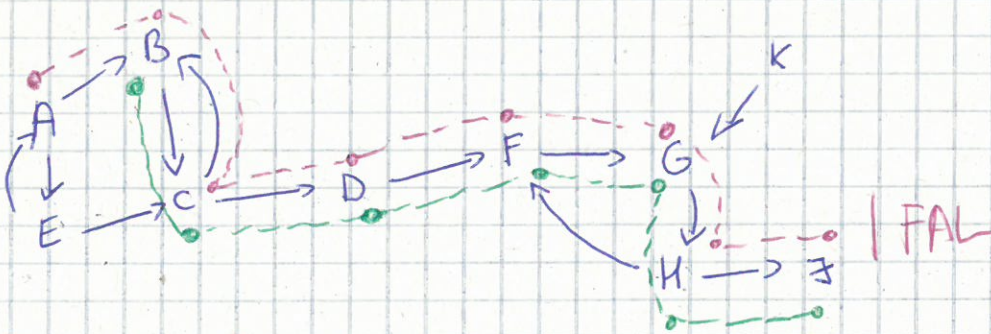


Mélységi keresés



A klasszikus algoritmus:

	A	B	C	D	E	F	G	H	J	K
Szin	Sz	Sz	Sz	Sz	Sz	Sz	Sz	Sz	Sz	Sz
Apa	—	A	B	C	A	D	F	G	H	—
d	1	2	3	4	16	5	6	7	8	19
f	18	15	14	13	17	12	11	10	9	20
	②	④	⑤	⑥	③	⑦	✓	✓	①	vissza

Verem

- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- A

Az összefüggő komponenssel vissza felé: UNILAK MEGF.

	A	B	C	D	E	F	G	H	J	K
Szin	Sz	Sz	Sz	Sz	Sz	Sz	Sz	Sz	Sz	Sz
Apa	—	—	B	—	A	—	H	F	—	—
d	3	7	8	11	4	13	15	14	19	1
f	6	10	9	12	5	18	16	17	20	2

Verem

- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~
- ~~X~~

Innen:

(C-B)

(A-E)

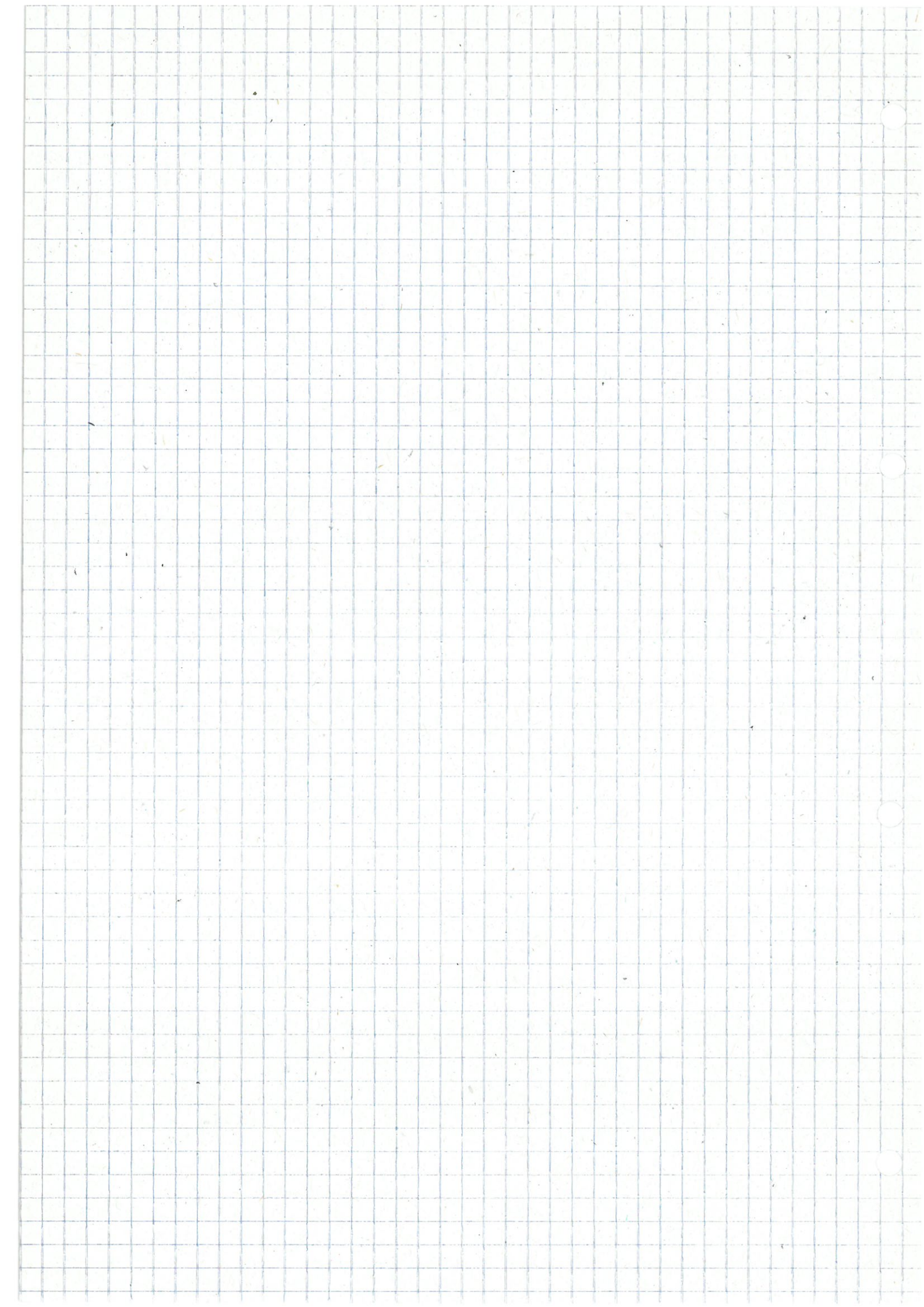
(G-H-F)

lesz

(D)

(J)

(K)





B-C A-E D F K
= G-H-I

Autoenkóder-alapú jellemzőreprezentáció mely neuronhálós, ultrahang-alapú némabeszéd-interfészekben

Gosztolya Gábor^{1,2}, Pintér Ádám¹, Tóth László¹, Grósz Tamás¹,
Csapó Tamás Gábor^{3,5}, Markó Alexandra^{4,5}

¹Szegedi Tudományegyetem, Informatikai Intézet
²MTA-SZTE Mesterséges Intelligencia Kutatócsoport
³Budapesti Műszaki és Gazdaságtudományi Egyetem,
Tavközlési és Médiainformatikai Tanszék
⁴Eötvös Loránd Tudományegyetem, Fonetikai Tanszék
⁵MTA-ELTE Lendület Lingvális Artikuláció Kutatócsoport
{ ggabor, tothl, groszt } @ inf.u-szeged.hu
csapot @ tmit.bme.hu, marko.alexandra @ btk.elte.hu

Kivonat A neurális hálón alapuló némabeszéd-interfészek általában a teljes ultrahangkép alapján becslik meg a spektrális paramétereket, melyekből a vokóder aztán beszédet generál. Habár ez a megközelítés igen kézenfekvő, és tapasztalataink szerint érthető beszédet képes generálni, több hátránya is van: egyrészt nehezen ragadja meg az egymáshoz közel eső területek (gyakorlatilag a pixelek) közötti összefüggéseket, másrészt igen pazarló. Könnyen belátható, hogy a képpontok egy jelentős része irreleváns a spektrális paraméterek becslése szempontjából, a szomszédos képpontok által tárolt információ nagyon redundáns, a mely háló mérete pedig jelentős a sok jellemző miatt. Jelen cikkünkben ezen problémák kezelésére egy autoenkóder neurális hálót tanítunk az ultrahangképre, és a szintézis spektrális paramétereinek becslését az autoenkóder háló bottleneck rétegének neuronjainak aktivációi alapján végezzük egy második mély hálóval. Kísérleti eredményeink alapján a javasolt eljárás hatékonyabb, mint a hagyományos megközelítés: a kapott átlagos négyzetes hibák minden esetben alacsonyabbnak, a korrelációértékek pedig magasabbnak voltak, mint a standard technikával kapottak. További előnye az eljárásnak, hogy a bottleneck réteg (relative) alacsony neuron-száma miatt, több szomszédos kép felhasználása a becslés során sem jár a paraméterszám lényeges növekedésével, miközben szignifikánsan javítja a paraméterbecslés pontosságát.

Kulcsszavak: némabeszéd-interfész, mely neuronháló, autoenkóder

1. Bevezetés

Az utóbbi évtizedben megnőtt az érdeklődés a beszédjel artikulációs jellemzőkből való helyreállítása iránt, ami az ún némabeszéd-interfészek (Silent Speech Interface, SSI) alapját képezi [1]. Ezen a területen a feladat a beszédjel rekonstrukciója az artikulációs szervek (pl. nyelv vagy ajkak) mozgásából anélkül, hogy az

alany valóban beszédjelet produkálna. A némabeszéd-interfészeknek kézenfekvő alkalmazási területeik lehetnek a beszédképzésben sérültek (pl. gégeeltávolítottakon átesett betegek) életminőségének javításában, illetve a beszéd továbbításában extrémén zajos környezetben (pl. katonai alkalmazásokban). Az artikulációs adatok rögzítése történhet ultrahangos képalkotással (ultrasound tongue imaging, UTI) [2,3,4,5,6], elektromágneses artikulográfiával (electromagnetic articulography, EMA) [7,8], állandó mágneses artikulográfiával (permanent magnetic articulography, PMA) [9], elektromiográfiával (electromyography, EMG) [10], vagy a fentiakat keverő multimodális megoldásokkal [11].

A jelenlegi legkorszerűbb SSI rendszerek a „közvetlen szintézis” alapvető alkalmazás, vagyis a beszédjelet közvetlen átalakítások (pl. beszédhangok felismerése) nélkül, közvetlenül az artikulációs szervek mozgásából kinyert jellemzőkből állítják elő, vokóder használatával [3,4,5,8,9]. Ebben a folyamatban egy hang-súlyos gépi tanulási lépés az artikulációs jellemzők (pl. ultrahangképből nyert vektorok) alapján a vokóder (spektrális) paramétereinek becslése, melyre általában mely neurális háló (Deep Neural Network, DNN, pl. [6,8,9]) vagy Gauss keverékmoddell (Gaussian Mixture Model, GMM, pl. [12,13]) szokás használni.

Speciálisan, az ultrahangkép-alapú SSI esetében a gépi tanuló eljárás bemenetét egy képkocka pixeljei jelentik. Könnyen látható, hogy ez a megközelítés, bár kézenfekvő és korábbi tapasztalataink (ld. pl. [6,14,15,16]) alapján érthető beszéd szintetizálását teszi lehetővé, több tekintetben is szuboptimális. A bemenetként használt, képenként több ezer képpont nagyméretben redundáns, valamint sok irreleváns jellemzőt is tartalmaz (bár ezen jellemzők kiválasztásával lehet segíteni [14]). A túl sok jellemző az alkalmazott mely háló hatékonyságára (tanítási és kiértékelési idő, tárolt súlyok száma) egyértelműen negatív hatással van, és a spektrális paraméterek becslését is rontja. Egy hatékony tömörítési eljárással mindkét területen javíthatunk.

Jelen cikkünkben a bemenetként használt ultrahangképet egy autoenkóder hálózat segítségével tömörítjük, és a beszédsszintézis spektrális paramétereit a bottleneck réteg aktivációit mint jellemzőket használva becsüljük egy második mely neurális hálóval. *Handwritten note: A képet az autoenkóder hálózat segítségével a bottleneck réteg aktivációi alapján a második hálóra becsüljük.*

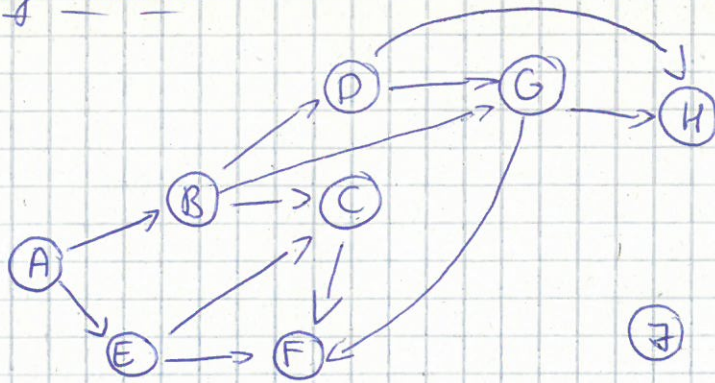
2. Némabeszéd-interfész spektrális paramétereinek becslése autoenkóder hálók használatával

Handwritten note: A képet az autoenkóder hálózat segítségével a bottleneck réteg aktivációi alapján a második hálóra becsüljük.

2.1. Autoenkóder neurális hálók

Az autoenkóder neurális hálózat egy olyan felügyelet nélküli gépi tanulási eljárás, mely a rejtett rétegekben az eredeti információ kódolt változatát állítja elő, majd ezt a kimeneti rétegre visszafejti [17]. Célja, hogy bejövő paraméterekből egy identitásfüggvényhez hasonló lekepezést tanuljon meg egy tömörebb vagy részletesebb reprezentáción keresztül. Technikailag általában egy olyan neurális hálóval valószínűsítjük meg, melynek az elvárt kimenete megegyezik a bemenettel. Tömörítéskor az egyik rejtett rétegnek a bemenő jellemzők számánál lényegesen

Szélességi keresés



	d	Apa	Szín
A	0	—	Szürke Fekete
B	1	A	Szürke Fekete
C	2	B	Szürke Fekete
D	2	B	Szürke Fekete
E	1	A	Fekete Szürke
F	2	E	Szürke Fekete
G	2	B	Szürke Szürke Fekete Fekete
H	3	D	Szürke
I	∞	nincs	Fekete

Rendezett tábla:

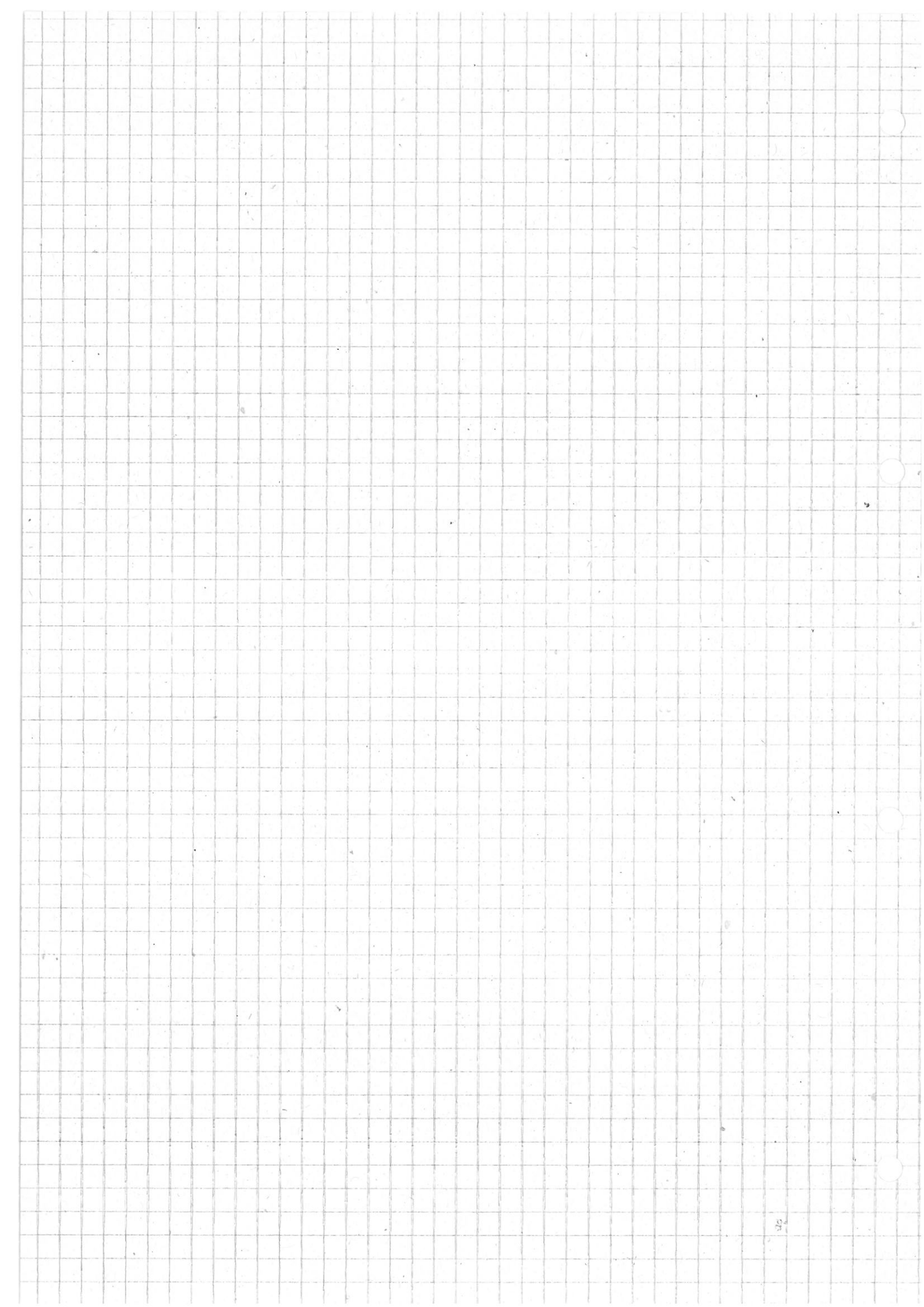
A	B	E	D	G	C	F	H
0	1	1	2	2	2	2	3

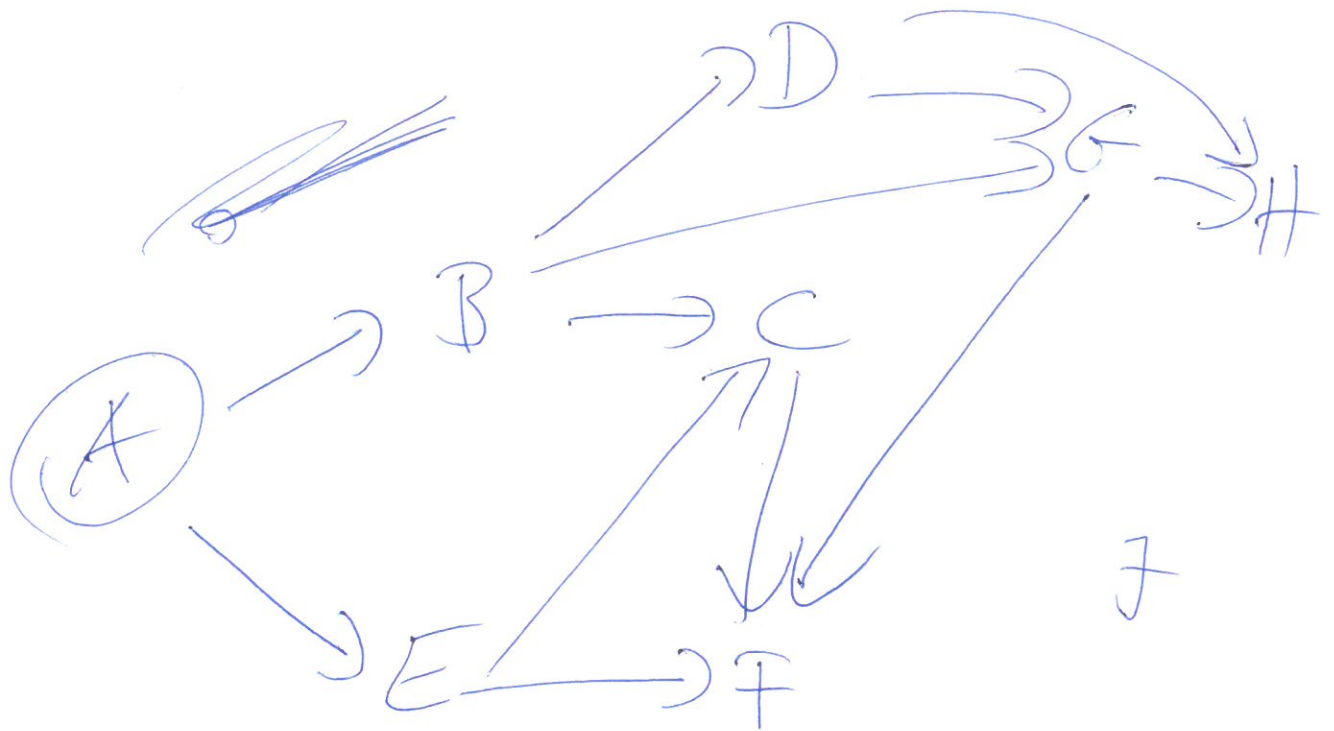
út H-ig:

A — B — D — H

út F-ig:

A — E — F





	d	Appa	Keep seen
A	0	—	✓ ✗
B	1	A	✗
C	2	B	✗
D	2	B	✗
E	1	A	✗
F	2	E	✗
G	2	B	✗
H	3	D	✗
7	∞	null	A - B - D - H

~~✗ ✗ ✗ ✗ ✗ ✗ ✗ ✗~~

1. Tömb és láncolt lista közül melyiket ideálisabb használni az elemek gyors elérésére és melyiket az elemek gyors beszűrésére?

2. Adott a 200, 180, 221, 101, 176, 110 számsorozat, amit ebben a sorrendben tárolunk el. Milyen sorrendben vehetjük ki az elemeket verem/sor/prioritási sor adatszerkezet esetén? (szerintem elég egy adatszerkezetet egy kérdésre)
3. Adj példát egy FIFO adatszerkezetre és indokold is meg, hogy miért FIFO!
4. Adj példát egy LIFO adatszerkezetre és indokold is meg, hogy miért LIFO!
5. Szeretnénk emberek magasságait letárolni, majd a legálacsonyabb magasságot gyorsan lekérdezni. A gyakorlaton tanult adatszerkezetek közül melyik adatszerkezetet ideális használnunk?

Hetedik gyakorlat

1. Építsen fel egy bináris keresőfát az alábbi számokból (ebben a sorrendben): 8, 9, 2, 4, 12, 17
2. Hogyan találjuk meg egy bináris keresőfa maximális elemét?
3. Hogyan találjuk meg egy bináris keresőfa minimális elemét?
4. n elemből építünk fel egy bináris keresőfát. Mekkora lehet a minimális és a maximális magassága?

Vanda kérdései:

5. Adj egy olyan 10 tagú számsorozatot, amit ebben a sorrendben bináris fába szűrve maximális magasságú lesz a fa (azaz 9 magasságú).
6. *egy tetszőleges gráfot felrajzolnék a táblára és megkérdezném, hogy * Mely tulajdonság igaz a felrajzott gráfra a következők közül: fa, bináris fa, teljes bináris fa, majdnem teljes bináris fa, kiegyensúlyozott bináris fa.

Nyolcadik gyakorlat

1. Mi a hash függvény?
2. Milyen hash függvényeket ismer?
3. Mi a hash tábla?

Vanda kérdései:

4. Hash táblák kapcsán mikor beszélünk ütközésről?
5. Milyen módszert használhatunk ütközésfeloldásra?
6. Hogy tároljuk el az ütközött elemeket láncolt hash táblában?
7. Láncolt hash táblánál mit jelent a kitöltési tényező/load factor?
8. Mi az ütközésfeloldásra használt nyílt címzés/open adressing módszer lényege?
9. Egy 8 hosszú hash tábla 4 elemet tartalmaz. Mi a hash tábla load factorja/kitöltési tényezője?
10. Szűrjük be az alábbi hash kódú elemeket egy 3 hosszú láncolt hash táblába!
 $h(\text{"tonya"}) = 4$
 $h(\text{"bonya"}) = 1$
 $h(\text{"afonya"}) = 2$
 $h(\text{"alga"}) = 9$
 $h(\text{"dimat"}) = 5$
 $h(\text{"mestint"}) = 5$
11. Szűrjük be az alábbi hash kódú elemeket egy 10 hosszú nyílt címzésű hash táblába lineáris probing/lineáris kipróbálás módszerrel!
 $h(\text{"tonya"}) = 14$
 $h(\text{"bonya"}) = 11$
 $h(\text{"afonya"}) = 31$
 $h(\text{"alga"}) = 55$
 $h(\text{"dimat"}) = 15$
 $h(\text{"mestint"}) = 65$

Kilencedik gyakorlat

1. Mi a rendezési feladat?
2. Hogyan működik a beszűrő rendezés és mennyi a műveletigénye?
3. Milyen $O(n \log n)$ -es rendezéseket ismer?
4. Milyen tulajdonsággal definiálunk egy kupacot?

Vanda kérdései:

5. Felrajzolnék egy bináris fát és megkérdezném, hogy kupac-e? Indoklással.
6. A 4. Feladatból könnyebbeket kérdeznék, mondjuk az első kettőt párban (vagy a harmadikat, illetve az utolsó!)
7. Egy konkrét tömb leszámoló rendezése, úgy hogy a köztes segéd tömböket is meg kell adni (tehát kellene a gyakoriságot és a kumulatív gyakoriságot tartalmazó tömbök, de a pszeudokód nem).

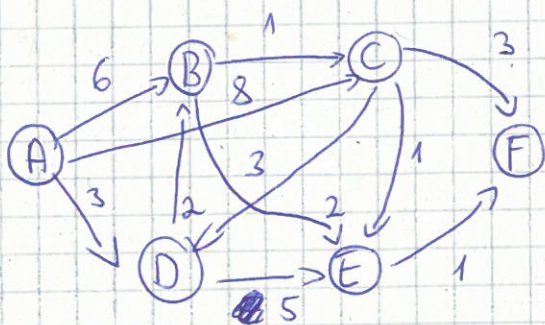
Tizedik gyakorlat

1. Adja meg egy ötszög gráfiának szomszédsági listás ábrázolását!
2. Adja meg egy ötszög gráfiának szomszédsági mátrix ábrázolását!
3. Adja meg egy ötszög gráfiának szélességi bejárását (tetszőleges csúsból)!
4. Adja meg egy ötszög gráfiának mélységi bejárását (tetszőleges csúsból)!

Vanda kérdései:

5. Igaz-e, hogy ha egy erősen összefüggő irányított gráf irányításait elhagyjuk, akkor egy összefüggő (irányítatlan) gráfot kapunk? A választ indokolni kell!
6. Felrajzolnék egy egyszerű irányított gráfot (mondjuk szemetszűrő hídral, meg egy izolált ponttal) és megkérdezném, hogy mik az erősen összefüggő komponensei (karikázni elég).
7. Feladatunk egy irányítatlan körmentes gráf egy topologikus rendezését megadni. Ez azonban értelmetlen feladat(nem lehet megcsinálni). Miért?
8. Adott a $G(V,E)$ irányított gráf, melynek csúcsai: $V = \{1, 2, 3, 4\}$ és élei: $E = \{(1, 2), (2, 4), (1, 3), (3, 4)\}$ (ezt felrajzolnám a táblára). Igaz-e, hogy G egy topologikus rendezése 2, 3, 1, 4? A választ indokolni kell!

Dijkstra - algoritmus:



		d	Apa	Kész?
①	A	0	-	✓
③	B	5	A	✓
④	C	6	B	✓
②	D	3	A	✓
⑤	E	7	B	✓
(6)	F	8	E	✓

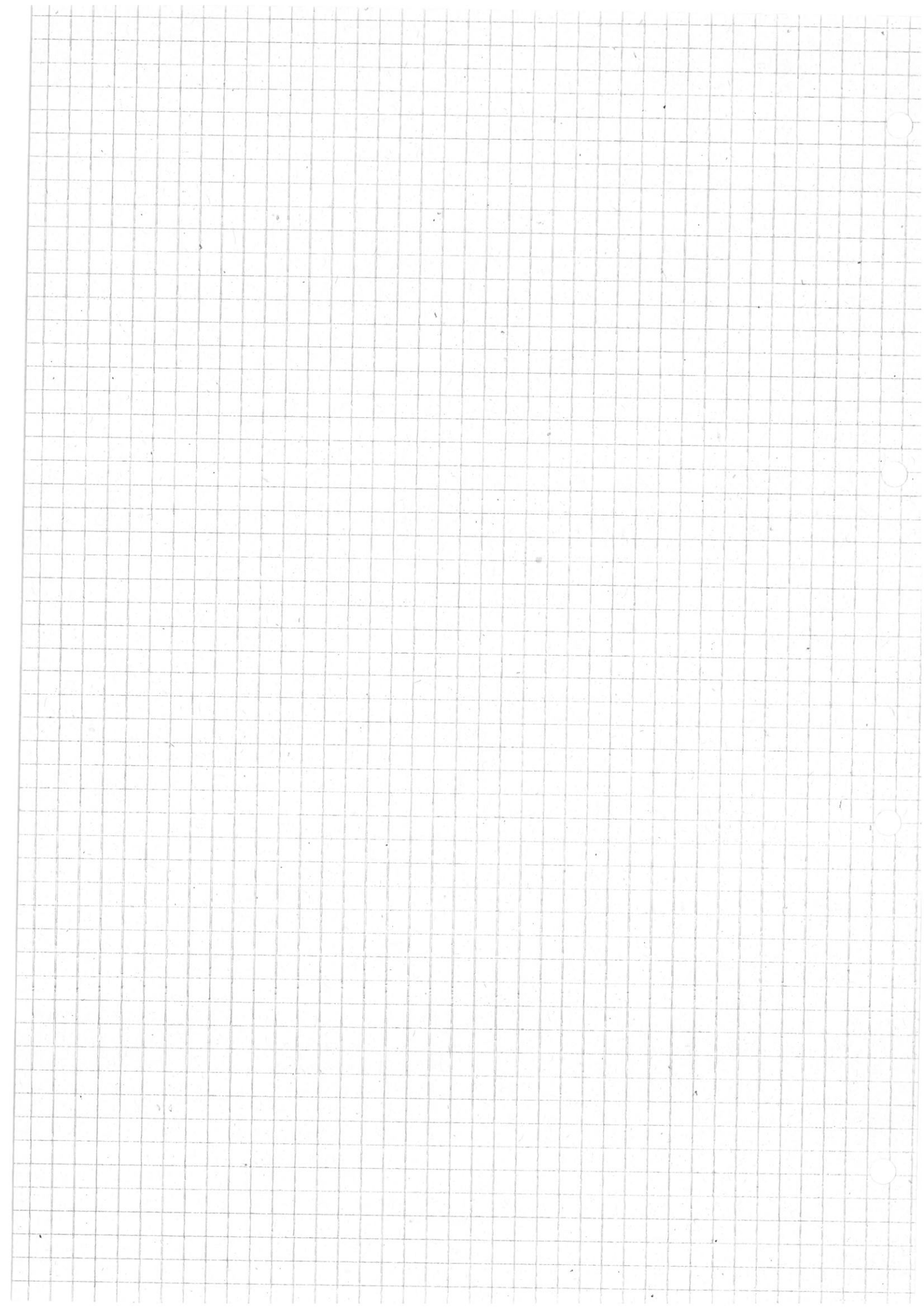
Legrövidebb út F-be:

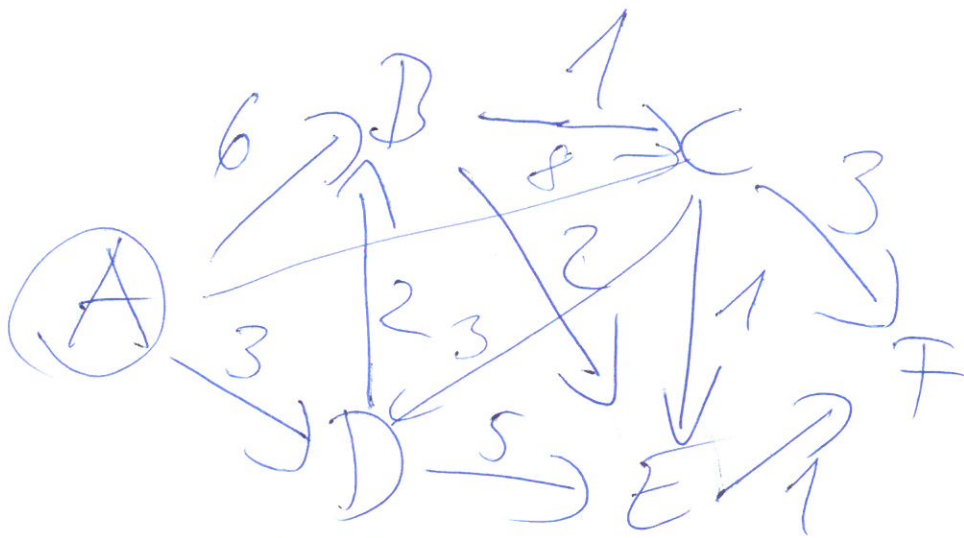
A - D - B - E - F

Legrövidebb út C-be:

A - D - B - C

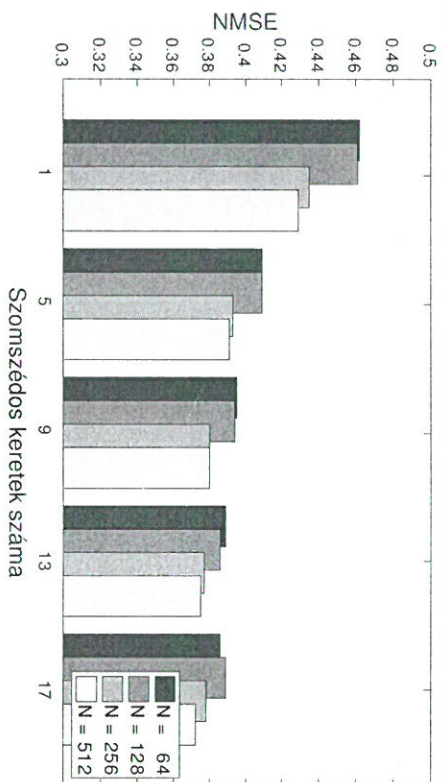
...





	d	Apa	Kész
A	0	-	✓
B	5	A D	✓
C	6	A B	✓
D	3	A	✓
E	7	A B	✓
F	8	A E	✓

A - D - B - E - F



4. ábra: A tesztalacson mért átlagos normalizált hibaértékek az autoenkóder háló bottle-nek rétegének neuron-száma (N) és a használt szomszédos keretek számának függvényében.

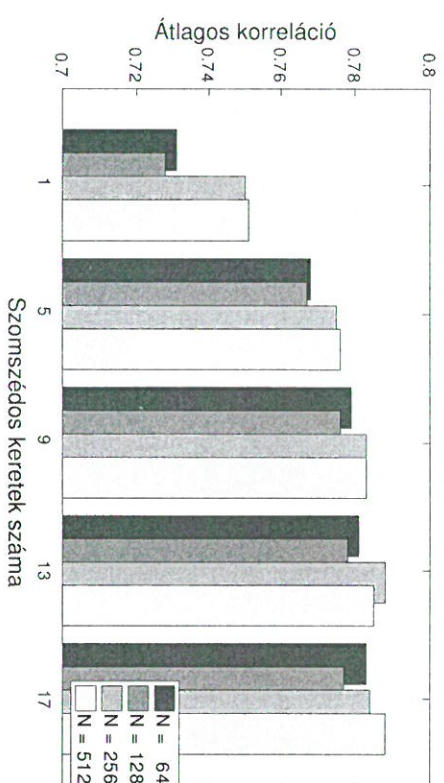
1. táblázat. Az elért...eredmények

Megközelítés	Szomsz. száma	Param. száma	NMSE		Korreláció	
			Fejl.	Test	Fejl.	Test
Standard	1	12,6M	0,529	0,534	0,680	0,676
	5	46,2M	0,523	0,530	0,684	0,680
Autoenkóder, $N = 64$	1	4,8M	0,459	0,462	0,731	0,729
	9	5,3M	0,390	0,395	0,779	0,776
Autoenkóder, $N = 256$	1	6,6M	0,432	0,435	0,750	0,749
	5	7,6M	0,395	0,393	0,775	0,777
	9	8,7M	0,384	0,380	0,783	0,786
	13	9,7M	0,376	0,377	0,788	0,787
Autoenkóder, $N = 512$	1	8,9M	0,430	0,429	0,751	0,752
	5	11,0M	0,394	0,391	0,776	0,778
	9	13,1M	0,382	0,380	0,783	0,785

5. Összegzés

Köszönetnyilvánítás

A kutatást részben a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatja (FK 124584). Tóth László munkáját az MTA Bolyai János Kutatási Ösztöndíja támogatja. Grósz Tamás az Emberi Erőforrások Minisztériuma UNKP-17-3 kódszámú Új Nemzeti Kiválóság Programja támogatja. A cikk elkészítéséhez használt Tian-X grafikus kártyát az NVIDIA Corporation adományozta.



5. ábra: A fejlesztési hálazon mért átlagos korrelációértékek az autoenkóder háló bottle-nek rétegének neuron-száma (N) és a használt szomszédos keretek számának függvényében.

Hivatkozások

- Denby, B., Schultz, T., Honda, K., Hueber, T., Gilbert, J.M., Brumberg, J.S.: Silent speech interfaces. *Speech Communication* **52**(4) (2010) 270–287
- Denby, B., Stone, M.: Speech synthesis from real time ultrasound images of the tongue. In: ICASSP, Montreal, Canada (2004) 685–688
- Hueber, T., Benaroya, E.I., Denby, B., Chollet, G.: Statistical mapping between articulatory and acoustic data for an ultrasound-based silent speech interface. In: Interspeech, Florence, Olaszország (2011) 593–596
- Hueber, T., Bailly, G., Denby, B.: Continuous articulatory-to-acoustic mapping using phone-based trajectory HMM for a silent speech interface. In: Interspeech, Portland, USA (2012) 723–726
- Jaumard-Hakoun, A., Xu, K., Lehoullenger, C., Roussel-Ragot, P., Denby, B.: An articulatory-based singing voice synthesis using tongue and lips imaging. In: Interspeech, San Francisco, USA (2016) 1467–1471
- Csapó, T.G., Grósz, T., Tóth, L., Markó, A.: Beszédszintézis ultrahangos artikulációs felvételekből mély neuronhálók segítségével. In: MSZNY 2017, Szeged (2017) 181–192
- Wang, J., Samal, A., Green, J.: Preliminary test of a real-time, interactive silent speech interface based on electromagnetic articulograph. In: SPLAT, Baltimore, USA (2014) 38–45
- Bocquet, F., Hueber, T., Girin, L., Savariaux, C., Yvert, B.: Real-time control of an articulatory-based speech synthesizer for brain computer interfaces. *PLOS Computational Biology* **12**(11) (2016) e1005119
- Gonzalez, J.A., Cheah, L.A., Green, P.D., Gilbert, J.M., Ell, S.R., Moore, R.K., Holdsworth, E.: Evaluation of a silent speech interface based on magnetic sensing and deep learning for a phonetically rich vocabulary. In: Interspeech, Stockholm, Svédország (2017) 3986–3990