a dk a skálázási faktor, a beágyazott vektorok dimenziószáma. A skálázásra (az osztásra) numerikus számítások stabilitásának fenntartása érdekében van szükség. A Softmax-függvény a szorzás során kapott értékeket valószínűségértékekké alakítja át. Az Attention-mátrix, figyelmi súlyokat tartalmaz, azaz minden query-token esetén, milyen súllyal kell figyelembe venni a kontextus elemeit (value vektorait). A V vektoraival megszorozva pedig megkapjuk a query-tokenek kontextuális reprezentációit is. Ezek a kontextuális beágyazott vektorok, illetve a beágyazott vektortérbe való transzformáltjai képezik a figyelmi modul kimenetét. A figyelmi modulhoz „hagyományos" feed-forward (többrétegű perceptron) hálózat is kapcsolódik, amely a figyelmi reprezentációk további feldolgozását végzi. A kódolómodul még tartalmaz normalizációs és reziduális hálózatra jellemző működéseket is. A bemeneti elemeket a sorozatban elfoglalt helyükkel kapcsolatos információval is ki kell egészíteni. Az encodermodul kimenete lehet egy következő hasonló kódolómodul számára bemenet, vagy seq2seq-architektúra esetén generatív feladatot ellátó decodermodul bemenete lehet.

A dekódermodul építőelemei megegyeznek a dekódermodulnál bemutatottakkal. Lényeges különbség, hogy a többfejes figyelmi rétegek esetén az aktuális input-tokent követő elemeket maszkolni kell, a rendszer csak a már beérkezett elemeket használhatja. Az első figyelmi réteg a dekódoló aktuális bemenetét dolgozza fel („maszkolt többfejes figyelem"), és ez kerül a második figyelmi rétegbe további feldolgozásra. Itt ez már kiegészül a dekóder szóreprezentációival, így a dekóder képes a decoder-input szövegek és a célszöveg (pl. fordítás esetén) közötti interdependencákat is megtanulni. Hogy a rendszer bemenetét minél alaposabban feldolgozhassa, ezért vertikálisan több transformer (encoder-decoder) blokkot is tartalmaz. Ezt azt jelenti, hogy a transformer, illetve az azt alkotó kódoló és dekódoló modulok kimenete egy újabb transformer modul bemenete lesz, ami tovább dolgozik az előző modultól kapott reprezentációkon. Az eredeti Transformer-model 6 emeletes, vagyis összesen a kódoló és dekódoló részben 12 transformer réteg volt. Az újabb modellek ennek számát is megtöbbszörözték. [11]

[11] Tunstall, L.–Werra, L.–Wolf, T. (2022): *Natural Language Processing with Transformers, Revised Edition 1st Edition.* Sebastopol: O'Reilly Media.

A transformer rétegek tulajdonképpen fokozatosan gazdagítják a szavak reprezentációját. A korai rétegek az egyszerűbb, elemi relációkat ragadják meg: rövid távú dependenciák, szomszédos kapcsolatok, szórend, lokális szintaktikai jellemzők (szófaj), egyszerű mondatszerkezet. A magasabban lévő transformer rétegek tanulják meg a szemantikai információkat, a frázisok közötti kapcsolatokat, tematikus szerepeket. A feldolgozás legmagasabb rétegei pedig az absztrakt kapcsolatok, diszkurzus-struktúra, egészleges emocionális jelentés (szentiment), és nagy távolságot áthidaló kölcsönös függőségi kapcsolatok megragadói.

5. ábra. Transformer



Forrás: Waswani et al., 2017.

Nagy nyelvi modellek

Láttuk a standard Transformer-modell két fő részből áll: decoder és encoder. Azonban a feladattól függően, amire rendszerünket szánjuk, a két összetevő egyike is elégséges lehet. A nagy nyelvi modelleknek architektúrájukat tekintve 3 csoportja van: csak encoder, csak decoder és encoder–decoder-modellek. Ezeket a típusokat egy-egy jellegzetes képviselőjükön keresztül mutatjuk be. [12]

[12] Rothman, D.–Gulli, A. (2022): *Transformers for Natural Language Processing: Build, train, and fine-tune deep neural network architectures for NLP with Python, Hugging Face, and OpenAI's GPT-3, ChatGPT, and GPT-4. 2nd ed.* Birmingham: Packt Publishing.

Csak-encoder modellek. Kétirányú figyemmel dolgoznak: a szavak reprezentációjának kialakításához a mondatban az adott pozíción túli elemeket is felhasználják. Autó-encoding rendszerek: a tanításukhoz nem kell előre címkézett tréninghalmaz. Bemenetük maszkolt: a mondatokból bizonyos szavakat kihagynak, ezeket kell a rendszernek megtanulni a kontextus alapján bejósolni. Lényeg a mondatok megértése, és a megfelelő nyelvi modell kialakítása. Elsődleges alkalmazhatóságuk a mondatok klasszifikációja (pl. Szentiment-analízis), entitások felismerése, információkeresés esetén. Számos nyelvi modellt építettek a csak-encoder architektúra elveinek megfelelően. A legismertebb a BERT (Bidirectional Encoder Representation from Transformer) és ennek különböző variációi. [13]

A BERT tanításánál a már említett Maszkolt Nyelvi Modell módszerét és a Következő Mondat Predikció-feladat módszerét alkalmazták. Ez utóbbiban a rendszer feladata az volt, hogy két mondat esetén eldöntse, az első mondatból következik-e a második. A BERT nagy modellje 12 fejet, és 24 encoder réteget tartalmazott. Tanítható paramétereinek száma 340 millió volt (ez ma már nem számít nagy modellnek).

Csak-decoder modellek. Ezek utóregresszív modellek: tanításuk során egyik bemenetük az előző fázisban saját maguk által generált kimenet (szó) van. A figyelmi réteg csak az aktuális bemenetet megelőző pozíciókat „látja": a tanulás csak egy irányban történik. Tanulás során a feladatok az input-sorozat következő elemére tett predikció. Leggyakoribb felhasználási területük: nyelvi tartalmak generálása, kérdés-válasz rendszerek. A GPT (Generative Pre-trained Transformer) család tagjai a csak-encoderek legismertebb képviselői. A GPT-3 96 decoder modult tartalmaz, és a fejek száma 96. A tanítható paraméterek száma 175 milliárd. [14]

Encoder–decoder-modellek: az eredeti Transformer-modellhez hasonló seq2seq-feldolgozás esetén használt architektúra. A decoder figyelmi rétege a bemeneti sorozat egészéhez hozzáfér, az encoderé csak a pozíciót megelőző részekhez. Autó-encoding és autóregresszív rendszerek, a szeparált rendszereknél megismert tanító algoritmusokkal. Alkalmazása olyan feladatok esetén, amikor a bemeneti sorozatokat (szövegeket) kimeneti szövegekké kell alakítani. Pl. Gépi fordítás, tartalmi összefoglalás, generatív válaszgenerálás. Jellegzetes képviselőjük a T5 és a BART. A BART (Bidirectional and Auto-Regressive Transformers) modell a BERT és a GPT-2 modell ötvözete, amely mindkét irányban tanulja meg a szöveg kontextusát, és maszkolt nyelvmodellezést használ az encoderben és az decoderben is.

[13] Devlin, J.–Chang, M.-w.–Lee, K.–Toutanova, K. (2018): *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* (arXiv:1810.04805v2. kötet). Letöltés dátuma: 2023. szeptember 17. Forrás: https://arxiv.org/abs/1810.04805v2

[14] Brown, T. (2020): *Language models are few-shot learners. Advances in neural information processingsystems.* (arXiv: 2005.14165. kötet). Letöltés dátuma: 2023. szeptember 17. Forrás: https://arxiv.org/abs/2005.14165

Jól teljesít többek között absztraktív összefoglalásban, természetes nyelvű generálásban és nyelvi következtetési feladatokban. 400 millió paraméter, 16 fej, 12 transformer-modul az encoderben és 12 transformer-modul a decoderben. A T5 (Text-To-Text Transfer Transformer) modell egy egységes keretrendszert alkalmaz, amely minden nyelvi feladatot szövegből-szövegbe transzformálásként kezel. Mindkét rész maszkolt nyelvmodellezést használ. A T5 modell kiemelkedő eredményeket ért el több mint 20 nyelvi feladatban, például gépi fordításban, összefoglalásban, kérdés-válaszadásban és közös értelem megértésében. A t5-tb alváltozata 24–24 transformer modult tartalmaz, 128 fejjel és 11 milliárd paraméterrel. [15]

Alapmodellek és finomhangolás

Számos betanított transzformer-modell hozzáférhető külső felhasználók számára is. Ezek legnagyobb gyűjteménye a Huggingface oldalán található. [16] Jónéhány esetben ezeket csak a már ismertetett szövegkiegészítéshez hasonló feladatokkal tanították. Ezeket alapmodelleknek (foundational model) nevezik, és ahhoz, hogy speciálisabb nyelvtechnológiai feladatokat oldjanak meg, finomhangolásra (fine tuning) van szükség. [17] A gépi tanulás irodalmában transzfer learningnek nevezik azt, amikor egy korábban betanított modellt használunk fel arra, hogy valamilyen új teljesítményre tanítsuk meg a már létező rendszerünket. [18] A finomhangolás során rendszerint az új feladattól függően, további rétegeket csatolnak a meglévő rendszerhez. A finomhangolás, avagy újratanítás során az új feladatnak megfelelő input adatokat kap a rendszer, ezeken történik a tréning. A folyamat során a tréning érintheti a rendszer meglévő rétegeit is, de ez általában hasonlóan forrásigényes, mint egy új modell tanítása. Gyakrabban előfordul, hogy az alapmodell rétegeinek egy részét, vagy akár egészét „befagyasztják", azaz ezek az utótréning során nem változhatnak. Ekkor csak a nem rögzített, és a kiegészítő rétegeken történik tanulás. Az újratanítás eredményessége általában arányos azzal, hogy hány réteget érint a tréning, de hatékonysági szempontok miatt a teljes hálózatot érintő eljárást ritkán alkalmazzák. A ChatGPT [1] esetén az alapmodell az OpenAI GPT-3.5 decoder modellje volt.

[1] OpenAI. (2023): *GPT-4 Technical Report. arXiv: 2303.08774 [cs.CL].* [Letöltés dátuma: 2023. szeptember 17.] Forrás: https://arxiv.org/abs/2303.08774

[15] Rafael , C.–Shazeer, N.–Roberts, A.–Lee, K.–Narang, S.–Matena, M.–Liu, P. J. (2020): Exploring the Limits of Transfer Learning with a Unified. *Journal of Machine Learning Research,* 21., pp. 1–67.

[16] *HuggingFace* (2023): Forrás: https://huggingface.co/

[17] Zhao, X. W.–Zhou, K. –Li, J.–Tang, T.–Wang, X.–Hou, Y.–Wen, J.-R. (2023): *A Survey of Large Language Models.* arXiv:2303.18223 [cs. CL]. Letöltés dátuma: 2023. szeptember 16. Forrás: https://arxiv.org/abs/2303.18223

[18] Azunre, P. (2021): *Transfer Learning for Natural Language Processing.* New York: Manning.

Finomhnagolása során, többek között képessé tették kérdések megválaszolására, történetek generálására, programkódok kiegészítésére. Az alkalmazott módszer a humánfeedback-alapú megerősítéses tanulás volt (Reinforcement Learning from Human Feedback): a rendszer humán felhasználók válaszaiból és a rendszer válaszaira adott értékelésekből tanult.

Prompting, prompt engineering

A modellek teljesítményét nem csak finomhangolással lehet növelni. In-context learningnek nevezik az eljárást, ha a rendszernek példát vagy példákat mutatunk be, hogy milyen jellegű kérésekre milyen jellegű válaszokat várunk, és azt milyen formában. [19] Ekkor a rendszer kapcsolatai (paraméterei) nem frissülnek, csak kiegészítő kontextust kap a feladathoz, amely alapján jobb minőségű válaszokat tud generálni. Zero-shot learning esetén a rendszer példa-prompt nélkül is minőségi választ ad. One-shot, vagy few-shot learning esetén ehhez egy vagy több példa bemutatására is szükség van a felhasználó részéről. Prompt engineeringnek nevezik azokat az eljárásokat, amelyekkel növelni lehet a rendszertől kapott válaszok minőségét. A ChatGPT esetén például segít, ha megadjuk a keresés célját, kontextusát és egyéb kiegészítő információkat, azt, hogy milyen szerepet (perspektívát) vegyen fel a rendszer (pl. mint tanár, mint laikus stb.) és milyen megszorításokat szeretnék (a válasz hossza, stílusa, nyelve stb.). Számítási és logikai következtetést igénylő feladatoknál sikeresebb a rendszer, ha megkérjük, hogy lépésekben végezze el a műveletet, illetve választásait indokolja is (Chain of Thought reasoning).

Alkalmazások

Már most számos olyan műszaki alkalmazás van, amelyek alapját a Transzformer alapú nyelvi modellek alkotják. [20] Az informatika területén a keresőmotorok szerves része lett, egyrészt nyelvi reprezentáción alapuló keresések beépítésével, másrészt a keresésekre adott válaszok generálásával és a felhasználóval folytatott kommunikáció megvalósításával. A szoftverfejlesztők munkáját a nyelvi modellek kódgeneráló funkciói segíthetik (Például a GitHub Copilot a Codex nevű finomhangolt GPT-3 modellt használja).

[19] DeepLearning. AI. (2023): ChatGPT Prompt Engineering for Developers. Course. Forrás: *DeepLearning.AI.* https://www.deeplearning.ai/short-courses/ChatGPT-prompt-engineering-for-developers/

[20] Kaddour, J.–Harris, J.–Mozes, M.–Bradley, H.–Reileanu, R.–McHardy, R. (2023): Challenges and Applications of Large Language Models. *arXiv:2307.10169 [cs.CL].* Letöltés dátuma: 2023. szeptember 17. Forrás: https://arxiv.org/abs/2307.10169

Fontos felhasználási területek az orvostudományi kutatásokban a molekulák (fehérjék, nukleinsavak) szerkezetének feltárása, illetve az egészségügyi dokumentumok feldolgozása. Az ügyfélszolgálatok, call centerek munkáját is várhatóan átalakítják a nagy nyelvi modellen alapuló chatbotok és dialógusrendszerek. Robotikában, pénzügyekben és az oktatásban is keresik az új technológia felhasználásának lehetőségeit. A nyelvi modellek további fejlődése és tulajdonságainak egyre alaposabb megértése a jövőben is számos új lehetőséget fog mind az alkalmazásfejlesztők, mind a felhasználók számára jelenteni.