Eötvös Loránd Tudományegyetem
Természettudományi Kar

Borbély Bernárd

Gépi tanulási modellek interpretálhatósága idősoros adatokon

Szakdolgozat
Alkalmazott Matematika MSc

Témavezető:
Csiszárik Adrián
HUN-REN Rényi Alfréd Matematikai Kutatóintézet

Budapest, 2025

Tartalomjegyzék

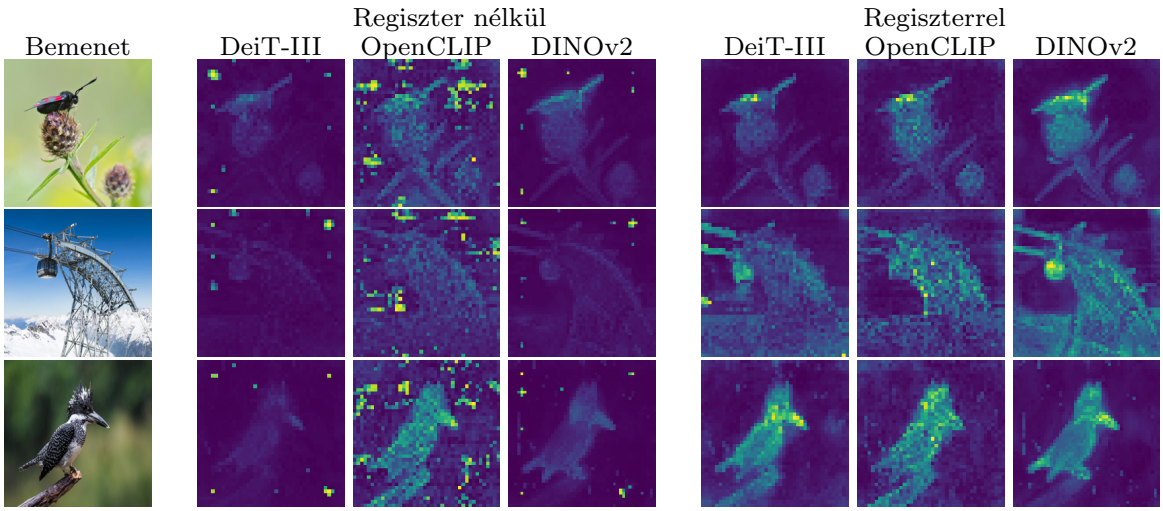
Ábrák jegyzéke

1. Motiváció és célkitűzés

Ahogy a mesterséges intelligencia egyre szélesebb körben elterjed, természetes igényként jelenik meg az **interpretálhatóság** (*interpretability*) és **indokolhatóság** (*explainability*). Nem csak az számít, hogy mi lett az eredmény, hanem az is, hogy a modell mi alapján hozta meg a döntést.

Az utóbbi években a képi és szöveges adatok feldolgozásában a **Transformer** [13] architektúrák érték el a *state-of-the-art* eredményeket. Ezek a modellek az úgynevezett *attention* (figyelem) mechanizmusra épülnek, amely segít kiemelni a bemenet azon részeit, amelyek a feldolgozás szempontjából különösen fontosak.
Az *attention* mátrixok a képi feldolgozásban az interpretálhatóságot is támogatják, mivel figyelmi térképként (*attention map*) értelmezhetőek. A bemeneti képeken jól vizualizálják, hogy a modell a kép mely régióira fókuszál, így a működése az ember számára is átláthatóbbá válik.
A *Transformer* és *attention* mechanizmus nem csak képeken, hanem idősoros adatokon is alkalmazhatóak, például *EKG*-jeleknél. Ebben az esetben azonban a kiemelt részek értelmezése gyakran nem egyértelmű, és jelentős mértékben támaszkodik domain-specifikus szakértelemre.

Az *interpretálhatóság* nem csak a modell utólagos értelmezésében fontos, hanem már a modell alkotás folyamatát is gazdagíthatják. Az *interpretálhatóságra* törekedve olyan módszereket és eszközöket alkalmazhatunk, amelyek által jobban megérthető a modell belső működése és a modellezés folyamatának részévé is válhatnak.
Ezek az eszközök nem csak a modell elemzésére szolgálhatnak, hanem akár a modellt is támogathatják, például úgy hogy fontos részminták megjelenését jelzik. Így a jól értelmezhető információk feltárása önmagában is hasznos lehet, egyfajta előfeldolgozásként.
Mivel az *interpretálhatóság* és az *indokolhatóság* fogalmai egymással szorosan összefonódnak, ezért röviden kitérek a kettő közti különbségre. A dolgozatom fókuszában azonban elsősorban az interpretálhatóság áll. A dolgozatom célja, neurális hálók interpretálhatóságának vizsgálata idősoros adatok esetén. Ennek érdekében áttekintem a gépi tanulás jelenlegi eszközeit az interpretálhatóságra, majd alkalmazhatóságát vizsgálom idősoros adatokon.



1. ábra. A mostanra már elhíresült „Vision Transformers needs registers" attention alapú megközelítésében, néhány token hozzáadásának segítségével az attention mapek interpretálhatóbbá tehetők. Az ábra a „Vision Transformers needs registers" [3] cikkből származik.

2. Interpretálhatóság a gépi tanulásban

2.1. Az interpretálhatóság fogalma

A gépi tanulási modellek interpretálhatóságának célja, hogy emberi módon értelmezhetővé tegye a modell döntését a felhasználó számára. Ez az értelmezhetőség származhat a modell építéséből azaz, hogy olyan elemeket építünk bele, amelyek interpretálhatóságot szolgáltatnak. Szintén származhat utólagos értelmezési módszerekből, amelyek már egy betanított modell működésére hívatottak rávilágítani.

A képi világban az *attention* mátrixok használata egy kiváló példa erre. Az attention mátrixok segítségével jól vizualizálható, hogy a kép egyes részei mennyire relevánsak a döntéshozatalban. Az interpretálhatóság ereje abban rejlik, hogy ezek a kiemelt régiók gyakran egybeesnek azokkal a képterületekkel, amelyekre egy ember is figyelmet fordítana egy hasonló döntés során — például egy orvos a röntgenkép elváltozásaira.

A nyelvi modellek esetében az interpretálhatóság egy másik formája lehet az információ forrásmegjelölése. Amikor a felhasználó kifejezetten a kutatás segítéséhez használja a nyelvi modelleket, akkor nem elégszik meg azzal, hogy hozzájut az információhoz, hanem referenciát is keres, amelyben önmaga is leellenőrizheti az információ helyességét.

Egy harmadik példa az interpretálhatóság fontosságára és módjára egy banki döntéstámogató rendszer. Képzeljünk el egy olyan helyzetet, ahol egy gépi tanulási modell segít a bankban eldönteni, hogy adjanak-e a hitelt az illetőnek vagy sem. Ehhez olyan adatokat használnak, mint életkor, jövedelem, meglévő tartozások, és kredit történet. Ha a modell elutasít egy hitelkérelmet, akkor az *interpretálhatóság* segít rámutatni a kapott adatok alapján, hogy mely információk alapján hozta meg ezt a döntést. Ez segít a modellt átláthatóbbá és megbízhatóbbá tenni.

2.2. Indokolhatóság és az interpretálhatóság

Az *interpretálhatósággal* szorosan összefűződő fogalom az *indokolhatóság*. Indokolhatóság esetén az a cél, hogy a modell döntését a bemenet és a modell működésével magyarázzuk, azaz technikai magyarázattal szolgáljunk. Az *interpretálhatósággal* szemben, itt nem feltétlenül szükséges, hogy emberek számára jól értelmezhető legyen az információ, sokkal inkább az érdekel minket, hogy a bemenet és a modell mely részei játszottak fontos szerepet a döntés meghozásában.

Például ha egy neurális háló belső rétegeiben azonosítjuk, hogy mely neuronok voltak legaktívabbak, akkor magyarázatot adunk a döntés meghozatalára. Azonban laikusok számára ezek az aktivációk nem értelmezhetőek.

Ebben a példában jól látszik, hogy az *interpretálhatóság* és *indokolhatóság* ugyan szorosan összefüggő fogalmak, nem teljesen fednek át.

2.3. Black box és white box modellek

Ahogy a gépi tanulásos modellek egyre jobb eredményeket érnek el, egyre bonyolultabb és emberek számára nehezen értelmezhető módon hozzák meg a döntéseiket.

Vannak olyan modellek, amelyek önmagukban jól interpretálhatóak. Ebbe a kategóriába tartoznak a döntési fák ahol hozzáférünk a csúcsokban kiértékelt kérdésekhez, amelyek alapján végül megtörténik a döntéshozatal. Az önmagukban jól értelmezhető gépi tanulási módszereket összefoglalóan **white-box** modelleknek is szokták nevezni.

Azokat a modelleket, amelyeknek a belső működése nem egyenesen interpretálható, hanem utólagos munkát igényel, gyűjtőnéven **black-box** modelleknek szoktuk nevezni. Azonban ezeknél sem kell teljesen lemondani az értelmezhetőségről. Egyrészt léteznek módszerek amelyek utólagosan segítenek értelmezni a modell működését. Illetve bizonyos architektúrákban, mint például a *Transformerek* esetén az *attention* mechanizmus, önmagukban is hasznos információt tudnak nyújtani a döntés indoklására.

2.4. Egy fontos black-box modell: a neuronhálók és ezek interpretálhatósága

Dolgozatom témája a neuronhálók köré szerveződik, ezért ebben az alfejezetben részletesebben bemutatom az ehhez kapcsolódó interpretálhatósági kérdéseket, mielőtt dolgozat fő témájára, az idősoros modellek interpretálhatóságára térek.

2.4.1. Computer vision.

A *computer vision* világában sokféle interpretálhatósági eszköz elérhető. A *state-of-the-art* modellek neuronhálókon alapulnak, tehát *black-box* modellek és utólagos interpretálásra szorulnak. Az egyik legismertebb utólagos interpretálási módszer a **Grad-CAM** [9]. A

Grad-CAM klasszifikálási feladatkörnyezetben lett kifejlesztve és egy adatpont klasszifikációját segíti megindokolni. Az indoklás osztályonként külön-külön értelmezhető, azaz a bemeneten mutatja meg, hogy az adott osztályba való klassszifikáláshoz a bemenet mely részeiből következtetett. Az indokláshoz deriválja az osztályra tett predikciót a bemenetre. Ez a derivált az ami lényegében megadja majd az egyes pixelek fontosságát, így létrehozva egy *heat-map*-et.

A szegmentáció feladat, például rákos sejtek szegmentációja, önmagában szintén egy interpretálási módszer lehet. Ebben a feladatban a betegnek tűnő részeket kell körberajzolnunk és ez önmagában egy bizonyítékul tud szolgálni a betegségre.

2.4.2. Nyelvi modalitás

A nyelvfeldolgozás egyik fő kihívása sokáig az volt, hogy egy-egy szó több jelentéssel is bírhat és a szövegkörnyezettől függ, hogy melyik jelentésére van szükségünk. Például a nyúl jelentheti a szőrös emlős állatot, az emberi karnak egy mozdulatát (pl. nyúl a süti után) és egy esemény időben történő elhúzódását (pl. a szakdolgozat írás az estébe nyúlt).

Komoly áttörést a *Transformer* architektúrák hoztak a beépített *attention* mechanizmusukkal, amelyek segítettek, hogy a bemenet elemeinek reprezentációját a kontextus függvényében pontosítani lehessen. A nagy nyelvi modellek, mint például a Lama3, GPT-2, GPT-3, Gemini[4, 8, 2, 11] *Transformer* architektúra épülnek. Az *attention* mátrixok nem csak a feldolgozásban segítenek, hanem az értelmezésben is fel lehet használni őket. A vizualizálhatóság révén, jól követhető, hogy a szöveg mely része volt fontos a modell számára az adott predikció, vagy döntés meghozatalában.

Hasonló figyelmi mintázatokat az idősoros adatokon is vizsgálhatók, bár ezek értelmezése gyakran kihívásokkal járhat.

3. Idősoros modellek interpretálhatósága

Az idősorok a legegyszerűbb formájukban egy folyamatnak az időben egymás után vett mérései, feljegyzései. Emiatt az idősorokat nagyon diverz területeken alkalmazzák. Egy idősorral le lehet írni szívritmus működését, tőzsdei folyamatokat és egy fúróhegy rezgését is, amelyek már természetükben is nagyon más folyamatok. Az idősor nagyon népszerű, és széles körben használt modellezési eszköz, amely gyakran vált gépi tanulási modellek célpontjává. Ahogy minden más domainben, az utóbbi években az idősoros gépi modellezésben is megfogalmazódott az igény az interpretálhatóságára.

3.1. Az idősor és kapcsolódó modellezési fogalmak

Ahhoz, hogy beszélni tudjunk az idősoros gépi tanulási modellek interpretálhatóságáról, először érdemes definiálni az alapfogalmakat, és megismerkedni az idősorok természetével. A legegyszerűbb formájukban az idősorok, szimplán egy megfigyelt folyamatnak az időben rendezett feljegyzései.

3.1. Definíció (Idősor)**.** *Az $X = \{x_1, x_2, \ldots\}$ rendezett feljegyzések sorozatát **idősornak** nevezzük, ha mindegyik x_t feljegyzés egy specifikus t időpontban készül, ahol $t \in T$ és T az **idősor indexhalmaza**. A T halmazról megköveteljük, hogy rendezhető legyen.*

Amennyiben T diszkrét, akkor **diszkrét idősorról**, ha pedig T intervallum (pl. $[0, 1]$), akkor **folytonos idősorról** beszélünk. A feljegyzett x_t értékekre általában úgy gondolunk mint egy véletlen folyamat megfigyeléseire, vagyis egy X_t valószínűségi változó egy realizációjára.

A diszkrét idősorok esetében a feljegyzések általában *egyenlő* időközönként történnek, de vannak olyan idősorok is, ahol *változó* a feljegyzések gyakorisága. Ilyen esetben akár a feljegyzési időpont is az idősor része lehet. Az idősorok esetében tehát az egyik kulcsfontosságú tulajdonság, hogy időben egymás után készülnek a feljegyzések, akár diszkrét, akár folytonos.

Az idősorokkal kapcsolatban is megfogalmazható többfajta feladat. Egyrészt értelmezhető az **idősor analízisének** feladata, ami a mögöttes folyamat tulajdonságainak feltárásával, megértésével foglalkozik. Illetve megfogalmazhatók explicit **modellezési** és **predikciós** feladatok is.

3.1. Feladat (Autogreresszív predikciós feladat)**.**
*Adott $X = (x_1, x_2, x_3, \ldots, x_n)$ idősor esetén egy olyan **függvényt** keresünk, amely az idősorból származó $x_1, \ldots, x_{k-1} \quad (k \in \mathbb{N}_+)$ mintából ad egy predikciót a x_k **értékre**, vagy **eloszlására**.*

Az x_k értékekre a korábbi értékekből következtetünk, ezt jelenti az **autoregresszivitás**. Az idősorok specialitása, hogy majdnem kizárólag mindig feltesszük, hogy a korábbi

értékektől függ valamilyen módon a jelenlegi k-adik érték. A függés módját a modellcsalád specifikálja, ez bármilyen bonyolult függés lehet, de gyakran lineáris.

3.2. Feladat (Analízis feladat)**.**
Adott $X = (x_1, x_2, x_3, \ldots, x_n)$ idősor esetén, az idősort ***generáló*** *megfelelő* ***modellcsaládot*** *szeretnénk kiválasztani, illetve a modellcsaládra jellemző tulajdonságokat meghatározni. Például szezonális trendek, és periódusaik, vagy növekvő trend meghatározása.*

A fent említett két feladat nem teljesen diszjunkt. Ahhoz, hogy egy predikciót végre tudjunk hajtani, szükségünk van az idősor valamilyen modellezésére. Az idősor analízis segíthet nekünk meghatározni, hogy milyen modellcsaládot válasszunk a modellezéshez. Miután kiválasztjuk a legjobb paraméterrel rendelkező modellt a családból, tudunk prediktálni. Megjegyezném, hogy amint kiválasztjuk, hogy melyik modellcsaláddal kívánjuk leírni az idősorunkat, lényegében máris kaptunk egy predikciós modellt.

A szakdolgozatban mind az idősort generáló folyamatra és mind a predikciós feladatot elvégző gépi tanulási módszerre „modelként" hivatkozunk. A fenti szemléletmód világít rá a kapcsolatra: amint kiválasztjuk a modellcsaládot, a gépi tanulási eljárás feladata a paraméterek kioptimalizálása, amelynek eredményeként a modellcsalád egy konkrét, már közvetlenül előrejelzésre használható modellé válik.

3.1.1. Példák és kapcsolatuk az interpretálhatósággal

Ebben a fejezetben néhány idősoros példával demonstrálom az interpretálhatóság sokszínűségét.

3.1. Példa (Napi hőmérséklet)**.** *Az egyik legtermészetesebb diszkrét idősor a meteorológiában, a hőmérséklet alakulásának feljegyzése. Ebben az idősorban megfigyelhető egy szezonális trend: a nyarak melegebbek, a telek hidegebbek.*
Az ilyen szezonális trendekkel rendelkező idősorokat, trigonometrikus függvények segítségével lehet jól modellezni: $X_t = cost(t/10) + Z_t$, ahol a $cos(t/10)$ szolgáltatja a szezonalitást, és Z_t pedig valamilyen 0 várható értékű valószínűségi változó, amely a véletlen ingadozást hivatott modellezni.

A 3.1.példában az interpretálhatóság eszközét a szezonalitással kapcsolatos megértéseink, és az ezekkel kapcsolatos modellezési döntseink adják. Azonosítjuk, hogy a napi középhőmérsékletben fellelhető egyfajta periodicitás. Ezt a periodicitást az évszakok váltakozásával magyarázzuk. A modellben pedig egy trigonometrikus függvénnyel reprezentáljuk.

3.2. Példa (Egy értékpapír tőzsdei árfolyama)**.** *Szintén szokás a tőzsdén egy értékpapír árfolyamát folytonos idősorként kezelni. Azonban egy árfolyam meghatározásánál nem feltélen szezonális, periodikus trend figyelhető meg, hanem sokkal inkább növekvő vagy csökkenő. Ebben az esetben érdemes az* $X_t = m_t + Z_t$ *típusú modellekkel modellezni az adatokat ahol* m_t *valamilyen monoton függvény,* Z_t *pedig a véletlen ingadozásért felelő komponenes.*

A 3.2. példában azonosítjuk, hogy az értékpapírnak van egy növekvő komponense, és az m_t közelítésével adunk magyarázatot, interpretálhatóságot az idősor viselkedésere. Habár a valós helyzet ennél sokkal komplexebb, és a piac tagjainak döntését számos tényező befolyásolhatja, de ebben az egyszerű példában is jól látható, hogy a modellezési döntés, hogyan adhat interpretálhatóságot.

Mivel egy tőzsdei folyamat működése bonyolultabb folyamat, ezért számtalan megközelítés és modell született. Az első intuíció az, hogy egy értékpapír árfolyama függ az aktuálistól és pár a mostani időpontot megelőző értéktől, hiszen ha egy árfolyam stabilan megy felfele, akkor a piac tagjai szívesebben fektetnek be, és a tőzsdéke emiatt megnő az árfolyama. Valósághűbb, ha úgy modellezünk, hogy a korábbi árfolyamok hatással vannak az aktuális árfolyam fejlődésére. Ezekkel el is érkeztünk az autoregresszív modellekhez, ahol a jelenlegi árfolyam a korábbiaktól függ.

3.3. Példa (Lineáris autoregresszív modell)**.** *Legyenek* $X_1, X_2, \ldots$ *valószínűségi változók, és* $p \in \mathbb{N}_+$*. A változók között fent áll az alábbi kapcsolat:*

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + Z_t,$$

ahol $\varphi_1, \ldots, \varphi_p$ *a modell változói, valós számok és* Z_t *egy* 0 *várható értékű (fehér zaj) valószínűségi változó. Ezt a modellt p-adrendű autoregresszív modellnek nevezzük és* $AR(p)$*-vel jelöljük.*

Ebben a modellcsaládban a modell illesztés, a család paramétereinek optimális meghatározása. Ennek az egy módszere a maximum-likelihood és az Expectation Maximization algoritmus, amelyekben iteratívan történik a paraméterek közelítése.

Ugyan egy ilyen bonyolultabb modell jobban le tudja írni egy idősor működését, de emberek számára már kevésbé értelmezhető. Az egyik lehetséges intrepretálhatósági megközelítés, ha meghatározzuk a paramétereket, de ezeknek nem feltétlen van az emberek számára intuitív értelmezése, így felmerül, hogy egyéb módszert keressünk az interpretálhatóságra.
Ezekből adódóan keresünk egyéb interpretálhatósági megközelítésetek.

A lineáris modellek gyakran túl egyszerűnek bizonyulnak, és nem tudnak bonyolult folymatokat modellezni. Ekkor jönnek szóba a bonyolultabb modellek, amelyek már képesek a folyamatot jól leírni. Azonban a bonyolultabb modellek általában nehezebben is értelmezhetőek. A bonyolultabb modellek nem csak az interpertálhatóságot tudják megnehezíteni, hanem a szerkezetük miatt gyakran a modell illesztés is lassabb. Ahhoz, hogy ilyen bonyolultabb modellek paramétereit jól tudjuk becsülni, egyre komplexebb és számításigényesebb algoritmusokra van szükség, amely nagyon le tudja lassítani a modell illesztés folyamatát.

A teljesség igénye nélkül, még szükséges megemlítenünk a gépi tanulással kapcsolatos idősoros modelleket.

3.4. Példa (Gépi tanulás alapú autoregresszív modellek)**.** *A gépi tanulás alapú modelleket az:*

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-p}) + Z_t,$$

összefüggés vezényli, ahol az f függvény lehet egy neuron háló, akár egy Transformer modell, és $Z_t \sim N(0, \sigma^2)$ a zaj változó, ahol $\sigma < \infty$. A p paraméter azt határozza meg, hogy mennyi megelőző megfigyelés lehet hatással a jelenlegi állapotra.

Ezeket a modelleket már szokás black-box modellnek tekinteni, és a modell illesztés már gradiens deszcens alapú módszerekkel történik. Az ilyen bonyolult modelleknél az interpretálhatóság kérdése már közel sem világos, ezért interpretálhatósági módszerek sokaságával igyekszünk mégis éretelmezést nyerni az illesztett modell predikciójára.
Érdemes megjegyezni, hogy a neurális hálókkal való modellezés több szempontból is megalapozott. Az első az előbb említett idősor modellezés. Egy másik érv, hogy a neurális hálókról és Transformerekről, elméleti szempontból is már születtek olyan eredmények, hogy tetszőleges függvényt tudnak közelíteni, feltéve ha lehet a háló végtelen szélességű és elérhető végtelen sok tanító adat.

3.1.2. Eseménysorok, mint idősorok egy-egy pillanatképeinek progressziója

Az idősorokból, kifejezetten a folytonos idősorokból lehet újabb idősorokat alkotni, ha mintavételezünk bizonyos időpontokban. Mivel egy folyamat monitorozásában a való életben nem tudunk ténylegesen folyamatos mintavételt végezni, így a legtöbb elérhető idősor valamilyen folyamatokból mintavételezett diszkrét idősor.

Előfordulhat, hogy nem számít nekünk egy mérésnek a pontos értéke, vagy nem tudunk pontos értéket mérni, ezért az X_t változó felvehető értékeinek tere nem egy folytonos, hanem diszkrét tér. Az ilyen folyamat modellezésében tehát van egy állapot, és a következő felvett állapot a jelenlegitől és pár megelőzőtől függ. Ha véges sok előzőtől függ a következő állapot, akkor Markovi folyamatról beszélhetünk. Ebben a fejezetben

ismertetem a fogalmat, és a pontosabb matematikai definícióval későbbi fejezetben foglalkozom.

Az ilyen modelleket mi eseménysoroknak, vagy esemény alapú modellezésnek hívjuk, és a szakdolgozatomban kiemelt helyet kap az interpretálhatóságuk. A fő kihívás ezen a területen is, hogy nem magától értetődő, hogy mi volna jól interpretálható információ.

3.2. A gépi tanulási modellek interpretálhatóságának attitűdjei

A gépi tanulási modellek interpretálhatóságára rengeteg módszert kifejlesztettek az évek alatt. Mivel az idősorok nagyon diverz domaint alkotnak, és nem egyértelmű milyen jellegű információ értelmezhető jól, ezért az interpretálhatósági módszerek is széles skálán mozognak. Ebben a fejezetben különböző osztályozásaival bemutatjuk az interpretálhatósági megközelítéseket, a 2. fejezetben már említett szempontok szerint. A fejezet tartalma nagy részben Theissler és munkatársai [12] munkájára alapul.

3.2.1. Post-hoc és ante-hoc módszerek

Az interpretálhatóság megjelenése történhet a modell előállítása során, és a már kész modell utólagos elemzésével is, előbbit *ante-hoc*, utóbbi *post-hoc* módszernek nevezzük. Az *ante-hoc* modellek önmagukban adnak értelmezhetőséget, a tervezésükből adódóan. Ide tartoznak például a **döntési fák**. A *post-hoc* interpretálhatósági módszereket, már a betanult modelleken végezzük és azt vizsgálják, hogy mit tanult meg a modell. *Post-hoc* módszerek például *Grad-CAM*.

3.2.2. Globális és lokális interpretálhatóság

Szintén megkülönböztethető globális és lokális interpretálhatóság. Globális magyarázatok estében a teljes tanítóhalmazra érvényes szabályokat keresünk. A lokális magyarázás azzal foglalkozik, hogy egy adott x instancián hogyan prediktált a modell.

3.2.3. Modellspecifikusság

A harmadik megkülönböztetés a **modell specifikus** és **modell független** módszerek. Vannak olyan módszerek, amelyek esetében nem számít, hogy az értelemzett modell az white-box, black-box modell.Ezek a **modell független** módszerek. Velük szemben bizonyos módszerekhez szükség van **modell specifikus** feltételekre, például a Grad-CAM csak differenciálható modelleken alkalmazható.

3.2.4. Saliency alapú megközelítések

A *saliency* alapú megközelítésekben a bemenet egyes részeinek a fontosságát szeretnénk meghatározni. Ennek a kivitelezésére többféle módszer is létezik.

A **SHAP módszer** [5] az egyik legelterjedtebb módszer egy instancen lokális interpretálhatóságot adni. A SHAP módszerben a játékelméletből ismert Shapley-érték segítségével mérjük fel egy-egy feature fontosságát. Nagy előnye, hogy ez egy modell független módszer, hiszen a bemenetet csupán black-boxként kezeli, így bármilyen gépi tanulási modellre alkalmazható. Nehézsége, hogy a valódi SHAP értékek kiszámolásához szükség volna a feature-ök minden részhalmazát vizsgálni, ezért inkább approximációkat alkalmaznak.

Léteznek még *attention* alapú *ante-hoc* módszerek, ahol a modell valamely részében megjelenik a Transformerekben elterjedt *attention* mechanizmus. Ezen módszerek csak akkor alkalmazhatóak, ha az attention mechanizmus bele van építve a modellbe, így az ilyen típusú módszerek *modell specifikusak*.

A **gradiens alapú** módszerek, mint a **Grad-CAM** [9], vagy a **Saliency** [10], a bemenetre vett gradiens segítségével határozzák meg az egyes jellemzők vagy pontok fontosságát. Ezek a módszerek gyakran csak kevés plusz számolást igényelnek, hiszen a tanítás közben már kiszámoltuk a szükséges deriváltakat. Azonban csak olyankor használhatóak ha a modell differenciálható.

Összefoglalva, a saliency alapú megközelítések a bemeneten jelölnek ki fontos részeket. Emiatt gyakran domain tudást igényelnek ahhoz, hogy értelmezhetőek legyenek, és az átlagos felhasználó számára továbbra is nehezen interpretálható az eredmény.

3.2.5. Részsorozat alapú megközelítések

Egy alternatív megközelítése az interpretálhatóságra, ha implicite valamilyen részsorozatot keresünk, amivel érvelni akarunk.

Maletzke és munkatársai [6] úgynevezett **motívumokat** keresnek, olyan fix hosszúságú részszekvenciákat, amelyek gyakran megjelennek az adathalmazban. Majd ezeket a *motívumokat* használják a klasszifikálási feladat megoldására egy döntési fa segítségével. Ez tehát **globális** mintákat azonosít az adathalmazban és így készít egy *ante-hoc* intrepretálható modellt. A *motívumos* megközelítés előnye, hogy implicite interpretálhatóak hiszen egy összefüggő részsorozat tartalmazásán mentén lehet magyarázni a klasszifikálást. Azonban nem sikerült velük *state-of-the-art* teljesítményt elérni, így teljesítményt áldozunk fel az interpretálhatóság kedvéért.

A *motívumok*-hoz hasonló ötleten alapul a **shapeletek** ötlete [14]. Ezeknél nem követeljük meg, hogy részsorozata legyen az idősoroknak. Tipikusan a *shapelet* alapú módszerekben olyan k *shapelet* megtalálása a cél, amelyek jól elkülönítik az egyes klasszifikálási osztályokat. Ezután egy távolság függvény segítségével egy instanciát transzformálnak a k shapeletre úgy, hogy megnézik az instancia tagjainak minimális távolságát az egyes shapeletektól, így kapva egy k hosszúságú, valós értékű transzformáltat. A klasszifikálás a transzformált adathalmaz segítségével történik.

Mind a *motívum* és *shapelet* megközelítésű módszereknél az optimális hossz megvá-

lasztása nehéséget jelent, mivel nagyban tudják befolyásolni a klasszifikálás pontosságát.

3.2.6. Eseménysorozatok felbontása

Az idősorok között a szakdolgozatomban megkülönböztetek egy osztályát az idősoroknak: az eseménysorokat. Ezen idősorok esetében azt feltételezzük, hogy több mögöttes generáló folyamat állapotait látjuk, amelyek valamilyen véletlen módon vannak összefésülve.

Egy ilyen eseménysorra egy lehetséges interpretálása lehetne, ha valahogy azonosítjuk azon részsorozatokat amelyek ugyanabból a generáló folyamatból származnak. A módszer előnye, hogy az egyes részsorozatok külön-külön értelmezhetőek lehetnek.

A hátrányuk azonban abban rejlik, hogy az idősorokról nem mindig feltételezhető, hogy több folyamat összefűződését látjuk, így a módszer az általánosságából veszít.

Arról, hogy mit értünk eseménysorozatok és a felbontásuk alatt, a 4. fejezetben tárgyalok bővebben.

4. Eseménysorozatok felbontása

4.1. Általában

Az eseménysorozatok felbontása amiatt is érdekes témakör, hogy ha sikerül több részsorozatra felbontani az eseménysort, akkor az egyes részsorozatok önmagukban interpretálhatóságot tudhatnak szolgáltatni. Erre kiváló példa lehet egy beteg orvosi élettörténetének követése.

4.1. Példa (Egészségügyi életút)**.** *Az* ***egészségügyi életút*** *alatt, egy adott illető, az egészségüggyel kapcsolatos életeseményeit értjük.*

Az *egészségügyi életútba* tartoznak az orvosi látogatások, a különböző kezelések, de még a felírt gyógyszerek kiváltása. Egy példa segítségével mutatom be, hogy az *egészségügyi életúttal* kapcsolatban, természetes módon jelenik meg a részsorozatokra bontás feladata, mint interpretálhatósági módszer.

Képzeljük el, hogy egy beteget kezel a háziorvosa vérnyomás problémákkal, és gyakori orvoslátogatás közben elkapja az influnezát. Ekkor *egészségügyi életútba* egymás után kerülnek be a vérnyomás és az influenza kezelésére az orvosi látogatások és gyógyszer kiváltások. Emiatt az egymást követő feljegyzések, már nem feltétlenül függnek össze. Ha a háziorvos ezután elemezni szeretné a páciense vérnyomás betegségének a progresszióját, akkor kiválasztja a a releváns feljegyzéseket és csak azokat veszi figyelembe. Azaz máshogy fogalmazva kiválasztja a releváns részsorozatot az orvosi élettörténetből, így nyerve interpretációt.
Hasonló módon amikor egy történész a Római Birodalom történéseit elemzi, akkor a fennmaradt feljegyzésekből a fontosakat kiválasztva magyarázza az aktuális uralkodó döntéseit. Vagyis az eseménysorból a releváns részsorozattal magyarázza az aktuálisan tárgyalt következményt. Így természetes módon jelenik meg az ötlet, miszerint az eseménysorozatokat ügyes módon részsorozatokra bontva tegyük értelmezhetőbbé a sorozat.

4.2. Matematikai modell

Ahhoz, hogy az eseménysorok felbonthatóságával dolgozni lehessen, szükséges egy matematika modell. Ennek a matematikai modellnek kell megragadnia, hogy a generálás komponensekből áll, amelyeket valamilyen kombinálási mechanizmus kever össze. A generatív modell megválasztása befolyásolja, hogy a szétválasztáshoz milyen attitűddel állunk hozzá. Aíg egy komplexebb modellel pontosabban tudjuk leírni az eseménysorokat, ha túl bonyolítjuk, a szétválasztás is nehezebbé válhat. Illetve a modellezésnél azt is szem előtt akarjuk tartani, hogy az interpretálhatósági eszköz, a részsorozatokra szedése legyen.
Mindezen célokat szem előtt tartva nézelődünk a matematika eszköztárában.

4.3. Egy spec eset: Markovi keverékmodellek

A speciális eset amely a fő témája a dolgozatnak: a markovi keverékmodellek. A célunk az eseménysorozatokat modellezni és interpretálhatósági módszereket vizsgálni rajtuk. Természetes választásnak tűnik a véletlent belevinni a modellbe, hiszen nem teljesen determinisztikus folyamatokat modellezünk.
Egészen egyszerű és jól megértett valószínűségi matematikai struktúrák a Markov-láncok, és különféle rokonaik. A Markov-láncok egyszerűségük, és analizálhatóságuk miatt gyakran használt modellezési eszközök, mi is ezért választottuk őket modellezésre. Több lehetséges megközelítéssel is foglalkozott már az irodalom, de mielőtt rátérünk a különböző lehetőségekre, fejtsük ki, a generálás menetét.
Az alábbi formuláció független gondolkodás eredménye, azonban lényegében megegyezik Batu és munkatársainak [1] a formulációjával, így a továbbiakban az ő jelölésrendszerüket használva ismertetjük a modellt. Van egy fő vezérlő és adott $M_1, M_2, \dots M_k$ Markov-láncok. A vezérlő valósítja meg a generálás folyamatát. Valamilyen véletlenszerű módon kiválaszt egy M_i Markov-láncot, amely "lép" egyet, vagyis folytatja a bolyongást az állapotterén, a megfelelő átmenet valószínűségek alapján. Az állapotokat azonosítjuk a generált karakterrel, és az új állapot karaktere lesz az eseménysorozat következő eleme.

A generáló modellt a Markov-lánc kiválasztás szerint két részre lehet szétválasztani.

- Az egyszerűbbik eset, amikor M_i Markov-láncot p_i valószínűséggel választhatjuk, és a p_i-k összege 1. Ez a probabilisztikus módszer.
- Az általánosabb modell, amikor a következő Markov-lánc választása, az aktuálisan kiválasztottól, azaz p_{ij} értékektől függ. Ezt szokták lánc-függő módszernek is nevezni.

Szintén modellezési döntés kérdése, hogy a Markov-láncok állapotterei hogyan viszonyulnak egymáshoz. Lehetnek

- páronként teljesen diszjunktak
- Vagy lehet átfedése az állapotterek között.

A kísérleteink nagy részében diszjunkt állapotterekkel és a probabilisztikus Markov-lánc választó módszerrel dolgozunk, azonban a jövőbeli kutatások és az alaposabb megértés szempontjából érdemes kidolgozni az általánosabb generatív modellek működését is.

4.4. Módszerek markovi eseménysorozatok szétszedésére

Miután kiválasztjuk az eseménysorozat generáló modellünk, azzal szeretnénk interpretálhatóságot nyújtani, hogy azonosítjuk azokat a részsorozatokat amelyek ugyanabból a

Markov-láncból származnak. Ez magában foglalja mind az egyes láncok állapotterének a meghatározását, mind pedig a mixture modellben a lánc-kiválasztási valószínűségeket. Azaz a probabilisztikus modellben a p_i értékeket, a lánc-függőben pedig a p_{ij} értékeket is meg kell határozni. Megjegyzendő, hogy a diszjunkt állapottér esetében, redukálható a feladat arra, hogy a teljes állapotteret partícionáljuk. Hiszen ha már a partícionálás adott, akkor mind a Markov-lánc átmeneti valószínűségei, mind a Markov-lánc választó valószínűségek és átmenetek azonosíthatóak.

4.5. Benchmark módszerek a klasszikus irodalomból

Ebben a fejezetben a szakirodalomból két ismert módszert tekintünk át a markovi keverékek szétválasztására. A saját módszerünk nem ezen algoritmusokra épül, de viszonyítási alapnak szolgálnak.

Kezdetek

A feladat formulációját Batu és munkatársai írták le először 2004-ben. Először azt vizsgálják mennyire pontosan fejthető vissza egy keverék modell, pusztán a megfigyelt szekvencia alapján. Ennek a formalizálására definiálják a **megfigyelési ekvivalenciát**.

4.1. Definíció (Megfigyelési ekvivalencia)**.** *Legyenek*

$$\mathcal{P} = \langle M^{(1)}, M^{(2)}, \ldots, M^{(k)}, \mathcal{I} \rangle \quad \text{és} \quad \mathcal{P}' = \langle M'^{(1)}, M'^{(2)}, \ldots, M'^{(k')}, \mathcal{I}' \rangle$$

markovi mixture-ök, ahol $\mathcal{I}$ és $\mathcal{I}'$ a keverékmodell keverési mechanizmusa.

Ekkor mondjuk, hogy $\mathcal{P}$ és $\mathcal{P}'$ **megfigyelési szempontból ekvivalensek***, ha $\mathcal{P}$ markov-láncainak tetszőleges inicializálási valószínűségeihez létezik $\mathcal{P}'$-nek olyan inicializálása, hogy bármely, n hosszúságú generált szekvencia esetében annak a valószínűsége, hogy $\mathcal{P}$ generálta, ugyanannyi mint $\mathcal{P}'$ valószínűsége.*

A definícióból adódóan, két modell statisztikailag nem megkülönöztethető csupán a megfigyelt szekvencia alapján.

Batu és munkatársai ezután prezentálnak egy algoritmust arra a feladatverzióra amikor az állapotterek diszjunktak, és az egyszerűbb, *probabilisztikus* módszerrel történik a keverés. Az algoritmus helyességéhez szükséges pár feltétel, például ergodikusság, azaz hogy minden állapotból el lehet jutni minden másik állapotba véges várható lépésszámban.

Fontos megfigyelés, hogy mivel a Markov-láncok állapottereik diszjunktak, ezért a feladat megoldásához elég, ha partícionáljuk a teljes állapotteret. Ugyanis, ha már meg vannak az egyes partíciók, egyszerűen a megfelelő állapotok részsorozatából rekonstruálni lehet a Markov-láncok átmeneti mátrixát. Az általuk adott algoritmus, tehát az állapottér partícionálását végzi el. Az algoritmus azon alapszik, hogy különböző típusú szimbólumsorozatok relatív gyakoriságát megszámoljuk a szekvenciában és ez alapján

tudunk következtetni arra, hogy egy Markov-láncba tartoznak vagy sem. Pontosabban fogalmazva, az i, ij és ipj típusú szimbólumsorozatok relatív gyakoriságának kiszámolására van szükség. A sikeres partícionálás után, a Markov-láncok rekonstrukciója egyszerűen elvégezhető. Az imént meghatározott állapottér partíciói szerint kiválasztjuk a partíció szimbólumait tartalmazó részsorozatot. Ebből a részsorozatból pedig rekonstruálható a Markov-lánc átmenet mátrixa. Illetve szintén egyszerű módon meg lehet határozni a Markov-láncok keveréséért felelős p_i értékeket. Összességében, az algoritmus segítségével nagy valószínűséggel elő tudunk állítani egy markovi-keverék modellt amely **megfigyelés szempontjából ekvivalens** az eredeti generáló folyamathoz.

Batu és munkatársai [1] az eddig említett alapötleteket kibővítve a diszjunkt állapotterek és lánc-függő keveréses modell illesztésére is adnak egy algoritmust. Ez már egy nehezebb feladat, de az algoritmus továbbra is polinomiális marad. Ha már átfednek az állapotterek és egyszerű keveréses a modell, akkor már csak bizonyos feltételek mellett lehet modellt illeszteni. Az az eset, amikor átfedhetnek az állapotterek és lánc-függő a keverés, az már nem visszafordítható modell.

Egy másik megközelítés

Minot és Yue [7] a Batu és munkatárasai[1] által megfogalmazott feladatot újraformulálják mint egy rejtett Markov-lánc (HMM) modellt. Ebben a megközelítésben a rejtett Markov-lánc állapotai az egyes komponens Markov-láncok, illetve plusz egy komponens az a változó amely kiválasztja, hogy melyik Markov-lánc fog generálni. A megfigyelt állapot amit a *HMM* kibocsát, az az utolsó kiválasztási változó, és a kiválasztott Markov-lánc állapotától függ.

Minot és Yue algoritmusa azt a feladatvariánst célozza meg, amelyben az állapotterek nem diszjunktak, de a keverési módszer az egyszerűbb *probabilisztikus* módszer. Az ő algoritmikus módszerük statisztikából ismeretes *Expectation-Maximization (EM)* algoritmus egyik változatára alapszik, így ennek az algoritmusnak a segítségével egy legvalószínűbb dekompozíciót kapunk. Azon szempontból fontos ez az eredmény, hogy ha az állapotterek nem diszjunktak akkor is lehetséges asszimptotikusan közelíteni a keverék modellt. Mivel azonban az *EM* algoritmus csak lokális minimumot talál, így a jelenlegi algoritmus is függhet a kezdeti értékektől.

4.6. Saját módszer

Az eseménysorozatok szétválasztására egy saját módszert is készítek. A feladat megoldásához elég ha a szekvencia elemeire meghatározzuk, hogy melyek származhatnak ugyanabból a Markov-láncból. A megközelítés a Transformer architektúrán alapul, és a modell belső reprezentációt felhasználva különíti el a komponenseket. Az módszer első lépésében egy Transformer modellt betanítunk be egy *next token prediction* feladatra. A tanítás

után kiválasztunk egy bemeneti szekvenciát és lekérjük a szekvenciában szereplő tokenek reprezentációját egy belső látens térből. Így minden tokenhez kapunk egy rejtett reprezentációs vektort. Ezután a kapott vektorokat csoportosítjuk, valamelyik klaszterező algoritmus segítségével. A hipotézis az, hogy egy klaszterba kerülő tokenek ugyanahhoz a markovi komponenshez tartoznak. A klaszter így implicit módon meghatározza a keverékmodell komponenseit. Miután a szekvencia tokenjeihez hozzárendeltük a komponensének címkéjét, a Markov-láncok átmenet mátrixa és a keverési arányok tapasztalati úton történő becsléssel meghatározhatóak.

A módszer eredményessége azon múlik, hogy a Transformer képes-e megtanulni a markovi keverékek szerkezetét és ennek megfelelően elválasztani egymástól a komponensek szimbólumait a reprezentációs térben.

5. Kísérlet

A kísérletek célja javasolt módszer korlátainak feltárása. Elsősorban arra keresem a választ, hogy a különböző keverési modelleket képes-e megtanulni a Transformer. A vizsgálatok során minden esetben *diszjunkt* állapotterekkel rendelkező keverék modellekkel dolgoztam. A keverési mechanizmus viszont kísérletenként változik.

5.0.1. Az predikció korlátai

A generatív modell egy sztochasztikus folyamat, így a predikciók pontosságának vannak bizonyos természetesen adódó korlátai. A sztochaszticitás miatt a predikcióban mindig van egy kis bizonytalanság, így egy bizonyos pontosság már nem várható el a predikciós modelltől. Még ha ismerem a generatív modell minden paraméterét: a Markov-lámcok átmenet mátrixait, a láncok kiválasztási valószínűségeit és még a láncok jelenlegi állapotához is van hozzáférésem; akkor is lehet bizonytalanság a jóslásban. Ebből adódik, egy alsókorlát a predikció lehetséges pontosságára. Ez pedig nem más, mint annak a valószínűsége, hogy pont azt a karaktert generálta a következőnek a keverék modell, mint ami a szekvenciába bekerült. Ezt az értéket **optimális alsó korlátnak** nevezem, és a tanítások során egyfajta alsó korlátnak tekintem.

A predikció pontosságára több féle felső korlát is adható. Az egyik a teljesen naiv becslés, amikor minden tokent ugyanolyan valószínűséggel tippel a modell következő tokennek. Ez a predikció egy triviális állapot, így természetes felső korlátként szolgál, amely összehasonlítási alap egy modell teljesítményére. Azonban ennél adható jobb, még mindig naív becslés. A becslés ötlete, hogy a tokenek tapasztalati eloszlásából adjuk a predikciónkat. Ez egy jóval pontosabb becslés, amelyet még mindig minimális számítási kapacitást igényel, így a kísérletek során ez a mérték fog szolgálni felső becslésnek.

Az alsó és felső becslések egy jó viszonyítási alapot szolgáltatnak a modellek valódi

teljesítményének felmérésében.

5.1. Alap kísérlet

Az első kísérletben azt vizsgáltam, hogy a Transformer architektúra képes-e általánosan megtanulni egy egyszerű markovi keverékmodell szerkezetét. Az ehhez használt adathalmazban szintetikusan generálok keverékmodelleket, amely két darab, egyenként két-két állapotú Markov-láncból áll. A keverékmodell egyetlen fix hosszúságú szekvenciát generál, ez a szekvencia lesz az adathalmaz egy sora. Az állapotterek diszjunktak, azaz nincs átfedés közöttük. A láncok közti váltás az egyszerűbb, probabilisztikus módszer alapján történik. Mind a Markov-láncok átmenet mátrixa, mind a Markov-láncok kiválasztási valószínűsége egyenletes eloszlásból generált véletlen eloszlások. A tesztadatok minden futásnál újragenerálódnak a fent leírt szabályok mentén. Így biztosítjuk, hogy a modell tényleges általánosítóképességét mérjük fel a tesztelés folyamán.

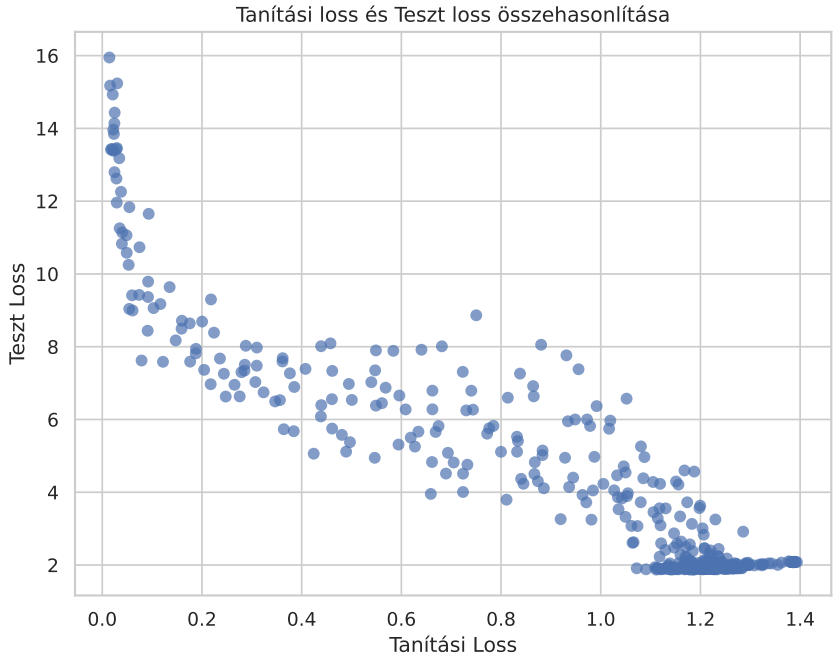
Paraméter	**Érték**
Markov-láncok száma	2
Állapotok száma lánconként	2
Keverési mechanizmus	Probabilisztikus

1. táblázat. A keverékmodell beállításai az alap kísérletben.

A kísérletben egy Transformer modell hiperparamétereit változtattam. Ezek a hiperparaméterek a Transformer rétegeinek számát, az *attention* fejek számát, a tanulási rátát, a generált szekvencia hosszát, a generált adathalmaz méretét, az optimizáló fajtája, illetve annak az ablaknak a méretét ahány tokenre „időben vissszafele" lát egy token, amikor az attention értékeket számoljuk.

2. táblázat. A Transformer modellek tanításához használt hiperparaméterek értékei

Hiperparaméter	**Érték / Megjegyzés**
Tanulási ráta	$[0.0001, 0.001, 0.01]$
Batch méret	32
Epoch-ok száma	5000
Szekvenciahossz	$[200, 400, 1000]$
Tanító adathalmaz mérete	$[100, 1000, 2000]$
Transformer rétegek száma	$[2, 3, 4]$
Fejek száma (multi-head attention)	4
Rejtett dimenziók száma	128
Optimizer	$[\text{"}SGD\text{"}, \text{"}adamW\text{"}]$
Loss függvény	Cross-entropy
Tanítandó cél	Következő token predikció



2. ábra. A hiperparaméter optimalizálás során a Tanítási loss összevetve a Teszt lossal.

5.1.1. Alap kísérlet eredményei

A kísérlet során azt tapasztaltam, hogy a Transformer architektúra nem tudja megtanulni a keverékmodell szerkezetét. A tanító adathalmazon ugyan valamennyire csökken a *loss* függvény, de a teszt adathalmazon a véletlenszerű tippelés szintjét éri el a modell. Sőt, ahogy a 2. ábrán látható minél jobb eredményeket ér el a Transformer az tanító adathalmazon, annál roszabb eredményeket ér el a teszt adathalmazon. A modell *overfittel*, azaz csak „megjegyzi" a tanítóadathalmaz elemeit, és nem általános struktúrát próbál megtanulni.

A sikertelenség mögött azonban más is állhat. Először is lehet, hogy a feladat túlságosan általános. Az egyeneletesen véletlenül generált keverék modellek lehetséges, hogy annyira diverzek, hogy már a modell nem tudja megtanulni az általános szerkezeti szabályokat. Egy másik lehetőség, hogy a generálás során az egymást követő karakterek ritkán vannak ugyanabból a láncból és emiatt nehezebb az egy láncba tartozó elemeket meghatározni. Az is előfordulhat, hogy a *Transformer* modell mérete, rétegeinek száma, rejtett reprezentációja nem voltak elegendően nagyok a struktúrák megtanulásához. A továbbiakban azt vizsgáltam, hogy az egyes nehézségek kiküszöbölése hogyan befolyásolja a tanítás eredményességét.

5.2. A lánchűség növelése

Amikor arra jutunk, hogy egy struktúrát nem sikerül megtanulnia a modellünknek, akkor érdemes lehet azt vizsgálni, hogy vajon a struktúra mely részei okozzák a nehézséget.

Ilyenkor a struktúra fokozatos egyszerűsítésével próbáljuk feltérképezni, melyik összetevő vagy tulajdonság teszi bonyolulttá a tanulási folyamatot.

Első lépésként a generálás során csökkentjük a Markov-láncok közti váltások gyakoriságát. Olyan keverékmodellt alakítunk ki, amelyben egy lánc aktiválódása utána, nagyobb eséllyel ugyanez a lánc marad aktív a következő lépésben. A keverési mechanizmus így önmaga is egy markovi folyamat lesz, amelynek az állapotai a különböző Markov-áncok, ez neveztük a *láncfüggő* keverésnek. Az így generált szekvenciákban lesz egyfajta konzisztencia, ahogy egy ideig az egyik lánc aktív, majd vált és a másik láncból kerülnek ki az új elemek. Ez struktúráltabb szerkezetet eredményez: a szomszédos elemek nagyobb valószínűséggel származnak ugyanabból a láncból, így megkönnyítheti a Transformer modellnek az összefüggések megértését. Annak érdekében, hogy a különböző szekvenciák keverési mechanizmusa ne legyen azonos, a keverékmodell generálásakor a láncválasztó folyamat átmeneti valószínűségeihez hozzáadunk egy kis zajt.

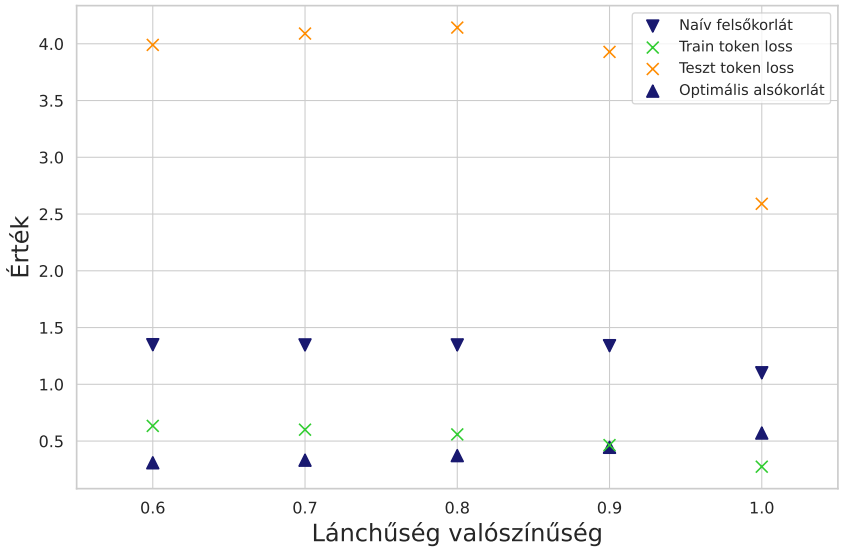
Paraméter	**Érték**
Markov-láncok száma	2
Állapotok száma lánconként	2
Keverési mechanizmus	Láncfüggő
Speciális szabály a keverésben	Nagy az esély ugyanabban a láncban maradni

3. táblázat. A keverékmodell beállításai a lánchűség kísérletben.

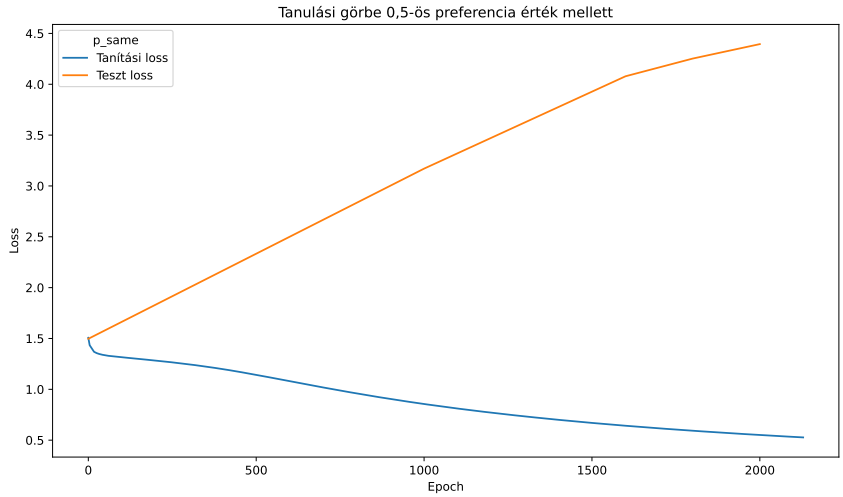
A kísérletekben a lánchűség paraméter változtatásával azt vizsgáltam, hogy az ilyen módon struktúráltabbá tett szekvenciák jobban tanulhatók-e a Transformer modellek számára. Fontos megjegyezni, hogy ebben és minden további kísérletben a betanítandó Transformer modell hiperparaméterei fixált értékek voltak. Ezekben a kísérletekben nem a modell kioptimalizálása volt a cél, hanem felmérni, hogy az egyes keverékmodellek milyen nehézségi szintet jelentenek a tanulása szempontjából.

5.2.1. A lánchűség növelésének eredményei

A 3. ábrán látható, hogy a lánchűség növelésével a *tanítási loss* is csökken. A tanítási *lossról* az is megfigyelhető, hogy a nagyobb lánchűség mellett közelebb kerül az alsó korláthoz, és a legnagyobb értékre már át is lépi azt. Az alsó korlát átlépése egyértelmű bizonyíték a túltanulásra, mivel információelméleti korlátokba ütközik az ilyen pontos előrejelzés. Ennek ellenére is, a *tanítási loss* csökkenő tendenciája arra utalhat, hogy a Transformer modell könnyebben megérti ezt a fajta szerkezetet. A megértés mellett szól az a megfigyelés is, hogy a lánchűség növekedésével a *teszt loss* is csökkenő tendenciát mutat. Ugyanakkor az is látható, hogy egyik esetben sem éri el a naív felső korlátot. További következtetéseket a tanítási görbe ábrájáról tudunk levonni.



3. ábra. A láncűség kísérletek eredményei. A láncűség értékek azt jelzik, hogy milyen valószínűséggel választjuk újra ugyanazt a markov-láncot aktívnak.



4. ábra. A $0,5$-ös láncűségű kísérlet tanulási görbéje.

A 4. ábra egy láncűséges futás tanítási görbéjét ábrázolja. Az ábráról az olvasható le, hogy ahogy a *tanítási loss* csökken, úgy a *teszt loss* folyamatosan nő. Ez a megfigyelés azt sugallja, hogy a Transformer modell, ilyen struktúrátabb szekvencia esetén is hajlamos túltanulni, és inkább „memorizálja” a tanító adathalmazt, semhogy a keverékmodell szerkezetét megértse.

5.3. A domináló Markov-lánc esete

A láncűséggel kapcsolatos megfigyeléseink arra utalnak, hogy a szerkezet továbbra is túlságosan komplex lehet, ezért további egyszerűsítéseket vizsgálunk. Az struktúra egyszerűsítésének az útján, következő lépésként olyan keverékmodellt hozunk létre, amelyben az egyik lánc domináns szerepet tölt be a generálás során. A modell tovább egyszerűsíthe-

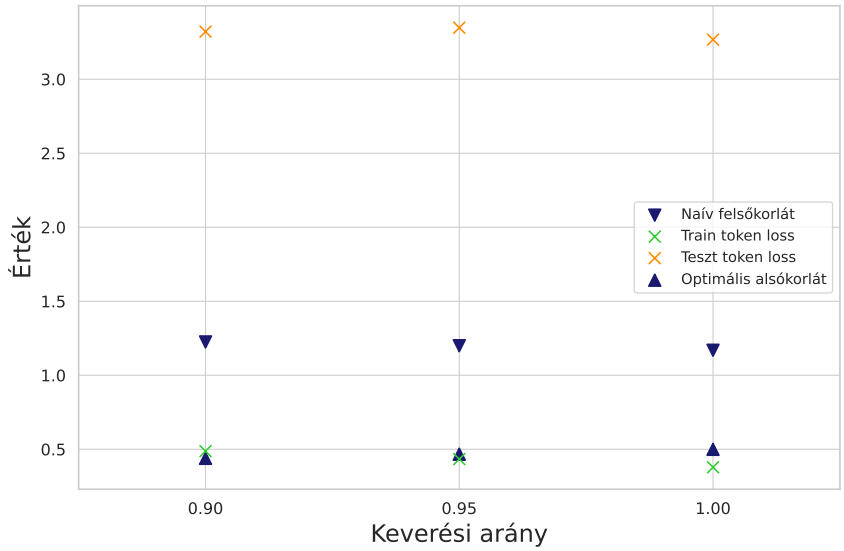
tő, ha a nem-dominálo komponenset tovább redukáljuk, mindössze 1 állapottal rendelkező Markov-láncra. A keverési mechanizmus ezúttal legyen megint probabilisztikus, azonban az egyik komponens lánc nagyobb valószínűséggel kerül kiválasztásra, mint a másik. Így a generált szekvenciák többnyire a domináns Markov-láncot követik, amelyet időnként megszakít egy egyszerű, determinisztikus folyamat. A domináns lánc preferenciájának a növekedésével így egyre jobban egy teljesen markovi folyamatra kezd hasonlítani a szekvencia. Annak érdekében, hogy az aktív Markov-láncok kiválasztásának valószínűségei ne egyezzenek meg a keverékmodellek között, ismét egy kis zajt adunk a kiválasztási valószínűségekhez. Így az adathalmaz szekvenciáit generáló keverékmodellekben a kiválasztási valószínűségek eltérnek.

Paraméter	**Érték**
Markov-láncok száma	2
Állapotok száma lánconként	$[2, 1]$
Keverési mechanizmus	Probabilisztikus
Speciális szabály a keverésben	A 2 állapotú lánc választási valószínűsége nagy

4. táblázat. A keverékmodell beállításai a dominálo Markov-lánc kísérletben.

A kísérletben a dominálo komponens preferencia súlyát változtattam, és azt vizsgáltam, hogy hogyan befolyásolja az adathalmaz tanulhatóságát.

5.3.1. A dominálo Markov-lánc eredményei



5. ábra. A dominálo Markov-lánc típusú keverék modellek tanításának eredményei. A „keverési arány” értékek azt jelzik, hogy a dominálo Markov-lánc milyen mértékben részesített előnyben a generálásra.

Az 5. ábrán megfigyelhető, hogy a *tanítási loss* mindegyik esetben az optimális becslés közelében van. A nagyobb keverési arányok kisebb *tanítási loss*-t eredményeznek, de ez a különbség minimális. A *teszt loss*-ok is közel hasonló értékeken állnak. A **??**. ábra a futások tanulási görbéjét ábrázolja, és egyértelműen leolvasható a túltanulás esete.

5.4. A teszthalmaz szűkítése

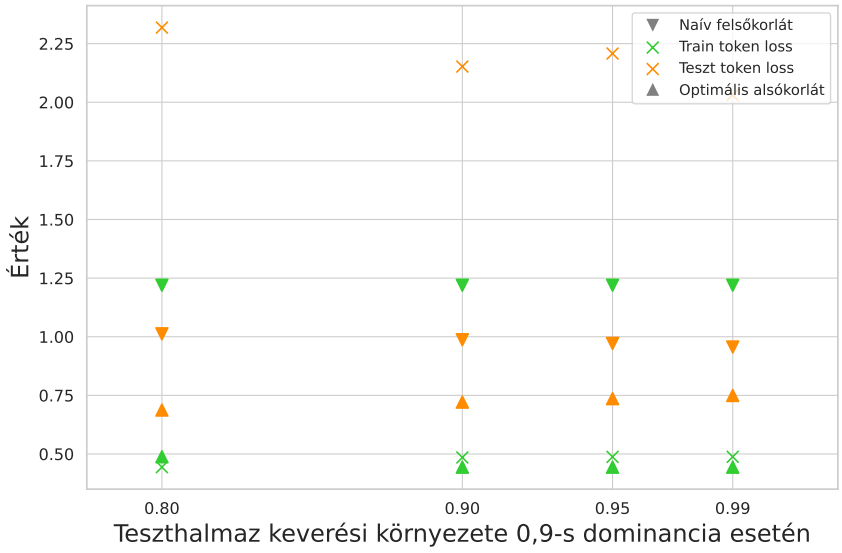
Az eddigi kísérletek alapjánúgy tűnik, hogy a *Transformer* modellek nem képesek olyan általános struktúrákat megtanulni, amelyek a teszt adathalmazon is jól működnek. Ez azonban még nem feltétlen jelenti azt, hogy semmilyen struktúrát nem értettt meg. Elképzelhető, hogy a tanulsás során a tanító adathalmazban szekvenciáinak a szerkezetét megértette, de a teszt adathalmazban látott szekvenciáknak a keverékmodellje nagyon eltér a tanító adathalmazbeliektől. Lehetséges, hogy a tanító adatokhoz hasonló keverékmodellből származó szekvenciákat nagyobb pontossággal tudná megoldani a betanított Transformer modell.

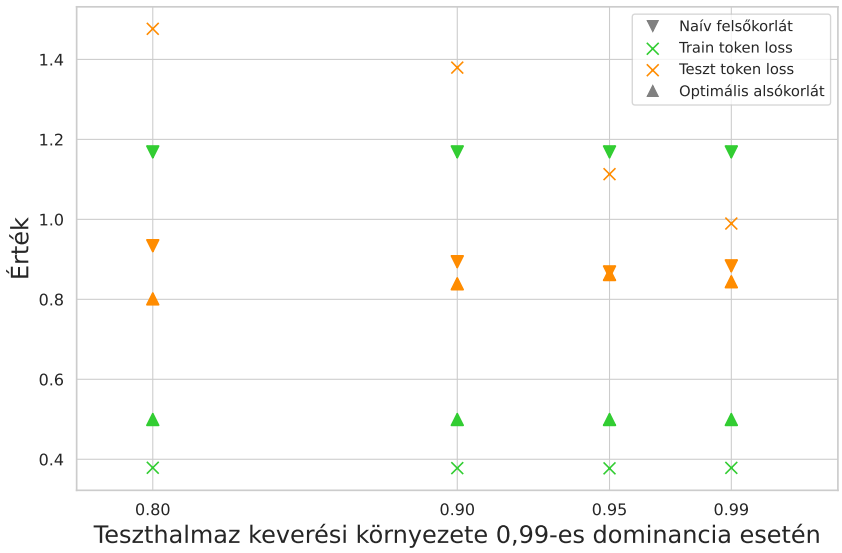
Annak vizsgálatára, hogy a modell képes-e jobban teljesíteni olyan teszthalmazon amely közelebb áll a tanítóhalmazhoz, ebben a kísérletben korlátozzuk a teszt halamaz diverzitását. A teszt adathalamz előállításához egy tanítóhalmazbeli keverékmodellet veszünk és mind a lánc választási valószínűségeket és a komponens Markov-láncok keverési valószínűségeit egy kis zajjal perturbáljuk. Az így kapott teszt szekvenciáink hasonló struktúrájú keverékmodellekből származnak, mint amik a tanítóadathalmazban szerepeltek, de mégsem teljesen azonosak.

Paraméter	**Érték**
Markov-láncok száma	2
Állapotok száma lánconként	$[2, 1]$
Keverési mechanizmus	Probabilisztikus
Speciális szabály a keverésben	A 2 állapotú lánc választási valószínűsége nagy
Speciális szabály a teszt adathalmazra	Egy tanító keverékmodell egy kis zajjal megperturbált változatai generálnak

5. táblázat. A keverékmodell beállításai a teszthalmaz szűkítése kísérletben.

A kísérlet fő paramétere a perturbálás mértéke. A cél annak a feltérképezése, hogy a *Transformer* modellek teljesítménye javul-e, ha a teszt adatokat generáló keverékmodellek szerkezete közelebb áll a tanítóhalmazbeliekéhez. A kísérletet többféle mértékben domináló markovi keverékmodellel is elvégeztem, hogy feltérképezzem, milyen hatással van a dominancia és a teszthalmaz szűkítésének eredményességére.





6. ábra. A teszthalmaz szűkítések eredménye, különböző dominancia értékek mellett.

5.4.1. A teszt adathalmaz szűkítésének eredményei

A 6. ábrán a teszthalmaz szűkítésének a hatása látható, három különböző esetben. A három eset a domináns markovi keverékmodell dominanciájának mértékében tér el egymástól. Az ábrán a zöld színű markerek, a tanítóadathalmazhoz, narancssárgák pedig a tesztadathalmazhoz tartoznak. A „tesztadathalmazok keverési környzete" alatt azt értem, hogy a tesztadathalmaz generálásakor, a keverékmodellek mennyire hasonlítanak hasonlóak a tanító halmazbeli példányhoz. Nagyobb keverési együttható kisebb eltérést, tehát szűkebb környezetet jelent. Az ábrákról több megfigyelés is leolvasható.

Elsőként észrevehető, hogy a tanítási és teszt korlátok eltérnek egymástól. Ennek oka, hogy a teszthalmaz szűkítésével egy sokkal kevésbé diverz adathalmazt hozunk létre. Emellett megfigyelhető, hogy a keverési környezet szükítésével a teszthalmaz naív felső és optimális alsó becslései közelednek egymáshoz.

A tanítási eredményből látszik, hogy a *tanítási loss* mindegyik esetben az optimális alsó becslés alá sülyed, amely egyértelműen a túltanulásra utal. Ezt a értelmezést az is megerősíti, hogy a *teszt lossok* minden esetben a naív felsőbecslés fölött maradnak.

Habár nem érik el a naív felső becslést, a *teszt lossok* a keverési környezet szűkítésével csökkenő tendenciát mutatnak. Ez a trend különösen szembetűnő 0,95-ös és 0,99-es dominanciájú kísérletek esetében. Ebből azt a következtetést vonjuk le, hogy minél nagyobb volt a dominancia mértéke, annál jelentősebben csökken a *teszt loss* a keverési környezet szűkítésével. Továbbá az is megfigyelhető, hogy a *teszt lossok* a dominancia mértékének növekedésével egyre közelebb kerülnek a naív felső korláthoz. Ez a két tendencia abban az esetben éri el a csúcspontját, amikor $0,99$-es a dominancia és a keverési környezet a lehető legszűkebb, ekkor a *teszt loss* szinte eléri naiv felsőbecslést.

Ezek a megfigyelések több következtetést is megengednek. Az első, hogy a dominancia növekedésével a *teszt loss* közelebb kerül a naív felső becsléshez. Ez arra utalhat, hogy a domináns komponens jelenléte ténylegesen segíti a *Transformer* modellt, abban hogy könyebben felismerje és megtanulja a mögöttes struktúrát. Másodszor az a tény, hogy keverési környezet szűkítésével csökken a *teszt loss*, arra utal, hogy a modell jobban boldogul a tanítóhalmazbelihez hasonló struktúrájú szekvenciákkal. Ez annak a jele, hogy valamennyit megérthetett a modell, a keverékmodell szerkezetéből.

Összességében tehát a teszt adathalmaz diverzitásának szűkítése kedvezően hatott a *Transformer* modell teljesítményére, különösen akkor, ha a domináló komponens önmagában erőteljesen meghatározta a szekvencia szerkezetét. Ez arra utal, hogy a modell bizonyos mértékig képes lehet a szekvenciák mögöttes generatív folyamataiból tanulni, de ehhez az is szükséges, hogy a struktúra kellően konzisztens legyen az adathalmazban.

Természetesen adódik tehát a kérdés: Vajon képes-e tanulni a modell, teljesen markovi szekvenciákból tanulni? A következő fejezetben ennek a kérdésnek jártam utána.

5.5. Teljes markovi generálás

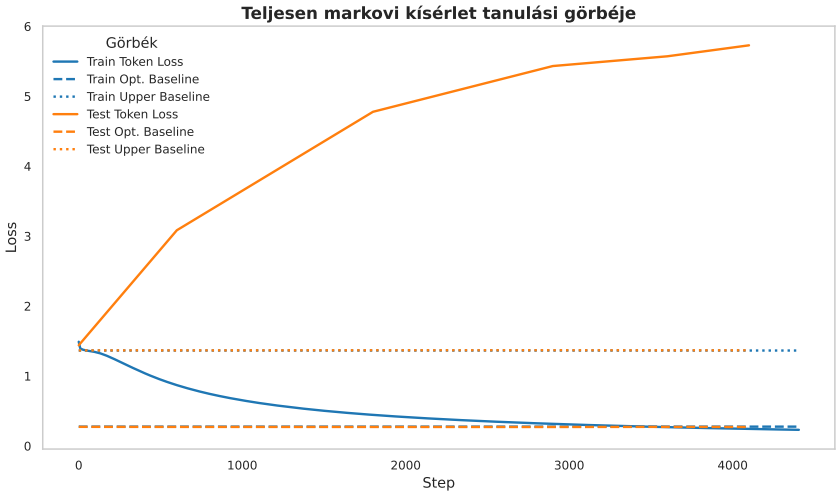
Az 5.3. és 5.4. alfejezetek kísérletei alapján megfigyelhető, hogy ahogy a keverékmodell egyre inkább markovi jellegűvé válik, úgy javult a *Transformer* modellek teljesítménye a teszt adathalmazon. Ugyanakkor az is világossá vált, hogy a Transformer modell még az egyszerű, dominált szerkezetű adathalmazokon is hajlamos túltanulni. A következő lépéseként az egyszerűsítés végső lehetőségét vizsgáltam. Olyan keverékmodelleket alkottam amelyek már csak egyetlen Markov-láncból álltak. Ez a domináló komponens típusú keverékmodellekszéldőséges esete, amikor a szekvencia már csak egyetlen teljesen markovi folyamatot követ. Ebben a kísérletben tehát mát azt vizsgálom, hogy egyszerű markovi folyamatokat sikerül-e a *Transformer* hálónak megtanulnia és a tudást a teszthalmazra általánosítania.

Paraméter	**Érték**
Markov-láncok száma	1
Állapotok száma lánconként	4
Keverési mechanizmus	Nincs

6. táblázat. A keverékmodell beállításai a teljesen markovi kísérletben.

A kísérletnek ebben az esetben nincs paramétere, a fókusz kizárólag a *Transformer* modell teljesítményére irányul. A cél annak megállapítása, hogy a modell képes-e megtanulni egy markovi folyamat szerkezetét, és ezt a tudást új szekvenciákon alkalmazni. Megjegyzendő, hogy ebben a kísérletben a naív felsőkorlát — amely a tapasztalati eloszlás alapján becsül — lényegében a Markov-lánc stacionárius eloszlását közelíti.

5.5.1. A teljes markovi generálás eredményei



7. ábra. A teljesen markovi adathalmazon a *Transformer* modell tanulási görbéje.

A 7. ábrán a *Transfomer* tanítási görbéje látható, amikor a szekvenciákat teljesen markovi folyamat generálja. Az ábra alapján megfigyelhető, hogy a *tanítási loss* folyamatosan és megbízhatóan csökken, ami arra utal, hogy a modell egyre pontosabban képes visszaadni a tanítóadathalmaz szekvenciáit. Ezzel szemben a *teszt loss* már a tanulás korai szakaszától kezdve emelkedni kezd, és a tanulás végére jelentősen meghaladja a kezdeti értékét. Ez a viselkedés arra utal, hogy nem a szekvenciák mögötti generatív struktúrát tanulja meg, hanem egyszerűen „memorizálja" az adott tanító példákat. A teszt teljesítmény romlása azt jelzi, hogy a modell nem képes általánosítani a teljesen markovi folyamat szabályosságaira, annak ellenére, hogy a determinisztikus szerkezet miatt erre számítottunk volna. A túltanulás jelei tehát már korán megjelennek, ami rávilágít arra, hogy a tanítópéldák ismétlődése hozzájárulhat a modell memorikus viselkedéséhez. Ez motiválja a következő kísérletet, amelyben a tanulás minden lépésében újonnan generált adathalmazt használunk, így kizárva az ismétlés lehetőségét.

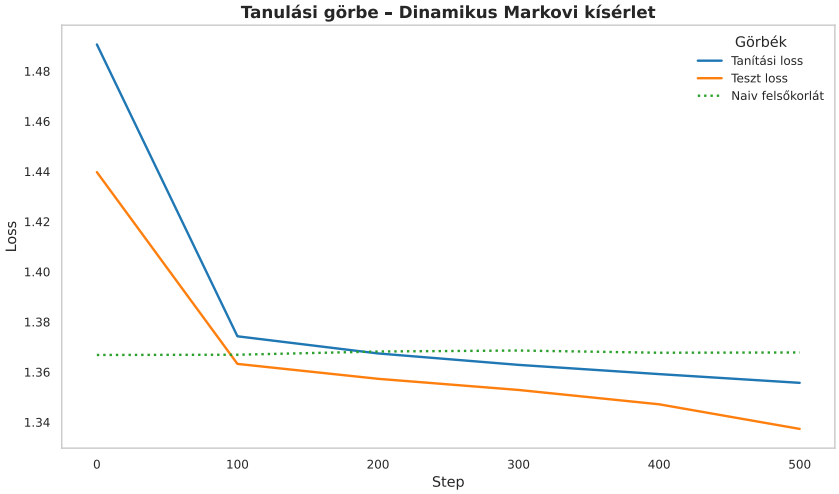
5.6. Dinamikus markovi tanulás

Az utolsó kísérletben továbbra is a teljesen markovi keverékmodelleket vizsgáltam. A 5.5. kísérlet tapasztalatai alapján a *Transformer* modell hajlamos volt túltanulni, és nem jutott el az általánosításig. A túltanulás elkerülésére egy gyakori stratégia a nagyobb tanító adathalmaz használata. Ennek egy szélsőséges esete, amikor minden tanítási körben (epochban) egy teljesen új tanítóadatokat generálunk. Ebben az esetben a modellnek nincs lehetősége túltanulni, hiszen gyakorlatilag sosem találkozik kétszer ugyanolyan példával, így nincs lehetősége azt memorizálni. Ezzel lényegében rákényszerítjük, hogy a szekvencia mögötti szerkezetet ismerje fel és használja.

Paraméter	**Érték**
Markov-láncok száma	1
Állapotok száma lánconként	4
Keverési mechanizmus	Nincs
Speciális tanító adathalmaz szabály	Minden epochban új tanítóadat

7. táblázat. A keverékmodell beállításai a teljesen markovi kísérletben.

A kísérletnek ebben az esetben sincs explicit paramétere. A figyelem középpontjában az a kérdése forog, hogy szélsőségesen nagy adathalmazon képes-e a *Transformer* modell elsajátítani a generáló struktúrát és a tesztadatokon alkalmazni. Amit vizsgálok az lényegében, hogy megtörténik-e az úgynevezett *in-context learning* jelensége.



8. ábra. A dinamikus markovi kísérletnek a tanulási görbéje.

5.6.1. Dinamikus markovi tanulás eredményei

A 8. ábra a dinamikus markovi kísérlet tanulás görbéjét mutatja. Jól látható, hogy mind a *tanítási*, mind a *teszt loss* szinte azonos ütemben csökken a tanítás során. A modell teljesítménye nagyjából a 200. iteráció után a naiv felső becslés szintja alá süllyed, a teszt adathalmazon is. Ez az eredmény arra utal, hogy a *Transformer* modell ebben a beállításban nem pusztán memorizál, hanem valóban képes elsajátítani a teljesen markovi szekvenciák mögötti generatív struktúrát. A *teszt loss* csökkenése, és annak együttmozgása a *tanító losszal*, a *generalizáció* és az *in-context-learning* jelenlétére utal.

5.7. Eredmények összegzése és értelmezése

A kísérletek elején az volt a célom, hogy egy *Transformer* modellt betanítva *következő token predikciós* feladatra megvizsgáljam, vajon a modell mechanizmusában explicit módon megjelenik-e a keverékmodell klaszterezése.

A betanítás kezdeti sikertelensége miatt a fókusz arra tolódott, hogy feltárjam: a keverékmodellek mely tulajdonságai, komponensei okozhatják a tanulási nehézségeket. Ezen az úton haladva több olyan egyszerűsítési lehetőséget is teszteltem, amelyek a struktúrák fokozatos egyszerűsítésével egyértelműen jobb tanulhatóságot eredményeztek.

A javuló tendencia ellenére a *Transformer* modellekben nem figyelhettünk meg valódi általánosítást: a hálók inkább memorizálták a tanítópéldákat, mintsem a mögöttes struktúrát tanulták volna meg.

Végül a tanító adathalmaz szélsőséges mértékű bővítése bizonyult hatékonynak: az utolsó kísérletben a modell már képes volt felismerni és alkalmazni a szekvenciák mögötti struktúrát a *teszt* adatokon is.

5.8. További kísérleti lehetőségek

Bár teljesen markovi adathalmazon sikerült elérni, hogy a modell általánosítható tudást sajátítson el, ennek a folyamatnak a konvergenciája lassú volt. Ez részben annak tudható be, hogy egy viszonylag kisméretű *Transformer* modellt használtam. Valószínűsíthető, hogy nagyobb kapacitású modellek gyorsabban és hatékonyabban tanulnának, és akár a naiv felsőbecslés alatti teljesítmény is elérhetővé válna.

A tanulás lassúságához az is hozzájárulhat, hogy a modell minden egyes iterációban teljesen új tanító adathalmazt látott. Ez ugyan meggátolja a túltanulást, azonban az ismerős minták ismételt megjelenésének hiányában csökken az esélye annak is, hogy a modell fokozatosan építse fel általános tudását.

Ezért érdemes lehet olyan tanítási stratégiát alkalmazni, amelyben nem minden iterációban cseréljük le az egész adathalmazt, hanem csak egy részét.

További lehetőség, hogy az új minták mindig a korábbiak egy szűk környezetéből kerüljenek ki, és így a tanulás strukturáltabb, fokozatosabb lehessen. Ez a megközelítés már közelít a *curriculum learning* szemlélethez, amelyben a modell az egyszerűbb struktúráktól halad a komplexebbek felé.

Miután a modell sikeresen betanulhatóvá válik a *következő token predikciós* feladatra, érdemes elkezdeni a *Transformer* belső működésének mélyebb vizsgálatát is. Például az *attention map*-ek elemzése során feltárható, hogy a modell mely bemeneti tokenekre „figyel” egy-egy predikció során. Ugyanígy érdekes lehet a tokenek belső reprezentációinak elemzése a látens térben: kialakulnak-e klaszterek, és ezek megfeleltethetők-e a keverékmodell komponenseinek?

Összességében ez egy lehetőségekkel teli, izgalmas kutatási irány, ahol még számos további kísérlet és elmélyülés vár a vizsgálódóra.

6. Összefoglaló

Dologzatomban az idősoros modellezés interpretálhatósági kérdéseit vizsgáltam. Bemutattam az lehetséges megközelítéseket, és ezeknek egy csoportosítását. Kiemelten foglalkoztam egy speciális struktúrájú idősoros modellel: a Markovi keverékmodellekkel. Felvezettem ezek egy lehetséges interpretációs megközelítését, a szekvencia szétválasztásának feladatát, valamint röviden bemutattam az irodalomból ismert klasszikus megoldást. Ezt követően ismertettem, majd ennek vizsgálatába kezdtem. A módszer implementálása során szembesültem az ilyen típusú szekvenciák predikciójának nehézségeivel. Ez vezetett a keverékmodellek tanulhatósági problémáinak mélyebb elemzéséhez. A vizsgálatok során különböző egyszerűsítési lépéseken keresztül kerestem olyan modellverziót, amely már sikeresen megtanulható a *Transformer* architektúra számára. Végezetül további kísérleti irányokat is bemutattam, amelyek a jövőben izgalmas és ígéretes kutatási lehetőségeket

kínálnak ezen a területen.

Irodalom

[1] Tuğkan Batu, Sudipto Guha és Sampath Kannan. „Inferring Mixtures of Markov Chains". *Learning Theory*. Szerk. John Shawe-Taylor és Yoram Singer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, 186–199. old. ISBN: 978-3-540-27819-1.

[2] Tom B. Brown és tsai. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[3] Timothée Darcet és tsai. „Vision Transformers Need Registers." *International Conference on Learning Representations*. 2024.

[4] Aaron Grattafiori és tsai. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[5] Scott M. Lundberg és Su-In Lee. „A unified approach to interpreting model predictions". *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 4768–4777. old. ISBN: 9781510860964.

[6] André Maletzke és tsai. „Time Series Classification using Motifs and Characteristics Extraction: A Case Study on ECG Databases". 2013. okt. ISBN: 978-90-78677-86-4. DOI: 10.2991/.2013.40.

[7] Ariana Minot és Yue M. Lu. „Separation of interleaved Markov chains". *2014 48th Asilomar Conference on Signals, Systems and Computers*. 2014, 1757–1761. old. DOI: 10.1109/ACSSC.2014.7094769.

[8] Alec Radford és tsai. „Language Models are Unsupervised Multitask Learners". (2019).

[9] Ramprasaath R. Selvaraju és tsai. „Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". *International Journal of Computer Vision* 128 (2016), 336–359. old. URL: https://api.semanticscholar.org/CorpusID:15019293.

[10] Karen Simonyan, Andrea Vedaldi és Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2014. arXiv: 1312.6034 [cs.CV]. URL: https://arxiv.org/abs/1312.6034.

[11] Gemini Team és tsai. *Gemini: A Family of Highly Capable Multimodal Models*. 2025. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[12] Andreas Theissler és tsai. „Explainable AI for Time Series Classification: A Review, Taxonomy and Research Directions". *IEEE Access* 10 (2022), 100700–100724. old. DOI: 10.1109/ACCESS.2022.3207765.

[13] Ashish Vaswani és tsai. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[14] Lexiang Ye és Eamonn Keogh. „Time series shapelets: a new primitive for data mining". *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: Association for Computing Machinery, 2009, 947–956. old. ISBN: 9781605584959. DOI: 10.1145/1557019.1557122. URL: https://doi.org/10.1145/1557019.1557122.

Nyilatkozat sablon – MI eszközök használatáról

Alulírott **Borbély Bernárd** nyilatkozom, hogy szakdolgozatom elkészítése során az alább felsorolt feladatok elvégzésére a megadott MI alapú eszközöket alkalmaztam:

Feladat	**Felhasznált eszköz**	**Felhasználás helye**	**Megjegyzés**
Irodalomkutatás	Deep Research	3.2-es és 4.5-ös alfejezetek	
Ábra készítés	GPT-4o	5.fejezet	Python kód
Szövegvázlat készítés	GPT-4o	4. és 5. fejezet	
Nyelvhelyesség	Writefull	Teljes dolgozat	

A felsoroltakon túl más MI alapú eszközt nem használtam.