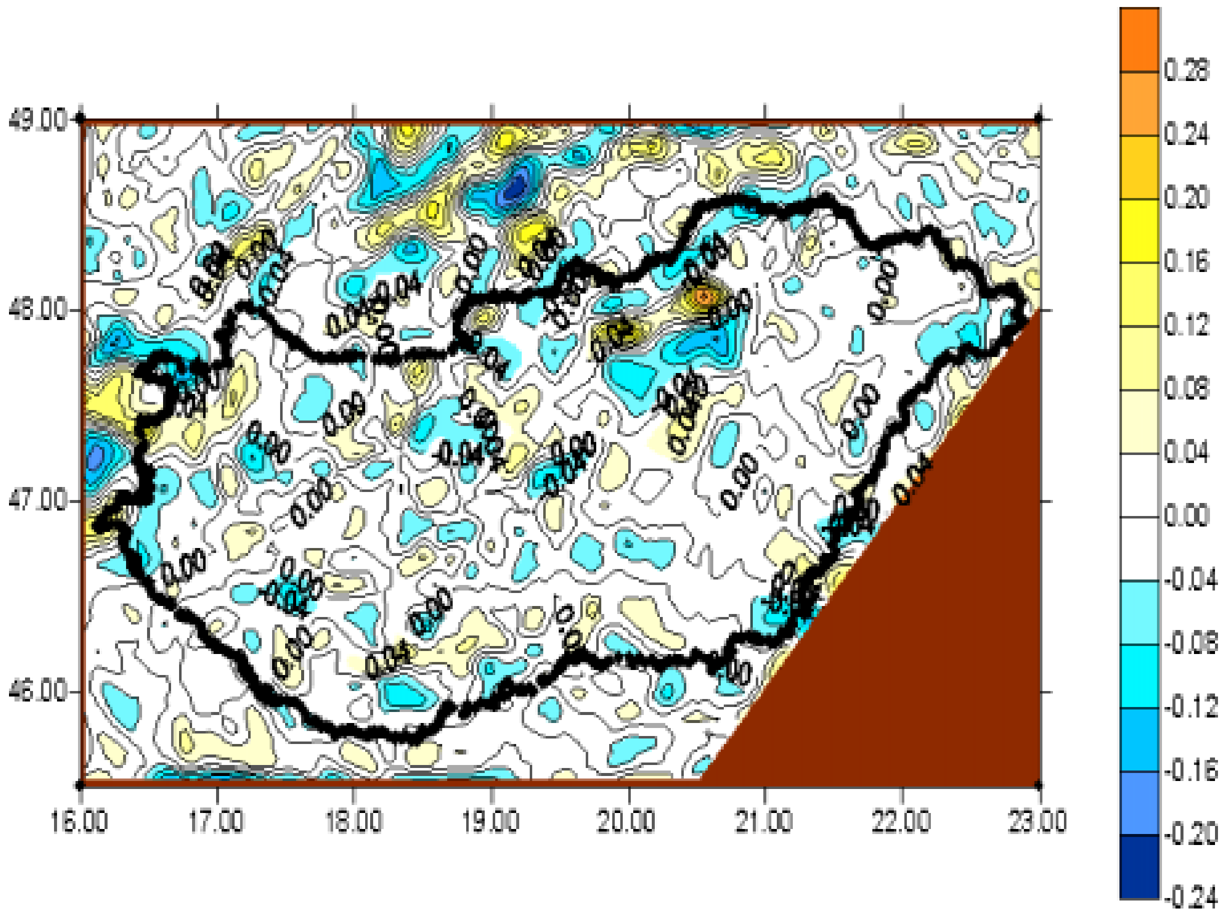
Magyarországi geoidfelület közelítése neurális hálózatokkal

Készítette: Zaletnyik Piroska
földmérő és térinformatikai hallgató

Konzulensek:
Dr. Völgyesi Lajos, Általános és Felsőgeodézia Tanszék,
Dr. Paláncz Béla, Fotogrammetria és Térinformatika Tanszék

Tudományos Diákköri Konferencia
Budapest, 2002. november



Tartalomjegyzék

Ábrajegyzék

1. Bevezetés

A dolgozatomban Völgyesi Lajos és Paláncz Béla kutatását folytattam (2002), amelyben azt vizsgálták, hogyan lehet neurális hálózatokkal közelíteni a magyarországi geoid felületét.

A korábbi kutatások alapján Magyarország területére meghatározott (HGTUB2000) gravimetriai geoidmegoldást (Tóth, Rózsa 2000) használtam fel kiindulási adatként. Ez a $45°30' \leq \varphi \leq 49°$, $16° \leq \lambda < 23°$ nagyságú területen tartalmazza a geoid-ellipszoid távolságokat $\Delta\varphi=0'30'' \times \Delta\lambda= 0'50''$ felbontással.

Munkám célja, az volt, hogy olyan függvényt állítsak elő, amely megfelelő pontossággal közelíti meg a geoid felületét Magyarország területén. Egy ilyen függvénynek igen jelentős gyakorlati haszna lehet. Az egyik olyan terület, ahol ezt lehetne használni, az a GPS mérés. GPS mérésből csak ellipszoid feletti magasságokat kapunk, holott mi többnyire tengerszint (geoid) feletti magasságokkal dolgozunk. Ahhoz hogy ellipszoid feletti magasságból tengerszint feletti magasságot kapjunk, ismerni kell a geoid-ellipszoid távolságát. Ez a probléma megoldható úgy is, hogy a mérés helyén interpolálunk az ismert rácsháló pontjait felhasználva. Ez az adatbázis viszont igen sok adatot tartalmaz (211680 pont), és így nem túl egyszerű a használata terepen. Sokkal könnyebb, ha csak egyetlen függvényt kell helyette alkalmazni, ami bármilyen programozási nyelven könnyen leírható és beépíthető akár egy GPS vevőbe is.

Felmerül a kérdés, hogy mért ne oldhatnánk meg ezt a feladatot polinommal való közelítéssel. Völgyesi Lajos vizsgálta ezt a lehetőséget. Ő azt találta, hogy a polinom fokszámának növelésével a hibák egy ideig csökkennek, egészen a nyolcad fokú polinomig, utána viszont rosszabbak lesznek az eredmények, mivel a megoldás rosszul kondicionált egyenleteket eredményez.

Új megoldási lehetőséget kínál viszont a mesterséges intelligencia körébe tartozó neurális hálózat, mellyel én is foglalkoztam.

2. Röviden a neurális hálózatokról

Nem véletlen, hogy a legtöbb embernek a neurális hálózatokról a biológiai tanulmányai jutnak eszébe és az emberi idegsejtre, a neuronra gondol. Az képezte ugyanis ennek az új számítási eszköznek az alapját, hogy megpróbálták az emberi gondolkodást ill. az idegsejtek működését utánozni.

A nagyon egyszerű felépítésű idegsejtet tanulmányozva igen érdekesnek tűnik, hogy ugyanazok ill. hasonló sejtek milyen sokféle feladatot képesek megoldani különböző hálózatokban. Ennek a számítástechnikai megoldása a neurális hálózat, amely, akárcsak az ember, tanulás útján képes megoldást találni különböző problémákra.

Ezek a rendszerek képesek olyan feladatokat megoldani, amelyek nem algoritmizálhatóak bonyolultságuk miatt, ill. bizonyos feladatokra sokkal gyorsabban, hatékonyabban képesek megoldást találni. Megoldhatóak velük pl. olyan feladatok is, amelyeknél nem ismerjük a kapcsolatot a bemenő és a kimenő adatok között csak sejtjük, hogy van valami összefüggés. Jellegzetes felhasználási területei pl. a szöveg-, beszéd- és hangfelismerés és optimalizálási feladatok. Már több geodéziai célú alkalmazás is készült. Az egyik koordinátatranszformáció meghatározására (Barsi 1999), a másik magasságok meghatározására (Veres 2002).

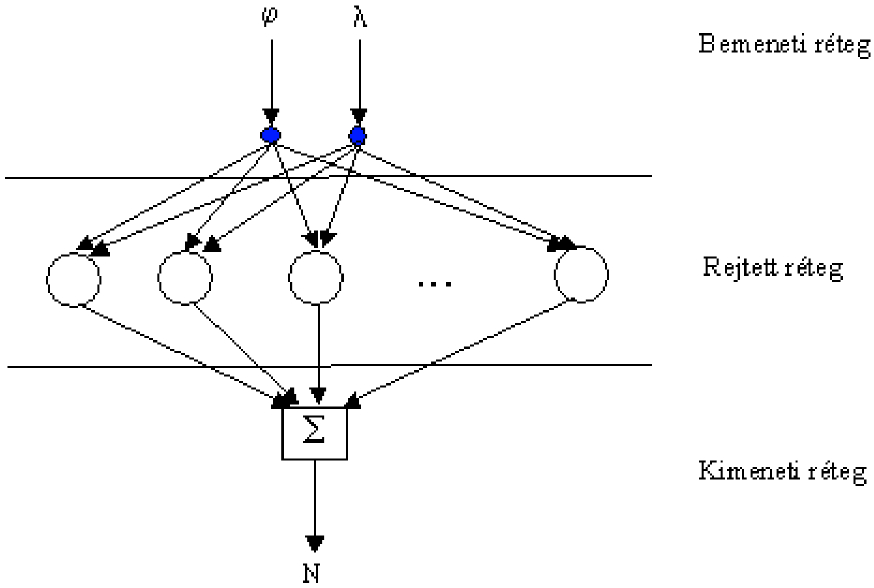
Igen fontos jellegzetessége ezeknek a hálózatoknak az approximációs vagy leképzést közelítő tulajdonság. Ezt úgy lehet elképzelni, hogy van egy rendszerünk, vagy feladatunk, aminek ismerjük a bemenő adatait és a kimenő adatait, de nem ismerjük az ezek közötti összefüggést, kapcsolatot, vagyis a leképzést. Legyen a bemenő adat x, a kimenő y, és a kettő közti leképzés *f*. Ekkor y=*f*(x), ahol *f* ismeretlen. Ilyenkor neurális hálózatokkal előállíthatunk egy modellt a leképzés közelítésére, ahol pl. a legkisebb négyzetek módszerét felhasználva minimalizáljuk a hibákat. Az eredeti rendszer x bemenő értékre adott válasza y, a közelítő modell válasza pedig legyen y'. Ilyenkor a cél, hogy y legyen egyenlő y'-vel, vagy, hogy a kettő közti eltérés (y-y') minimális legyen. A modellt tanító adatok segítségével lehet előállítani (Horváth 1995).

Ezt a tulajdonságát lehet felhasználni valamilyen felület, jelen esetben a geoid felület, közelítése során is. Az eltérés itt abban van, hogy most két bemenő adatunk van, a két vízszintes koordináta, és egy kimenő adat, a geoid -ellipszoid távolság. A kettő közötti összefüggést keressük neurális hálózatokkal.

A feladat megoldásához a Mathematica szoftvert ill. az ehhez tartozó neurális hálózatok kiegészítő modult használtam. Jelenleg több szoftver felhasználásával is lehet ilyen rendszereket készíteni. A Mathematica előnye, hogy nem csak numerikus, hanem szimbolikus számításra is képes. A legtöbb rendszer úgy működik, hogy meg kell adni a bemenő számadatokat, és a rendszer kiadja az eredményt szám formában, anélkül, hogy tudnánk közben pontosan mi is történt a hálózatban. Ezzel szemben a Mathematica-ban a bemenő adatok lehetnek változók is (pl. x,y) és ilyenkor eredményként a leképzés függvényét kapjuk meg (*f*(x,y)), amit később máshol is közvetlenül felhasználhatunk.

3. A hálózatok felépítése, RBF hálózatok

A neurális hálózatokat úgy lehet szemléltetni, mint egy több rétegből álló rendszert, ahol a rétegekben csomópontok, azaz neuronok helyezkednek el. Ezek a rétegek a következőek: bemeneti réteg, egy vagy több rejtett réteg és egy kimeneti réteg. A következő ábra az általam használt, egy rejtett réteget tartalmazó hálózatot ábrázolja.



1. ábra, Egy rejtett rétegű neurális hálózat felépítése

Egy neuron[1] több bemenettel és egy kimenettel rendelkezik. A rejtett rétegben lévő neuronok számát mi határozhatjuk meg az adott feladat függvényében.

A neuron a bemeneti komponensek súlyozott összegét határozza meg és ezen hajt végre valamilyen nem lineáris leképzést. Ez utóbbit nevezik aktivációs függvénynek. A végeredmény a neuron kimeneti jele.

A hálózat súlyait tanítással állíthatjuk elő. A tanulás során a rendszer az ismert és összetartozó be- és kimeneti adatok között keresi meg a kapcsolatot leíró függvényt. Konvergens, iteratív eljárással határozza meg a különböző súlyokat, paramétereket (Horváth 1995).

Az általam alkalmazott aktivációs függvény az RBF (Radial Basis Function = radiális bázisú függvény). Ez egy igen hatékony és gyorsan tanuló eljárás, amit nagyon jól lehet alkalmazni szabálytalan felületek közelítésére. Maga a függvény egy Gauss-féle haranggörbére emlékeztet. A függvénynek két paramétere van: λ, β.

$$e^{-\lambda^2 (x-\beta)^2} \qquad (1)$$

Mint korábban említettem a hálózat eredménye a neuronok kimenő adatainak a súlyozott összege, két bemenő adat (x,y) esetében ez a következő:

$$z(x,y) = \sum_{i=1}^{n} w_i e^{-\lambda 1_i^2 (x-\beta 1_i)^2 - \lambda 2_i^2 (y-\beta 2_i)^2} \qquad (2)$$

ahol n az alkalmazott neuronok száma és w_i a súlyok értéke. A függvény bonyolultsága tulajdonképpen a neuronszámtól függ. Ezt a rendszer tanítása előtt mi adhatjuk meg. Minél több neuront alkalmazunk, annál bonyolultabb felületeket tudunk

[1] A neuronokat szokták csomópontoknak is nevezni, de jelen esetben ezt az elnevezést inkább kerülöm, mivel könnyen összekeverhető az alkalmazott tanító ill. tesztpontokkal, amelyek ténylegesen létező földrajzi pontok, koordinátákkal, míg a rejtett rétegbeli neuronok csupán egy képzeletbeli réteg elemei, amelyek az alkalmazott függvény bonyolultságát szabják meg.

leírni. Azonban nem érdemes a neuron számot túlságosan megnövelni, mert ez túl bonyolult hálózatot eredményez. A tanulási folyamat mindenképpen egy lassú iterációs eljárás, aminek az időszükséglete a neuronszám növelésével többszörösére nő, ugyanakkor a rendszer pontossága egy idő után már nem nő tovább arányosan, és a tanítás túl lassúvá, gazdaságtalanná válik. Az ideális neuronszámot próbálgatással lehet meghatározni az egyes hálózatokhoz, figyelembe véve a pontossági követelményeket és a rendelkezésre álló időt, számítógép kapacitást.

4. Az alkalmazott hálózat

Az általam használt hálózatban, mint azt már korábban említettem két bemenő és egy kimenő adat van. A kettő között egy rejtett réteg van (ld. 1. ábra) és abban 35 neuron. Az alkalmazott aktivációs függvény az RBF függvény.

Fontos jellegzetessége ennek a rendszernek, hogy nem egy neurális hálózattal közelíti a felületet, hanem neurális hálózatok sorozatával.

Ezt a módszert Paláncz Béla javasolta, és én ezt a modellt teszteltem le és igyekeztem tovább pontosítani. Abban a korábbi kutatásban, amit ő és Völgyesi Lajos végeztek, a már korábban említett 211680 rácsháló pontot tartalmazó adatbázisból a területen egyenletes eloszlásban kiválasztott kb. 8000 ponttal dolgoztak (Paláncz, Völgyesi 2002). Ez azt jelenti, hogy minden huszonötödik pont került bele a tanítópontok közé. Az eredeti adatok a 45°30'≤ φ ≤ 49° , 16°≤ λ < 23° nagyságú területre voltak megadva Δφ=0'30" × Δλ= 0'50" felbontással, a kiválasztott 8484 pont ugyanezen a területen helyezkedik el Δφ=2'30" × Δλ= 4'10" felbontással.

Ezekkel a pontokkal készült el a neurális hálózat tanítása, majd a szimbolikus kiértékeléssel a leképzési függvényt ki is fejezték. Ennek az alakja a (2) képletnek is megfelelően a következő:

35 db $w_i e^{-\lambda 1_i^2 (x-\beta 1_i)^2 - \lambda 2_i^2 (y-\beta 2_i)^2}$ alakú tag + a*x+b*y+c (konstans).

35 db (2) képlettel egyező alakú tagot kaptunk az eredményben, mivel a rejtett rétegben összesen 35 neuront alkalmaztunk. Itt látszik tehát, hogyan változik a felületet leíró függvény bonyolultsága a neuronszámnak megfelelően. Mintaként álljon itt az egyik részlete a függvénynek, ahol a bemenő adatok x1 és x2 voltak:

$$5.22702 \cdot 10^{6} \cdot e^{-0.753045 \cdot (-47.7342 + x1)^2 - 0.0753045 \cdot (-21.925 + x2)^2}$$

Maga a tanítási folyamat ilyen nagyszámú (kb. 8000) pontnál igen hosszadalmas feladat, ami igen sok órát, esetleg több napot is igénybe vehet, a számítógép kiépítettségének (memória, processzor) függvényében. Az iterációk számát megadhatjuk mi is, illetve egy bizonyos minimumhatárt elérve a program magától leáll és befejezi a tanítást. Célszerű megadni egy iterációs számot, mivel nem tudhatjuk előre, hogy a gép magától hány iteráció után állna le. Persze ilyenkor is lehet az iterációt tovább folytatni, ha még nem értük el a hibák minimumát, de legalább nem szükséges egyszerre az egész tanítást végrehajtani. Az általam készített hálózatnál ezt az iterációs határt 200-ra állítottam és ezt a 200 iterációt a gép 3 óra alatt hajtotta végre (a gép adatai: 1 GHz processzor, 256 Mb RAM, ami jelenleg nem számít lassú gépnek). A hivatkozásban szereplő hálózatnál összesen kb. 920 iterációra volt szükség a minimumhatár eléréséhez, az általam továbbfejlesztett hálózatnál pedig kb. 350-re. Ezekből az adatokból is látszik, hogy ez az egész tanulási folyamat mennyire időigényes, viszont, ha már egyszer kész a hálózat ill. a leképzési függvény, akkor ennek a használata már rendkívül gyors.

A közelítés javításának persze lehetnek más módjai is, példa erre az alkalmazott neurális hálózat sorozat. Ez azt jelenti, hogy miután elkészítettük az első hálózatot a 8000 pont alapján, ki lehet számítani ugyanezekre a pontokra a hibákat, levonva a tényleges értékekből a leképzési függvénnyel előállított értékeket. Ha következő lépésként megpróbáljuk megtanítani a rendszert ezekre a hibákra, és utána hozzáadjuk az első neurális hálózat hibás értékeihez a hibákat, akkor elvileg a helyes értékeket kapjuk. Persze ez a gyakorlatban nem így van, mivel az új, a hibákat tanuló hálózat sem lesz száz százalékig tökéletes, lesznek ennek is hibái, amikre persze újabb hálózatokat készíthetünk. Ennek a sorozatnak a határértéke a tényleges kimeneti érték. Persze annak is megvan a határa, hogy meddig érdemes elmenni ilyen hálózat sorozat kialakításánál, mivel egyre hosszabb és bonyolultabb függvényeket fogunk kapni, összeadva a sok neurális hálózat leképzési függvényét. Mivel célunk elsősorban az, hogy csökkentsük a felhasznált adatok mennyiségét, hogy ne kelljen mindig a több mint 200000 adatot használnunk, ez megszabja azt is, hogy lehetőleg a függvényünk se legyen ilyen bonyolult, mivel akkor semmivel sem jutottunk előbbre, mintha az iterációs folyamatot választottuk volna.

A korábbi kutatások során egy négy tagból álló neurális hálózat sorozatot alkalmaztak, ahol látszik, hogy a hibák nagysága jelentősen csökkent az első és a negyedik hálózat után. Jellemző adatai e kétféle hálózatnak a szórása és a hiba maximális értéke, mint láthatjuk mind a két adat jelentősen lecsökkent a 4. fokú hálózatnál:

	σ (szórás)	Maximális hiba
1. fok	9.77 cm	58.54 cm
4. fok	6.58 cm	36.23 cm

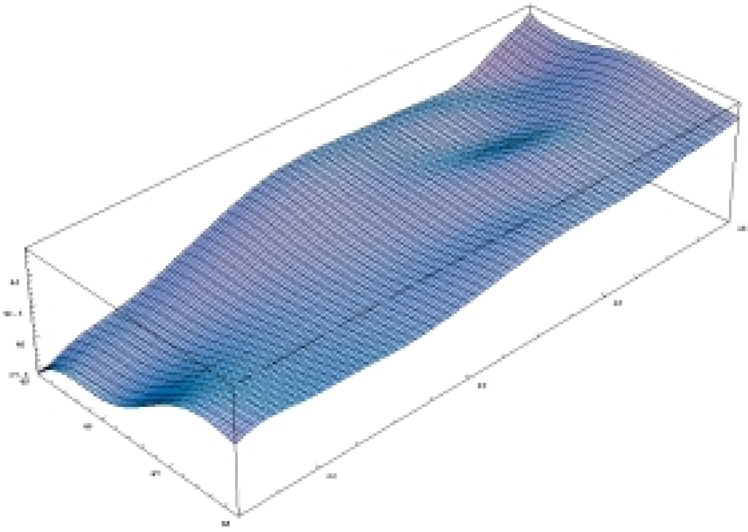
Ha azonban figyelembe vesszük, hogy maga az adatbázis ±3-4 cm pontosságú, akkor nekünk is ezen a pontosságon belül kellene maradnunk, és szükséges még tovább csökkenteni a középhibákat, a szórást. Ez az egyik megoldandó feladat, amivel én tovább próbálkoztam, illetve teszteltem az elkészült hálózatokat a összes (211680) pontra, és megvizsgáltam, hogy ez a kiválasztott 8000 pont megfelelően reprezentálja-e az eredeti kétszázezres adathalmazt.

5. A tesztelés eredménye

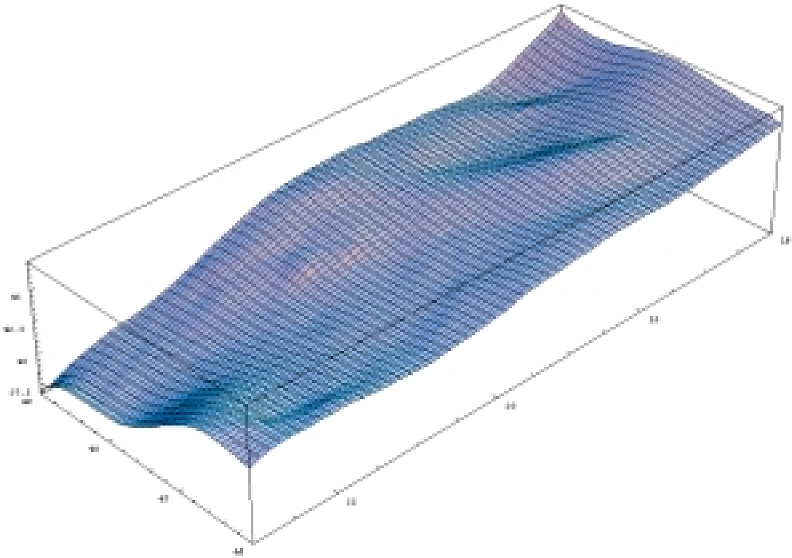
A hálózat pontosságát mutató adatok közül a legfontosabbak: a maradék hibák szórása, maximális értékei (pozitív, negatív maximumok), és a várható értéke. Ezeknek az alakulását ismertetem a különböző tesztelések során.

Az eredeti 211680 pontból a tanításhoz összesen 8484 pont került felhasználásra. A következőkben meg fogom nézni, hogyan alakultak a fent említett értékek az elsőrendű neurális hálózat tanítása után, ill. a négy tagból álló neurális hálózat sorozat alkalmazása után. Az előbbi eredményeként kapott függvényt f1-nek neveztem el, az utóbbit F-nek (F=f1+f2+f3+f4). Továbbá a későbbiekben alkalmazható javításokhoz ábrázolom a fenti függvényeket, illetve a hibákat.

Ha az elsőrendű ill. a negyedrendű neurális hálózatok leképzési függvényét ábrázoljuk a tanítópontokra, akkor a következő felületeket kapjuk:



2. ábra A geoidfelület közelítése 1. rendű neurális hálózattal



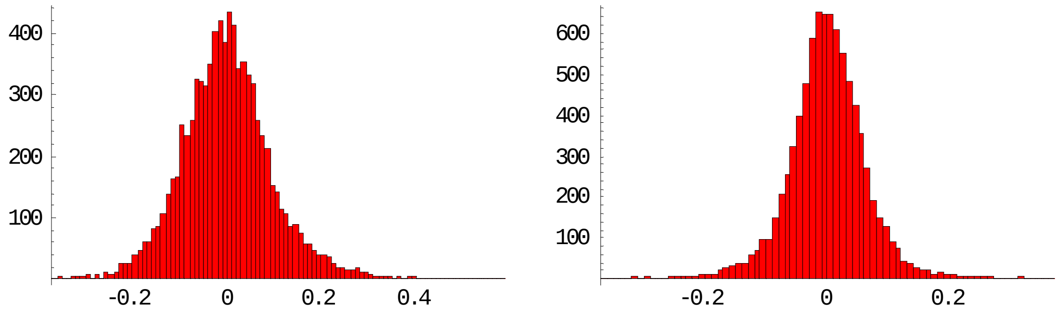
3. ábra A geoidfelület közelítése 4. rendű neurális hálózattal

A 2., 3. ábrából is látszik, hogy a negyedrendű neurális hálózattal egy sokkal bonyolultabb felületet lehet ábrázolni, ami sokkal közelebb áll a tényleges geoid felületéhez.

Ha elvégezzük a **teszteléš**t a **8484 tanítópontra**, akkor a következő eredményeket kapjuk:

	σ (szórás)	*Pozitív hiba maximum*	*Negatív hiba maximum*	*A hiba várható értéke*
1. fok	9.77 cm	58.54 cm	-36.66 cm	0.00 cm
4. fok	6.58 cm	36.23 cm	-36.74 cm	0,00 cm

Az eltérések hisztogramjai a következőek:



4. ábra 1. és 4. rendű hálózatok hibáinak hisztogramja

A **tesztelés**t elvégezve a **211680 pontra** a következő eredmények adódnak:

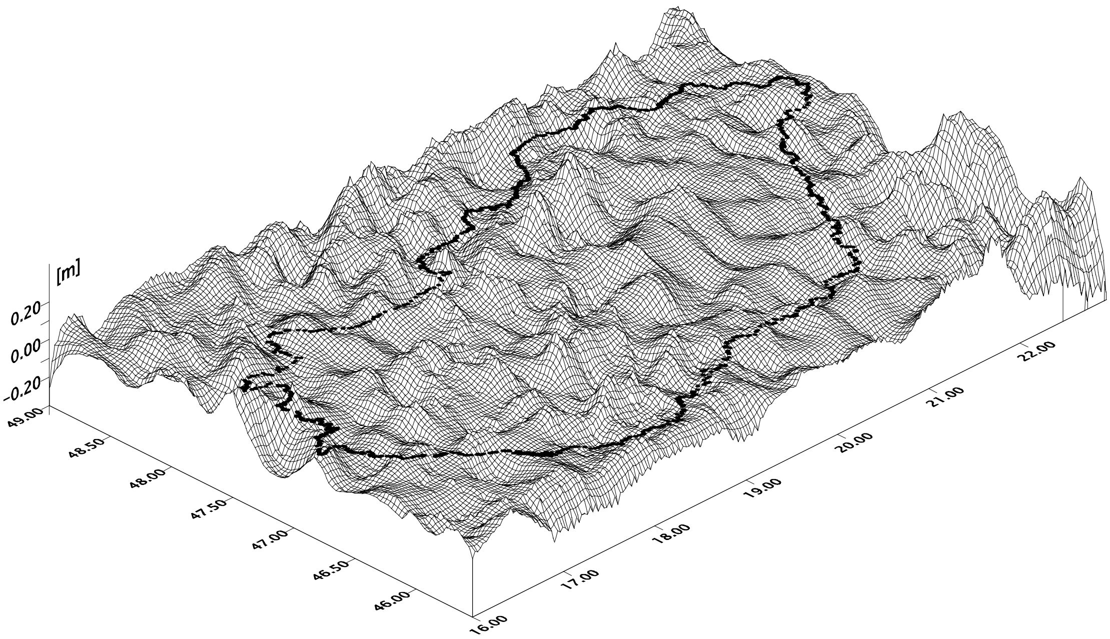
	σ (szórás)	*Pozitív hiba maximum*	*Negatív hiba maximum*	*A hiba várható értéke*
1. fok	9.87 cm	60.03 cm	-41.61 cm	0.01 cm
4. fok	6.76 cm	43.33 cm	-50.62 cm	0,02 cm

A két adatsor összehasonlításából látszik, hogy a szórás közel változatlan maradt a 8000 és a 200000 pontra végzett teszteléskor is. A várható érték pedig mindkét esetben nullának tekinthető. Tehát úgy tűnik, hogy a kiválasztott 8000 pontos mintaállomány megfelelően reprezentálja az összes adatot. Persze a maximális értékek nagyobbak lettek a teljes állományra végzett teszteléskor, de magának az eloszlásnak, a haranggörbének az alakja változatlan maradt.

Persze ezek az eredmények még igen távol állnak pontosságban a kívánatos 3-4 cm-es értéktől. Ezért ki kellett találni valamit, amivel tovább lehetne csökkenteni a közelítés hibáját.

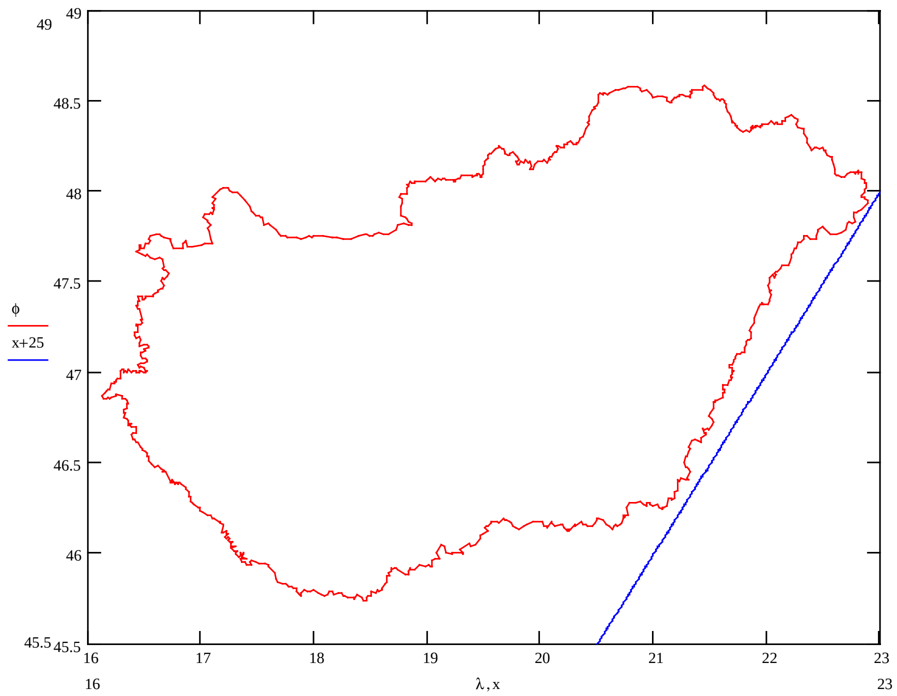
Először felmerül, hogy el kellene végezni a tanítást egy nagyobb adathalmazra, nem csak 8000 pontra. Így a javítás érdekében először ezzel próbálkoztam. Nem minden 25., hanem minden 4. pontot vettem bele a tanítópontok közé, azaz összesen 52920 pontot. Azonban ez a módszer nem vezetett eredményre a futási idők nagyságrendekkel történő megnövekedése miatt. A 200 iterációt a számítógép a 8000 pontra 3 óra alatt hajtotta végre, ugyanez a 200 iteráció az 50000 pontra már legalább egy hétig tartott volna, és persze ez valószínűleg nem is lett volna elég, hanem jóval több iterációt kellett volna alkalmazni. Így kénytelen voltam ezt a kutatási irányt abbahagyni.

Egy másik javítási lehetőség kínálkozik, ha megfigyeljük a nagy hibák eloszlását.



5. ábra A 4. rendű neurális hálózattal közelített felület eltérései a geoidfelülettől

Az 5. ábrán észrevehetjük, hogy a legnagyobb hibák, nem is Magyarország területére esnek, hanem a romániai részen találhatóak. Ezt a részt egy egyenessel le lehet vágni. MathCAD program segítségével meghatároztam annak az egyenesnek az egyenletét, amely a levágandó adatokra vonatkozó feltételt reprezentálja.



6. ábra A romániai nagy eltérésű területet levágó egyenes

Ennek az egyenesnek az egyenlete igen egyszerű lett: $\varphi = \lambda + 25$. Ezután a Mathematica programmal kiválogattam azokat az adatokat, amelyek e fölött az egyenes felett találhatóak, és a továbbiakban csak ezekkel dolgoztam.

Kiválogatás után a 8484 pontos állományból 7438 pont maradt, a 211680 pontos állományból pedig 184910. Ez után elvégeztem ezekre az állományokra is a tesztelést, hogy megvizsgáljam, mekkora mértékben javulnak ettől az eredmények, illetve, hogy egyáltalán javulnak-e.

Most a tesztelést már csak az F függvénnyel (a negyedrendű hálózattal) végeztem el. Összehasonlításképpen leírom a korábbi, levágás előtti hálózatok eredményeit is.

Tesztelés 8484 ill. 7438 pontra

		σ (szórás)	*Pozitív hiba maximum*	*Negatív hiba maximum*	*A hiba várható értéke*
8484 pont	*4. fok*	6.58 cm	36.23 cm	-36.74 cm	0,00 cm
7438 pont	*4. fok*	5.63 cm	26.80 cm	-25.67 cm	0,01 cm

Tesztelés 211680 ill. 184910 pontra

		σ (szórás)	*Pozitív hiba maximum*	*Negatív hiba maximum*	*A hiba várható értéke*
211680 pont	*4. fok*	6.76 cm	43.33 cm	-50.62 cm	0,02 cm
184910 pont	*4. fok*	5.68 cm	29.14 cm	-35.13 cm	0,00 cm

Ha a fenti eredményeket összevetjük a korábbi 4. fokú eredményekkel, akkor láthatjuk, hogy tényleg igen jelentős mértékben javultak. A hibák maximális értékei például 10-15 cm-rel lettek jobbak, és a sok nagyon nagy hibájú adat levágásának eredményeképpen a szórás is kb. 1 cm-rel kevesebb lett.

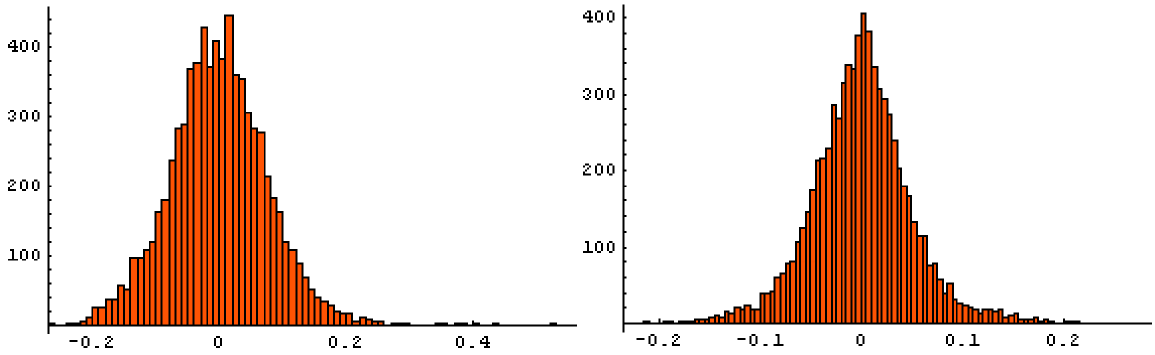
6. Új neurális hálózat

A korábbi vizsgálatokból megállapítható tehát, hogy a legnagyobb hibák nem is Magyarország területére esnek, hanem Romániában találhatóak. Érdemes lenne készíteni egy új neurális hálózatot, annak az adatállománynak a felhasználásával, amely nem tartalmazza az ezeken a területeken levő adatokat, és így valószínűleg jobb eredményeket kaphatnánk.

Hogy ezt a feltevést kipróbáljam újra elvégeztem a 4 neurális hálózat tanítását, ezekkel a leválogatott adatokkal.

Az elkészült Mathematica program megtalálható a mellékletben, és ebben szerepelnek az új f1, f2, f3, f4 függvények is, aminek a meghatározása a tulajdonképpeni cél jelen esetben.

Az új neurális hálózatok hisztogramjai a következőek:



7. ábra Az új 1. és 4. rendű hálózatok hibáinak hisztogramja

Az **új hálózat** eredményei a következőek lettek, a 7438 tanítópontra vonatkoztatva:

	σ (szórás)	*Pozitív hiba maximum*	*Negatív hiba maximum*	*A hiba várható értéke*
1. fok	7.96 cm	55.12 cm	-26.21 cm	0.00 cm
4. fok	4.97 cm	28.37 cm	-23.36 cm	0,00 cm

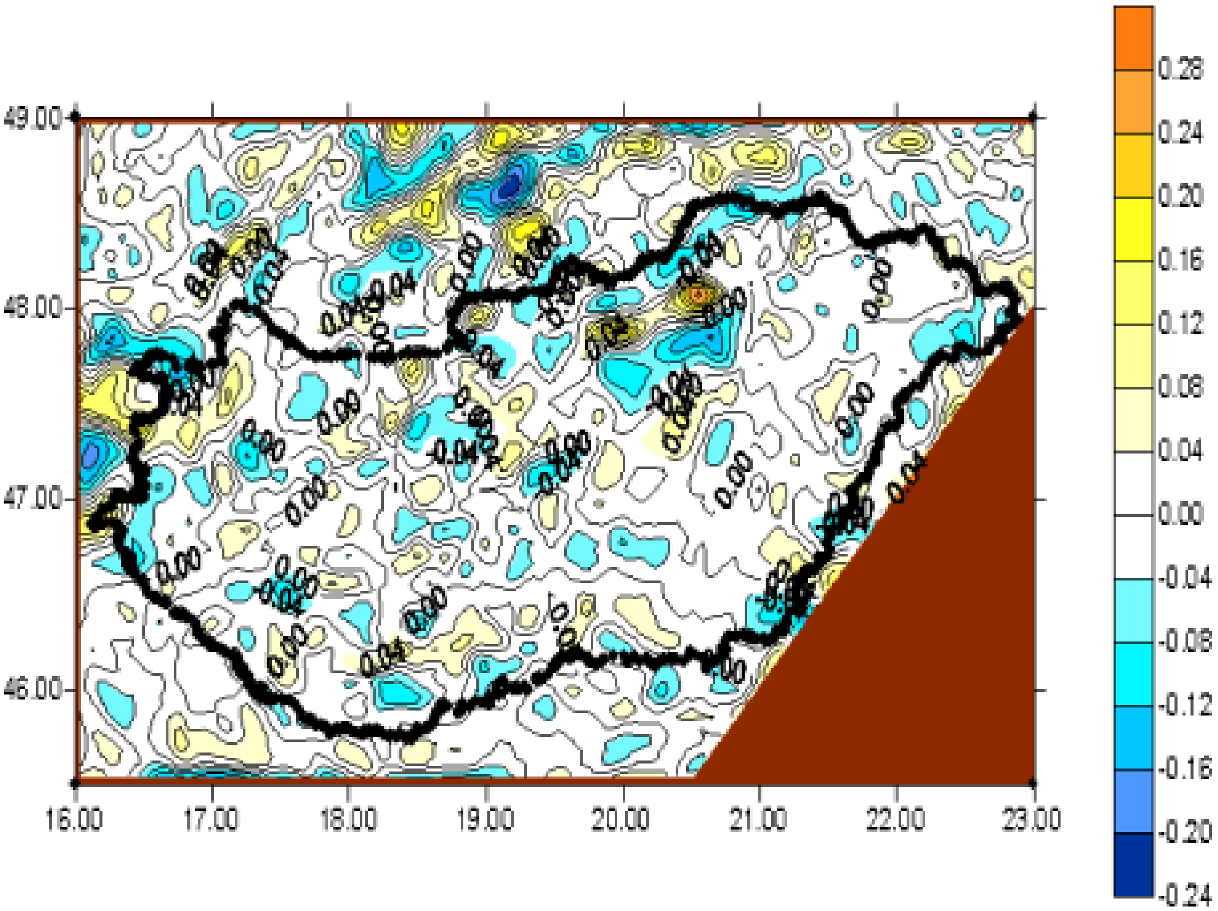
A könnyebb összehasonlításhoz összefoglalom a **korábbi** 8484 tanítópontra készített hálózat 7438 pontra tesztelt eredményeit.

	σ (szórás)	*Pozitív hiba maximum*	*Negatív hiba maximum*	*A hiba várható értéke*
4. fok	5.63 cm	26.80 cm	-25.67 cm	0,01 cm

Itt is némi javulás figyelhető meg a szórásban a korábbi hálózathoz képest (5.63 cm-ről lecsökkentek 4.97 cm-re), a hibák maximális értékei viszont maradtak körülbelül ugyanazok. Ez a jelenség a hálózat robosztusságára utal, azaz néhány „rossz” adat a tanulóhalmazban csak csekély mértékben befolyásolja a tanulás minőségét.

7. Eredmények, további lehetőségek

Surfer program segítségével ábrázoltam az eltéréseket, az eredeti (HGTUB2000) geoidmodell adataiból levonva az új neurális hálózattal kapott eredményeket.



8. ábra Eltérések az eredeti (HGTUB2000) geoid magasságok és a neurális hálózattal közelített magasságok között

A 8. ábrán már csak a leválogatott adatokat ábrázoltam, kihagyva az általam levágott romániai részt. Ha megnézzük ezt az ábrát, látható, hogy a nagy hibák egy jó része ismét csak az országhatáron kívülre esik, illetve az is feltűnő, hogy nagyobb hibákat elsősorban a hegyvidékeken találunk, Magyarországon a Mátra, Bükk és az Alpokalja területén.

Az elérni kívánt pontosság 3-4 cm volt. Az ábrán fehérrel jelöltem a 4 cm alatti eltéréseket. Ugyan ezt nem sikerült mindenütt elérni, de az ország jelentős részén már igen. Ebből az a következtetés vonható le, hogy érdemes még ezzel a témával a későbbiekben is foglalkozni, és tovább pontosítani a közelítést, mivel az eddigi eredmények bíztatóak.

A fentiek alapján igen valószínű, hogy a magyarországi hegyvidéki területeken a tanítópontok sűrítése javíthat az eredményeken. Érdemes lenne kipróbálni ezt egy új neurális hálózat elkészítésével.

Hivatkozások

1. B. Paláncz-L. Völgyesi (2002): High accuracy data representation via sequence of neural networks, megjelenés alatt
2. Barsi Árpád (1999): Koordináta transzformáció megoldása neurális hálózatokkal, Geodézia és Kartográfia, Budapest, LI, No. 10. 12-18.
3. Veres Gábor (2002): RBF neurális hálózat alkalmazása magasság meghatározására, Geodézia és Kartográfia, Budapest LIV. No. 7. 25-30.
4. Horváth Gábor (1995): Neurális hálózatok és műszaki alkalmazásaik, Műegyetemi kiadó, Budapest
5. Tóth Gy, Rózsa Sz (2000): New Datasets and Techniques – an Improvement in the Hungarian Geoid Solution. Paper presented at Gravity, Geoid and Geodynamics Conference, Banff, Alberta, Canada July 31-Aug 4, 2000.
6. Tóth Gy, Rózsa Sz, Andritsanos V D, Ádám J, Tziavos I N (2000): Towards a cm-Geoid for Hungary: Recent Efforts and Results. Phys. Chem. Earth (A), Vol. 25. No. 1. pp. 47-52.