

UNIVERZITA J. SELYEHO – SELYE JÁNOS EGYETEM
FAKULTA EKONÓMIE A INFORMATIKY–
GAZDASÁGTUDOMÁNYI ÉS INFORMATIKAI KAR

APLIKÁCIA HOPFIELDOVÝCH SIETÍ-
HOPFIELD HÁLÓZATOK ALKALMAZÁSA

Diplomová práca -Diplomamunka

UNIVERZITA J. SELYEHO – SELYE JÁNOS EGYETEM
FAKULTA EKONÓMIE A INFORMATIKY–
GAZDASÁGTUDOMÁNYI ÉS INFORMATIKAI KAR

APLIKÁCIA HOPFIELDOVÝCH SIETÍ-
HOPFIELD HÁLÓZATOK ALKALMAZÁSA

Diplomová práca -Diplomamunka

Študijný program:	Aplikovaná Informatika
Tanulmányi program:	Alkalmazott Informatika
Názov študijného odboru:	18. informatika
Tanulmányi szak megnevezése:	18. informatika
Vedúci záverečnej práce:	prof. RNDr. Tibor Kmet', CSc.
Témavezető:	prof. RNDr. Kmet' Tibor, CSc.
Školacie pracovisko:	KINF - Katedra informatiky
Tanszék megnevezése:	KINF - Informatikai Tanszék
Autor:	Bc.František Ribní
Szerző:	Bc.Ribní František

Komárno, 2024

Čestné vyhlásenie – Becsületbeli nyilatkozat

Čestne vyhlasujem, že som svoju diplomovú prácu vypracoval samostatne na základe svojich vedomostí s použitím uvedenej literatúry pod odborným vedením vedúceho práce. Som si vedomý, že plagiáty univerzita neakceptuje, zaväzuje ma na vypracovanie novej témy a v danom akademickom roku stratím možnosť absolvovať príslušnú časť štátnych skúšok.

Alulírott, Ribní František kijelentem, hogy a diplomadolgozatomat önállóan dolgoztam ki, az abban feltüntetett forrásokból merítve, a témavezetőm szakmai irányításával. Tudatában vagyok annak, hogy a plagizált dolgozatot az egyetem visszautasítja, új téma kidolgozására kötelez, és az adott tanévben megfoszt az államvizsga ezen része teljesítésének lehetőségétől.

Trstice, 7. 5. 2024

.....

Nádszeg, 2024. 5. 7.

Bc. František Ribní

Köszönetnyilvánítás

Köszönetet szeretnék nyilvánítani prof. RNDr. Kmet' Tibor, CSc. professzor úrnak, amiért a diplomamunkám témavezetését vállalta, javaslataival, illetve szakirodalmak biztosításával hozzájárult annak elkészítéséhez. Ugyanakkor a családomnak is szeretném megköszönni, hogy támogattak egyetemi tanulmányaim alatt.



Univerzita J. Selyeho
Fakulta ekonómie a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. František Ribní
Študijný program: Aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 18. - informatika
Typ záverečnej práce: Diplomová práca
Jazyk záverečnej práce: maďarský
Sekundárny jazyk: slovenský
Téma: Hopfield hálózatok alkalmazásai
Anotácia: A diplomamunka elméleti részében átfogó elemzést tartalmaz a Hopfieldhálózatokról, és azok alkalmazási területeiről, illetve a Hopfield-hálózatok alapelveit és működési mechanizmusát tárgyalja. Ez magában foglalja a neurális hálózatok fejlődésének rövid áttekintését is. A diplomamunka második, gyakorlati része a Hopfield-hálózatok alkalmazási területeivel foglalkozik: mint például a mintázatfelismerés és a kombinatorikus problémák megoldása. Az implementált alkalmazásokat a szerző Matlab- környezetben valósítja meg. A diplomamunka eredményeit és következtetéseit a Hopfield- hálózatok hatékonyságára, alkalmazási lehetőségeire és korlátjaira vonatkozóan foglalja össze. A diplomamunka végén javaslatokat tesz a Hopfield-hálózatok továbbfejlesztésére és alkalmazásainak bővítésére.
Kľúčové slová: Hopfield-hálózatok és alkalmazási területeik, Matlab
Vedúci: prof. RNDr. Tibor Kmet', CSc.
Katedra: KINF - Katedra informatiky
Vedúci katedry: Ing. Ondrej Takáč, PhD.
Dátum schválenia: 12.06.2023

prof. RNDr. Tibor Kmet', CSc.

osoba zodpovedná za realizáciu študijného programu

Abstrakt

František Ribník: *APLIKÁCIA HOPFIELDOVÝCH SIETÍ*. [Diplomová práca] Univerzita J. Selyeho. Fakulta ekonómie a informatiky, Katedra matematiky a informatiky, 2.ročník, Stupeň odbornej kvalifikácie: *Magister*, Študijný odbor: *Aplikovaná informatika*. Vedúci: prof. RNDr. Tibor Kmeť, CSc., Komárno, 2024

Teoretická časť práce obsahuje komplexnú analýzu *Hopfieldových* sietí a ich aplikácií a rozoberá základné princípy a mechanizmy fungovania *Hopfieldových* sietí. Obsahuje aj stručný prehľad vývoja neurónových sietí. Druhá, praktická časť práce sa zaoberá aplikáciami *Hopfieldových* sietí, napríklad rozpoznávaním vzorov a riešením kombinatorických problémov. Autor implementuje aplikácie v prostredí *Matlab*. V práci sú zhrnuté výsledky a závery o účinnosti, aplikačnom potenciáli a obmedzeniach *Hopfieldových* sietí. V závere práce sú uvedené návrhy na ďalší rozvoj *Hopfieldových* sietí a rozšírenie ich aplikácií.

Kľúčové slová: *Hopfieldove siete a ich aplikácie, Matlab*

Absztrakt

Ribnı František: *HOPFIELD HÁLÓZATOK ALKALMAZÁSA*. [Diplomamunka] Selye János Egyetem. Gazdaságtudományi és Informatikai Kar, Matematika és Informatika Tanszék, 2.évfolyam, Szakmai képesítési szint: *Magiszter*, Tanulmányi szak: *Alkalmazott informatika*. Témavezető: prof. RNDr. Kmet' Tibor, CSc., Komárno, 2024

A diplomamunka elméleti részében átfogó elemzést tartalmaz a *Hopfield*-hálózatokról, és azok alkalmazási területeiről, illetve a *Hopfield*-hálózatok alapelveit és működési mechanizmusát tárgyalja. Ez magában foglalja a neurális hálózatok fejlődésének rövid áttekintését is. A diplomamunka második, gyakorlati része a *Hopfield*-hálózatok alkalmazási területeivel foglalkozik: mint például a mintázatfelismerés és a kombinatorikus problémák megoldása. Az implementált alkalmazásokat a szerző *Matlab*- környezetben valósítja meg. A diplomamunka eredményeit és következtetéseit a *Hopfield*- hálózatok hatékonyságára, alkalmazási lehetőségeire és korlátjaira vonatkozóan foglalja össze. A diplomamunka végén javaslatokat tesz a *Hopfield*-hálózatok továbbfejlesztésére és alkalmazásainak bővítésére.

Kulcsszavak: *Hopfield*-hálózatok és alkalmazási területeik, *Matlab*

Abstract

František RibnÍ: *APPLICATION OF HOPFIELD NETWORKS*. [Master thesis] Selye János University. Faculty of Economics and Informatics, Department of Mathematics, and Informatics, second year, Professional qualification level: Master, Study course: Applied Informatics. Supervisor: prof. RNDr. Tibor Kmet', CSc., Komárno, 2024

The theoretical part of the thesis contains a comprehensive analysis of Hopfield networks and their applications, and discusses the basic principles and operating mechanisms of Hopfield networks. It also includes a brief overview of the development of neural networks. The second, practical part of the thesis deals with applications of Hopfield networks, such as pattern recognition and solving combinatorial problems. The author implements the applications in a Matlab environment. The thesis summarizes the results and conclusions on the efficiency, application potential and limitations of Hopfield networks. At the end of the thesis, suggestions are made for further development of Hopfield networks and for extending their applications.

Keywords: Hopfield networks and their applications, Matlab

Tartalomjegyzék

Bevezetés.....	13
Munka célja.....	14
1 A neurális hálózatok technológiai forradalma	15
1.1 Visszatekintés a múltba	15
1.2 Az ezredforduló és napjaink.....	19
2 A Hopfield hálózatok részletes áttekintése	22
2.0.1 Hebb tanulási szabálya	22
2.0.2 Gyakorlati példa Hebb tanulásra.....	24
2.1 Demisztifikálás.....	26
2.1.1 Diszkrét hálózatok	26
2.1.2 Folytonos hálózatok.....	26
2.1.3 Architektúra	26
2.1.4 Energiafüggvény a Hopfield hálózatban.....	26
2.1.5 Tanulási modell Hopfield hálózattal	27
3 A következő generáció irányítói: Hopfield hálózatok a XXI. században	29
3.1 Alacsony költségű digitális képalkotó és feldolgozó rendszer	29
3.1.1 Implementálás.....	31
3.2 Hopfield kaotikus neurális hálózaton alapuló színes képtitkosítási algoritmus....	36
3.2.1 Titkosítási folyamat lépései	37
3.2.2 Legfontosabb megállapítások	38
3.3 Kemoinformatika és a Hopfield hálózatok	39
3.4 Végezetül, de nem utolsó sorban	41
3.4.1 Hopfield hálózat, mint optimalizáló algoritmus	41
3.4.2 Kapacitás növelése?	42
4 Hopfield hálózatok implementálása a gyakorlatban.....	43

4.1	Fejlesztői környezet	43
4.2	Bevezető példa	44
4.3	Hopfield hálózatok kapacitása	46
4.3.1	Program felépítése	47
4.3.2	Felhasznált képletek.....	48
4.3.3	Program kimenete	48
4.2.4	Algoritmusok pontossága.....	50
4.4	Képszegmentálás.....	51
4.4.1	Előnyök és hátrányok, alkalmazásunkban	52
4.4.2	Összegzés.....	52
4.5	Kriptográfiai megközelítés.....	54
4.6	Optimalizálási probléma Hopfield logikával	57
4.6.1	Hopfield hasonmás?.....	58
4.6.2	Részletes leírás.....	59
4.6.3	Kimenetek, tesztek.....	60
5	Összegző weboldal	64
5.1	Fejlesztői környezet	64
	Befejezés	66
	Resumé.....	67
	Hivatkozások.....	68
	Mellékletek	75

Ábrajegyzék

1. ábra: Neurális hálózatok idővonala Forrás: saját szerkesztés (visme.com).....	21
2. ábra: A Hebb-háló architektúrája Forrás: saját szerkesztés.....	23
3. ábra: A program kimenete balról jobbra haladva a j, k, b, y, a1, a2, an, w1_n, w2_n, b_n	24
4. ábra: McCulloch-Pitts Forrás: saját szerkesztés	25
5. ábra: Architektúráis grafikon egy N=4 neuronból álló Hopfield hálózatra Forrás: saját szerkesztés (Simulinkben)	28
6. ábra: Példák jó (a) és romlott (b) narancsképekre, amelyeket az osztályozási folyamat során értékelt az osztályozás előtt (c) és után (d). Forrás: a felhasznált tanulmányból..	32
7. ábra: Az implementált rendszer Forrás: felhasznált tanulmányból.....	35
8. ábra: Az eredeti kép, a rejtjelezett kép, illetve a visszanyert kép Forrás: tanulmányból	39
9. ábra: A rendszer felépítése Forrás: az említett tanulmány	40
10. ábra: A visszakapott ábra, illetve annak Hamming távolsága iterációként Forrás: saját szerkesztés	45
11. ábra: A mintáink reprezentálása Forrás: saját szerkesztés.....	45
12. ábra: Command windowson belüli, iterációként változó értékek Forrás: saját szerkesztés	45
13. ábra: A mintáink reprezentálása Forrás: saját szerkesztés	46
14. ábra: A kapacitás maximális kihasználása, pixelhiba nélkül Forrás: saját szerkesztés	48
15. ábra: Rekonstruált betűk, pixelhiba nélkül Forrás: Saját szerkesztés	49
16. ábra: 15 beolvasott betűnél a kapacitás megtelik ezért több pixelhiba keletkezik Forrás: Saját szerkesztés	49
17. ábra: Hiba léptékének növekedése 15 betű esetén Forrás: Saját szerkesztés.....	50
18. ábra: Eredeti képek Forrás: saját szerkesztés.....	52
19. ábra: Levágott és átméretezett bináris képek Forrás: saját szerkesztés	53
20. ábra: Zajjal ellátott, illetve visszanyert képek Forrás: saját szerkesztés.....	53
21. ábra: Hisztogram az eredeti majd a visszakapott képekre, a fehér szín hivatott jelölni a szemmel nem látható pixel különbségeket Forrás: saját szerkesztés	54

22. ábra: Command windowson belüli entrópia értékek Forrás: Saját szerkesztés	55
24. ábra: Bit shift titkosítás Forrás: Saját szerkesztés.....	56
23. ábra: Hopfield, Xor titkosítás Forrás: Saját szerkesztés.....	56
25. ábra: Substitution titkosítás Forrás: Saját szerkesztés	57
26. ábra: Futás 10*10 adathalmazra Forrás: Saját szerkesztés	60
27. ábra: Végző állapot 10*10 halmazra Forrás: Saját szerkesztés.....	60
29. ábra: Iterációnkénti command window értesítések Forrás: Saját szerkesztés	61
28. ábra: 31200 iteráció utáni eredmény 20*20 adathalmazra Forrás: saját szerkesztés..	61
30. ábra: Végző állapot 20*20 adathalmazra Forrás: saját szerkesztés.....	62
31. ábra: A weboldal fogadóképe Forrás: saját szerkesztés	65
32. ábra: 4.1. fejezet folyamatábrája Forrás: saját szerkesztés (draw.io).....	75
33. ábra: 4.3. fejezethez saját értelmezésű pseudokód Forrás: saját szerkesztés	76

Bevezetés

Az informatika berkein belül egy nagyon fontos terület a neurális hálók témaköre. Az emberi agy működése mindig is lenyűgözte az emberiséget, és sok évtizeden át inspirációt nyújtott a számítástechnika és a mesterséges intelligencia terén. Az idegsejteken alapuló hálózatok, az úgynevezett neurális hálózatok lehetővé teszik az információ feldolgozását és reprezentálását olyan módon, amely hasonlít az agy működéséhez. A környezet megfigyelése, értelmezése és a megfigyelések implementálása vagy épp felhasználása a tudományok területén nem mondható újkeletűnek, a történelem során ez a minta megfigyelhető és ha az idő vasfoga alatt, *ad absurdum* az évmilliók alatt a természetben valami bevált, azt miért ne használhatná fel az ember a saját maga segítségére, vagy egy ő általa alkotott rendszerre, amely a természetben alapszik.

Egy nagyon gyorsanfejlődő és napjainkban a figyelem központjában álló témakörrel beszélünk. Számos felhasználási területtel, móddal, illetve eszközökkel valósítják meg mind ezt a gyakorlatban. Mi főleg az informatika szempontjából fogjuk a neurális hálózatokat elemezni, majd a dolgozat másik felében bemutatni az egyes hálózatok alkalmazásait.

Lépésről-lépésre haladva, az oldalakat lapozva tárul majd elénk e általam „csodálatosnak” definiált világ, amely túlmutat a matematikai képleteken és a végén megjutalmaz a kívánt eredménnyel. A lejjebb látható idézet az általam pozitívnak vett fejlődés másik oldalát is említi, de a tényleges konklúziót az olvasóra bízom.

„ ... Nyitottak arra a lehetőségre, hogy az új technológiák új matematikai távlatokat nyithatnak és újfajta matematikai megértést. Az a kilátás, hogy a matematikai gondolkodásban betöltött jelentős szerepének átadása a számítógépnek, lehet, hogy elbizonytalanító, de egyben felemelő is, és várakozással kell tekinteniünk arra, hogy lássuk, hová jutunk el, hová vezet minket a technológia¹ (Hanna, Larvor, 2020, 1144). ”

Az említett technológiai fejlődés milyenségének elgondolása után a következő rész magát a diplomamunka célját fogja ismertetni.

¹ Eredeti idézet angolul: „ ... Open to the possibility that new technologies can open new mathematical vistas and afford new types of mathematical understanding. The prospect of ceding a substantial role in mathematical reasoning to the computer may be disconcerting, but it should also be exhilarating, and we should look forward to seeing where the technology takes us. ”

Munka célja

A cél magában hordozza a *Hopfield* hálózatok elméleti áttekintését a gyakorlatival párhuzamosan, valamint egyes megvalósításait a modern kori, legrelevánsabb tanulmányok alapján. Emellett elkerülhetetlennek érzem a neurális hálózatok történeti áttekintését is. Ugyancsak ez az áttekintés sok szakirodalomban megtalálható, de hiányoltam egészen a végletekig, a modern rendszerek jellemzéséig való eljutást, célom ezzel is szemléltetni, hogy milyen pontról jutottunk vagy fejlődtünk egészen napjainkig.

Az elméleti áttekintés a szakirodalmak által, amelyet több tudományos cikk, kiadvány alapjául formáltam össze, jó gerincet ad szerintem annak a megalapozott tudásnak, amelyre majd szükségünk lesz, hogy megértsük a munka második, azaz gyakorlati részét.

A gyakorlati rész érdekessége vagy színessége pont azon az elven alapszik, hogy megpróbáltam többféle síkon implementálni a *Hopfield* hálózatokat, így ezzel azok variabilitására rámutatva. Számptalan felhasználási módjukat ismerjük, mint például az optimalizációt, mintázat felismerő rendszert, vagy afféle kombinatorikus² problémákat, mint például az utazóügynök probléma³, de beszélhetnénk akár a klaszterezés, rekonstrukció és restaurálás, kriptográfiáról⁴ is, ezen témakörök közül néhányat meg is jeleníték alkalmazásaim során.

Mindenek felett a fő cél az előbb említettek összességéből egy homogén munka megalkotása, amely fejezetei részeszesei egy egymásra épülő, okkal felépített koncepciónak.

Remélem a felállított célokkal sikerrel járok és a megvalósításom is kellően kerek lesz, a továbbiakban belevetjük magunk a neurális hálózatok technológiai forradalmának fergetegébe.

² Ugyanakkor a Knapsack problémára is alkalmazható.

³ Angolul: Traveling Salesman problem

⁴ „A kriptográfia ógörög eredetű szó, jelentése: titkosírás. Ez a tudomány azóta foglalkoztatja az embert, amióta a tulajdon, mint fogalom, a birtoklás részévé vált az emberiségnek, azonban tényleges kutatása csak néhány évtizeddel ezelőtt kezdődött meg. Régebben a nyelvtudomány témakörébe sorolták, azonban mára már erősen matematikai és informatikai jelleget kapott. Rejtjelezéssel, titkosírással, kódolással foglalkozik, melyek előállításával mellett egy másik fő cél azok megfejtése is (Varga Péter, 2008, 3).”

1 A neurális hálózatok technológiai forradalma

A diplomamunkám fő témája a *Hopfield* hálózatok, de számomra elengedhetetlennek tűnik, hogy e témakör kapcsán ne jelenjen meg valamely formában magának a neurális hálózatoknak jelene és jövője mellett a múltja is.

Sok ember elméjében fennáll az a kép, hogy a neurális hálózatok valamely „*kézzel megfoghatatlan*”, bonyolult rendszerek. Diplomamunkám rész céljának is tekinthető, hogy ezt a fogalmat szeretném demisztifikálni⁵.

Én a következőképp definiálnám⁶: a neurális hálózat egy komplex struktúra, amely több mesterséges neuronból áll, és adatok feldolgozására és mintázatok tanulására képes. Tanulás közben az adatokból optimalizálja a súlyokat és a kapcsolatokat, hogy prediktív és osztályozási feladatokat végezzen.

Más megfogalmazásban a neurális hálózatok mint koncepció, valahol ott kezdődik, hogy az ember, mint gondolkodó lény, megfigyelte környezetét és mint sok tudományágon belül biológiai rendszerek analógiájára próbált meg mesterséges rendszereket alkotni. E rendszerek sajátosságait, tulajdonságait figyelte meg és ezekből próbált következtetéseket levonni, mint például a nagy számú, de egyenként kicsiny alapegységből való felépítés, valamint az egységek közötti kapcsolatok, és nem utolsósorban ezen egységek tanulási képessége. Főleg az élő természetből, csak bizonyos általános elveket sikerült ellesni, a gyakorlatban megvalósított mesterséges neurális hálózatok nem rendelkeznek a biológiai rendszerek mélyebb tulajdonságaival (Fazekas, 2013, 9-14).

1.1 Visszatekintés a múltba

A neurális hálózatok története több időszakra osztható. Az első lépés a mesterséges neurális hálózatok felé 1943-ban történt, amikor *Warren McCulloch*, egy neurofiziológus és egy fiatal matematikus, *Walter Pitts* kifejlesztette az első neurális hálózatok modelljeit. Megírták

⁵ Az a folyamat, amely során egy dolgot vagy jelenséget, koncepciót megmagyarázunk, megértjük a részleteit, ezáltal eloszlatjuk az általa keltett bonyolultságot.

⁶ Hosszabban ugyanígy általam megfogalmazott definíció: A neurális hálózat egy olyan gépi tanulási modell, amely inspirációt merít az emberi agy felépítéséből. Ez egy összetett hálózati struktúra, amely több egymással kapcsolódó mesterséges neuronból áll, és képes információkat feldolgozni, mint például adatokat vagy mintákat. A neurális hálózatok az adatokból tanulnak és alkalmazkodnak az ismétlődő mintázatokra, és képesek prediktív vagy osztályozó feladatokat végrehajtani. A folyamat során a súlyok és a kapcsolatok a neuronok között folyamatosan finomításra kerülnek az optimális eredmények eléréséhez.

a *The Logical Calculus of the Ideas Immanent in Nervous Activity*⁷ (Az idegi tevékenységben rejlő eszmék logikai számítása) című tanulmányukat, arról, hogyan működhetnek a neuronok. Hálózataik a következők voltak: egyszerű elemeken alapultak, amelyeket bináris eszközöknek tekintettek, rögzített küszöbértékekkel. Modelljük eredményei egyszerű logikai függvények voltak: „mindent-vagy-semmit” jellegű idegtevékenységet mutattak.

1944-ben *Joseph Erlanger* és *Herbert Spencer Gasserrel* együtt számos idegrostfajtát azonosított, és megállapították az akciós potenciálok közötti kapcsolatot vezetési sebesség és a rostátmérő között.

1949-ben *Hebb* pszichológus megírta *The Organization of Behavior*⁸ (A viselkedés szerveződése) című művét, amelyben rámutatott arra a tényre, hogy az idegpályák minden egyes alkalommal megerősödnek, amikor használjuk őket, ez a koncepció alapvetően fontos az emberi tanulás szempontjából.

1958-ban *Rosenblatt* pszichológus korai munkát végzett a *perceptronokról*. A *Perceptron* egy olyan elektronikus eszköz volt, amelyet a biológiai elvek alapján készült, és tanulási képességet mutatott. Ő írta a neuroszámítástechnikáról szóló korai könyvet is, *Principles of Neurodynamics* címmel. Egy másik rendszere az *ADALINE* (*ADaptive LInear Element*) volt, amely 1960-ban fejlesztett ki két villamosmérnök, *Widrow* és *Hoff*. Az alkalmazott módszer a tanuláshoz más volt, mint a *Perceptroné*, a *Least-MeanSquares*⁹ tanulási szabályt alkalmazta. 1962-ben *Widrow* és *Hoff* kifejlesztett egy tanulási eljárást, amely az értéket vizsgálja, mielőtt a súlyt módosítaná.

A kezdeti lelkesedés után a terület túlélt egy frusztrációval teli időszakot és kegyvesztettséget. 1969-ben *Minsky* és *Papert* könyvet írt *Perceptrons: An Introduction to Computational Geometry*¹⁰. Ez egy neurális hálózatokat lejárató kampány része volt. A kutatások, amelyek számos alapvető problémát mutattak fel, és amelyben általánosították az

⁷ Itt elérhető az egész tanulmány: <https://link.springer.com/article/10.1007/bf02478259>

⁸ Itt elérhető a teljes: https://pure.mpg.de/rest/items/item_2346268_3/component/file_2346267/content

⁹ *LMS* algoritmus egy adaptív szűrő, amelyet a *Widrow és Hoff* (1960) fejlesztett ki villamosmérnöki alkalmazásokhoz. Ezt olyan alkalmazásokban használják, mint a visszhangszűrés távolsági hívásoknál, vérvizsgálatoknál, vérnyomás-szabályozás, és zajszűrő fejhallgató. Az *LMS* algoritmusban a hiba visszakerül az algoritmusba, minden egyes minta bevitele után és a súlyok beállítása megtörténik (Russel, 1-10).

¹⁰ Itt elérhető a teljes tanulmány: <https://www.britannica.com/topic/Perceptrons-An-Introduction-to-Computational-Geometry>

egyrétegű *perceptronok* korlátait. Bár a szerzők jól tudták, hogy a nagy teljesítményű *perceptronok* több réteggel rendelkeznek, és *Rosenblatt* alapvető *feed-forward perceptronjai* három réteggel rendelkeznek, a *perceptront* kétrétegű gépként definiálták, amely csak lineárisan szeparálható problémákat képes kezelni, és például nem tudja megoldani az úgynevezett *exclusiveOR (XOR)*¹¹ problémát. Mivel a közérdeklődés és a rendelkezésre álló finanszírozás minimális lett, csak néhány kutató folytatta a munkát az olyan problémákon, mint például a mintafelismerés. De ebben az időszakban számos olyan paradigma született, amelyekkel a modern munka továbbra is foglalkozik.

Klopf 1972-ben, kifejlesztett egy alapot a tanulásra mesterséges neuronokban egy biológiai elv alapján. *Paul Werbos* 1974-ben kifejlesztette a *back-propagation* tanulási módszert, bár annak jelentőségét csak 1986-ban ismerték fel teljesen.

Fukushima kifejlesztett egy lépcsőzetesen betanított többrétegű neurális hálózatot a kézzel írt karakterek értelmezéséhez. Az eredeti munka *Cognitron: A selforganizing multilayered neural network*¹² (Egy önszerveződő többrétegű neurális hálózat) 1975-ben jelent meg.

Grossberg 1976-ban az *Adaptive pattern classification and universal recoding*¹³ (Adaptív mintaosztályozás és univerzális rekódolás) mutatta be az adaptív rezonanciát, mint az emberi kognitív információfeldolgozás elméletét.

Az 1980-as években több esemény is újbóli érdeklődést váltott ki. *Kohonen* nagyban hozzájárult, illetve számos munkát közölt a mesterséges neurális hálózatok területén. Ő vezette be a mesterséges neurális hálózatot, amelyet néha *Kohonen-térképnek* vagy -hálózatnak is neveznek.

Most érkeztünk el a történelem fergetegében munkánk témaköréhez. *Hopfield*, a *Caltech*¹⁴ munkatársa 1982-ben bemutatott egy tanulmányt *Neural Networks and Physical Systems*

¹¹ Az *XOR*, vagy „kizárólagos vagy” probléma az *ANN (Artificial Neural Network, Mesterséges neurális hálózat)* -kutatás klasszikus problémája. Ez az a probléma, amikor egy neurális hálózat segítségével két bináris bemenet esetén megjósoljuk az *XOR* logikai kapuk kimeneteit. Az *XOR* függvénynek igaz értéket kell visszaadnia, ha a két bemenet nem egyenlő, és hamis értéket, ha egyenlőek. Mára ez az úgynevezett probléma rég megoldódott, de a múltban elég nagy fejtörést okozott ez a szakmának (Forrás: Google,2023).

¹² Itt elérhető az egész tanulmány: <https://link.springer.com/article/10.1007/BF00342633>

¹³ Itt elérhető a teljes tanulmány: <https://link.springer.com/article/10.1007/BF00344744>

¹⁴ *A California Institute of Technology* (magyarul: Kaliforniai Műszaki Egyetem, röviden angolul *Caltech*) az Egyesült Államok (USA) egyik magánegyeteme, a kaliforniai *Pasadena* városban található *Los Angeles* közelében. Vezető kutatóegyetem, fő kutatási területei a természettudományok és a műszaki tudományok. (Forrás: Google,2023)

with Emergent Collective Computational Abilities¹⁵ (Neurális hálózatok és fizikai rendszerek feltörekvő kollektív számítási képességekkel). Hopfield leírta a rekurrens mesterséges neurális hálózatot, amely tartalomcímezhető memóriarendszerként szolgál. Az ő munkái több száz magasan képzett tudóst, matematikust és egyéb tudományág képviselőjét győzték meg arról, hogy csatlakozzanak a neurális hálózatok feltörekvő területéhez.

Az eredetileg Werbos által 1974-ben felfedezett *backpropagation*¹⁶ algoritmust 1986-ban fedezték fel újra a *Learning Internal Representation by Error Propagation*¹⁷ című könyvében Rumelhart és társai A *backpropagation a gradient descent* (gradiens ereszkedés) algoritmus egy formája, amelyet mesterséges neurális hálózatoknál használnak hibaminimalizálásra.

1985-re az *American Institute of Physics*¹⁸ (Amerikai Fizikai Intézet) elkezdte az azóta évente megrendezett *Neural Networks for Computing* (Neurális hálózatok a számítástechnikában) elnevezésű találkozót. 1987-ben az első nyílt konferencia a neurális *IEEE International Conference on Neural Networks* (IEEE Nemzetközi Neurális Hálózatok Konferenciája) volt *San Diegóban* tartották, és megalakult a Nemzetközi Neurális Hálózati Társaság (INNS). 1988-ban megalapították az *INNS Neural Networks* című folyóiratát, amelyet 1989-ben a *Neural Computation*, 1990-ben pedig az *IEEE Transactions on Neural Networks* követett. Ez az esemény sorozat is nagyban hozzájárult a neurális hálózatok és témakörük fejlődéséhez.

Carpenter és Grossberg 1987-ben az *A massively parallel architecture for a self-organizing neural pattern recognition machine*¹⁹ leírta az ART1 kifejezetten bináris minták felismerésére tervezett, felügyelet nélküli tanulási modellt (Saeed, Homenda, 2016, 3-5).

¹⁵ Itt elérhető az egész tanulmány: <https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554>

¹⁶ A *backpropagation* alapvetően, egyszerűen egy hatékony és pontos módszer egyetlen célmennyiség (például a mintaosztályozási hiba) összes deriváltjának kiszámítására egy nagyszámú bemeneti mennyiség (például egy osztályozási szabály paraméterei vagy súlyai) függvényében. A *backpropagation* kiterjeszti ezt a módszert, így dinamikus rendszerekre is alkalmazható. Ez lehetővé teszi a szükséges deriváltak kiszámítását egy iteratív elemzési eljárással, egy memóriával rendelkező neurális hálózat vagy egy olyan vezérlőrendszer optimalizálásakor, amely időben maximalizálja a teljesítményt (Werbos, 1990).

¹⁷ Itt elérhető: <https://ieeexplore.ieee.org/document/6302929>

¹⁸ Az AIP-t 1931-ben alapították, válaszul a tudományok finanszírozásának hiányára a nagy gazdasági világválság idején. 1932-ben hivatalosan is megalakult, öt eredeti „tagtársaságból” állt, és összesen négyezer tagot számlált. Az 1960-as évek közepétől kezdődően újabb tagtársaságokkal bővült. Az AIP megalakulását követően azonnal tudományos folyóiratokat kezdett kiadni. (Forrás: wiki, 2023, Elérhető: https://en.wikipedia.org/wiki/American_Institute_of_Physics)

¹⁹ Elérhető: <https://www.sciencedirect.com/science/article/abs/pii/S0734189X87800142>

1.2 Az ezredforduló és napjaink

A neurális hálózatok területén jelentős előrelépés történt, elég nagy ahhoz, hogy figyelmet vonzzon maga köré és további kutatásokat finanszírozzon. Ma a neurális hálózatokkal kapcsolatos viták mindenütt zajlanak. A jelenlegi kereskedelmi alkalmazásokon túlmutató fejlődés lehetségesnek tűnik, és a kutatás számos fronton halad előre a területen. A neurális elméleten alapuló *chipek* egyre inkább megjelennek, és összetett problémákra való alkalmazások fejlődnek. A neurális hálózati technológia napjainkban egyértelműen az átmenet időszakát éli.

2009 és 2012 között a *rekurrens* neurális hálózatok és a *deep feedforward* (mély előre csatolt) neurális hálózatokat *Schmidhuber* kutatócsoportjában fejlesztették ki. 2014-ben az *IBM*²⁰ tudósai bemutatták a processzort (*TrueNorth*), amely architektúrája hasonló az agyban létezőhöz. Az *IBM* továbbá bemutatott egy olyan integrált áramkört, amely mérete egy bélyeghez hasonló áramkör, amely képes szimulálni több millió neuron és 256 millió szinapszis működését valós időben. A rendszer 46 és 400 milliárd között képes végrehajtani szinaptikus műveletet másodpercenként (Saeed, Homenda, 2016, 6-7).

Ian Goodfellow 2014-ben megtervezi a *Generative Adversarial Network (GAN)* rendszert. A *GAN* képes a képzési adatokkal azonos statisztikájú új adatokat létrehozni. A *GAN* lényege, hogy két hálózatot használ, amelyek egymással versenyeznek. Az első a generátor, új adatokat hoz létre. A másodiknak az úgynevezett diszkriminátornak a valós és a generált adatok kerülnek bemutatásra, és eldönti, hogy melyik kategóriába tartoznak, így javul a generált adatok minősége.

Olaf Ronneberger és társai létrehozták az *U-netet* (U-hálót) 2015-ben az orvosi biológiai képek szegmentálására. Az *U-Net* egy kódoló konvolúciós hálózattól áll, amely egy dekódoló hálózattal van összekötve a kép felfelépítésére. Az *U-net* fontos jellemzője, hogy ez a két rész úgynevezett kihagyásos kapcsolatokkal van összekötve.

Kaiming He, Xiangyu Zhang, Shaoqing Ren és *Jian Sun* 2015-ben szintén bemutatják a *Residual Neural Network (ResNet)* nevű neurális hálózatot. A *ResNet* egy mély konvolúciós

²⁰ Az *IBM*, vagyis *International Business Machines* egy egyesült államokbeli (*USA*) székhelyű, multinacionális informatikai cég. A cég részvénye egyike annak a harminc részvényből álló kosárnak, amelyből a *Dow Jones* Ipari Átlagot számítják. A céget szokták „*Kék Óriás*” (*Big Blue*) néven is emlegetni. (Forrás: wiki, Elérhető: <https://hu.wikipedia.org/wiki/IBM>)

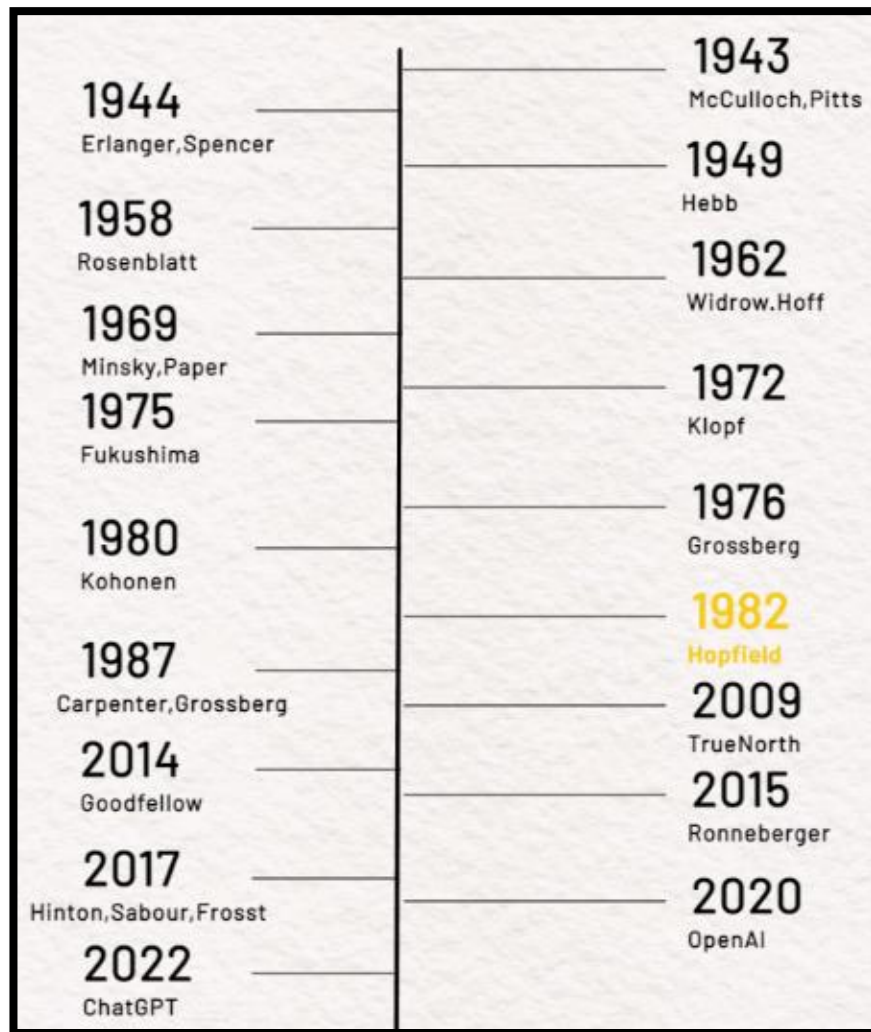
hálózat, amely az úgynevezett kihagyott kapcsolatok vagy rövidítések bevezetésével javítja az eltűnő gradiens problémát.

Geoffrey Hinton, Sara Sabour és Niholas Frosst bemutatja 2017-ben a *Capsulenet*-et, egy olyan neurális hálót, amely kapszulákból áll, nem pedig egyes neuronokból. Szándékuk szerint figyelembe veszik a képen belüli tájolást és térbeli viszonyokat. Ugyanebben az évben a *Google Brain* csapata bemutatta a *Transformer* modellt. Egy mélytanulási modellt, amelyet főként természetes nyelvfeldolgozásra használnak. Párhuzamosítja a szekvenciális adatokat, és az önfigyelési mechanizmuson alapul.

Bidirectional Encoder Representations from Transformers egy transzformátor alapú modell a természetes nyelvfeldolgozáshoz 2018-ban kerül bemutatásra.

Napjaink nagy csodája és annak előzménye: 2020-ban az *OpenAI* közzéteszi a *Generative Pre-trained Transformer 3*-at (közismert nevén a *GPT-3*), egy mély tanulási modellt, amely emberhez hasonló szövegek előállítására szolgál. A bemenet egy kezdeti szöveg, és az adott bemenetet folytató szöveget állít elő.

Napjainkban, 2022 környékén az *OpenAI* kifejleszti és közzéteszi a *ChatGPT* fejlett chatbotot. A korábbi *GPT* modellekre épül, de sokkal fejlettebb. Képes emlékezni a beszélgetés korábbi részeire, és képes összetett válaszokat adni. Még programkódot is képes generálni (Pumalin,2023).



1. ábra: Neurális hálózatok idővonala **Forrás:** saját szerkesztés (visme.com)

Nagyon színes az a spektrum, amely a neurális hálózatok tudományágát érinti. Ebben a fejezetben a legrelevánsabb, legnagyobb impulzust keltő tudósokat vettem górcső alá.

Az alul olvasható idézettel zárnám ezt a fejezetet, amely szerintem kellően reprezentálja magát a neurális hálózatok történetét is, hogy az ember a kreativitásával karöltve, mindig azon van, hogy újabbnál újabb módokat találjon fel saját maga támogatására.

„Computer science is not about building computers or writing programs. It's about finding new ways to support human thinking.”

(„Az informatika nem arról szól, hogy számítógépeket építünk és programokat írunk. Az informatika arról szól, hogy új módokat találunk az emberi gondolkodás támogatására.”)

Alan kay²¹

²¹ Az idézet az azquotes.com oldaláról származik.

2 A Hopfield hálózatok részletes áttekintése

„Hopfield hálózata²² (Hopfield, 1982), (Hopfield & Tank, 1985) egy rekurrens neurális hálózat, amelynek rögzített szinaptikus súly értékei vannak, ahol az aktiválási értékek egy relaxációs folyamaton mennek keresztül a visszacsatolás során. Diszkrét vagy folyamatos dinamikus rendszerként jellemezhetjük őket, és differenciálegyenletek segítségével írhatóak le. Gyakran attraktív, vagy autoasszociatív neurális hálózatnak nevezik őket (Kmet', Kmet'ová, Annuš, 2021, 86).”

Ugyanakkor egy másik tanulmány szerinti definícióban a Hopfield hálózatokat vagy Hopfield asszociatív memóriát úgy definiálták, mint egyrétegű, szimmetrikus, nem lineáris, autoasszociatív, rekurrens, legközelebbi szomszédos mintázat kódoló (*nearest neighbor pattern encoder*), amely bináris/bipoláris térbeli mintákat tárol.

$$A_k = (a_1^k, \dots, a_n^k) \quad k = 1, 2, \dots, m \quad (3.1)$$

Hebb tanulással, a diszkrét esetben, míg a folyamatos idejű esetben a következőképpen határozza meg: egyrétegű, autoasszociatív, legközelebbi szomszédos kódoló (*nearest neighbor encoder*), amely folyamatos időben működik, tetszőleges analóg térbeli mintákat tárol.

$$A_k = (a_1^k, \dots, a_n^k) \quad k = 1, 2, \dots, m \quad (3.2)$$

A hálózat offline tanul, aszinkron módon frissíti az úgynevezett feldolgozó elemeit (*PE*) (Sulehria, Zhang, 2007, 125-126).

2.0.1 Hebb tanulási szabálya²³

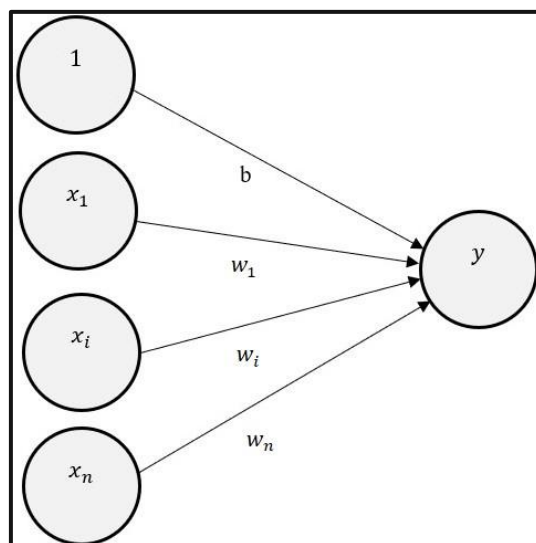
Ha már említésre került pár mondattal feljebb a Hebb féle tanulási szabály, akkor az algoritmust és a hálózat architektúrát ismertetném.

A Hebb féle tanulási szabály az egyik legkorábbi és legegyszerűbb tanulási szabály a neurális hálózatok számára. Donald Hebb javasolta, hogy ha két összekapcsolt neuron egyszerre „be van kapcsolva”, akkor a köztük lévő súlyt növelni kell (más szóval élve, ha egy időben aktív két neuron, a kapcsolatuk erősödik a Hebb tanulás által). A Hebb hálózat egy egyrétegű neurális hálózat, amely egy bemeneti rétegből, sok bemeneti egységgel,

²² Shun'ichi Amari leírta már 1972-ben ezt a modellt, de John Hopfield népszerűsítette 1982-ben (Forrás: Wiki, 2023, Elérhető: https://en.wikipedia.org/wiki/Hopfield_network).

²³ Angolul: *Hebbian learning*

valamint egy kimeneti rétegből és egy kimeneti egységből áll. Ezt az architektúrát általában mintázatosztályozásra használják, ezért alkalmazható jól a *Hopfield hálózatoknál* is.



2. ábra: A *Hebb*-háló architektúrája **Forrás:** saját szerkesztés

Hebb tanulási algoritmus lépésenként:

- I. Inicializáljuk az összes súlyt és a torzítást nullára

$$w_i = 0, \quad \text{az } i = 1 - \text{től } n^{24} - \text{ig}, b = 0$$

- II. Minden egyes bemeneti képzési vektor (*training vector*) és célkimeneti (*target output*) páros, $S: t$, esetében végezzük el a 2-5. lépést.
- III. A bemenet aktiválásának beállítása: $x_i = S_i, i = 1, \dots, n$.
- IV. A kimenet aktiválásának beállítása: $y = t$.
- V. Állítsuk be a súlyokat és a torzítást:

$$w_i(\text{új}) = w_i(\text{régi}) + x_i y \text{ ahol } i = 1, \dots, n,$$

$$b(\text{új}) = b(\text{régi}) + y$$

Ha a torzítást (*bias*) mindig 1-es bemeneti jelnek tekintjük, akkor a súlyváltozása következőképpen írható fel

$$w(\text{új}) = w(\text{régi}) + \Delta w, \text{ ahol } \Delta w = xy$$

(Chakraverty, Sahoo, Mahato, 2019, 175-176).

²⁴ Az n a bemeneti neuronok száma.

2.0.2 Gyakorlati példa Hebb tanulásra

```
clc,
close all
%%
j=[ 1 1 -1 -1 ];% 1.bemenet
k=[1 -1 1 -1 ];% 2.bemenet
b=[1 1 1 1 ];% bias
y=[-1 1 1 -1 ];% cél kimenet
w1=0;% 1. kezdeti súly
w2=0;% 2. kezdeti súly
b0=0;% kezdeti bias súly
%%
for i=1:4
a1(i)=j(i)*y(i);% súlymódosítás az első bemeneti szerint
a2(i)=k(i)*y(i); % súlymódosítás a második bemeneti szerint
an(i)=b(i)*y(i);% súlymódosítás a bias szerint
w1_n(i)=w1+a1(i);% 1. új súly
w2_n(i)=w2+a2(i);% 2. új súly
b_n(i)=b0+an(i);% új bias súly
w1=w1_n(i);% frissített súly
w2=w2_n(i); % frissített bias súly
b0=b_n(i);
end
%%
disp(' j k b Y dw1 dw2 an w1 w2 B ');% bemenetek, cél kimenet, bias vektorok
transzponáltjai,
% a súlymódosítások transzponáltjai, új súlyok és bias súlyok transzponáltjai
B=[j' k' b' y' a1' a2' an' w1_n' w2_n' b_n'];
disp(B)
```

(Chakraverty, Sahoo, Mahato,2019,178, (alapján)).

A fent látható kód, *Matlab*²⁵ környezetben lett elkészítve, az előbb elméleti síkon ismertetett *Hebb* algoritmus implementációja, amely egy *XOR* kaput tanít meg. A kód inicializálja a bemeneti változókat és súlyokat, majd egy ciklusban iterálva lépésenként végrehajtja a súlymódosításokat és frissíti azokat. Végül kiírja a bemeneteket, cél kimenetet, súlymódosításokat és az új súlyokat.

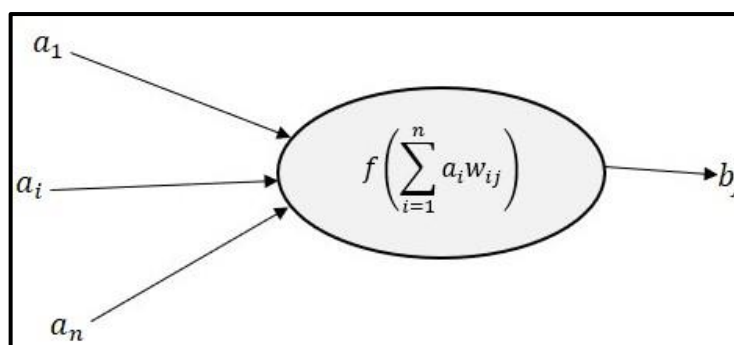
1	1	1	-1	-1	-1	-1	-1	-1	-1
1	-1	1	1	1	-1	1	0	-2	0
-1	1	1	1	-1	1	1	-1	-1	1
-1	-1	1	-1	1	1	-1	0	0	0

3. ábra: A program kimenete balról jobbra haladva a j, k, b, y, a1, a2, an, w1_n, w2_n, b_n

²⁵ A MATLAB egy programozási és numerikus számítási platform, amelyet mérnökök és tudósok milliói használnak adatok elemzésére, algoritmusok fejlesztésére és modellek készítésére, MathWorks fejleszti, számtalan beépített alkalmazással rendelkezik, mint például a Simulink...

Folytatva, azonban a *Hopfield* hálózatokat, minden bemenet egy jelet hordoz, amely összeadódik, (Σ) és az összeadás után a jelet egy $f()$ küszöbértékfüggvényen keresztül feldolgozzuk, amely egy kimeneti jel(ek).

Az alkalmazott neuronmodell a *McCulloch-Pitts*²⁶ a diszkrét esetben, a következővel bináris/bipoláris értékekkel, amelyeket a lent látható ábra szemléltet, az $f()$ pedig lépcsőzetes küszöbérték függvény (*threshold function*).



4. ábra: McCulloch-Pitts **Forrás:** saját szerkesztés

A *McCulloch-Pitts*-modell modell bináris/bipoláris értékeket használ, míg a *Hopfield* egy folytonos, determinisztikus neuronhálózatot, amely egymással összekapcsolt neuronokból áll, fokozatos reakcióval, amely ugyanolyan jól működik, mint a kétállapotú. A neuron itt lényegében analóg, vagyis a küszöbfüggvény (*threshold function*) $f()$ egy *ramp*²⁷ (rámpa) vagy egy *sigmoid*²⁸ (S alakú).

Ugyanakkor a modellek tanulása két kategóriába sorolható, felügyelt (*supervised*) és a felügyelet nélküli (*unsupervised*), bár mindkettőnek létezhetnek egyidejűleg aspektusai egy adott architektúrában. A felügyelt tanulás magában foglal egy külső tanárt és/vagy globális információt (*external teacher and/or global information*) tartalmaz, és a továbbiakban még két alkategóriába sorolható: strukturált és időbeli vagy ideiglenes tanulás (*temporal learning*). A felügyelet nélküli tanulás esetében, szintén önszerveződésnek is nevezik, ez egy olyan folyamat, amelyben nincs külső tanító, és például a helyi információktól függ.

A definíció szerint a diszkrét *Hopfield* hálózatban, a *Hebb* tanulás kerül leírásra, amelyet már az előbb tárgyaltunk (Sulehria, Zhang,2007,127-129).

²⁶ Az első fejezetben említésre kerül.

²⁷ Nagyon hasonlít a *sigmoid* aktiválási függvényhez, a bemeneteket egy (0,1) tartományban képezi le a kimenetre, a sima görbe helyett a rámpa éles görbét kap. Ez a lineáris függvény csonka változata (Kumar,2022).

²⁸ A *sigmoid* függvény egy olyan korlátos, differenciálható valós függvény, amely minden valós bemeneti értékre definiálva van, és amelynek mindenhol pozitív deriváltja van (Han, Moraga,1995,195).

2.1 Demisztifikálás

Az imént ismertetett tudományosabb megközelítés után egy egyszerűbb gondolatmenettel szeretném zárni a *Hopfield* hálózatok elméleti áttekintését mielőtt rátérnék a felhasználási területeikre. Az alapvetően már említett diszkrét, illetve folytonos típusú *Hopfield* hálózatokkal kezdeném.

2.1.1 Diszkrét²⁹ hálózatok

Ezek a hálózatok a két diszkrét kimenet bármelyikét adják. A kapott kimenet alapján további két típus lehetséges:

- I. **Bináris:** Ebben a típusban a kimenet vagy 0 vagy 1.
- II. **Bipoláris:** A kétpólusú hálózatokban a kimenet vagy -1 (ha a kimenet < 0) vagy 1 (ha a kimenet > 0).

2.1.2 Folytonos hálózatok

A bináris vagy bipoláris kimenet helyett a kimeneti érték 0 és 1 között van.

2.1.3 Architektúra

A *Hopfield* hálózat felépítése a következő elemekből áll:

- I. Az egyes csomópontok megőrzik állapotukat, amíg frissítésre nincs szükség.
- II. A frissítendő csomópont véletlenszerűen kerül kiválasztásra.
- III. Minden csomópont saját magán kívül minden más csomóponttal kapcsolatban áll.
- IV. Az egyes csomópontok állapota 0/1 vagy 1/-1.
- V. A *Hopfield* hálózat szerkezete szimmetrikus, azaz $w_{ij} = w_{ji}$ minden i és j esetén (amely már említésre került az előző fejezetek során).

2.1.4 Energiafüggvény a Hopfield hálózatban

A *Hopfield* hálózatokban két különböző típusú frissítés létezik.

- I. **Szinkron:** Az összes csomópont frissítése minden alkalommal egyszerre történik.
- II. **Aszinkron:** Egyszerre csak egy csomópont frissítése. Ez a csomópont véletlenszerűen vagy meghatározott szabályok alapján kerül kiválasztásra.

²⁹ A diszkrét adatok olyan numerikus adattípusok, amelyek egész, konkrét számokat tartalmaznak, meghatározott és rögzített adatértékekkel, amelyeket számolással határoznak meg. A folyamatos adatok összetett számokat és egy adott időintervallumban mért változó adatértékeket tartalmaznak (Harry Lees, 2022).

Aszinkron frissítésnél a *Hopfield* hálózatok minden egyes állapotához egy energiaérték tartozik. Az értéket egy függvényből kapjuk, és ezt a függvényt energiafüggvénynek³⁰ nevezzük. Ez a függvény a frissítés során csökkenhet vagy változatlan maradhat. A *Hopfield* hálózat energiafüggvényét a következőképpen határozzuk meg:

$$E = -1/2 \sum_{i=1}^N \sum_{j=1}^N w_{ij} x_i x_j$$

Egy hálózat akkor tekinthető stabil állapotúnak, ha az energiafüggvény a minimum felé vagy a minimumra konvergál.

2.1.5 Tanulási modell Hopfield hálózattal

A *Hopfield* hálózat tanítása az egyes állapotok energiájának csökkentését jelenti. A tanulási modell egy tanító és egy tesztelési algoritmusból áll. Beszéljünk mindegyikről egyenként. A tanítási algoritmusról már szó esett (*Hebb*).

Tesztelési algoritmus a kiszámított hálózati bemenet és a küszöbérték alapján számítsa ki a kimenetet hasonló képpen:

$$y_{ij} = x_i + \sum_{i=1}^N y_i w_{ij}$$

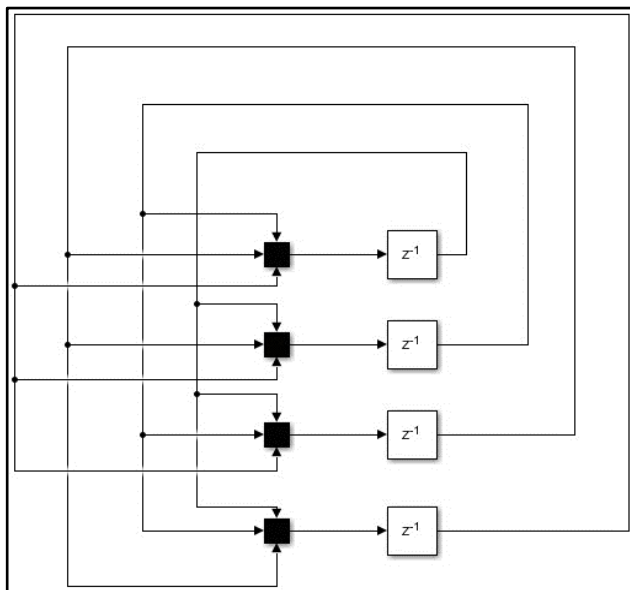
(Kmeť, Kmeťová, Annuš, 2021, 86-121) & (Haykin, 2008, 690-700).

Érdekességgént egy más tanulmány a *Hopfield* hálózatok architektúráját neurodinamikai szemszögből vizsgálta pár mondat erejéig említésre kerülne.

A *Hopfield* hálózatokat hasonlóan definiálja, az asszociatív memória példája, rejtett neuronok nélkül; az asszociatív memória fontos erőforrás az intelligens viselkedéshez. Valamint a *Hopfield* hálózat (modell) neuronok és megfelelő egységnyi időbeli késleltetések halmazából áll, amelyek egy többhurkos visszacsatolt rendszert alkotnak, (lásd 5. ábra). A visszacsatolási hurkok száma megegyezik a neuronok számával. Alapvetően minden egyes neuron kimenete egy egységnyi időkésleltetési elemen keresztül (*unit-time delay element*)

³⁰ Az energiafüggvényt olyan függvényként definiáljuk, amely a rendszer állapotának kötött és nem növekvő függvénye. Az energiafüggvény, más néven *Ljapunov*-függvény határozza meg a diszkrét *Hopfield*-hálózat stabilitását. Egy stabil hálózatban, amikor a csomópontok állapota megváltozik, az energiafüggvény csökken. Az energiaváltozás attól függ, hogy egyszerre csak egy egység tudja frissíteni aktiválását (Artificial N.N., 2023).

visszajuttatódik a hálózat minden egyes neuronjához. Más szóval, a modellben nincs ön-visszacsatolás (Haykin,2008,690).



5. ábra: Architektúráis grafikon egy $N=4$ neuronból álló *Hopfield* hálózatra **Forrás:** saját szerkesztés (*Simulinkben*)

Ezek mellett számos implementációjuk létezik, amelyről még nem esett szó például a többretegű *Hopfield* hálózatok, *Hopfield* hálózat tanulása tanár-minta alapján vagy még sorolhatnám, de a tudományosabb, illetve egyszerűsített magyarázat kellően megalapozza számunkra azt a tudást, amelyre szükséges lesz majd a következő fejezetek megértéséhez, illetve a még nem megtárgyalt részek ismertetésre kerülnek majd az alkalmazásuk során.

A gyakorlat és az elmélet között sokszor hézag keletkezik, amely az orvostudomány, mezőgazdaság, oktatás és jog területén dolgozó szakemberek számára tartós és nehéz problémát jelent és ez alól például a matematika se különbözik.

Azonban a gyakran kritizált elmélet és gyakorlat közötti hézag alapján ez a küldetés egy megfoghatatlan ideál marad. Néhány vezető tudományos folyóiratban megjelent különkiadás rámutat a növekvő aggodalmakra, hogy az akadémiai kutatás kevésbé hasznos a gyakorlati problémák megoldására, valamint, hogy az elmélet és gyakorlat közötti szakadék egyre növekszik.

Egyre több kritika éri az akadémiai tanulmányok eredményeit, amelyek nem hasznosak a gyakorló szakemberek számára, és nem kerülnek alkalmazásra. Pont ebből eredeztetve szeretném, hogy az ismertetett elméleti részek a továbbiakban gyakorlativá legyenek varázsolva, illetve a következő fejezet kifejezetten gyakorlati tudományos cikkekről fog szólni (Van De Ven, Paul E. Johnson,2006,802-804).

3 A következő generáció irányítói: Hopfield hálózatok a XXI. században

A *Hopfield* hálózatok napjainkra az informatika területén belül számos helyre beivódottak, felhasználási területeik számottevő, a következő részben a modern, illetve általam legérdekesebbnek vélt tanulmányok kerülnek majd bemutatásra.

3.1 Alacsony költségű digitális képalkotó és feldolgozó rendszer

A narancsfeldolgozó ipar a közelmúltban világszerte vertikálisan nőtt, miután a narancsból készült termékek olyan jövedelmező piacra kerültek, amely számos országban a gazdaság jelentős részét teszi ki. Általában a narancs osztályozása az iparágakban még mindig kézzel vagy drága technológiák alkalmazásával végzik. A legújabb kutatások célja az volt, hogy olyan rendszereket fejlesszenek ki, amelyek képesek e feladat elvégzésére. Ez a tanulmány egy alacsony költségű narancsosztályozó rendszer képfeldolgozással és mesterséges neurális hálózatokkal van megvalósítva, narancsok osztályozására és ezen gyümölcsök fő jellemzőinek előrejelzésére készült és azt írja le. Egy szisztematikus fényképészeti és módszertani eljárást alkalmaztak a képfeldolgozásra és a *Hopfield* rekurrens³¹ mesterséges neurális hálózatok megvalósításával egy megbízható kiválasztási rendszert kaptak.

A kapott eredmények elfogadható, átlagosan 85%-os százalékos arányt értek el a következők esetében: a helyes válaszokat, figyelembe véve a minőség és a méret kritériumait, ami azt jelenti, hogy a megvalósított rendszer hasonló vagy jobb eredményeket ér el, a hasonló munkákhoz képest. Emellett a gazdasági elemzés kedvező megtérülést mutatott ki, 3 és 5 hónap között, ami a megvalósíthatóságát igazolja. Összességében ez a munka biztosítja a hatékony narancskiválasztó rendszert, amely minimális emberi érintkezéssel és alacsony költséggel jár a narancsipar számára.

A továbbiakban a roncsolásmentes osztályozó és jellemző rendszerek fejlesztését tárgyalja a tanulmány, amelyek az ipari minőségi követelmények teljesítése érdekében érzékelőket és vezérlőeszközöket kombinálnak. Ezek a rendszerek, érzékelő eszközöket használnak a tárgyak valós képének rögzítésére és jellemzőik automatikus osztályozására.

³¹ Másnéven visszacsatolt. A *rekurrens* neurális hálózatok (*RNN*) olyan dinamikus rendszerek, amelyek hatékonyan használják fel a bemeneti sorozat időbeli információit, mind a klasszifikációhoz, mind pedig a kategorizáláshoz előrejelzésre. Ez azt jelenti, hogy a betanítás után a kölcsönhatások az aktuális bemenet és a belső állapotok közötti kapcsolatokat dolgozzák fel a kimenet előállításához és a belső állapotok releváns múltbeli információinak reprezentálására (Hüsken,2001,223).

A citrusfélék, mint a narancs, a mandarin, a citrom és a lime könnyen *hibridizálódhatnak* (keresztezhetőek), ami kihívást jelent a fizikai jellemzők alapján történő azonosításuk és osztályozásuk során.

A narancs, mint a legelterjedtebb citrusféle, világszerte számos fajtaival rendelkezik. A narancságazat az évek során jelentős növekedést tapasztalt, a legnagyobb termelők Brazília³², az Egyesült Államok és Kína. A narancs és a belőle készült termékek, mint például a gyümölcslé és az illóolajok, jövedelmező piaccal rendelkeznek, és jelentősen hozzájárulnak számos ország gazdaságához.

A narancsok homogenitásának és jó megjelenésének biztosítása érdekében elengedhetetlen a romlott vagy éretlen gyümölcsök korai felismerése. Ezek a hibás gyümölcsök befolyásolhatják a származtatott termékek ízét, és potenciálisan fertőzéseket terjeszthetnek más gyümölcsökre. A narancsok osztályozási szabványai általában a méretet, a színt és a minőséget veszik figyelembe, és az olyan hibák, mint a gombák, a foltok és a rothadás befolyásolják a minőségértékelést.

Hagyományosan a gyümölcsök válogatását és osztályozását kézzel végzik emberi kezelők, vagy olyan drága és nem teljesen pontos technológiák segítségével, mint az ultrahangos vizsgálat, a mágneses rezonanciás képalkotás és a komputertomográfia. A kézi osztályozás hibalehetőségekkel jár és többletköltségeket okoz, míg a fejlett technológiák magas költségei korlátozzák széles körű alkalmazásukat. Ezért pontos, gyors és alacsony költségű automatizált módszerekre van szükség a gyümölcsök válogatására és osztályozására.

A narancsok esetében képfeldolgozási technikák alkalmazhatók a gyümölcsben jelen lévő különböző színek intenzitási szintjeinek elemzésére, és a gyümölcs geometriai alakjának figyelembevételére az osztályozáshoz. A tanulmány a narancsosztályozó rendszer teljesítményének javítására *Hopfield rekurrens* modellen alapuló mesterséges neurális hálózatok (*ANN*) használatát javasolja. Ez a módszertan a gyümölcsök minőségi és

³² Brazília a világ legnagyobb narancstermelő országa, amelynek teljes ültetvényterülete meghaladja a több mint 820 000 hektárt. A termesztett narancs nagy részéből Braziliában (70%) fagyasztott narancssűrítményt állítanak elő (*FCOJ-frozen concentrated orange juice*) feldolgozó nagyvállalatok által. *Export* a teljes termelt *FCOJ* mintegy 97%-át teszi ki, ami Brazília a világ legnagyobb *FCOJ*-termelője és -exportőre. A brazil citruságazat adja a világ citrusfélékből előállított gyümölcsök felét, narancslé kínálatának felét, és a Braziliában forgalmazott gyümölcslé 80%-át (Mourad,2009,655).

méretparamétereinek elemzésére összpontosít, és alacsony költségű alternatívát javasol az összegyűjtött gyümölcsképek adatbázisának elemzésére.

Az *ANN*-alapú rendszert a hagyományos távolsági módszerekkel hasonlítják össze, hogy felmérjék a pontosságát és előnyeit. A cél egy pontos és költséghatékony képfeldolgozási technikákon és mesterséges neurális hálózatokon alapuló gyümölcskiválasztó rendszer kifejlesztése (Fermo, Cavali, 2021, 181-182).

3.1.1 Implementálás

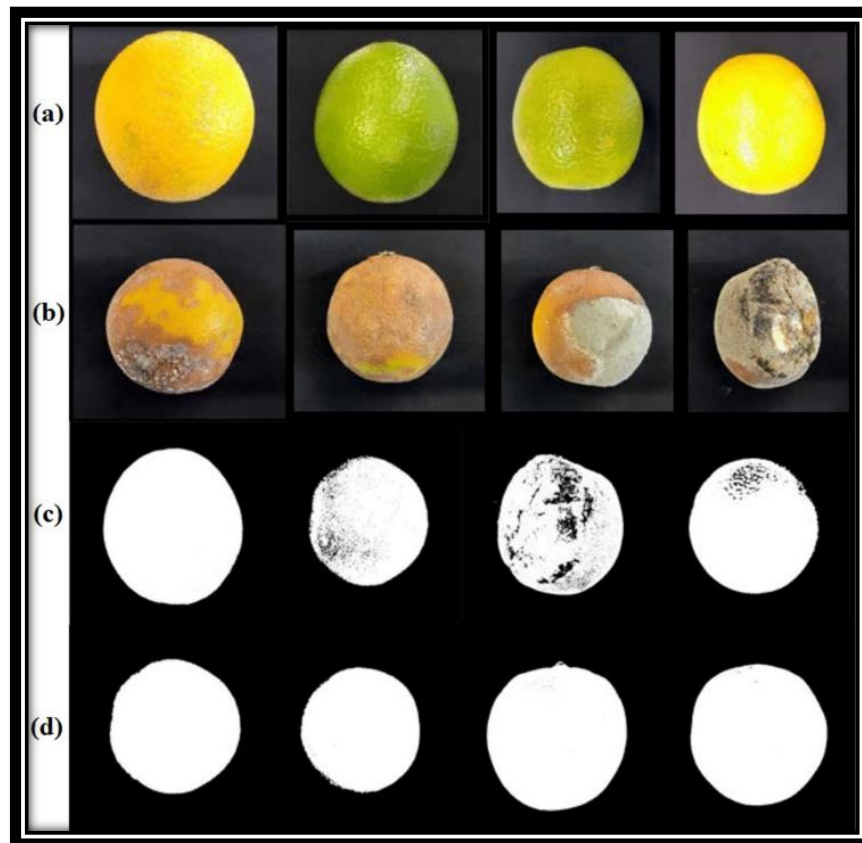
Az ebben a munkában javasolt módszertan a *Hopfield rekurrens ANN*-t használja egy pixel-összehasonlítási technika megvalósítására, amely képes a következőkre a narancssárga képek közötti hasonlóság elemzésére. Ez a megvalósítás egy algoritmus kifejlesztésével történik a *Matlab* szoftver segítségével.

Az esettanulmányban felhasznált narancsok a *citrus sinensis*, *osbeck* fajba tartoznak, és a *Pêra* fajtához tartoznak, amely a legfontosabb brazil citrusfajta. A begyűjtött mintákat a kísérletben a következő módszerekkel osztályozták: különböző méret, szín és simaság (a narancs felülete) szerint osztályozták, hogy minden hatást reprezentáljanak a fizikai jellemzőkből adódóan. Ezután a narancsokat a megfelelő romlási időnek megfelelően tartósították, hogy különböző romlott mintákat kapjanak.

Később a narancsokat háromszor fotózták le különböző konzerváltsági állapotban, hogy kialakítsák az elemzendő képi adatbázist a *Hopfield rekurrens ANN* segítségével. Ezután egy mobiltelefon digitális kamerájával, készültek a képek fekete háttér előtt. A fényképezőgép és a narancsok között 15,3 cm volt, a gyümölcsöket pedig úgy helyezték el, hogy mindig a kép közepén helyezkedjenek el. A megvilágításhoz egy *LED*-szalagot használtak körülbelül 32 cm távolságból ezek a körülmények mind kellettek a jó képminőség eléréséhez. Lépések:

- I. **Adatkészlet gyűjtése:** Kezdetben 59 különböző tulajdonságokkal rendelkező narancs képét rögzítették.
- II. **Adatbővítés:** Az osztályok kiegyensúlyozása érdekében adatnövelő algoritmust alkalmaztak az adathalmaz méretének növelésére.
- III. **Az adathalmaz felosztása:** Az adathalmazt két részre osztották, az egyiket a méret, a másikat a minőség szempontjából. Mindkét rész 100 narancsból állt, amelyeket egyenlően osztottak két osztályra (50/50).

- IV. **Képkivágás:** A rögzített képeket a *Matlab* segítségével egységesen levágtak, hogy kiválasszanak egy területet, amely a teljes narancsot mutatja.
- V. **Mintakiválasztás:** Két narancsot választottak ki mintának, az egyik a jó, a másik a romlott narancsot képviselte, mind a méret, mind a minőség szempontjából.



6. ábra: Példák jó (a) és romlott (b) narancsképekre, amelyeket az osztályozási folyamat során értékelték az osztályozás előtt (c) és után (d). **Forrás:** a felhasznált tanulmányból

- VI. **Méretosztályozás:** A méretosztályozáshoz a küszöbértéket 6,23 cm átlagos átmérőnél állapítottuk meg. Az átlagnál kisebb vagy azzal megegyező átmérőjű narancsokat normálisnak, míg az átlagnál nagyobb átmérőjűeket nagynak tekintették.
- VII. **Képfeldolgozás:** A *Matlab* képfeldolgozó eszköztárának segítségével elemezték a narancsok bináris képeit. A bináris képen a fehér (a narancs felületét jelképező) pixelek megszámlálását végezték el.
- VIII. **Méretmeghatározás:** A narancsonkénti fehér képpontok teljes számát összehasonlították az átlagossal. Az átlagnál nagyobb pixelszámú narancsokat nagynak, míg a többit kicsinek minősítették.

IX. **Képjavitás:** Morfológiai záróműveletet ³³ (*morphological closing operation*) alkalmaztak a narancshéj foltjainak és textúrahibáinak eltávolítására. Ez a művelet megszüntette a fekete foltokat a bináris képen belül (Fermo, Cavali, 2021, 183-185).

A 6. ábra (c) és a (d) példákat mutat a jó és a romlott narancsokon végzett képfeldolgozási műveletekre. Ezek a kezdeti lépések a narancsok előfeldolgozását és osztályozását célozták méretük és minőségük alapján, megalapozva ezzel a további elemzést és az osztályozási rendszerhez szükséges mesterséges neurális hálózat (*ANN*) kifejlesztését. A második szakasz a narancsok jellemzőik alapján történő osztályozására javasolt rendszer megvalósítási eljárását ismerteti, amely egy *Hopfield* rekurrens mesterséges neurális hálózatot (*ANN*) használ. Az alábbiakban összefoglaltam az érintett lépéseket:

- I. **A minták betöltése:** A rendszer a különböző értékelendő jellemzőket reprezentáló minták képének betöltésével kezdődik. Ezek a minták referenciaként szolgálnak az összehasonlításhoz.
- II. **Képkivágás:** Az eredetileg 3120×4160 képpont méretű betöltött képeket egy 2900×2200 képpont méretűre vágjuk. Ez a lépés a narancs részleteinek megtartására összpontosít, miközben a felesleges háttérrel eltávolítjuk.
- III. **Szürkeárnyaltos átalakítás:** A levágott képeket a *Matlab Image Processing Toolbox RGB2GRAY* függvényével 8 bites szürkeárnyaltúvá alakítjuk.
- IV. **Binarizálás:** A szürkeárnyaltos képek binarizálási folyamaton mennek keresztül a *Matlab imbinarize* függvényének használatával. Ebben a lépésben minden egyes képpontot 1-re (fehér) vagy 0-ra (fekete) az előtér polaritása alapján beállítunk. E paraméterek megfelelő értékeit kiértékeléssel határozzuk meg.
- V. **A kép átméretezése:** A binarizált képeket 40×40 pixel méretű mátrixra méretezzük át. A mátrix minden egyes pozíciója egy képpontot jelöl, és az értékek a binarizáció alapján 1 vagy -1 (fehér vagy fekete) értéket kapnak.
- VI. **Tömbi reprezentáció:** Az átméretezett mátrixot ezután egy 1600 pozíciót tartalmazó 1 dimenziós tömbre alakítjuk át. Ez a reprezentáció leegyszerűsíti az összehasonlítási folyamatot a következő lépésekben.
- VII. **ANN-elemzés:** A rendszer két külön *ANN* segítségével elemzi a képeket reprezentáló tömböket, egy minőségre és egy méretre vonatkozó *ANN* segítségével. A *Hopfield*

³³ A matematikai morfológia a képfeldolgozás matematikai megközelítése, amely a halmazelméleten alapul. Az 1985-ös év óta a képi ábrázoláshoz és leíráshoz hasznos geometriai jellemzők kinyerésének eszközévé vált. Általában a hézagok, hiányosságok kitöltése és a kép visszanyerése egyetlen lépéssel elvégezhető, a záró művelettel a morfológiában (Zang, 2009, 83).

rekurrens *ANN* pixelről pixelre értékeli a képeket, összehasonlítva azokat a hálózathoz korábban bemutatott mintákkal.

VIII. **Osztályozási eredmény:** A mintakép és a minták közötti hasonlóság alapján az *ANN* visszajelzést ad arról, hogy a gyümölcs jóváhagyott vagy elutasított. A rendszer 1 vagy 2 értéket jelenít meg, ami az 1. számú mintához (jó vagy nagy), illetve a 2. számú mintához (romlott vagy kicsi) való hasonlóságot jelzi.

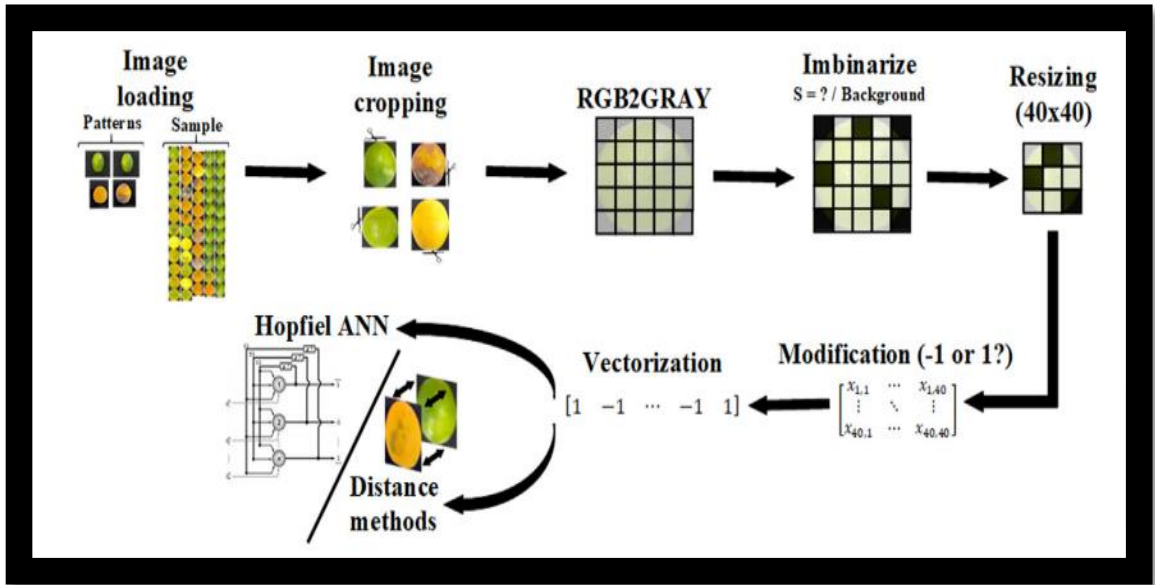
A javasolt rendszer célja a narancsfeldolgozó iparban az osztályozási folyamat automatizálása. A méretezett forgatókönyvben nyolc narancsadagoló szalagot automatizálnak, kamerák, mozgásérzékelő érzékelők, világító lámpák és csapóajtó-eltávolító mechanizmusok segítségével. Egy szalagot megtartanak a kidobott narancsok kézi szelektálására. A rendszer karbantartásához és fejlesztéséhez egy alkalmazott általi felügyeletre és irányításra van szükség.

A megtérülési idő, amely a kezdeti beruházás megtérüléséhez szükséges időt jelenti, körülbelül 3,5 hónapra becsülhető. Ez azt jelzi, hogy a tőkebefektetés viszonylag rövid idő alatt térül meg. A tanulmány az osztályozási rendszer teljesítményértékelését is tárgyalja. A *Hopfield* rekurrens *ANN* 85%-os pontosságot ér el mind a minőségi, mind a méretkritériumok tekintetében. Az osztályozási eredményeket összehasonlítják más, távolsági módszereken alapuló képhasonlósági metrikákkal, és a *Hopfield*-hálózat hasonló vagy jobb teljesítményt mutat.

Összességében a képfelismerésen alapuló javasolt osztályozási rendszer gazdaságos megvalósíthatóságot és kielégítő teljesítményt mutat. Olcsó alternatívát kínál az olyan fejlett technológiákkal szemben, amelyekhez drága berendezésekre, például *spektrofotométerekre*³⁴ és *hiperspektrális*³⁵ érzékelőkre van szükség. A módszer rugalmassága lehetővé teszi a különböző gyümölcsökhöz való könnyű alkalmazkodást és a változó minőségi szabványok vagy regionális eltérések alapján történő potenciális fejlesztéseket (Fermo, Cavali, 2021, 186-191).

³⁴ A spektrofotométer olyan műszer, amely a mintaoldaton való áthaladás után elnyelt fotonok mennyiségét (a fény intenzitását) méri. A spektrofotométerrel egy ismert kémiai anyag mennyisége másnéven koncentrációja is meghatározható az észlelt fény intenzitásának mérésével (Google, 2023).

³⁵ A *hiperspektrális* képalkotás az egész elektromágneses spektrumból gyűjti és dolgozza fel az információkat. A *hiperspektrális* képalkotás célja, hogy egy képnek minden egyes pixelére megkapjuk a spektrumot, azzal a céllal, hogy megtaláljuk a tárgyakat, azonosítsuk az anyagokat vagy felismerjük a folyamatokat (Wikipedia, 2023).



7. ábra: Az implementált rendszer **Forrás:** felhasznált tanulmányból

A munkában használt pixelenkénti összehasonlítási módszerek, amelyek közül párat a későbbiek folyamán mi is fogunk használni:

$$d_{st} = \sqrt[2]{(x_s - x_t)(x_s - x_t)^t}^{36}$$

$$d_{st} = \sum_{j=1}^N |x_{sj} - x_{tj}|^{37}$$

$$d_{st} = (\#(x_{sj} \neq x_{tj})/n)^{38}$$

$$PSNR = \frac{1}{mn} \sum_{i=1}^{M-1} \sum_{j=0}^{N-1} |I(i,j) - K(i,j)|^2^{39}$$

³⁶ *Euklideszi távolság:* A matematikában az euklideszi térben két pont közötti euklideszi távolság a két pont közötti vonalszakasz hossza. A pontok *Cartesianus*-koordinátáiból a *Pitagorasz*-tétel segítségével számítható ki, ezért időnként *Pitagorasz*-távolságnak is nevezik. Ezek az elnevezések az ókori görög matematikusoktól, *Eukleidész*től és *Pitagorasz*tól származnak, bár *Eukleidész* a távolságokat nem számként ábrázolta, és a *Pitagorasz*-tétel és a távolságszámítás közötti kapcsolat csak a 18. században jött létre (Wikipedia,2023). Ha például A koordinátái (3, 0) és B koordinátái pedig (0, 4), akkor a két pont közötti euklideszi távolság 5.

³⁷ A *Cityblock distance* metrika a pixelek közötti utat méri egy 4 összekapcsolt szomszédság alapján. Azok a pixelek, amelyek élei érintkeznek, 1 egység távolságra vannak egymástól; az átlósan érintkező pixelek 2 egység távolságra vannak egymástól (MathWorks,2023).

³⁸ A *Hamming*-távolság lényegében két karakterlánc azon szimbólumainak vagy pozícióinak száma, amelyekben a megfelelő karakterek eltérnek egymástól (Yang,2021).

³⁹ A *PSNR* kiszámítja a két kép közötti maximális jel-zaj arányt decibelben. Ez az arány az eredeti és a tömörített kép közötti minőség mérésére szolgál. Minél magasabb a *PSNR*, annál jobb a tömörített vagy rekonstruált kép minősége (MathWorks,2023).

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - Y'_i)^2 \quad 40$$

$$d_{st} = 1 - \frac{(x_s x'_t)}{\sqrt{(x_s x'_s)(x_t x'_t)}} \quad 41$$

$$SSIM = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)^{42}}{(\mu_x^2 \mu_y^2 + c_1)(\sigma_x^2 \sigma_y^2 + c_2)}$$

(Fermo, Cavali, 2021, 187).

Ez a tanulmány is mutatja, hogy sokszor a vélt fejlettebb eszközök, módszerek, technikák helyett a gazdaságosság felé kell hajlanunk és ez a példa is jól szemlélteti azt, ha kellő energiát fektetünk egy adott rendszerbe, az felveszi a versenyt a modernebbnek vélt rendszerekkel, a költséghatékonyságról pedig nem is beszélve.

3.2 Hopfield kaotikus neurális hálózaton alapuló színes képtitkosítási algoritmus

A következő tanulmány tovább szemlélteti a *Hopfield* hálózatok színességét. A bevezetőben említettekre reflektálva épp a kriptográfia terén alkalmazza a *Hopfield* hálózatokat.

A számítógépes ipar gyors fejlődésével, az interneten keresztül multimédiás információcsere egyre nagyobb mértékben növekszik, és az információk titkosságára vonatkozó követelmények is egyre nagyobbak. Mivel a digitális képek fontos részét képezik a multimédiás információcserének, a jó képtitkosítási algoritmus elengedhetetlen. Az utóbbi időben egyre több *kaotikus*⁴³ titkosítási rendszert használnak képtitkosításra.

⁴⁰ *Mean squared error*: A teljes átlagos négyzetes hiba kritériuma egy módja a becslések összehasonlításának az egy torzított és egy torzítatlan modell paramétereinek összehasonlítása (Dodge,2008,141).

⁴¹ *Cosine distance*: A koszinusz hasonlóság egy belső szorzatú tér két vektora közötti hasonlóságot méri. A két vektor közötti szög koszinuszával mérhető, és azt határozza meg, hogy két vektor nagyjából ugyanabba az irányba mutat-e. A szövegelemzésben gyakran használják a dokumentumok hasonlóságának mérésére (Han,2012).

⁴² *SSIM*: Az *SSIM Index* minőségértékelési index három kifejezés, nevezetesen a fényerősség, a kontraszt és a strukturális kifejezés kiszámításán alapul. A teljes index a három kifejezés többszörös kombinációja (MathWorks,2023).

⁴³ Teljesen zavaros vagy rendezetlen.

Fontos volt a tanulmány szerzőinek elvégezniük véletlenszerűségi tesztet, amely eredményei azt mutatják, hogy a *Hopfield* kaotikus neurális hálózat, a súlymátrix alapján *pszeudo-véletlenszerűséggel*⁴⁴ rendelkezik.

Ez a cikk a *Logistic map*⁴⁵ (logisztikus térkép) felosztását és a *Tent map*⁴⁶ (sátortérképpel) való kombinálását javasolja, hogy egy új, fázisos kaotikus térképet alkosson.

3.2.1 Titkosítási folyamat lépései

- I. **A kulcs inicializálása:** Az algoritmushoz több kulcsparaméterre van szükség, többek között. Ezek a paraméterek a kaotikus térképek és a *Hopfield* neurális hálózat vezérlőértékeiként és kezdeti feltételeiként működnek.
- II. **Confusion Process** (Zavarodási folyamat): A kulcsparamétereket az iteratív érték előállítására használjuk. Ezután ezt az értéket kiterjesztjük és transzformáljuk, hogy a kiterjesztett F-transzformáció segítségével p és q értékeket kapjunk.
- III. **Arnold Cat Map**⁴⁷ **transzformáció:** A zavarási folyamatból kapott p és q értékeket egy *Arnold Cat* térkép transzformáció paramétereiként használjuk, amely a bemeneti kép pixeleit keveri.
- IV. **A színes kép keverése:** A színes kép *RGB*-csatornás jeleit bevezessük a rendszerbe, és az *Arnold Cat*-térkép transzformációval kapott transzformált képet tekintjük a titkosított színes képnek. Az így kapott mátrixot S -nek jelölték.
- V. **Diffúziós folyamat:** A kulcsparaméterek segítségével és azok iterálása után. Ezeket az értékeket ezután a *Hopfield* kaotikus neurális hálózat kezdeti feltételeiként használjuk. Egy $N \times N$ iterációt hajtunk végre, amelynek eredményeképpen az XR , XG és XB szekvenciákat kapjuk. Az X mátrixot ezen szekvenciák felhasználásával képezzük, hasonlóan az S mátrixhoz.

⁴⁴ Szekvenciák, amelyeket determinisztikus algoritmusok generálnak, hogy szimulálják a valóban véletlenszerű szekvenciákat, *pszeudorandom* (*PR*) szekvenciáknak nevezünk. Egy *pszeudorandom* sorozatot a $[0, 1]$ egységnyi intervallumban *pszeudorandom* számok sorozatának (*PRN*) nevezzük (Topuzoğlu, 2006, 135).

⁴⁵ A logisztikus térkép egy nemlineáris dinamikus egyenlet. A logisztikus modellt eredetileg *Pierre François Verhulst* demográfiai modellként vezette be (Rani, 2009, 447).

⁴⁶ A logisztikus térképhez hasonlóan a sátortérkép is egy egydimenziós, diszkrét idejű kaotikus rendszer, amely számos területen alkalmazható (Luca, 2009).

⁴⁷ Az *Arnold Cat* térkép a hiperbolikus struktúrából adódó kaotikus viselkedés jól ismert prototípusa. Valójában ez egy *Bernoulli*-rendszer. *Arnold* macskatérképe egy kaotikus térkép, *Vladimir Arnold*-ról nevezték el, aki az 1960-as években egy macska képével demonstrálta hatását, innen eredeztethető maga a név (Antoiou, 1997, 78-80).

VI. **Kulcsfolyam generálása:** Az X mátrixot feldolgozzuk, a kulcsfolyam létrehozásához, a kulcsfolyamot minden egyes színcsatornára (R, G, B) és minden egyes pixelhelyezetre (i, j) megkapjuk.

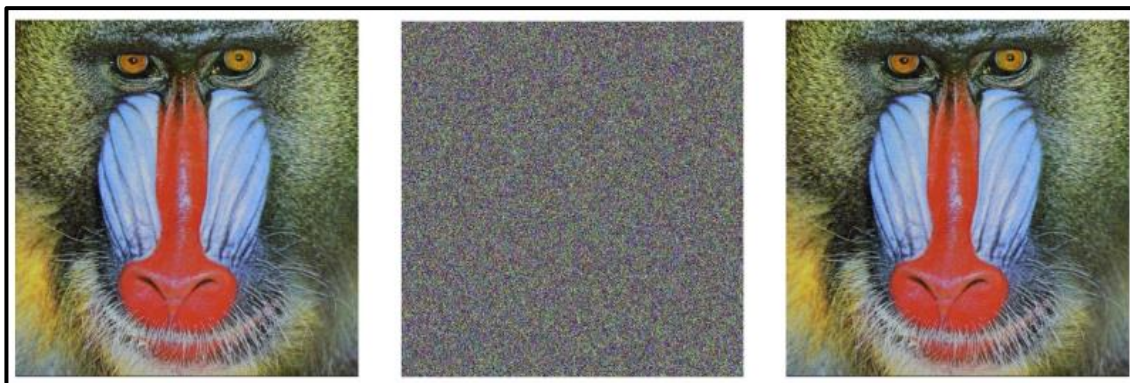
VII. **Bitenkénti XOR a titkosított képpel:** A kulcsfolyam mátrix és az S kódolt képmátrix elemeit bites XOR-ozással a kódolt mátrixot kapjuk meg.

A titkosítási folyamat ezen a ponton befejeződik. A visszafejtéshez ugyanazokat a kulcsparamétereket használjuk, és a titkosított képet XOR-ozzuk a kulcsfolyamhoz, hogy visszanyerjük a titkosított képet. Ezután az *Arnold Cat*-térkép inverz transzformációját alkalmazzuk a p és q paraméterek felhasználásával, hogy megkapjuk az eredeti képet (Wanga,2019,107-115).

A szöveg ismertet néhány kísérleti eredményt és a titkosítási algoritmus biztonsági elemzését is, beleértve a kulcssterelemzést, az érzékenységelemzést, a statisztikai elemzést, a támadáselemzést és az információs entrópielemzést. Ezen elemzések célja a titkosítási algoritmus biztonságának és hatékonyságának értékelése különböző támadásokkal és statisztikai elemzési technikákkal szemben.

3.2.2 Legfontosabb megállapítások

Az algoritmust a vágási és zajtámadásokkal szembeni ellenálló képességét értékelték. A különböző pozíciókban és sebességgel ($1/16$, $1/4$ és $1/2$) végrehajtott vágási támadások esetén a titkosítási algoritmus jó vágási ellenállást mutat. Még akkor is, ha a vágási arány eléri az $1/2$ -et, a kép fő tartalma látható marad. Az algoritmus a zajtámadásokkal szemben is jó ellenállást mutat, különösen a *Gauss*-zajjal és a só- és bors-zajjal szemben. Összességében a kísérleti elemzések, beleértve a statisztikai információelemzést, a differenciális támadáselemzést, a vágási támadást és a zajtámadást, bizonyítják a javasolt algoritmus fölényét és a különböző támadásokkal szembeni ellenállását. Egyszóval az algoritmus ígéretes eredményeket mutat a biztonság és a különböző támadásokkal szemben (Wanga,2019,115-118).



8. ábra: Az eredeti kép, a rejtjelezett kép, illetve a visszanyert kép **Forrás:** tanulmányból

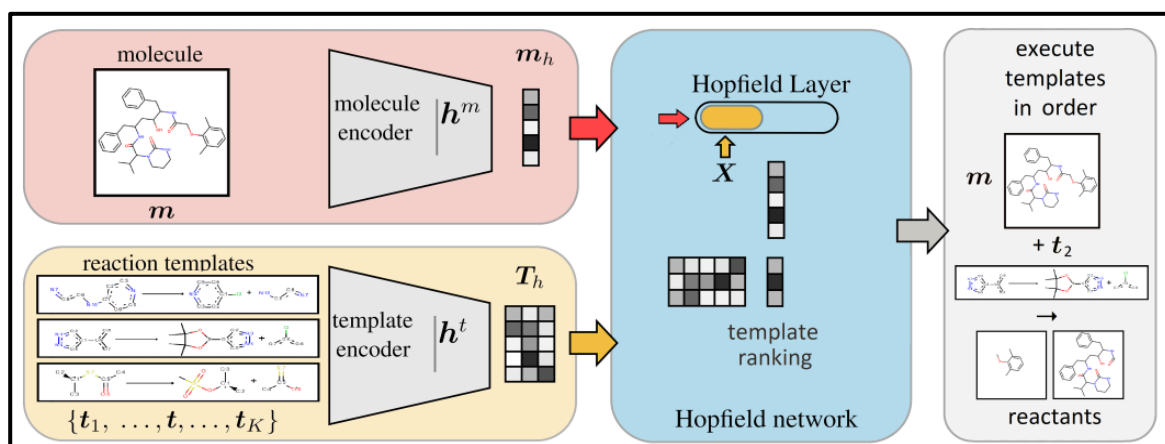
Az internet és a multimédiás rendszerek, szociális platformok gyors növekedése és fejlődése miatt a közelmúltban nagymértékben megnőtt a képek és videósorozatok használata. A biztonságos, hatékony és rugalmas kommunikációval kapcsolatban számos tanulmány született.

A multimédiás adatok védelmére a bizonyítható biztonságú teljes titkosítás a legbiztonságosabb megoldás. Számos multimédiás alkalmazás azonban kompromisszumot keres a biztonság terén, hogy más követelményeket is lehetővé tegyen, pl. alacsony feldolgozási igényeket, a bitfolyam-megfelelőség megtartását (Chuman,2017,858). Nyilván nem a mi posztunk álláspontot foglalni, de ebből is látható, hogy van piaca és fontossága a titkosításnak, talán még ennyire sosem volt, mint napjainkban. Ebből kifolyólag a fenti kép titkosítási tanulmány is mutatja, hogy van ennek fontossága, az csak pozitívumnak tekinthető, hogy a *Hopfield* hálózatok segítségével hajtották végre a titkosítási folyamatot.

3.3 Kemoinformatika és a Hopfield hálózatok

Az informatika a tudományágak között növekszik, mondhatni sok helyre beivódik és hatással van a kémia, a biológia és az orvosbiológiai tudományok számos területére. A jól bevált bioinformatikai diszciplína mellett az idők során más informatikai alapú interdiszciplináris területek is fejlődtek, mint például a kemoinformatika és az orvosbiológiai informatika. További kapcsolódó kutatási területek, mint a farmakoinformatika, élelmiszerinformatika, anyaginformatika és neuroinformatika újabban jelentek meg, és önálló részdiszciplínákként fejlődnek tovább (López-López,2020,26). A kemoinformatika egy felhasználási példáját mutatja be az alábbi fejezet is és egy újabb alkalmazási módját a *Hopfield* hálózatoknak.

A neurális hálózatok és az informatika terjedését minden tudományágon belül szemlélteti a következő etalon is, amely, a számítógépes szintézistervezés új modelljét írja le a kémiában. A molekulák szintézisútvonalainak megtalálását foglalja magában, ami egy összetett folyamat. A javasolt modell mintaalapú megközelítést alkalmaz, és egy modern *Hopfield*-hálózatot (*MHN*) használ a releváns minták és a termékmolekulák összekapcsolására. A korábbi *CASP*⁴⁸-módszerek szabályalapú szakértői rendszereket vagy rögzített reakciókon betanított gépi tanulási módszereket használtak a kémiai reaktivitás modellezésére. Ezeknek a módszereknek azonban vannak korlátait, például, a kevés adatot tartalmazó rendszerek és a ritka reakcióminták esetében gyenge a teljesítmény. A javasolt modell ezeket a kihívásokat egy mintaalapú megközelítés és egy modern *Hopfield* hálózat alkalmazásával kezeli. Ahelyett, hogy a mintákat külön kategóriaként kezelné, a modell megtanulja a releváns mintákat a termékmolekulához társítani, a molekulák és a minták kódolásával és reprezentálásával. Ez lehetővé teszi a minták feletti általánosítást és javítja a teljesítményt, különösen a kevés gyakorló mintával rendelkező vagy nem látott minták esetében. A modellt *sztochasztikus gradiens* ereszkedéssel optimalizálták. A javasolt modell javulást mutat a prediktív teljesítményben, és képes kezelni az alacsony adatmennyiséget, így ígéretes megközelítés a számítógépes szintézistervezéshez a kémiában (Seidl, 2023,4-28).



9. ábra: A rendszer felépítése **Forrás:** az említett tanulmány

⁴⁸ A szerkezet-előrejelzés kritikus értékelése (*CASP- Critical Assessment of Structure Prediction*) kísérlet. A *CASP* egy olyan szervezet, amely közösségi szintű kísérleteket végez, például a fehérjék szerkezetének aminosav-szekvenciából történő modellezése terén elért eredmények mérésére. A *CASP* alapelve a szerkezet-előrejelzési módszerek teljesen vakon történő tesztelése, és ezt a *CASP* 1994 óta kétévente végzi (Mault,2013,1-2).

„Az informatika nem szól a számítógépekről, mint ahogy a csillagászat a távcsövekről, a biológia a mikroszkópokról vagy a kémia a mérőpoharakról és kémcsövekről. A tudomány nem az eszközökről szól. Hanem arról, hogyan használjuk őket, és mit tudunk meg, amikor használjuk őket.”⁴⁹

Edsger Dijkstra

A sajátos értelmezésben a fenti idézet kellően reprezentálja a tudományok közti átmenet szépségét, illetve ez esetben az eszközökre úgy tekintenek, mint egyes tudományok eszközeinek az áthidalására más tudományterületekre. Amely a XXI. században kifejezetten kurrens módon jelen van, hogy az informatika részese minden tudománynak, ez esetben épp a kémia szakemberei hívták segédletül eme informatikai eszközöket.

3.4 Végezetül, de nem utolsó sorban

A fejezet keretein belül számos alkalmazását a *Hopfield* hálózatoknak (főként a gyakorlatban) „láttuk vendégül” és ismertettük az olvasóval, a kriptográfia, kemoinformatika vagy a képfeldolgozás terén. A bemutató utolsó alfejezete a nem megemlített felhasználási módszereket hivatott ismertetni rövidebb és tömörebb alakban.

3.4.1 Hopfield hálózat, mint optimalizáló algoritmus

A *Hopfield* neurális hálózat akár statikus optimalizáló eszközként is alkalmazható. A hálózat súlyainak megtanulására, kiküszöbölve a manuális súlymeghatározás szükségességét. A módszertan gradiens ereszkedést használ a kényszer súlyozási együttthatótérben az együttthatók frissítésére és az új súlyértékek kiszámítására. Az adaptációs eljárás sikeresen vezeti a hálózatot az utazóügynök probléma, lokális optimális megoldásai felé. Egyfajta adaptációs módszert mutat be a *Hopfield* hálózat statikus optimalizálási problémákhoz való konfigurálására. A módszer kiküszöböli a kézi hangolás szükségességét azáltal, hogy automatikusan meghatározza a súlyozási együttthatók megfelelő értékeit. Az eredmények azt mutatják, hogy az adaptív *Hopfield* hálózat jobb

⁴⁹ Eredeti: Computer science is no more about computers than astronomy is about telescopes, biology is about microscopes or chemistry is about beakers and test tubes. Science is not about tools. It is about how we use them, and what we find out when we do (quotlr.com,2023).

megoldási minőség mellett képes konvergálni a lokális optimális megoldásokhoz. Mindez hozzájárul a *Hopfield* hálózatok mint optimalizációs algoritmus alkalmazhatóságának növeléséhez, és automatizált megközelítést biztosít a hálózat konfigurálásához (Serpén,2008,2-13).

3.4.2 Kapacitás növelése?

A kvantum *Hopfield* hálózatok új megközelítésére és tárolókéességének tárgyalására is napjainkban sorkerült. Az efféle modell elsődleges célja, hogy a kvantumszámítási elvek kihasználásával javítsa a klasszikus *Hopfield* hálózatok tárolókapacitását.

A hagyományos *Hopfield* hálózatoknak tárolókapacitása korlátozott, különösen akkor, ha véletlenszerűen választunk ki mintákat. A javaslat erre az lenne, hogy a *Hopfield* hálózatok kvantumtartományba való átültetésével a tárolókapacitás jelentősen növelhető. Ha a *Hopfield* hálózatot úgy kezeljük, mint egy *feed-forward* hálózatot akkor a *feed-forward* értelmezés lehetővé teszi a klasszikus neurális hálózatok kvantum analógjainak megvalósítását. Ezen kvantum *Hopfield* hálózatoknak szimulációja során klasszikus memóriamintákból és e minták zajos változataiból álló tréningkészletet hoznak létre. A gyakorlóhalmaz segítségével becsüljük meg a teljesítményt, és iteratív módon frissítjük azokat. Az eredmények azt mutatják, hogy a kvantum *Hopfield* hálózat több mintát képes tárolni, mint a klasszikus *Hopfield* hálózat a *Hebb* tanulással. A hibaarányok a tárolt minták számával nőnek, de a kvantum *Hopfield* hálózatok esetében viszonylag alacsonyak maradnak a klasszikus *Hopfield* hálózatokhoz képest. A számítási költségek miatt azonban nagyobb rendszerméreteket nem lehetett szimulálni. A betanított kvantummodellben rejlő lehetőségekben és a jövőbeli kutatási irányok a kvantum *Hopfield* hálózatok irányába látják a jövőt a kutatók (Niels,2020,576-589) &(Valeria,2020). Talán a neurális hálózatok jövője is ebben rejlik, bátorkodásul a lent látható idézettel zárnám ezt a gondolatmenetet:

„Az emberi agy 100 milliárd neuronjának viselkedését klasszikus számítógéppel nem lehet szimulálni, de a kvantum gépi tanulás ígéri, hogy teljesíti ezt a követelményt.”⁵⁰

Amit Ray

⁵⁰ Eredeti: “Simulating the behavior of 100 billion neurons of human brain is not feasible by classical computer but quantum machine learning promises to fulfill that requirement (goodread,2023).”

4 Hopfield hálózatok implementálása a gyakorlatban

Az elméleti rész lezárta után, amely egy átfogó képet biztosított számunkra, diplomamunkánk második része az általam implementált, illetve megvalósított programok ismertetéséből fog állni. A kezdeti példák tekinthetőek általánosabbnak, melyek főleg a mintafelismerésre fognak fókuszálni.

4.1 Fejlesztői környezet

Az annotációnk részeként, illetve munkánk során említésre került fejlesztői/szoftver környezetünk a már ismertetésre került *MATLAB®* lesz. Azon belül is a *MATLAB® R2021b* verziójában folytattam, illetve implementáltam program kódjaimat, amelyet a legvégén integráltam egy egységes applikációvá.

Fontosnak tartom kiemelni a verziószámot, mivel egyes funkciók, illetve függvények másként fordulhatnak elő más verziószámokban és az esetleges nehézségek elkerülése miatt érzem fontosnak a kihangsúlyozását.

Idővel fejlődött ki a mostani stádiumává a *MATLAB®*. Kezdetekben az első *MATLAB®* nem programozási nyelv volt, hanem egy egyszerű interaktív mátrixszámológép. Nem voltak programok, nem voltak eszköztárak, nem volt grafika.

A *MATLAB* első verziójának matematikai alapja *J. H. Wilkinson* és 18 kollégája 1965 és 1970 között publikált, később a *Wilkinson* és *C. Reinsch* által szerkesztett *Handbook for Automatic Computation, Volume II, Linear Algebra* című kézikönyvben összegyűjtött kutatási cikkei voltak. Ezek a dolgozatok *Algol 60*-ban implementált algoritmusokat mutatnak be mátrix lineáris egyenlet és sajátérték problémák megoldására (Moler,2018). Valahol itt kezdődött a szoftver múltja, hasonlatosan, mint az általam ismertetett neurális hálózatok technológiai forradalma fejezet keretein belül egy hatalmas mondhatnám monumentális dologgá vált megannyi felhasználási területtel.

Ez okból tökéletes környezetet ad/adott számunkra diplomamunkánk gyakorlati részének megvalósítására. Továbbiakban nem is szaporítanám a szót térjünk rá az implementációkra.

4.2 Bevezető példa

A bemeneti mintákat egy „*patterns*” nevű mátrixként definiáljuk. A mátrix minden sora egyetlen mintát, az oszlopok pedig az egyes minták egyes elemeit jelölik. A hálózat súlyait nullára inicializáljuk, majd a *Hebb* tanulási szabály segítségével betanítjuk. Ennek során a súlyokat a bemeneti minták közötti korreláció alapján frissítjük. Az így kapott súlymátrixot a „*weights*”(„súlyok”) változóban tároljuk. Kiválasztunk egy bemeneti mintát a minták halmazából, majd zajjal korruptáljuk, hogy létrehozzunk egy „*noisy_input*” mintát. Fontos még kiemelni. Ha x_i és x_j a neuronok állapotai i és j , illetve a képzési mintákban, akkor:

$$W(ij) = \sum_P x_i^{(P)} \cdot x_j^{(P)}$$

Ahol az összeg az összes P képzési mintára vonatkozik. A súlymátrix átlós elemeit nullára állítjuk ($W(i,i) = 0$), hogy elkerüljük az önkapcsolatokat.

A *Hopfield* hálózatot ezután iteratív módon frissítjük, amíg egy stabil kimeneti mintához nem konvergál. Minden egyes iterációnál a kimeneti mintát az aktuális súlyok és az előző kimeneti minta alapján frissítjük. A zajos bemeneti és a kimeneti minta közötti *Hamming*-távolságot minden egyes iterációnál kiszámítjuk és egy tömbben tároljuk. A kimeneti mintát és a *Hamming*-távolságot az iterációk során ábrázoljuk. Ezt egymás után öt másodperc eltéréssel kétszer futtassuk le. A második futás egy kétszer akkora adathalmazra szól, de ugyanazzal a technikákkal, algoritmusokkal járunk el ennél is.

A *Hamming* távolságból (lásd 10. ábra) illetve az iterációk értékeiből is (lásd 12. ábra) látható, hogy neurális hálózatunk megtalálta vagy mondhatnám helyreállította a zajos ábránkat.

Ámbár 35 kocka-felbontású ábrára kiválóan működik algoritmusunk (akár 90%-os zaj esetén is), emeljük a tétet és duplázzuk meg az ábránkat 70 kocka-felbontásúvá.

The screenshot shows a software window titled 'Figure 1' with a menu bar (File, Edit, View, Insert, Tools, Desktop, Window, Help) and a toolbar. The main area displays three binary images:

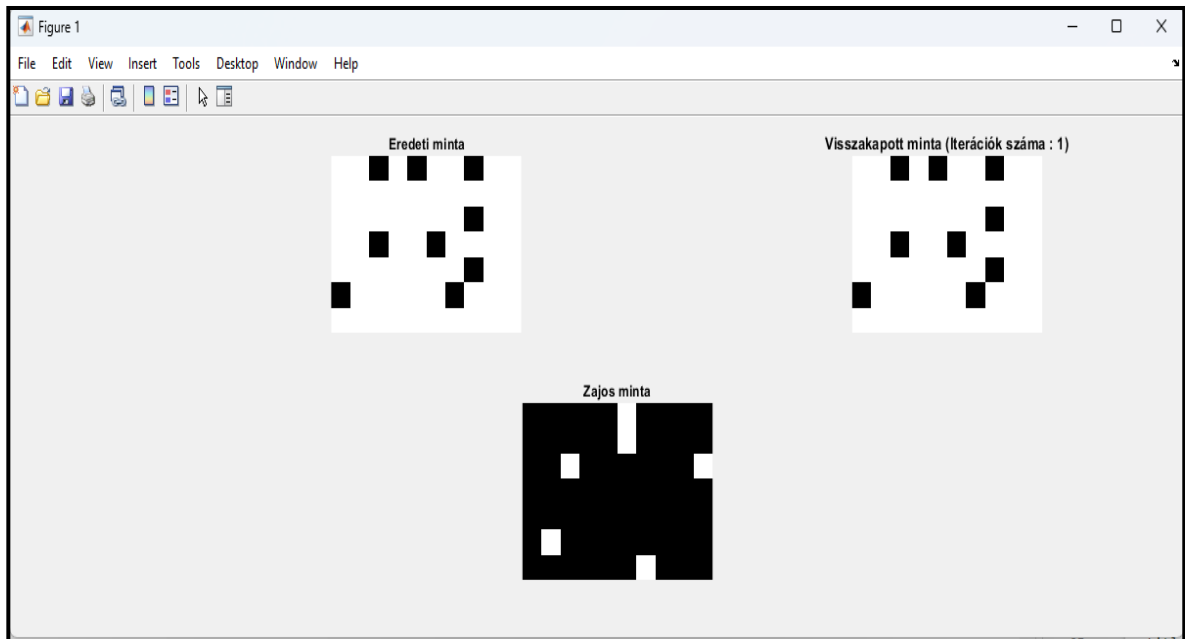
- Eredeti minta**: A 10x10 pixel binary image showing a complex, noisy pattern of black and white pixels.
- Visszakapott minta (Iterációk száma : 4)**: A 10x10 pixel binary image showing a complex, noisy pattern of black and white pixels, similar to the original sample.
- Zajos minta**: A 10x10 pixel binary image that is almost entirely black, with a single white pixel in the top right corner.

The screenshot shows a MATLAB window titled "Figure 2". It contains two subplots. The left subplot, titled "Visszakapott minta (Végleges)", displays a 10x10 binary matrix with a complex pattern of black and white pixels. The right subplot, titled "Hamming Távolság (Iterációk száma : 5)", shows a line graph of Hamming distance versus iteration number. The y-axis is labeled "Hamming Távolság" and ranges from 0 to 35. The x-axis is labeled "Iterációk száma" and ranges from 0 to 100. The graph shows a sharp drop in distance from approximately 7 at iteration 0 to 0 at iteration 5, with subsequent iterations remaining at 0.

Iteráció 1, Kimenet:	[0 -1 0 0 0]
Iteráció 2, Kimenet:	[-1 1 -1 0 0 0 0 -1 -1 0 -1 0 1 -1 -1 -1 -1 1 1 -1 1 1 -1 -1 -1 1 -1 1 1 -1]
Iteráció 3, Kimenet:	[-1 1 -1 -1 1 -1 1 -1 -1 1 -1 1 1 -1 -1 -1 -1 1 1 -1 1 1 -1 -1 -1 1 -1 1 1 -1]
Iteráció 4, Kimenet:	[-1 1 -1 -1 1 -1 1 1 -1 1 -1 1 1 -1 -1 -1 -1 1 1 -1 1 1 -1 -1 -1 1 -1 1 1 -1]
Iteráció 5, Kimenet:	[-1 1 -1 1 1 -1 1 1 -1 1 -1 1 1 -1 -1 -1 -1 1 1 -1 1 1 -1 -1 -1 1 -1 1 1 -1]

45

Sok minden függ a tanítási halmaztól is, mennyire jó az adat, nincs e túl sok rossz adat, így például nem a pontos mintát kapjuk vissza. Íme a 7*10 kimenete:



13. ábra: A mintáink reprezentálása **Forrás:** saját szerkesztés

Ennek ellenére egy hasonló de mégis különböző algoritmust készítettem, amely bemenete a *character recognition plotchar* a *MATLAB*-ban *prprob()*⁵¹ lesz amely mellett, tetszés szerint annyi betűt zajosíthatunk el amennyi csak definiálva van, illetve kiválóan szemlélteti majd számunkra a *Hopfield* hálózatok kapacitásának korlátosságát (Beale,2004a).

4.3 Hopfield hálózatok kapacitása

Alapos kutatás után arra az elhatározásra jutottam, hogy a bevezető példák keretein belül mindenképp beszeretném mutatni a mintázat felismerést, illetve a *Hopfield* hálózatok kapacitásának korlátát mielőtt a kevésbé megszokott „vizekre eveznénk”. A tanulmányokban rendszerint számokon kerülnek szemléltetésre, valamint jellemzően bemutatásra, pont ebből kifolyólag én a *Matlab* által definiált *prprob*⁵² mintahalmazt vettem alapul, amely kifejezetten az angol *abc*-n alapszik.

⁵¹ Mátrix 26 oszloppal, egy az ábécé minden egyes betűjéhez.

⁵² PRPROB Character recognition problem definition, Mark Beale, Copyright 1992-2002 The MathWorks, Inc. ALPHABET - 35x26 mátrix 5x7 bittérképekből álló mátrix minden egyes betűhöz. TARGETS - 26x26 célvektor.

4.3.1 Program felépítése

Karakter meghatározás: Az ábécé 26 karakterét (A-tól Z-ig) 5x7-es bittérképként definiálja. Minden karakter 0 és 1 mátrixként van ábrázolva, ahol a 0 egy fehér pixelt, az 1 pedig egy fekete pixelt jelent. Ezek a karakterminták az ábécé mátrixban vannak tárolva.

Célvektorok: A *Hopfield* hálózat képzésének megkönnyítése érdekében a kód 26 célvektort definiál, amelyek mindegyike az ábécé egy-egy karakterét képviseli. A célvektorokat a célmátrixban tároljuk, ahol minden sor egy adott karakterhez tartozó célvektornak felel meg.

Hopfield neurális hálózat beállítása: Létrehozunk és betanítunk egy *Hopfield* neurális hálózatot. A karaktermintákat bemenetként és a megfelelő célvektorokat kimenetként használva. A *newhop* függvényt a *Hopfield* háló inicializálására, a *sim* függvényt pedig a megadott mintákkal és célpontokkal (*targets*) történő betanítására használjuk.

Karakterfelismerés zajjal: A kód zajt visz be a karaktermintákba, majd teszteli a *Hopfield* hálózat képességét a zajos karakterek felismerésére. A zaj mennyiségét a zaj változóval szabályozzuk, amely megadja az egyes karakterekben a véletlenszerű pixelfordítások számát.

Az eredmények vizualizálása: Ezután vizualizálja az eredeti karaktereket, a zajos karaktereket és a *Hopfield* hálózat által rekonstruált karaktereket. Minden egyes iterációhoz (betűk száma) három karakterkészletet jelenít meg egymás mellett, külön ábrákon:

- a. **Eredeti karakterek:** Megjeleníti a *Hopfield* hálózat képzéséhez használt karaktermintákat.
- b. **Zajos karakterek:** A *Hopfield* hálózat teszteléséhez használt, hozzáadott zajjal ellátott karaktereket jeleníti meg.
- c. **Rekonstruált zajos karakterek:** A *Hopfield* hálózat által a zajos bemenetből rekonstruált karakterek megjelenítése.

Teljesítményértékelés: Minden egyes iterációra kiszámítja az eredeti karakterek és a *Hopfield* hálózat által rekonstruált karakterek közötti pixelkülönbségek számát (teljesítményhibák). Ezeket a teljesítményhibákat a *teljesitmenyhiba* tömbben tárolja.

Tárolási kapacitélelemzés: A teljesítményhibákat a *Hopfield* hálózat képzéséhez használt betűk számának függvényében ábrázolja. Ez a grafikon betekintést nyújt a *Hopfield* hálózat tárolókapacitásába, azaz abba, hogy hány betűt képes pontosan felismerni a betanítási minták száma alapján. A teljesítményelemzés segít megérteni a hálózat korlátait és tárolni, illetve pontosan felidézni képes minták számát illetően (Beale, 2004b).

4.3.2 Felhasznált képletek

Súlymátrixra:

$$W = \sum_i^P x_i x_i^T - P \cdot I$$

ahol W a súlymátrix, P a minták száma, x_i az i -edik minta, és I az azonossági mátrix.

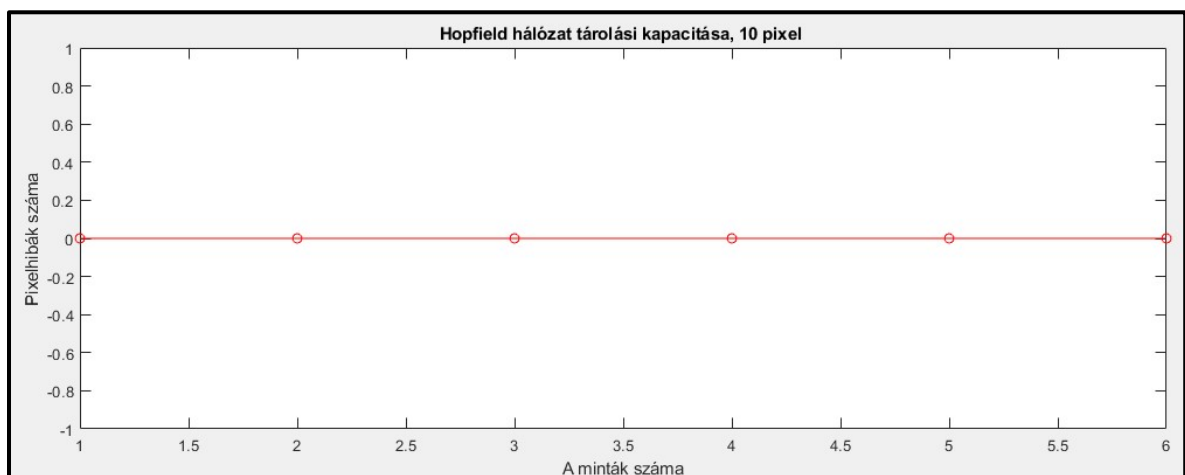
A *Hopfield* hálózat frissítési szabálya:

$$x_j^{(t+1)} = \text{sgn} \left(\sum_i^N w_{ij} \cdot x_i^{(t)} \right)$$

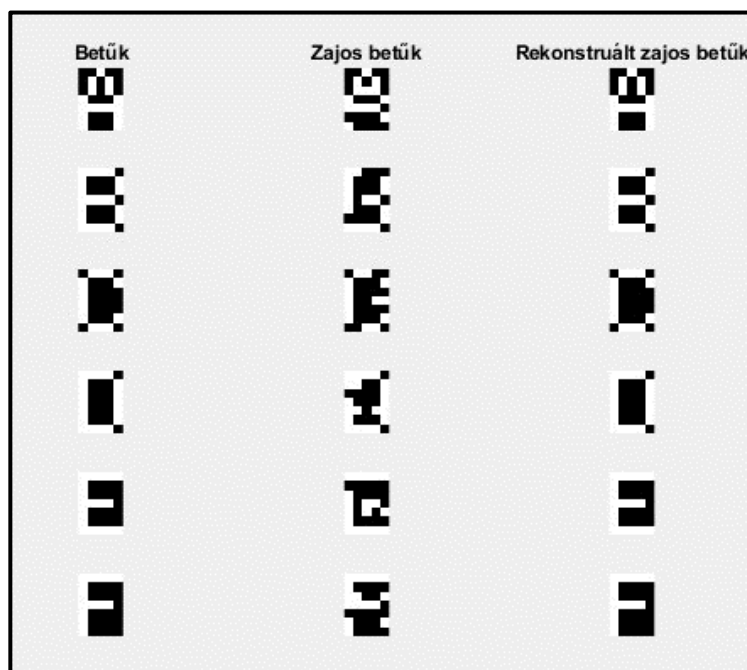
ahol a $x_j^{(t+1)}$ a neuron frissítése a j időpillanatban $t + 1$, N a neuronok száma, w_{ij} az i és j neuronok közötti súly és sgn a *signum* függvény.

4.3.3 Program kimenete

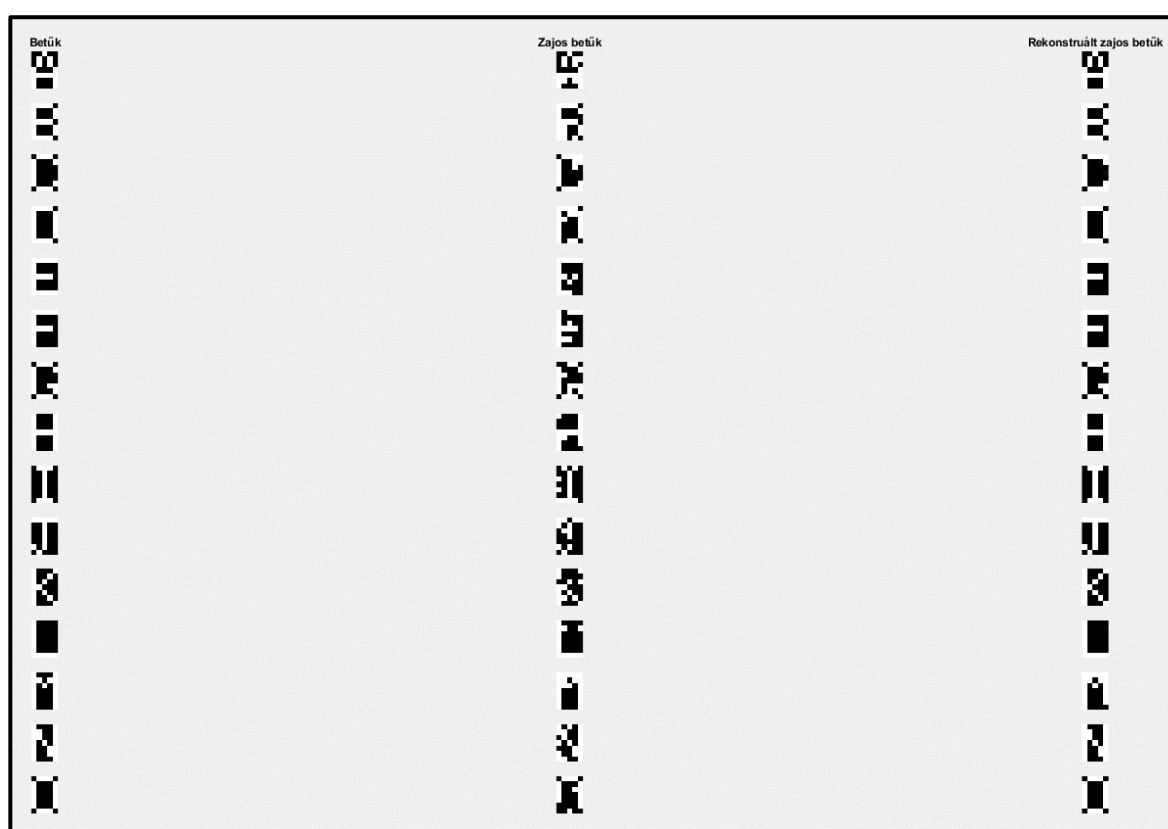
A program hasznossága főleg a *Hopfield* hálózatok kapacitásának szemléltetésében látom.



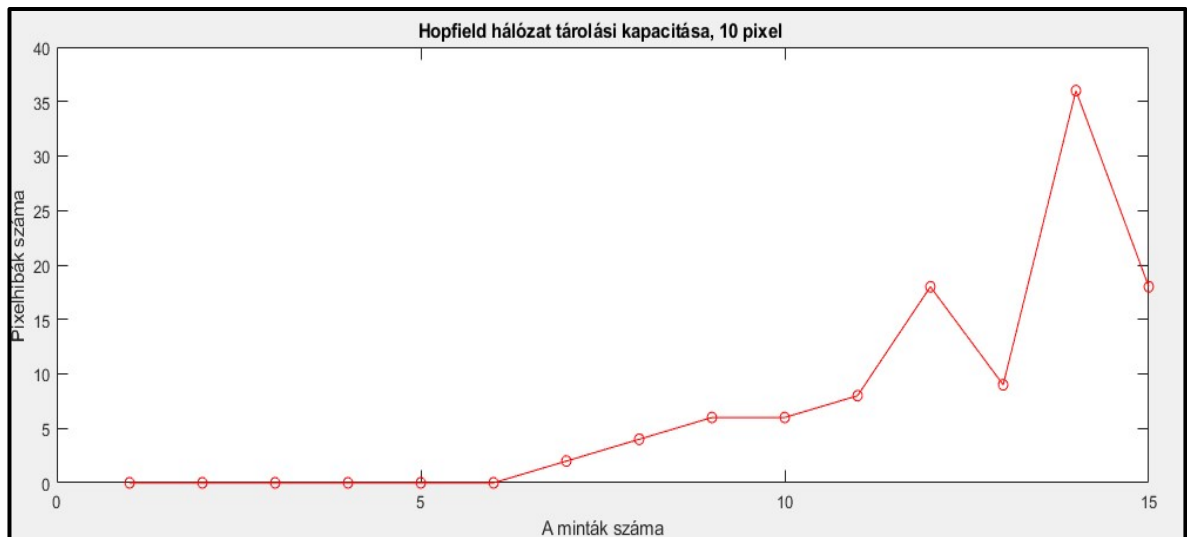
14. ábra: A kapacitás maximális kihasználása, pixelhiba nélkül **Forrás:** saját szerkesztés



15. ábra: Rekonstruált betűk, pixelhiba nélkül **Forrás:** Saját szerkesztés



16. ábra: 15 beolvasott betűnél a kapacitás megtelik ezért több pixelhiba keletkezik
Forrás: Saját szerkesztés



17. ábra: Hiba léptékének növekedése 15 betű esetén **Forrás:** Saját szerkesztés

4.2.4 Algoritmusok pontossága

Fontos kiemelni, hogy bizonyos algoritmusok jobb teljesítménnyel rendelkeznek ebből kifolyólag jobb eredményeket is érnek el. A lent látható részlet egy tanulmányból vett teljesítmény próbán alapszik, amelyet az angol *abc* betűin hajtottak végre,

Az eredményeket elemezve megállapíthatjuk, hogy a *Hebb*-módszer több mint 40%-os mintázat felismerésnél nem sikerült, ami nem túl kielégítő eredmény. Az eredmények főleg azért okoznak csalódást, mert a felhasznált minták egyszerű, mindössze 56 bitből álló minták, amelyekkel manapság ritkán találkozunk valós környezetben. Érezhetően jobb megbízhatóságot mutat az *Oja* tréningmódszer⁵³: ezek a tesztek 14 esetben nem sikerültek, ami a következő eredményekhez vezet 28%-os a téves osztályozás. Ez még mindig nem tökéletes, de jelentősen jobb. Az utolsó módszer: a *pseudoinverz* tökéletesnek tűnik. Ennek ellenére a 4%-os téves felismerés ezt a módszert minősíti a legjobbnak a három módszer közül. Azt is érdemes megemlíteni, hogy az összes egyszerű osztályozók 85% feletti

⁵³ *Oja* tanulási szabálya, vagy egyszerűen *Oja* szabálya, amelyet először a finn *Erkki Oja* 1982-ben, a *Simplified neuron model as a principal component analyzer*, művében jegyezte le. Ez egy modell arra, hogy az agyban vagy a mesterséges idegsejtekben lévő neuronok hogyan működnek. A neurális hálózatokban idővel megváltoztatják a kapcsolatok erősségét, vagyis tanulnak. Ez egy a standard *Hebb*-szabály módosítása, amely a multiplikációs normalizálással megoldja az összes stabilitási problémát, és egy algoritmust generál a főkomponens-elemzéshez. Ez egy olyan hatás számítási formája, amely a biológiai neuronokban is előfordul (Szandala, 2015, 71).

pontossággal általában kielégítőek, a tesztelt *pszeudoinverz*⁵⁴ pedig az elérte a 96%-os pontosságot a mintafelismerés területén (Szandala,2015,68-75).

4.4 Képszegmentálás

A következő program az előbbieken a minták helyett képeken szemlélteti a *Hopfield* hálózatokat. Pontosabban a programkód négy képet tölt be (*petofi.jpg*, *pufi.jpg*, *feri.jpg* és *bill.jpg*), és az *im2bw* függvény segítségével bináris képekké alakítja őket. Minden képet levágunk és átméretezünk, hogy létrehozzuk a képzési mintákat (*ob2*, *tr2*, *br2* és *wr2*). Majd a levágott és átméretezett bináris képeket képzési mintázatként jeleníti meg egy 2x2-es *subplotban*. A képzési mintákat ellaposítjuk és egy *P_matrix* nevezetű mátrixba koncipiáljuk. A *P_mátrix* képzési adatot a *W_mátrix* súlymátrix kiszámításához használjuk. A *zajtcsinál* függvény a gyakorló minták zajos változatait generálja, amelyek tesztadatként szolgálnak. A zajos teszt adatokat a *zajos_mintak* változóban tároljuk.

A *Hopfield* hálózat képzése a neuronok állapotainak iteratív frissítésével történik megszokott módon a képzési minták tárolása érdekében. Ez a folyamat addig folytatódik, amíg a hálózat el nem ér egy stabil állapotot, ahol a frissítés nem változtatja meg a neuronok állapotát. A zajos teszt adatokat a betanított *Hopfield* hálózat elé tárjuk majd a zajos bemenet és a hálózat kimenete közötti különbséget ábrázoljuk. A zajos teszt adatokat és a rekonstruált mintákat egymás mellett jelenítjük meg egy 2x4-es *subplot*-ban. A hálózat minden neuronja egy bináris állapotot reprezentál. Ebben a kódban minden neuron a bináris képek egy pixelének felel meg, és állapota +1 (*ON*) vagy -1 (*OFF*) lehet. A *W_matrix* súlymátrixot a képzési adatokból számítjuk ki. Az *i* és *j* neuron közötti súlyt a következő adja meg:

$$w_{ij} = \frac{1}{N} \sum_{k=1}^N P_{ki} P_{kj}$$

Ahol *N* a neuronok (a képek képpontjainak) teljes száma, és *P* a képzési adatmátrix. A neuronok állapota iteratív módon frissül egy előjelfüggvény segítségével, amelyet a kódban a *jeleles2* függvény ír le. A frissítési szabály a következő:

$$x_i^{t+1} = \text{sign}(\sum_j w_{ij} x_j^t)$$

⁵⁴ Ennek a furcsán hangzó megoldásnak meglehetősen egyszerű alapjai vannak. Feltételezzük, hogy minden minta a bemeneten pontosan önmagát adja a kimeneten. Ha van egy súlymátrixunk (*W*) és a kapott mintát megszorozzuk vele (*X*), akkor megkapjuk magát a mintát (Szandala,2015,71).

Ahol x_i^{t+1} az i neuron frissített állapota $t + 1$ időpontban, w_{ij} az i neuron és a j neuron közötti súly, és x_j^t a j neuron állapota t időpontban.

4.4.1 Előnyök és hátrányok, alkalmazásunkban

Előnyök: *Hopfield* hálózatok képesek felismerni és kiegészíteni a mintákat, még akkor is, ha hiányos vagy zajos adatokkal találkozunk, robusztusan képesek több mintát tárolni és előhívni, garantáltan stabil állapotba konvergálnak, ami egyes alkalmazásokban vonzóvá teszi őket.

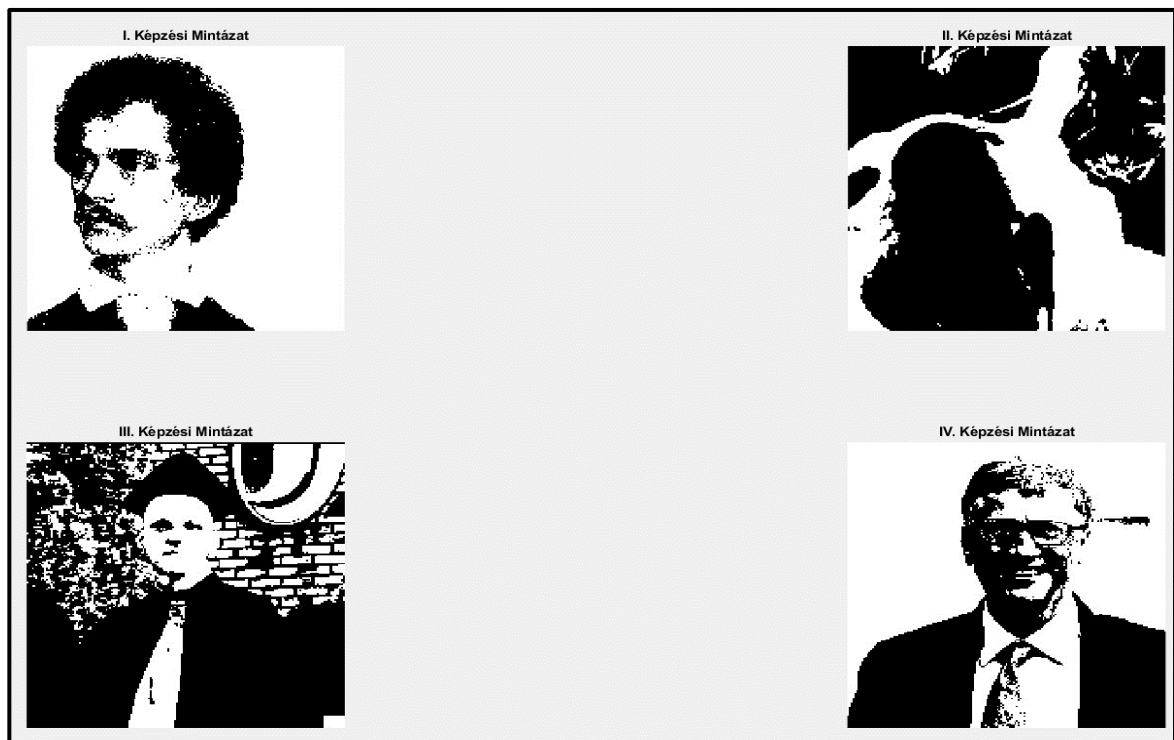
Hátrányok: *Hopfield* hálózat kapacitása a minták tárolására korlátozott, és előfordulhatnak hamis állapotok vagy a hasonló minták közötti zavarok. A stabil állapot eléréséhez szükséges konvergenciaidő lassú lehet, különösen nagy hálózatok vagy összetett minták esetén. Valamint a súlymátrixának szimmetrikusnak kell lennie, ami nem mindig megvalósítható vagy alkalmas bizonyos alkalmazásokhoz. Illetve a hálózat lokális minimumokhoz konvergálhat, ami helytelen vagy hiányos mintafelidézéshöz vezethet (Cheng,1996).

4.4.2 Összegzés

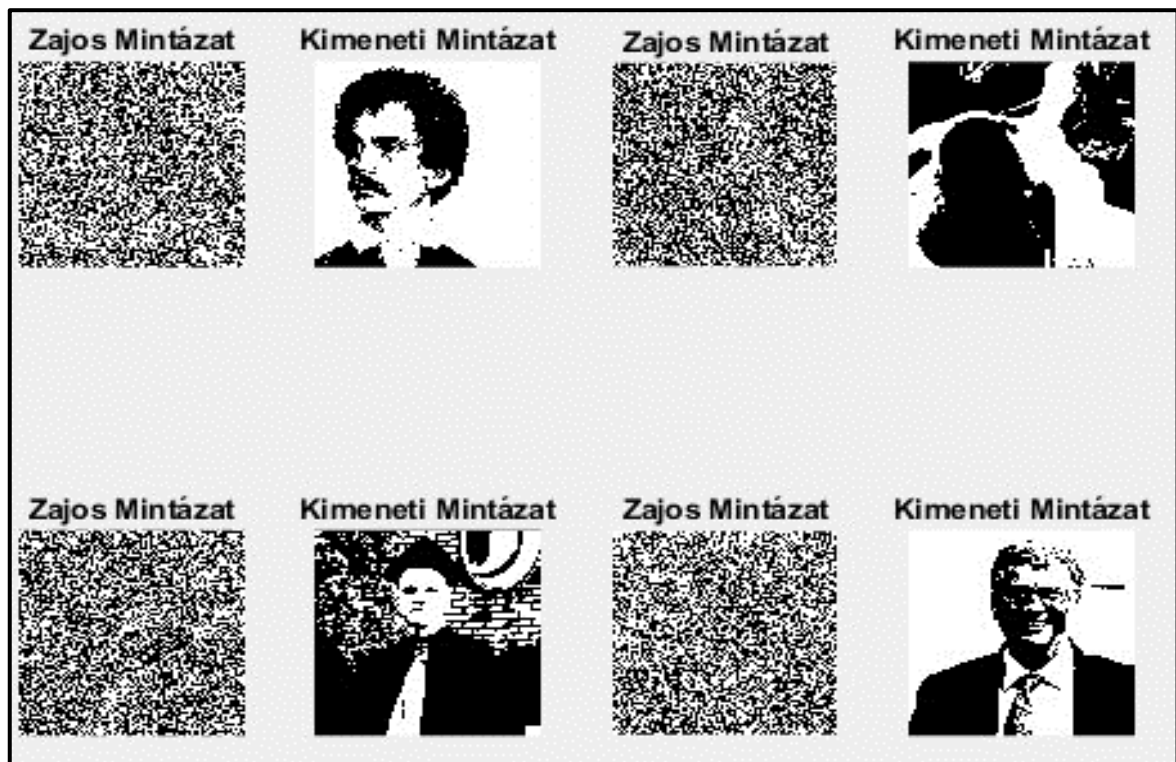
Összefoglalva, a program bemutatja, hogy egy *Hopfield* neurális hálózat hogyan használható mintafelismerésre és kiegészítésre, konkrétan képek rekonstruálására zajos vagy sérült bemenetekből. A hálózat súlymátrixa kulcsfontosságú a minták tárolásában és visszakeresésében, az iteratív frissítési szabály pedig segít a hálózat stabil állapotba való konvergálásában.



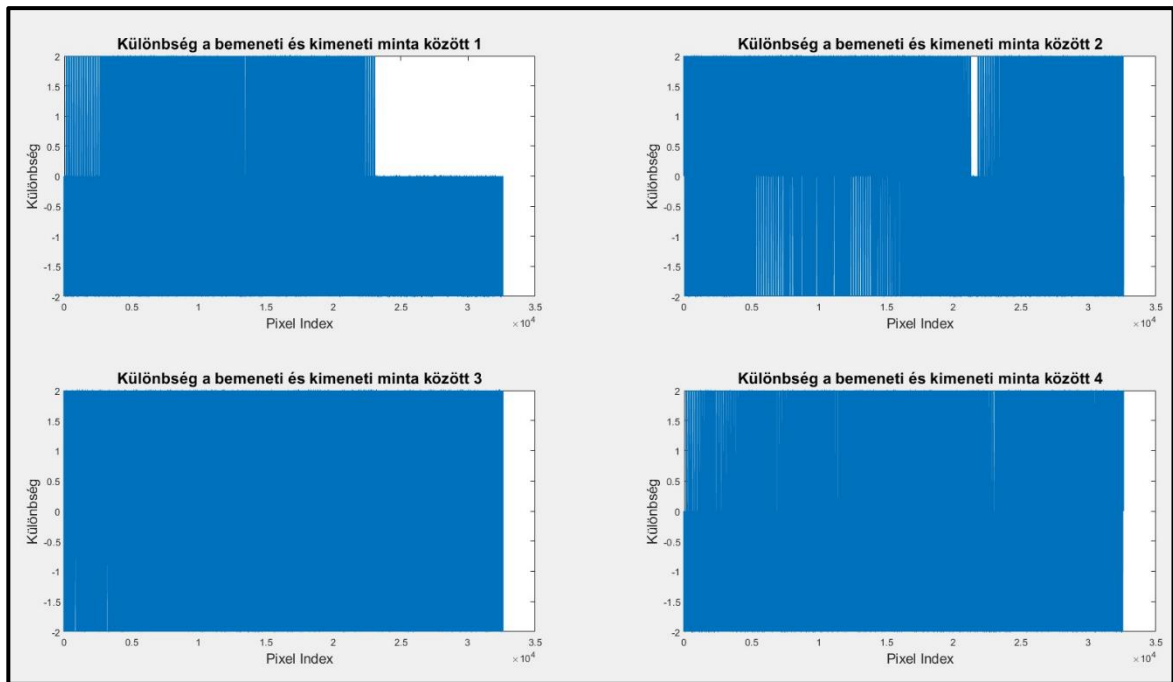
18. ábra: Eredeti képek **Forrás:** saját szerkesztés



20. ábra: Zajjal ellátott, illetve visszanyert képek **Forrás:** saját szerkesztés



19. ábra: Levágott és átméretezett bináris képek **Forrás:** saját szerkesztés



21. ábra: Hisztogram az eredeti majd a visszakapott képekre, a fehér szín hivatott jelölni a szemmel nem látható pixel különbségeket **Forrás:** saját szerkesztés

4.5 Kriptográfiai megközelítés

Jelen fejezet a kevésbé megszokott alkalmazási módját hivatott elénk tárni a *Hopfield* hálózatoknak a kriptográfia területén. Megfigyelésünk azt mutatja, hogy a *Hopfield* hálózatok *pseudo* véletlenszerűséggel rendelkeznek ezért alkalmazhatónak véltem „bevetni” a kriptográfia terén is.

Három képtitkosítási módszer vizsgálatát mutatja be a program, különös tekintettel a *Hopfield* hálózatok alkalmazására. Ebben a konkrét megvalósításban egy *pseudo* véletlen szekvencia létrehozására használjuk, amely a titkosítási folyamat részeként szolgál.

A kód először is inicializál egy *Hopfield* hálózatot egy véletlenszerű súlyok és állapotok készletével. Ezután a hálózat dinamikáját iteratív módon alkalmazzuk, ami egy véletlenszerű szekvenciát alkotó frissítések sorozatát eredményezi. Ezt a szekvenciát normalizáljuk, és egy olyan kép létrehozására használjuk, amely a *XOR*-titkosítás kódjaként szolgál.

Az eredeti képet beolvassuk, és kiszámítjuk az entrópiáját. Ezt követően a kód három különböző titkosítási módszert alkalmaz: *XOR*-titkosítás (a *Hopfield* által generált szekvencia felhasználásával), helyettesítési titkosítás és *XOR*-ral történő biteltolódásos

titkosítás. Mindegyik módszer entrópiája is kiszámításra kerül, ami tükrözi a titkosított képek összetettségét vagy véletlenszerűségét.

A *XOR*-titkosítási módszer esetében, ahol a *Hopfield*-dinamika döntő szerepet játszik, az eredeti kép 7,01-es entrópiája a titkosítás után 7,50-re nő. Az entrópia növekedése azt jelzi, hogy a titkosítási folyamat sikeres volt, és a képet véletlenszerűvé tette. A visszafejtés visszaállítja az eredeti entrópiát, ami tovább erősíti a módszer hatékonyságát.

A helyettesítő kódolási módszer⁵⁵ entrópiája az eredetivel megegyező, 7,01 marad, mivel a módszer csupán átrendezi a pixelértékeket anélkül, hogy azok általános eloszlását megváltoztatná.

A biteltolódásos titkosítási módszer⁵⁶ jelentős változást mutat, a titkosított kép entrópiája 8,00. A visszafejtett kép entrópiája azonban 5,90-re csökken, ami arra utalhat, hogy a titkosítási vagy visszafejtési folyamat során bizonyos információk elveszthetnek.

Az eredeti, a titkosított és a visszafejtett kép vizuális ábrázolása látható, a megfelelő hisztogramok pedig betekintést nyújtanak a pixelértékek mögöttes eloszlásába.

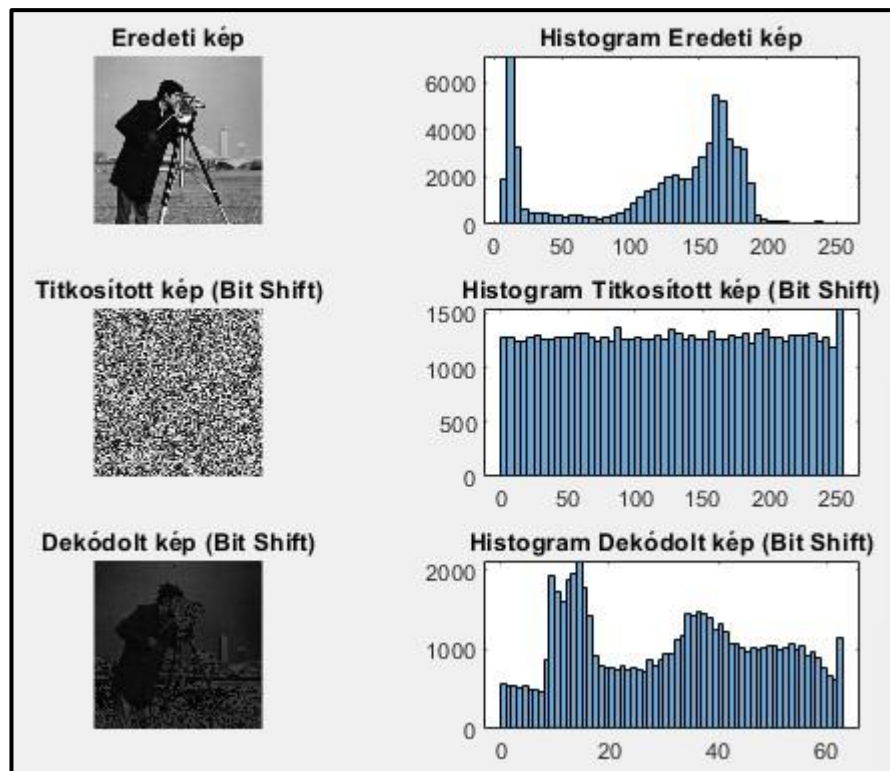
Összefoglalva, a programkód a neurális hálózatok és a képfeldolgozás metszéspontját vizsgálja, *Hopfield* hálózatokat alkalmazva a titkosítási technikák javítására. A *Hopfield*-dinamikát kihasználó *XOR*-módszer kifejezetten jól teljesít, míg a biteltolódásos módszer további finomítást igényelhet az információ integritásának biztosítása érdekében. Ez a feltárás érdekes lehetőségeket nyit meg a neurális hálózatok dinamikájának és a kriptográfiának az integrálására, és alapjául szolgálhat a fejlettebb kutatásoknak és alkalmazásoknak (Kesler,2015) & (Yang,2014).

```
Eredeti kép entrópia: 7.01
Titkosított kép entrópia(XOR): 7.50
Titkosított kép entrópia (Substitution): 7.01
Titkosított kép entrópia (Bit Shift): 8.00
Dekódolt kép entrópia (XOR): 7.01
Dekódolt kép entrópia (Substitution): 7.01
Dekódolt kép entrópia (Bit Shift): 5.90
```

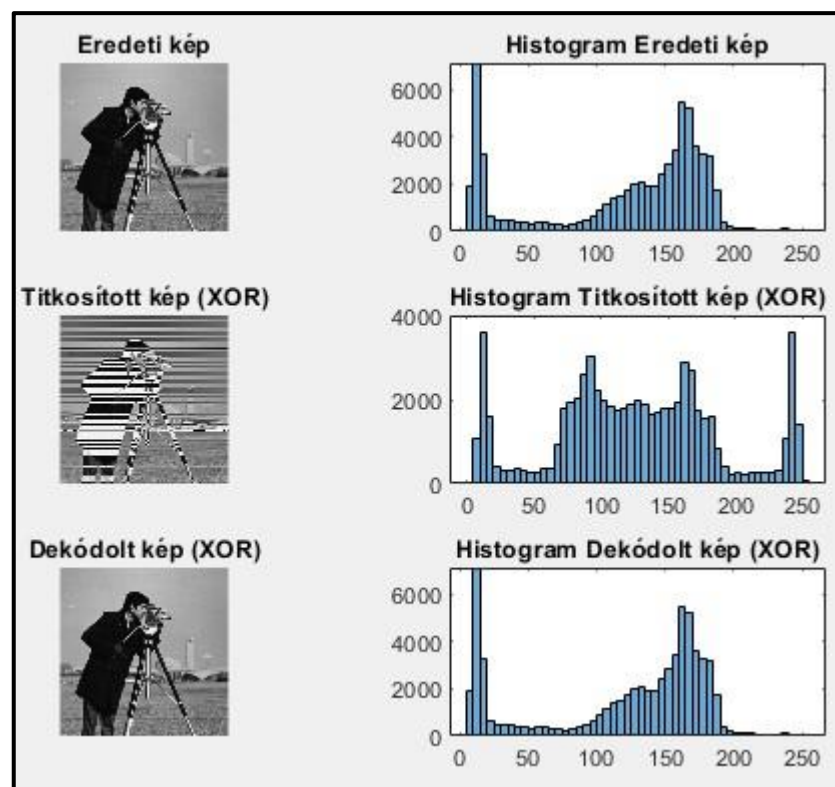
22. ábra: *Command window*on belüli entrópia értékek **Forrás:** Saját szerkesztés

⁵⁵ A helyettesítő kódok úgy titkosítják a nyílt szöveget, hogy a kulcs által meghatározott módon minden egyes betűt vagy szimbólumot más szimbólummal cserélnék fel a nyílt szövegben (Shimeall,2014,160).

⁵⁶ Bit Shift transzformációs módszer. A titkosított rejtjelezett szövegekét tovább lehet blokkokra osztani, majd biteltolással nemlineárisan átalakítani, ezáltal javítva a titkosítás biztonságát (Google,2023).



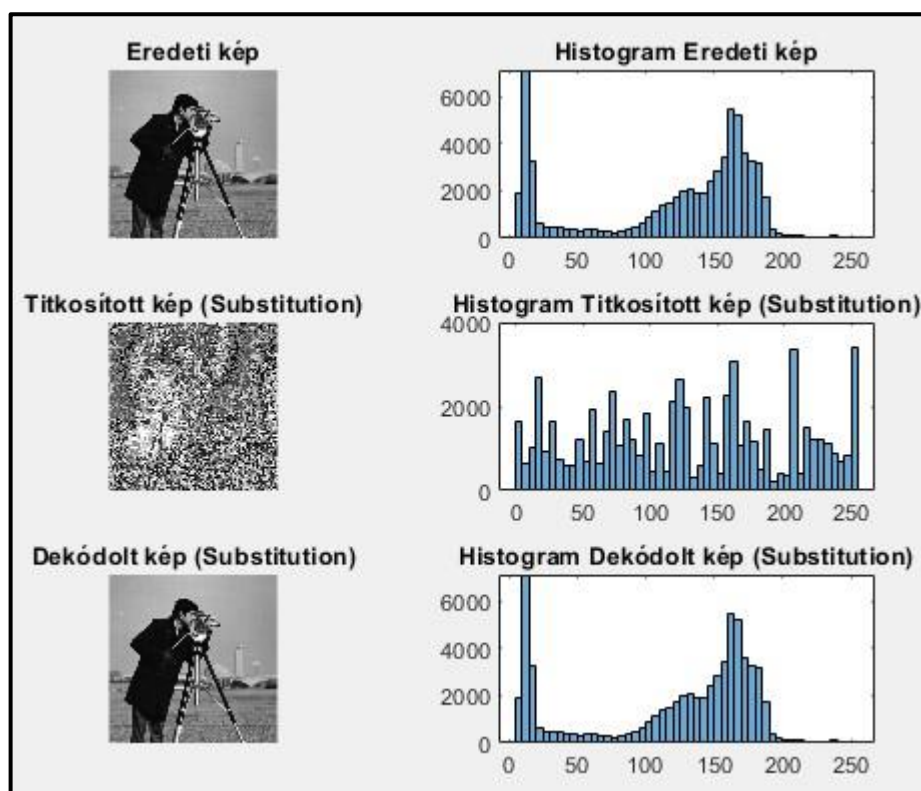
24. ábra: *Bit shift* titkosítás **Forrás:** Saját szerkesztés



23. ábra: *Hopfield, Xor* titkosítás **Forrás:** Saját szerkesztés

A titkosított kép *histogramja* általában kevésbé strukturált, és az entrópia is magasabb. A magas entrópia azt jelenti, hogy a kép nehezebben rekonstruálható, ami jó a titkosítás

szempontjából. A magas entrópia azt is jelenti, hogy a kép információtartalma jobban el van osztva, ami megnehezítheti a támadó számára a kulcs vagy az algoritmus megtalálását.



25. ábra: *Substitution* titkosítás Forrás: Saját szerkesztés

4.6 Optimalizálási probléma Hopfield logikával

Az előző fejezet is kifejezetten szokatlan alkalmazási területet hivatott szemléltetni a *Hopfield* hálózatokra. De meggyőződésem és tapasztalatom által, amit a téma kutatása alatt szereztem, hogy a majoritás alkalmazásai helyett sajátos megoldásokat találjak ki a *Hopfield* hálózatok alkalmazására. Ezen diplomamunka is a legelterjedtebb, alkalmazásaival indított, de amint már említettem szeretnék „idegen vizekre evezni”, megnézni ezen alkalmazásokban való helytállását a *Hopfield* hálózatoknak.

A sorban következő alkalmazás előzménye a teljes *Hopfield* logika implementálása egy optimalizálási problémában nem bizonyult megfelelőnek, mondhatnám nem kifejezetten alkalmazható e célra. Pont ezért a *Hopfield* logikát átfontolva alkalmaztam az optimalizálásban azokkal a részekkel, amelyek jól működnek és vegyítettem más

algoritmusokkal mind ahhoz, hogy egy jobban működő programkódot nyerjek ezáltal. A kevésbé jól működő részeket pedig elvetettem.

A programkód egy kombinatorikus optimalizációs algoritmust valósít meg, amelyet a már említett *Hopfield* neurális hálózatok ihlettek, és a kvadratikussal kapcsolatos problémák (QAP) megoldására alkalmaztam. Hozzá teszem az utazóügynök probléma (*TPS Traveling sales man problem*) helyett azért választottam ezt a problémakört mivel kifejezetten kimerítve éreztem a *TPS* problémát és jól bejártam.

A *QAP* egy klasszikus optimalizálási probléma, amely rendelkezik n létesítménnyel és n helyszínnel, és a cél a létesítményeknek és a helyszíneknek egy olyan leképezésének meghatározása, amely minimalizálja a költséget. A költséget jellemzően két mátrix határozza meg: a távolság és az áramlás.

A távolsági mátrix $D = [d_{ij}]$ ⁵⁷ a helyszínek közötti távolságot jelöli, illetve az áramlás mátrix $F = [f_{kl}]$ ⁵⁸ a létesítmények közötti áramlást (kölcsonhatást) jelenti k és l . A cél a teljes költség minimalizálása:

$$C = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot f_{state(i),state(j)}$$

ahol, $state(i)$ a helyhez rendelt létesítményt jelenti i (Burkard, 1984, 283-289).

4.6.1 Hopfield hasonmás?

Az említett algoritmus a *Hopfield* neurális hálózatok által inspirált iterációk sorozatát használja a *QAP* megoldásának megtalálására. Íme egy összehasonlítás a programkód és egy hagyományos *Hopfield* hálózat között:

Az aktuális megoldást egy permutációs vektor $state$ (állapot) reprezentálja, amely leírja a létesítmények helyhez rendelését. A *Hopfield* hálózatokban a neuronok bináris állapotokat reprezentálnak, és az energiatájkép irányítja ezen állapotok alakulását. Valamint a kódban szereplő (*calculate_energy* által kiszámított) költségfüggvény szolgál energiafüggvényként, amelyet az algoritmus minimalizálni próbál. Az energiafüggvény egy *Hopfield* hálózatban a hálózat állapotának mértékegysége és a hálózat dinamikája egy alacsony energiájú állapot megtalálására törekszik. Az *update_hopfield* függvény véletlenszerűen kiválaszt két elemet

⁵⁷ Angolul *distance* ezért jelölöm d-vel.

⁵⁸ Angolul *flow* ezért jelölöm f-vel.

a permutációban, és felcseréli őket, ha ez csökkenti az energiát (költséget). Az energiacsökkentésen alapuló választás, hogy elfogadjuk vagy elutasítjuk a cserét, hasonló a *Hopfield* hálózatok determinisztikus frissítési szabályához, ahol a neuronok az energiafüggvény minimalizálása érdekében frissítik állapotukat. Az algoritmus figyeli, hogy az egymást követő iterációk során nem változik-e az energia, és megáll, ha 150 iteráción keresztül nem észlel változást. Ez hasonlít a *Hopfield* hálózat stabil állapothoz való konvergenciájához. A *visualize_state* függvény segít megjeleníteni a rendszer aktuális állapotát. Bár nem része a hagyományos *Hopfield* hálózatoknak, a vizualizáció hasznos lehet a komplex rendszerek viselkedésének megértésében.

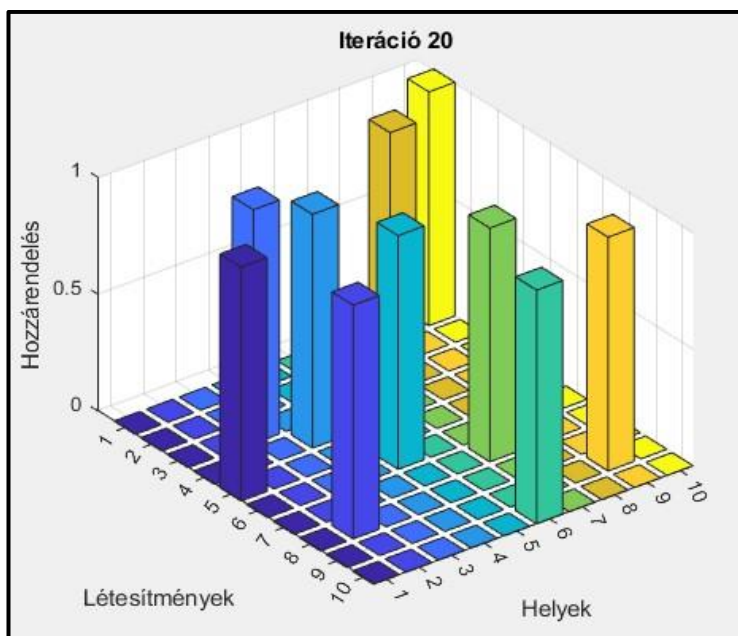
A *Hopfield* neurális hálózatok által inspirált technikát alkalmaztuk a *QAP* megoldására a létesítmény hozzárendelések iteratív cseréjével az energia (költség) minimalizálása érdekében. Az energiafüggvény használata, az energiát minimalizáló frissítési szabály és a stabil állapotokhoz való konvergencia a legfontosabb hasonlóságok az algoritmus és a *Hopfield* hálózatok között. A *Hopfield* hálózat asszociatív memóriefunkciója itt nem alkalmazható közvetlenül, de az energiaminimalizálás módszere és a rendszer állapotának dinamikus frissítése egyértelműen a *Hopfield* féle neurális hálózatok modelljéből merít ihletet. Kifejezetten a használd, ami működő, dobd el, ami hatástalan elvet használtam az algoritmusnál mind végig a témánknál maradva.

4.6.2 Részletes leírás

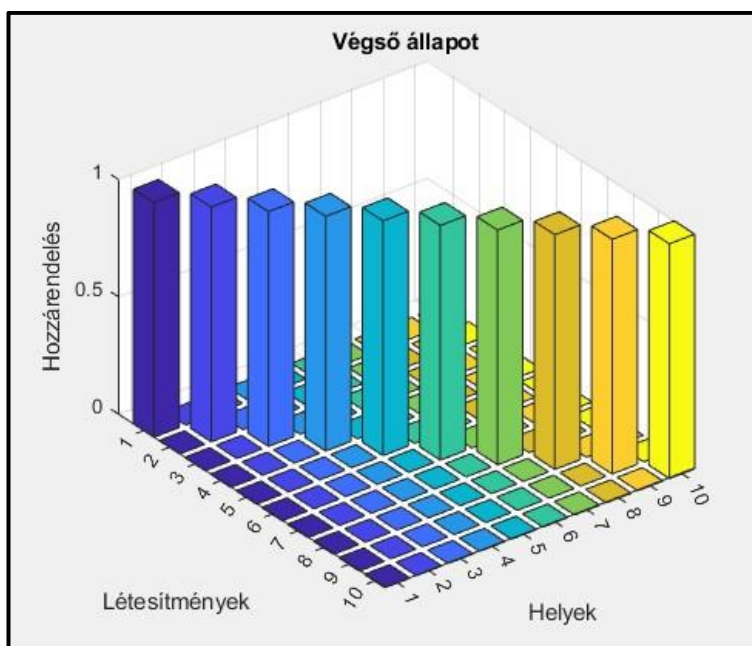
Az eddigiek folyamán még nem került említésre, de a *clc*, *close all* parancsokkal kezdünk, amelyek törlik a parancsablakot és bezárják az összes ábrát. Utána betöltjük a távolságmátrixot szöveges fájlból, illetve betöltjük szintűgy az áramlási mátrixot. A következőkben meghatározzuk a helyek és létesítmények számát a távolságmátrix méretéből. Biztosítjuk, hogy a távolság- és az áramlási mátrixok négyzet alakúak és azonos méretűek legyenek. Ha nem így van, hibaüzenetet kapunk. Valamint véletlenszerű permutációt generálunk az állapot inicializálásához. Beállítjuk az iterációk teljes számát és meghatározzuk a *printf*-ek gyakoriságát. Inicializálunk egy számlálót a változatlan energiájú iterációkhoz. Utána kiszámítjuk a rendszer kezdeti energiáját. Majd meghívjuk a frissítési függvényt két elem felcseréléséhez. Később újra számítjuk az energiát az új állapothoz, növeljük a számlálót, ha az energia változatlan marad akkor kilépünk a ciklusból, ha eléri a

150-et (amit preferálunk). 100 iterációnként kiírjuk az iterációs számot és az aktuális energiát, az optimalizálás sikerességének követése szempontjából, egyszerűen megjelenítjük az aktuális állapotot. Végezetül vizualizáljuk a végső állapotot és megjelenítjük a végső energiát.

4.6.3 Kimenetek, tesztek



26. ábra: Futás 10*10 adathalmazra **Forrás:** Saját szerkesztés



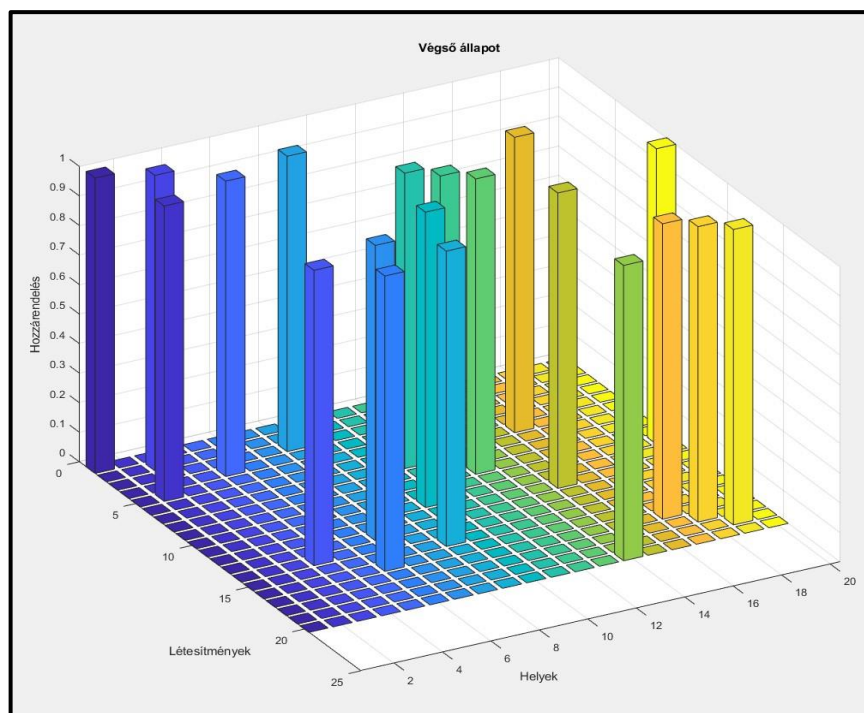
27. ábra: Végső állapot 10*10 halmazra **Forrás:** Saját szerkesztés

```
Kezdeti energia: 8580
Iteráció 10: Energia = 8160
Iteráció 20: Energia = 8160
Iteráció 30: Energia = 8090
Iteráció 40: Energia = 8020
Iteráció 50: Energia = 7720
Iteráció 60: Energia = 7720
Iteráció 70: Energia = 7720
Iteráció 80: Energia = 7510
Iteráció 90: Energia = 7510
Iteráció 100: Energia = 7510
Iteráció 110: Energia = 7510
Iteráció 120: Energia = 7510
Iteráció 130: Energia = 7510
Iteráció 140: Energia = 7510
Iteráció 150: Energia = 7510
Iteráció 160: Energia = 7510
Iteráció 170: Energia = 7510
Iteráció 180: Energia = 7510
Iteráció 190: Energia = 7000
Iteráció 200: Energia = 7000
Iteráció 210: Energia = 7000
Iteráció 220: Energia = 7000
Iteráció 230: Energia = 7000
Iteráció 240: Energia = 7000
Iteráció 250: Energia = 7000
Iteráció 260: Energia = 7000
Iteráció 270: Energia = 7000
Iteráció 280: Energia = 7000
Iteráció 290: Energia = 7000
Iteráció 300: Energia = 7000
Iteráció 310: Energia = 7000
Iteráció 320: Energia = 7000
Iteráció 330: Energia = 7000
Az állapot 100 iteráció óta nem változik. Leállítás.
Végző energia: 7000
```

29. ábra: Iterációnkénti *command window* értesítések **Forrás:** Saját szerkesztés

```
Iteráció 30750: Energia = 29240
Iteráció 30900: Energia = 29240
Iteráció 31050: Energia = 29240
Iteráció 31200: Energia = 29240
Az állapot 100 iteráció óta nem változik. Leállítás.
Végző energia: 29240
```

28. ábra: 31200 iteráció utáni eredmény 20*20 adathalmazra **Forrás:** saját szerkesztés



30. ábra: Végző állapot 20*20 adathalmazra **Forrás:** saját szerkesztés

A heurisztikus algoritmus az adathalmaz nagyságára tekintettel, vagyis minél nagyobb a keresési tér exponenciálisan növekedik a bonyolultsága úgymond megnövekedik az esély a lokális minimumok megtalálására a globális minimum helyett. Ezért a 10*10 adathalmazunk tökéletes megoldást mutat akár 150 iteráció eltelte után is (szemmel láthatóan) viszont, ha növeljük a tétet és megduplázzuk az adathalmazt, keresési teret szemmel sem tudnánk megmondani jól döntött-e az algoritmus, nem is beszélve az említett lokális minimumokról, illetve az időről, amit igénybe vesz a lefutása. Ezáltal a hosszas várakozás sem garantálhat számunkra megfelelő, optimális eredményt. Ennek ellenére az optimális eredmény nincs kizárva (Kokash,2005,1-8) & (Desale,2015, 296-304).

Fontos kihangsúlyozni, hogy ez a jelenség nem egyedi a megadott kódhoz, hanem általános jellemzője a heurisztikus keresési algoritmusoknak, amelyek nagy adathalmazokon dolgozva, kompromisszumot kell kötniük az optimális megoldás megtalálása és a futási idő között. Az alfejezetet az alábbi idézettel zárom:

„Innovation is taking two things that exist and putting them together in a new way.”

Tom Freston⁵⁹

⁵⁹ Az idézet a quora.com weboldalról származik.

5 Összegző weboldal

Kiegészítésként, valamint bátorkodva, hogy az interneten is elérhető legyen diplomamunkám és könnyen hozzáférhető, megalkottam egy *weboldalt* amely a *Microsoft word*-ben megírt munkám tükörképe.

5.1 Fejlesztői környezet

A weblapot a *Microsoft* által kifejlesztett *Visual Studio Code* forráskód-szerkesztővel hoztuk létre, amely támogatja a hibakeresést, a beágyazott *Git*-vezérlést, a *GitHub*ot a zökkenőmentes verziókezeléshez, a szintaxis-kiemelést, az intelligens kódkiegészítést és még sok mást. Rugalmas és sokoldalú, a programozási nyelvek és keretrendszerek széles skáláját támogatja.

A *Visual Studio Code*-t a weboldal *HTML* ⁶⁰ -, *CSS*- és *JavaScript*-programkódjának megírására és szerkesztésére használtuk. Mindegyik nyelv külön célt szolgál. A *HTML* (*HyperText Markup Language*) a weboldal tartalmának strukturálására szolgál, a *CSS* (*Cascading Style Sheets*) a weboldal elrendezéséhez és megjelenéséhez, a *JavaScript* pedig az interaktív elemeket teszi lehetővé.

A weboldal egy információs oldal a neurális hálózatokról és a *Hopfield*-hálózatokról. A webhelynek van egy fő céloldala, amely tartalmaz egy fejléct egy navigációs sávval, egy különböző fejezeteknek szentelt részt, valamint egy lábléct, amely szerzői jogi információkat és egy kapcsolatfelvételi oldalra mutató linket tartalmaz. A webhely témája esztétikai kialakításában nagymértékben támaszkodik a *CSS*-re, különösen a *Google Montserrat* betűtípusára és a különböző színsémákra és átalakításokra.

JavaScript része interaktivitást visz a weboldalba. Ez egy bizonyos görgetési távolság után megjelenő „*vissza a tetejére*” gomb és egy öt másodpercenként változó, forgó képek

⁶⁰ *Web* a leggyorsabban növekvő erőforrás, amelyet gyorsan és folyamatosan használnak szinte minden platformon. Sok szoftver gyártó saját szabadalmaztatott technikát hozott létre. Pl. *Adobe System Flesh*, *Microsoft silver light*, *Oracle JAVA FX*, *Google Gears*, *Apple Quick Time*, webes alkalmazások futtatására. Mivel a *HTML5* a *HTML* újabb verziója, segít nekünk abban, hogy interaktív és gazdag weboldalakat hozzunk létre. A *HTML* drasztikusan nőtt és fejlődött az egyszerűen csak a weboldalak előállításától egészen a hang, videó és animációk előállítására, valamint, hogy offlinefunkcionalitást, helyi tárolást és földrajzi helymeghatározást bármely kliensen oldali adatbázisban is létre lehet hozni általa (Ratha,2018,551).

funkcióit tartalmazza. Mindkét funkció növeli a felhasználói élményt és a weboldal általános használhatóságát.

A *HTML*- és *CSS*-fájlok a *link tag* segítségével kapcsolódnak egymáshoz, a *JavaScript*-kód pedig a *HTML*-dokumentum *script*-tagjain belül található. A problémák ilyen szétválasztása a kódot jobban kezelhetővé és szervezetté teszi.

Ami a verziókezelést és az „együtműködést” illeti, a projekt a *GitHub*ot használja. A *GitHub* egy webalapú platform a *git* segítségével történő verziókezeléshez, amely lehetővé teszi, hogy több ember dolgozzon egy projekten anélkül, hogy egymás módosításait felülírnák. A projektet egy *GitHub*-tárhelyen tároljuk. A *repository* az összes kód központi helyeként szolgál, amelyet a *git* parancsok segítségével klónozni (letölteni), elágazni (külön területet létrehozni új funkciók kipróbálására anélkül, hogy a fő kódot befolyásolná) és *commitolni* (a változtatásokat elmenteni) lehet.

A *Github* biztosítja, hogy a kódbázis naprakész és szinkronizált maradjon a projekten dolgozó összes fejlesztő számára. Emellett a változások előzményeit is biztosítja, így könnyebben nyomon követhető a fejlődés, és szükség esetén vissza lehet térni a korábbi verziókhoz. Valamint magát a linket is számunkra a *Github* szolgáltatja <https://ferencribni.github.io/>.



31. ábra: A weboldal fogadóképe **Forrás:** saját szerkesztés

Befejezés

Resumé

Hivatkozások

Dr. Fazekas István, *Neurális hálózatok*. [online]. Debrecen, 2013. [cit.2023.05.29.] Elérhető: <https://gyires.inf.unideb.hu/GyBITT/19/Neuralis_halozatok_v8.pdf>

Khalid Saeed, Władysław Homenda, *Computer Information Systems and Industrial Management*. [online]. Vilnius, Lithuania, 2016. szeptember 14-16. [cit.2023.05.30.] Elérhető: <<https://link.springer.com/book/10.1007/978-3-319-99954-8>>

Brian Russell, *The Least-Mean-Square (LMS) algorithm and its geophysical applications*. [online]. Hampson-Russell. [cit.2023.06.01.]

Elérhető: <<https://www.crewes.org/Documents/SlideShows/2017/CSS201730.pdf>>

Pumalin, *A Brief History of Neural Nets (Important dates in the History of Neural Nets)* [online]. 2023. január 29. [cit.2023.06.01.] Elérhető: <<https://pub.towardsai.net/a-brief-history-of-neural-nets-472107bc2c9c>>

Kmet' Tibor, Kmet'ová Mária, Annuš Norbert, *Neurális hálózatok programi megvalósítása MATLAB-ban*. [online]. 2021. [cit.2023.06.05.] Elérhető: <https://elearning.ujs.sk/pluginfile.php/40771/mod_resource/content/0/N_H_Kmet_07_12_21_final.pdf>

Hanna, G., & Larvor, B. *As Thurston says? On using quotations from famous mathematicians to make points about philosophy and education*. [online]. 2020 március 24. [cit.2023.06.05.] Elérhető: <<https://link.springer.com/article/10.1007/s11858-020-01154-w>>

Varga Péter, *ECC alapú elektronikus szavazási séma*. [online]. 2008. [cit.2023.06.05.] Elérhető: <<https://dea.lib.unideb.hu/server/api/core/bitstreams/e1f80fce-dd9a-472e-a040-7a02fa08bce9/content>>

Humayun Karim Sulehria, Ye Zhang, *Hopfield Neural Networks—A Survey*. [online]. School of Electronics and Information Engineering Harbin Institute of Technology, Harbin PR China 2007. február 16-19 [cit.2023.06.07.] Elérhető: <<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ae1103b32c83c9489f974d605d974d4d98c2562d>>

Snehashish Chakraverty, Deepti Moyi Sahoo, Nisha Rani Mahato, *Concepts of Soft Computing Fuzzy and ANN with Programming*, [online]. Springer Nature Singapore. ISBN 978-981-13-7430-2. 2019. [cit.2023.06.07.]

Elérhető: <https://link.springer.com/chapter/10.1007/978-981-13-7430-2_12>

Simon Haykin, *Neural Networks and Learning Machines Third Edition*, [online]. McMaster University Hamilton, Ontario, Canada. ISBN-13: 978-0-13-147139-9. 2008. [cit.2023.06.08.]

Elérhető: <<https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf>>

Van De Ven, Paul E. Johnson, *Knowledge for Theory and Practice. Academy of Management Review*, [online]. University of Minnesota, 2006. [cit.2023.06.13.]

Elérhető:< <https://journals.aom.org/doi/abs/10.5465/amr.2006.22527385>>

Harry Lees, *Discrete vs. Continuous Data*, [online]. TrustRadius. 2022 július 29. [cit.2023.06.14.]

Elérhető:< <https://www.trustradius.com/buyer-blog/discrete-vs-continuous>>

Artificial Neural Network - Hopfield Networks, [online]. tutorialspoint. 2023.06.14. [cit.2023.06.14.]

Elérhető:<https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_hopfield.htm>

Jun Han, Claudio Moraga, *The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning*, [online]. Research Group Computational Intelligence Dept. of Computer Science, University of Dortmund D-44221 Dortmund, Germany.1995. [cit.2023.06.14.]

Elérhető:< https://link.springer.com/chapter/10.1007/3-540-59497-3_175 >

Mr. Bhanari Kumar, *What is Activation Functions, Neural Functions?*, [online]. 360DigiTMG.2020 május 22. [cit.2023.06.14.]

Elérhető:< <https://360digitmg.com/blog/activation-functions-neural-networks#ramp> >

Igor R. Fermo, Thiago S. Cavali, Lucas Bonfim-Rocha, Caio L. Srutkoske, Franklin C. Flores, Cid M.G. Andrade, *Development of a low-cost digital image processing system for oranges selection using hopfield networks*, [online].2021 január [cit.2023.06.15.]

Elérhető:< <https://www.sciencedirect.com/science/article/abs/pii/S0960308520305642>>

Michael Husken, Peter Stagge, *Recurrent neural networks for time series classification*, [online]. Institut für Neuroinformatik, Ruhr-Universität at Bochum, 44780 Bochum, Germany.2001 január 25. [cit.2023.06.15.]

Elérhető:< <https://www.sciencedirect.com/science/article/abs/pii/S0925231201007068>>

Leda Coltro,Anna Lúcia Mourad, Rojane M. Kletecke, Taíssa A. Mendonça,Sílvia P. M. Germer, *Assessing the environmental profile of orange production in Brazil*, [online]. Springer-Verlag, 2009 június 4. [cit.2023.06.15.]

Elérhető:< <https://link.springer.com/article/10.1007/s11367-009-0097-1>>

Dakun Zhang, *Extended Closing Operation in Morphology and Its Application in Image Processing*. [online]. School of Computer Engineering and Automation, Tianjin Polytechnic University.2009. [cit.2023.06.15.]

Elérhető:< <https://ieeexplore.ieee.org/abstract/document/5190022>>

Euclidean distance, Wikipedia. [online].2023 május 8. [cit.2023.06.16.] Elérhető:< https://en.wikipedia.org/wiki/Euclidean_distance >

Distance Transform of a Binary Image. MathWorks,Help Center. [online].2023. [cit.2023.06.16.] Elérhető:<<https://www.mathworks.com/help/images/distance-transform-of-a-binary-image.html>>

Xin-She Yang, *Nature-Inspired Optimization Algorithms, Second Edition*. [online] Elsevier Inc. ISBN: 978-0-12-821986-7.2021. [cit.2023.06.16.]

Elérhető:<<https://www.sciencedirect.com/book/9780128219867/nature-inspired-optimization-algorithms>>

PSNR. MathWorks,Help Center. [online].2023. [cit.2023.06.16.] Elérhető:<<https://www.mathworks.com/help/vision/ref/psnr.html>>

Yadolah Dodge, *The Concise Encyclopedia of Statistics*. University of Neuchâtel Switzerland, Springer is part of Springer Science+Business Media, ISBN: 978-0-387-32833-1. [online].2008. [cit.2023.06.16.]

Elérhető:< https://link.springer.com/referenceworkentry/10.1007/978-0-387-32833-1_251>

Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining: Concepts and Techniques*, Elsevier Inc. ISBN: 978-0-12-381479-1. [online].2012. [cit.2023.06.16.]

Elérhető:<<https://www.sciencedirect.com/book/9780123814791/data-mining-concepts-and-techniques#book-description>>

ssim. MathWorks,Help Center. [online].2023. [cit.2023.06.16.] Elérhető:<<https://www.mathworks.com/help/images/ref/ssim.html#bt7y34q-1>>

Paul J. Werbos, *Backpropagation Through Time: What It Does and How to Do It*. Proceedings of the IEEE.1990 október [cit.2023.06.17.] Elérhető:<<https://ieeexplore.ieee.org/abstract/document/58337>>

Hyperspectral imaging, Wikipedia. [online].2023 június 13. [cit.2023.06.17.] Elérhető:<https://en.wikipedia.org/wiki/Hyperspectral_imaging>

Xing-Yuan Wanga, Zhi-Ming Li, *A color image encryption algorithm based on Hopfield chaotic neural network*. National Natural Science Foundation of China. [online].2019. [cit.2023.06.17.]

Elérhető: < <https://www.sciencedirect.com/science/article/abs/pii/S0143816618309497>>

Topuzoğlu, A., & Winterhof, A. *PSEUDORANDOM SEQUENCES. Topics in Geometry, Coding Theory and Cryptography*, vol 6. Springer, Dordrecht. ISBN 978-1-4020-5334-4. [online].2006. [cit.2023.06.17.]

Elérhető: < https://link.springer.com/chapter/10.1007/1-4020-5334-4_4 >

Mamta Rani, Rashi Agarwal, *Generation of fractals from complex logistic map. Chaos, Solitons & Fractals*. Published by Elsevier Ltd. [online].2009. [cit.2023.06.17.]

Elérhető: < <https://www.sciencedirect.com/science/article/abs/pii/S0960077909000137>>

Luca A., Vlad A., Badea B., Frunzete M., *A study on statistical independence in the tent map. 2009 International Symposium on Signals, Circuits and Systems*. IEEE. ISBN 9781-4244-3786-3 [online]. 2009. [cit.2023.06.17.]

Elérhető: < <https://ieeexplore.ieee.org/abstract/document/5206145/metrics#metrics>>

ANTONIOU, BI QIAO, Z. SUCHANECKI, *Generalized Spectral Decomposition and Intrinsic Irreversibility of the Arnold Cat Map*. Chaos, Solitons & Fractals Vol. 8, Elsevier Science Ltd Printed in Great Britain. [online].1997. [cit.2023.06.17.]

Elérhető: < <https://www.sciencedirect.com/science/article/abs/pii/S0960077996000562>>

Chuman T., Iida K., Kiya H., *Image manipulation on social media for encryption-then-compression systems*. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). [online]. 2017. [cit.2023.06.18.]

Elérhető: < <https://ieeexplore.ieee.org/abstract/document/8282153/metrics#metrics>>

López-López, E., Bajorath, J., & Medina-Franco, J. L. *Informatics for Chemistry, Biology, and Biomedical Sciences*. Journal of Chemical Information and Modeling, 61(1), 26–35. [online]. 2020. [cit.2023.07.07.]

Elérhető: < <https://pubs.acs.org/doi/abs/10.1021/acs.jcim.0c01301>>

Philipp Seidl, Philipp Renz, Natalia Dyubankova, Paulo Neves, Jonas Verhoeven, Jörg K. Wegner, Marwin Segler, Sepp Hochreiter, Günter Klambauer, *Supporting Information for Improving few- and zero-shot reaction template prediction using modern Hopfield networks*. [online]. 2023. [cit.2023.07.07.]

Elérhető: < <https://pubs.acs.org/doi/full/10.1021/acs.jcim.1c01065#>>

Moult, J., Fidelis, K., Kryzhtafovich, A., Schwede, T., & Tramontano, A. *Critical assessment of methods of protein structure prediction (CASP) - round x. Proteins: Structure, Function, and Bioinformatics*, [online]. 2013. [cit.2023.07.07.] Elérhető: < <https://onlinelibrary.wiley.com/doi/full/10.1002/prot.24452>>

Computational Chemistry Quotes. quotlr.com. [online].2023. [cit.2023.07.08.] Elérhető:< <https://quotlr.com/quotes-about-computational-chemistry> >

Gursel Serpen, *Hopfield Network as Static Optimizer: Learning the Weights and Eliminating the Guesswork*, Springer Science+Business Media, LLC. 2007. [online]. 2008. [cit.2023.07.09.] Elérhető: < <https://link.springer.com/article/10.1007/s11063-007-9055-8>>

Valeria V. Krzhizhanovskaya, Gábor Závodszy, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, João Teixeira . *Lecture Notes in Computer Science 12142*. Springer International Publishing; Springer, ISBN: 9783030504328,9783030504335. 2020.[online]. [cit.2023.07.11.] Elérhető: < <https://link.springer.com/book/10.1007/978-3-030-50433-5> >

Niels M. P. Neumann, Frank Phillipson, *Quantum Hopfield Neural Networks: A New Approach and Its Storage Capacity*, Part of the Lecture Notes in Computer Science book series (LNTCS, volume 12142). 2020.06.15. [online]. [cit.2023.07.11.] Elérhető: < https://link.springer.com/chapter/10.1007/978-3-030-50433-5_44 >

Quantum Computing Quotes, goodread.com, [online]. 2023. [cit.2023.07.15.] Elérhető:< <https://www.goodreads.com/quotes/tag/quantum-computing> >

Szandala, T. (2015). *Comparison of Different Learning Algorithms for Pattern Recognition with Hopfield's Neural Network*. Procedia Computer Science, 71, 68–75. doi: 10.1016/j.procs.2015.12.205. 2015. [online]. [cit.2023.07.30.] Elérhető: < <https://www.sciencedirect.com/science/article/pii/S1877050915036662>>

Mark Beale, *Neural Network Toolbox, For Use with Matlab , User's Guide Version 4*, The MathWorks, 1-846, 2004. október. The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098. [online]. [cit.2023.07.30.] Elérhető: < http://cda.psych.uiuc.edu/matlab_pdf/nnet.pdf >

Shimeall, T. J., & Spring, J. M. (2014). *Resistance Strategies. Introduction to Information Security*, 155–186. doi:10.1016/b978-1-59749-969-9.00008-0. 2014. [online]. [cit.2023.07.30.] Elérhető: < <https://www.sciencedirect.com/science/article/abs/pii/B9781597499699000080> >

Gary C. Kessler. *An Overview of Cryptography*. 1998-2015. Auerbach. 2015.04.16. [online]. [cit.2023.07.30.] Elérhető: < https://d1wqtxts1xzle7.cloudfront.net/38411944/An_Overview_of_Cryptography.pdf?1438962673=&response-content-disposition=inline%3B+filename%3DAn_Overview_of_Cryptography.pdf&Expires=1690741998&Signature=dHIFDRAj1SFuMK7X1FN8Wa1I1KMQuYb4UQJ49GxziK3RXm0WDj4DTYz-82v9BmVNUH08h4MTpbZlY3JX8RtPTmX10VLQCMw03bVOdrldZGqOVooGfRcvccjr0aeRx6BkjRRbWjXmdzq293rtZK3hadX3SwFRr3PKrnLRHpk87uKBA~rtnyWC8Ov2PKrblMD8Y1Fbn8xlDWVpEJUoOh5YBXkh9grz75bsseE0BGstMaXl3KtuHB0tXrEpRt0ZdT7E7Mg8aaWCAjaYe2LsFnwdAUZSWg7PHYGMKvzSEfl3codup6bUga4fXI6PBAFlrh5oBlAHKzkzxW0rwmGPlnEOg_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA >

Yang, C.-N., & Wang, D.-S. (2014). *Property Analysis of XOR-Based Visual Cryptography*. IEEE Transactions on Circuits and Systems for Video Technology, 24(2), 189–197. doi:10.1109/tcsvt.2013.2276708 2014. [online]. [cit.2023.07.30.] Elérhető: < <https://ieeexplore.ieee.org/abstract/document/6575153> >

Kuo-Sheng Cheng, Jzau-Sheng Lin, & Chi-Wu Mao. (1996). *The application of competitive Hopfield neural network to medical image segmentation*. IEEE Transactions on Medical Imaging, 15(4), 560–567. doi:10.1109/42.511759. 1996. [online]. [cit.2023.07.30.] Elérhető: < <https://ieeexplore.ieee.org/abstract/document/511759> >

Burkard, R. E. (1984). *Quadratic assignment problems*. *European Journal of Operational Research*, 15(3), 283–289. doi:10.1016/0377-2217(84)90093-6 1984. [online]. [cit.2023.08.01.] Elérhető: < <https://www.sciencedirect.com/science/article/abs/pii/0377221784900936> >

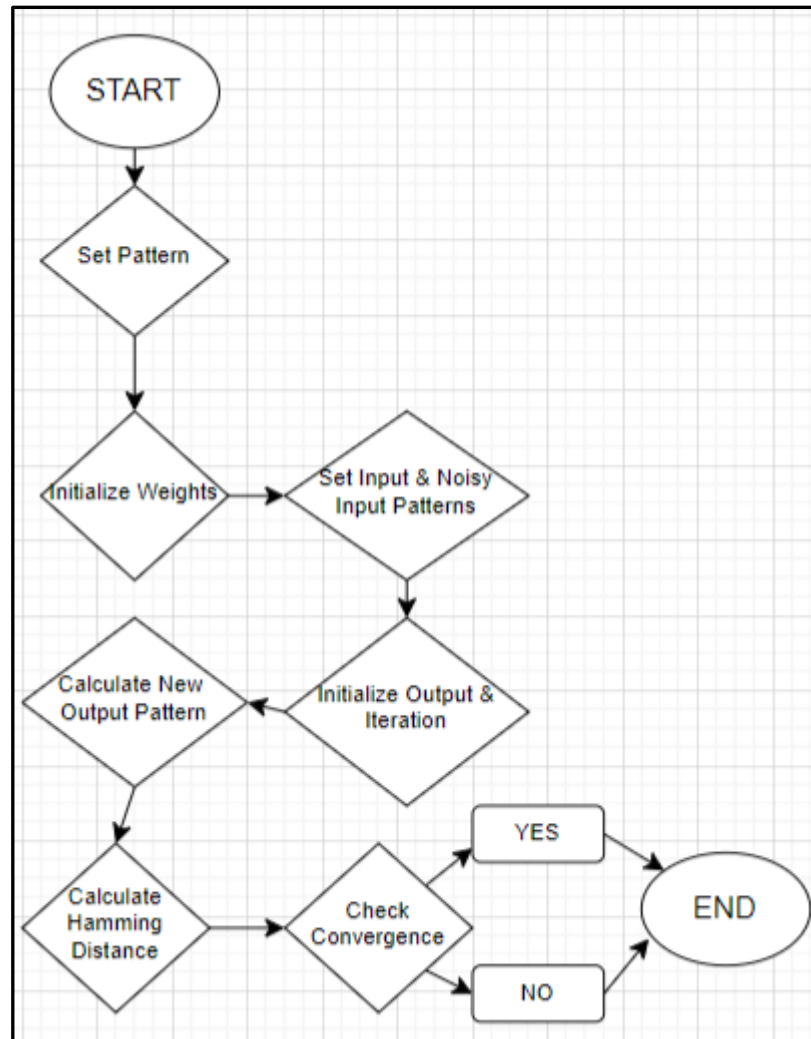
Natallia Kokash. *An introduction to heuristic algorithms*. Department of Informatics and Telecommunications University of Trento, Italy. 2005. [online]. [cit.2023.08.01.] Elérhető: <<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8314bf30780871868076775ba62759f1faf8c9f0>>

Sachin Desale, Akhtar Rasool, Sushil Andhale, Priti Rane. *Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey*. INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS VOLUME 2, ISSUE 5, MAY 2015, PP 296-304. ISSN (ONLINE): 2349-7084. 2015. [online]. [cit.2023.08.01.] Elérhető: < <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=039da802e218c89a3c49e235af8e555b2fdfa063> >

Cleve Moler, *A Brief History of MATLAB*, mathworks.com. [online]. 2018. [cit.2023.08.06.] Elérhető:< <https://www.mathworks.com/company/newsletters/articles/a-brief-history-of-matlab.html> >

Ashis Kumar Ratha, Shibani Sahu, Priya Meher. *HTML5 in Web Development: A New Approach*. Asst. Prof, Department of Computer Science & Engg., VIT, Bargarh, Odisha, INDIA Student Researcher, Department of Computer Science & Engg. VIT, Bargarh Odisha, INDIA, International Research Journal of Engineering and Technology (IRJET), ISSN: 2395-0072. 2018. március. [online]. [cit.2023.08.06.] Elérhető: < https://d1wqtxts1xzle7.cloudfront.net/56779892/IRJET-V5I3123-libre.pdf?1528799028=&response-content-disposition=inline%3B+filename%3DHTML5_in_Web_Development_A_New_Approach.pdf&Expires=1691336829&Signature=EAjFisVY8aSOQ5LZWYBLkvzo87X~uVqm1NmNwpysDHJ4YDkve0sIK5m-IOC0Piild0R-CXljevqdkTybJjTEeoNCJjrSnEanabdJGOpfNEByZQ8zje33Q~ZRms0afJ31ObrflRFoJ0HBM-euDmCIGp2ByUiUHBvnO7DUX9wGuvFKB5We~wDSZ-Mj0WomRyC-lgY3vlsvsim7kO-suRBZkpNXSy0E2J7DtmB~XC8ff8cMi0~WSssxfGdQJgeLp4OBpYwKumg39SjqGMWd62b2WDii-erHo7~3hLY6Z9dKjO7vtmRVmeV8DkahOFodNYMdLMuqWuq6RWe3RRuo4m-xpgQ_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA >

Mellékletek



32. ábra: 4.1. fejezet folyamatábrája **Forrás:** saját szerkesztés (draw.io)

```

FÜGGVÉNY Main
    imgOriginal, entropyOriginal = BetöltKépÉsEntrópia()
    randomSequence = PszeudoVéletlenSzekvencia()
    TitkosításDekódolás('XOR', imgOriginal, randomSequence)
    TitkosításDekódolás('Substitution', imgOriginal, randomSequence)
    TitkosításDekódolás('Bit Shift', imgOriginal, randomSequence)
    ÁbrázoldAzEredményeket()
VÉGE FÜGGVÉNY

FÜGGVÉNY PszeudoVéletlenSzekvencia
    Inicializáld a Hopfield hálót
    randomSequence = 0
    CIKLUS i = 1-től n-ig
        randomSequence[i] = HopfieldFrissítés(i)
    VÉGE CIKLUS
    VISSZATÉR randomSequence
VÉGE FÜGGVÉNY

FÜGGVÉNY TitkosításDekódolás(módszer, imgOriginal, randomSequence)
    imgEncrypted = Titkosít(imgOriginal, randomSequence, módszer)
    imgDecrypted = Dekódol(imgEncrypted, randomSequence, módszer)
    KiírEntrópia(imgOriginal, imgEncrypted, imgDecrypted)
VÉGE FÜGGVÉNY

```

33. ábra: 4.3. fejezethez saját értelmezésű pszeudokód **Forrás:** saját szerkesztés