
Production Systems and Information Engineering
Volume 11 (2), pp. 33–42
doi: 10.32968/psaie.2023.2.3

TANDEM RÉSZSZTRINGEK DETEKTÁLÁSA NEURÁLIS HÁLÓ ALKALMAZÁSÁVAL

László Kovács
University of Miskolc, Hungary
Institute of Information Technology
laszlo.kovacs@uni-miskolc.hu

Dávid Polonkai
University of Miskolc, Hungary
Institute of Information Technology
david.polonkai2@gmail.com

Abstract. A sztringek és a listák alapvető szerepet játszanak az adatstruktúrák között. A sztringekhez kapcsolódóan az egyik központi művelet az egy adott mintára illeszkedő részek keresése. A részsztring mintakeresés egyik speciális típusa az ismétlődő részek meghatározása. Ez a feladat kiemelkedő fontosságú többek között a biológiában (gének vizsgálata) vagy a folyamatmodellezésben (ciklusok keresése). A cikk a hagyományos, direkt mintakeresési módszerek egy sajátos alternatíváját mutatja be, amely a neurális hálók alkalmazásán alapszik. A kidolgozott módszerrel végzett összehasonlító elemzések azt mutatják, hogy a neurális háló alapú módszerek elsősorban a közelítő keresések esetén bizonyulnak hatékony megoldásnak.

Keywords: mintaillesztés, részszting keresés, neurális hálók

1. Bevezetés

A sztringek több területen játszanak alapvető szerepet, mint például a génkutatásban [1] , a szabadszöveges dokumentumok elemzésében vagy a folyamatbányászatban [2]. A szekvenciák esetében a minta alapú rész keresés mellett kiemelt szerepű az ismétlődések feltárása is. A szövegben egy szakasz ismételt megjelenésének azon esetét, amikor a szakasz duplikálódik, tandem részsztringeknek is nevezik. A tandem résztringek feltárása több különböző módon és eszközzel valósulhat meg. A hagyományos illesztés alapú megoldásoknak is több típusa létezik, kezdve a naív módszertől a hatékony rekurzív módszerekig széles a módszer paletta. Ebben a palettában viszont nem találunk neurális hálós módszereket. A jelen cikk fő feladata a neurális háló alapú

megoldások lehetőségeinek az elemzése és ennek keretében egy megoldási változat bemutatása. A javasolt rendszer a sztringet egy kétdimenziós mátrixra vetíti le, majd egy CNN háló segítségével osztályozza ezen képet, hogy hol tartalmaz ismétlődő szakaszokat a mátrix.

A feladat formálisan egy adott ABC halmaz A esetén a következőkép fogalmazható meg. Az ABC eleminek egy véges sorozatát nevezzük sztringnek:

$$s = a_1, a_2, ..., a_m, a_i \in A$$

Egy

$$s' = a'_1, a'_2, .., a'_k$$

sztring részhalmaza s-nek, ha

$$\exists i : a_i = a'_1, ..., a_{i+j-1} = a'_j, ..., a_{i+k-1} = a'_k$$

Egy s' részszting tandem részsztring, ha

$$\exists i : a_i = a_{i+k} = a'_1, ..., a_{i+j-1} = a_{i+k+j-1} = a'_j, ..., a_{i+k-1} = a_{i+2*k-1} = a'_k.$$

2. Mintaillesztés alapú módszerek

Az ismétlődő részsztring szakaszok meghatározásának triviális módszere egy egymásba ágyazott ciklus, melyben a külső ciklus a ciklusmag kezdő pozícióját hordozza, a belső ciklus a mag hosszát jelöli. Ez alapján egy N hosszú sztringnél $O(N^2)$ ciklusmag jelölt kerül tesztelésre. A tesztelés során megnézzük, hogy ismétlődik-e az aktuális mag jelölt. A teljes vizsgálat í azgy $O(N^3)$ költségig is felmehet. Mivel az egyes biológiai alkalmazásokban igen hosszú szekvenciákat kell vizsgálni és a hosszabb sztringek esetén a triviális megoldás már nem alkalmazható, több javaslat is született a ciklusmagok hatékonyabb detektálására.

Az egyik első megoldás az időbeli hatékonyság növelésére a [3] publikációban bemutatott javaslat. A kidolgozott algoritmus az induló sztinget elemi darabokra bontja. Ezt követően a szomszédos részek összefűzésével hosszabb részekre lépünk tovább. A kidolgozott CONCATENATE függvény ezen részsztringeket átvizsgálva a fellelt ismétlődő mintákat begyűjti. A módszer végrehajtási költsége $O(NlogN)$.

Napjainkban a leghatékonyabb algoritmusok már $O(N)$ költségű megoldást is el tudnak érni [4] [5]. Az algoritmus egy rekurzív feldolgozást végez a beérkező sztringen. Az első szinten kettébontja a sztringet, majd meghatározza, hogy van-e olyan részsztring ismétlődés, amely metszi a felvett vágásvonalat.

Az ilyen típusú tandem sztringek feltárása után a két sztring féltéren különkülön rekurzívan elvégzi az oda tartozó tandem részek feltárását.

3. CNN alapú módszer

A CNN neurális hálók alkalmazása 1980-as években fejlődött ki [6]. A háló szerkezetét és működését tekintve biológiai eredetű [7]. A CNN azaz Konvolúciós Neurális Hálók a hagyományos hálók közötti különbség, hogy a hagyományos neuronrétegeken kívül úgynevezett konvolúciós rétegeket is tartalmaznak. A konvolúciós rétegek szűrő ablakokat visznek végig a mátrixon. A szűrőablak tekinthető egy $n \times n$-es mátrixként, amit elhelyezünk a kiszámítandó adat mátrixán. A művelet következő lépésében az egyes értékeket összeszorozzuk, majd összeadjuk őket. Ezt követően az ablakmátrixot eltoljuk egy egységgel. A műveletsort megismételjük a teljes adatmátrixra. Ezen ablakmátrix mérete tetszőleges lehet. A tanítás előtt a szűrőablak értékei tetszőlegesek. A tanítási folyamat alatt a szűrőablak értékei úgy változnak, hogy megtanuljanak kiemelni egy lényeges információt a mátrixból. A konvolúciós rétegeket általában egy pooling réteg követi, amely a konvolúció által készített mátrix dimenzióját csökkenti valamilyen módon. A konvolúcióhoz hasonlóan egy meghatározott ablakmérettel végighalad a mátrixon és az ablakban található értékeket képzi le valamilyen függvény segítségével, általában a maximum kiválasztás vagy az átlagolás használatos a CNN-ek esetében. Napjainkban egyre több területen alkalmazzák a konvolúciós neurális hálókat. A legnagyobb felhasználási területe a képfeldolgozás [8][9]. Azonban hatékonyságuk olyan magas a képfeldolgozásban, hogy gyakran más problémakörökre is alkalmazzák az adatok képpé transzformálásával [10]. Jelen cikkben ezzel a problémával is fogunk foglalkozni, hiszen az adataink lineáris sztringszekvenciák.

A neurális háló tanításához nagy mennyiségű példaadatra van szükség, ezért egy adathalmaz generátort készítettünk. A generátor feladata olyan sztringek generálása, amelyben található ismétlődés, és amelyekben nem jelenik meg ismétlődés. Ismétlődés alatt azonban nem csak az egyetlen karakter ismétlődését kell generálnunk, hanem a több karakterből álló ismétlődő alszekvenciákat is készítenünk kell a CNN megfelelő minőségű jóslása érdekében. Azonban nem lehet tetszőlegesen hosszú ismétlődő szakaszokat választani, ha az ismétlődést tartalmazó sztringrész hossza adott. Egy $s, m = |s|$ sztring esetében, csak abban az esetben lehet egy j hosszú tandem részsztring helyes, ha az ismétlődő részsztring hossza i és igaz, hogy $j \mod i = 0$ és $j < m$. A generátor, ezért mindenek előtt a megadott m esetén kiszámítja a lehetséges j és i értékeket. Ezt követően egy próbadobás segítségével a generátor eldönti, hogy ismétlődést tartalmazó sztringet, vagy ismétlődés mentes sztringet generál az adott ciklusban.

Ha ismétlődés menteset, akkor úgy generáljuk a karaktereket, hogy ne lehessen véletlen ismétlődés. Ha ismétlődő résszel rendelkező sztringet, először meghatározzuk az ismétlődés pozicióját és az ismétlődő rész hosszát, majd az ismétlődés pozíciójáig az előző módon generáljuk a szekvenciát. Az ismétlődő részt elérve kiválasztjuk az ismétlődő részsztringben használatos karakter szekvenciát és egymás után összefűzünk $\rfloor \frac{j}{i}$ darabot a karakterszekvenciából. A maradék részét a sztringnek ismétlődés mentesen generáljuk.

3.1. Előfeldolgozás

Mivel jelen cikkben a tandem részsztringek megtalálása a feladatunk, amelyek lineáris egydimenziós adatok, ezeket át kell alakítanunk CNN számára értelmezhető mátrix formátumra. Az előfeldolgozással, mátrixos formátumra alakítással kapcsolatos követelmények:

- A tandem részsztringek jellegzetesen jelenjenek meg a mátrixon.
- A bemenetet minél kisebb dimenziójura csökkentsük.
- A neurális háló egy hosszúságú sztringben minden lehetséges tandem részsztringet megtalálhasson.

Az általunk használt megoldás egy egyértelmű leképzést jelent. A teljes $s = a_1, a_2, \ldots, a_m$; $a_i \in A$ sztringet egy $m \times m$ mátrixxá alakítjuk. Az M egyezés mátrixban egy elemet az $i, j \in \mathbb{N}_0$ számokkal azonosítunk. Az $M[i, j] = 0$, ha $a_i \neq a_j$, $M[i, j] = 1$, ha $a_i = a_j$.
Például (1 táblázat):

	a	l	a	l
a	1	0	1	0
l	0	1	0	1
a	1	0	1	0
l	0	1	0	1

	a	l	m	a
a	1	0	0	1
l	0	1	0	0
m	0	0	1	0
a	1	0	0	1

Table 1. Egyezés mátrix

Ha az előbb bemutatott táblázatot képként fogjuk fel (1 ábra):

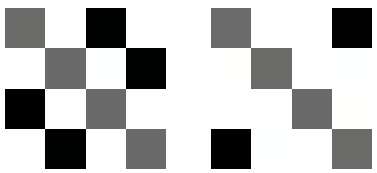


Figure 1. Egyezés mátrix képként

Az ismétlődés a képen az főátlóval párhuzamos pixelsorként jelenik meg. Ezen megoldás segítségével a CNN könnyedén felismerheti az ismétlődéseket. A mátrix előállítását a főátlóra való szimmetrikusságának kihasználásával lehet optimalizálni, így a számítási költség a felére csökken.
A probléma azonban, hogy egy m hosszú sztring esetében m^2 lesz a neurális háló bemeneti neuronjainak száma. Így viszonylag hamar elérjünk a hardveres korlátainkat a CNN tanítása vagy ismétlődés keresés alatt. A probléma megoldására különböző redukciós megoldásokat dolgoztunk ki:

- Bitmap
- Átlag
- Reduct

A Bitmap egy olyan leképzés, amely a mátrixban egy $n \times n$ almátrixot, ahol $n < m$, egy valós számmal helyettesítünk. Az $n \times n$-es mátrixot, mint n^2 hosszúságú szekvenciát fogjuk fel, és a belőlük megalkotott bináris szám 10-es számrendszer beli értékét elosztjuk a $2^{n^2} - 1$ számmal. Az így kapott 0 és 1 közötti értéket adjuk a neurális hálónak. Ez a megoldás azért ideális, mert így visszakaphatjuk az eredeti értéket is. Mégis képesek vagyunk vele $n = 2$ esetén a felére csökkenteni a memóriaigényét a neurális hálónak. Például adott az alábbi $n = 2$ almátrix $\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$, melyet 1001-ként fogjuk fel, ami 10-es számrendszerben 9. A behelyettesítendő érték pedig $9/(2^{2^2} - 1) = 0.6$.

Az Átlag módszer a szekvencia mátrixból egy $n \times n$-es almátrix értékeit átlagolja, majd a mátrixban az átlag értékével helyettesíti az $n \times n$-es részt. A feljebb bemutatott példa esetében ez az érték $\frac{1+0+1+0}{4} = 0.5$. Ezzel a módszerrel is n-ed részére csökkenthetjük a memóriaigényét a rétegeknek.
Mind az átlagolásos, mind a bitmap alapú módszernél a mátrixból nem átlapoltan vesszük a mintákat, mint a hagyományos konvolúció vagy pooling esetében, hanem a mintavételezési ablak ablakméretének megfelelő egységet fog mozogni minden irányba.

A reduct módszer esetében a hasonló szomszédokat keressük meg és nagyobb értéket adunk a mátrixban azoknak a csoportosításoknak, ahol az almátrix főátlója 1 értékű.

4. Kísérletek

A cikk célja összehasonlítani a naív tandem részszring keresést és a CNN alapú tandem részszring keresést. Jelen esetben a következő tényezőket vizsgáltuk:

- Pontosság
- Megoldáshoz szükséges idő
- Erőforrás igény

Fontos megjegyezni, hogy a megoldáshoz szükséges idő alatt a CNN jóslási idejét tekintjük.

4.1. CNN architektúra paraméterezése

Az cikkben bemutatott módszer által használt konvolúciós neurális háló egyszerű szerkezetű. A háló elkészítéséhez az egyszerűség és a számunkra szükséges könyvtárak elérhetősége miatt a python nyelvet választottuk. A keras keretrendszer segítségével sikerült könnyedén felépíteni a neurális hálót és tanítani azt. Bemeneti rétegeinek a száma m^2. 3 konvolúciós egység követi egymást, amelyet teljesen kapcsolt (dense) rétegek követnek. A konvolúciós egységek a $Conv2D$ rétegeket használják fel. Ez a réteg egyszerre több szűrőt használ fel a képen. Az első konvolúciós egységben a szűrők száma 64, a másodikban 128, majd az utolsó egységben 256. Az ablakméret minden esetben 3×3-mas. A felhasznált aktivációs függvény a konvolúciós egységekben a LeakyReLU, melynek képlete:

$$f(x) = \begin{cases} \alpha * x \text{ ha } x < 0 \\ x \text{ ha } x \leq 0 \end{cases}$$

Esetünkben $\alpha = 0.1$. Ezt követi egy konvolúciós egységben egy 2×2-es méretű maximum pooling, amely csökkenti az adat dimenzióját. Végül két $Dense$ réteg segítségével hozzuk létre a megoldást. Veszteségfüggvénynek a kereszttentrópiát használtuk, softmax aktivációs függvénnyel. A tanító függvény az Adam a cikkben bemutatott háló esetében. A 2 példában látható neurális háló bemeneti sztring hossza $m = 150$ volt. Látható, hogy a konvolúció miatt

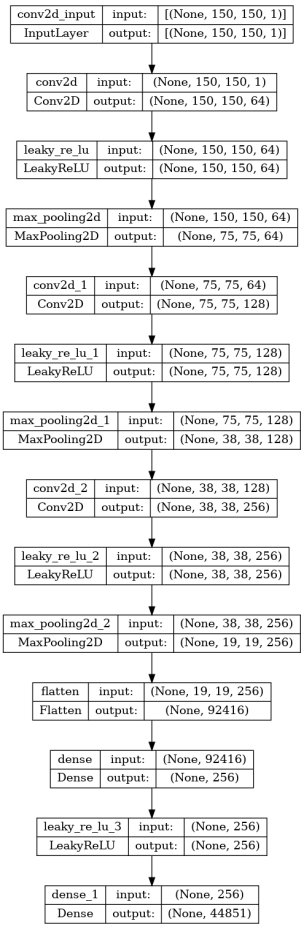


Figure 2. CNN model

a belső rétegek száma is nagy marad, ezért is fontos a redukció az előfeldolgozás esetében.

4.2. Futtatási konfiguráció

A teszteket egy Nvidia GPU-val szerelt gépen futtatuk 16GB videómemoriával. A naív tesztek a CPU-n futottak.

5. Teszt eredmények

Minden tesztesetet 3 szor futtatunk le, majd átlagoltuk az eredményeket. A tesztek 40 és 100 hosszúságú sztringek esetében 10000 sztringből álltak, a többi esetben 5000 sztringből. A tanító halmazból 80%-ot használtunk fel tanításra,

a maradék 20%-ot a hálók pontosságának vizsgálatára, és a megoldási idők vizsgálatára használtuk. Epochok szempontjából több epochal futtatuk a kísérletet:

- 20
- 40
- 100
- kilépési feltételes futtatás (Early Stopping)

A kísérleteket lefuttatuk redukciós módszerekkel és anélkül is. A sztringek hosszai a következőképp alakultak:

- 40
- 100
- 200
- 300
- 400
- 500
- 600

A teszt eredményeket a 3,4,5 ábrákban foglaltuk össze.

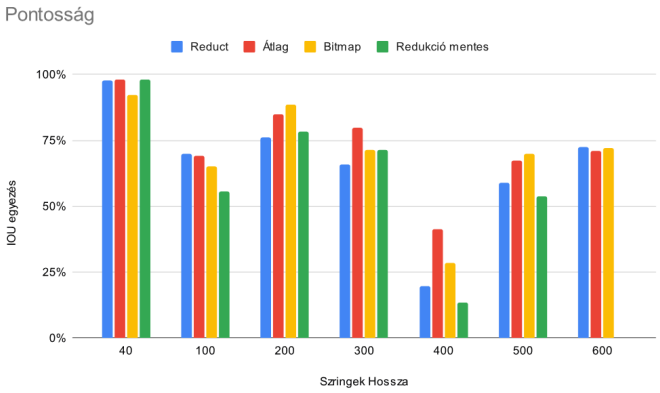
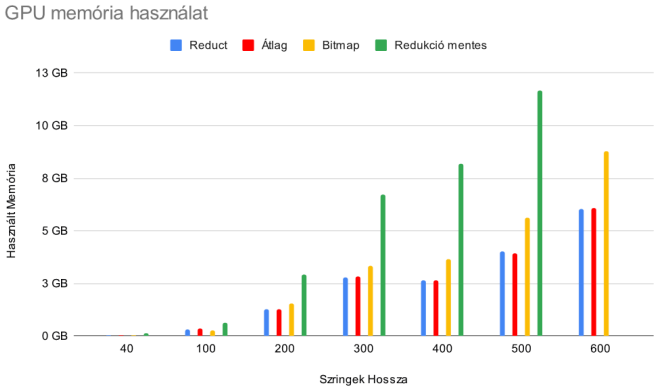


Figure 3. Pontosság

Több ábrán nem látható a redukció mentes futás a 600 hosszúságú sztringek esetében, ennek oka a videómemoriából való kicsordulás. Látható, hogy a neurális háló pontossága meglehetősen jó. A sztringek hossza nem jelentősen befolyásolja a pontosságot a CNN esetében, még így is viszonylag távol vagyunk a 100%-os pontosságtól, de nagy közelséggel megtalálja a CNN a tandem részsztringet. Erőforrásigény tekintetében a redukcióval sikeresen megfeleztük



usage.png

Figure 4. GPU memória használat

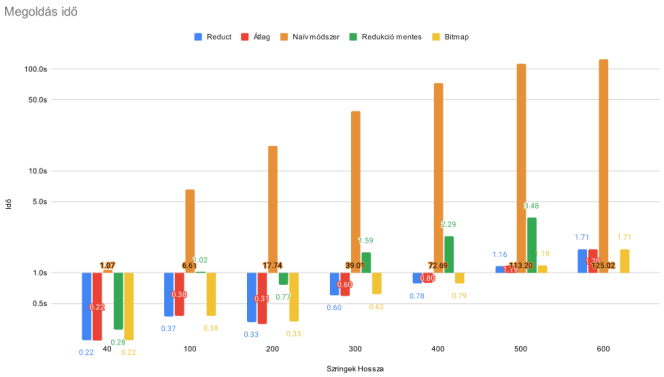


Figure 5. Megoldási idő

az erőforrás igényeket az egyes futtatások folyamán és a pontoság nem csökkent. A megoldási idő tekintetében körülbelül 100-as nagyságrendű a különbség. Míg a neurális hálós módszer 1 másodpercet igényelt, a naív módszer 100 környékit. Az epochok számát tekintve érdemes megemlíteni, hogy a 26. epochnál megállt a tanítás azoknál a futtatásoknál, amelyeknél kilépési feltételt alkalmaztunk.

6. Összegzés

tandem részsztring keresés egy fontos művelet a biológia, kémia és a folyamatelemzés területén. Ugyan napjainkban a klasszikus illesztés alapú módszerek dominálják a területet, érdekes kérdés, hogy mennyire lehetnek hatékonyak a gépi tanulási módszerek ennél a feladatnál. A cikkben ezen vizsgálat keretében

bemutatunk egy javasolt CNN alapú architektúrát a tandem részsztringek feltárására. A kidolgozott módszerrel végzett összehasonlító elemzések azt mutatják, hogy a neurális háló alapú módszerek működőképesek és elsősorban a közelítő keresések esetén bizonyulnak hatékony megoldásnak.

References

[1] J. G. Chen andC. T. Lee, C. T.: Finding all tandem arrays in dna sequences. In *23rd Workshop on Combinatorial Mathematics and Computation Theory*, 2006.

[2] Dakic, D., Sladojevic, S., Lolic, T., and Stefanovic, D.: Process mining possibilities and challenges: a case study. In *IEEE 17th International Symposium on Intelligent Systems and Informatics*, 2019, pp. 161–166, URL https://doi.org/10.1109/SISY47553.2019.9111591.

[3] Main, M. and Lorentz, R.: An o(n log n) algorithm for finding repetitions in a string. *Journal Algorithms*, **5**, (1984), 422–432.

[4] Amir, e. a., Amihood: Repetition detection in a dynamic string. In *27th Annual European Symposium on Algorithms*, 2019.

[5] Crochemore, M., Iliopoulos, C. S., Kubica, M., Radoszewski, J., W.Rytter, and Walen., T.: Extracting powers and periods in a word from its runs structure. *Theor. Comput. Sci.*, (521), (2014), 29–41, URL https://doi.org/10.1016/j.tcs.2013.11.018.

[6] Fukushima, K. and Miyake, S.: Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982, URL https://doi.org/10.1007/BF00344251.

[7] Fukushima, K.: Neocognitron. *Scholarpedia*, **2**(1), (2007), 1717, URL http://dx.doi.org/10.4249/scholarpedia.1717.

[8] Denker, J., Gardner, W., Graf, H., Henderson, D., Howard, R., Hubbard, W., Jackel, L. D., Baird, H., and Guyon, I.: Neural network recognizer for hand-written zip code digits. *Advances in neural information processing systems*, **1**.

[9] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[10] Sharma, A., Vans, E., Shigemizu, D., Boroevich, K. A., and Tsunoda, T.: Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific reports*, **9**(1), (2019), 1–7, URL https://doi.org/10.1038/s41598-019-47765-6.