



# RBF neurális hálózat alkalmazása magasság meghatározására<sup>1</sup>

Veres Gábor, a Budapesti Műszaki és Gazdaságtudományi Egyetem  
Általános- és Felsőgeodéziai Tanszék doktorandusza  
(E-mail: tsoka@sc.bme.hu)

## Bevezetés

A mesterséges neurális hálózatok (Artificial Neural Networks – ANNs) alkalmazásai napjainkban egyre szélesebb körben jelentkeznek. A mesterséges neurális hálózatok ötlete, működési elve az idegsejtekből építkező, biológiai neurális hálózatokból származtatható. Az első neuron modellt 1943-ban publikálta *W. S. McCullogh* és *W. Pitts*. A róluk elnevezett egyszerű neuron a beérkező információkat súlyozva összegzi, és ezt az értéket az aktivációs függvényvel vizsgálja. Az utóbbi években jelentős előrelépések történtek a mesterséges intelligencia – pl. alakfelismerés, hangfelismerés – megvalósításában. A „neurális hálózatok“ inkább egyfajta modell családot jelentenek, mint konkrét eljárást. A mesterséges neurális hálózatoknak egy nagyon fontos tulajdonsága az, hogy adaptívak, a problémát nem „programozással”, hanem a példákából, tanulással oldják meg. [4] Az adatokból nyert információkból képeznek eredményt, melynek előnye, hogy a megoldáshoz szükséges köztes információk száma csökkenthető. Ennek megfelelően neurális hálózatot akkor célszerű alkalmazni, ha a probléma explicit módon nem írható le; ha valamilyen (sejtett) kapcsolat áll fenn a bemenő (input:  $\mathbf{x}$ ) adatok és a kimenő (output:  $\mathbf{y}$ ) adatok között:  $\mathbf{y}=f(\mathbf{x})$ , de az  $f$  függvény ismeretlen. A megoldás itt nem az  $f$  függvény lesz, hanem egy olyan hálózat, amely hasonló kimenő adatokat szolgál, mint az  $f$  függvény, ugyanolyan bemenő adatok mellett. A [2] szerint a neurális hálózatok három legfontosabb tulajdonsága:

- a neuronok rendezett topológiájú, összekapcsolt rendszeréből áll;
- rendelkezik tanulási algoritmussal;
- rendelkezik a megtanult információ felhasználásával

nálását lehetővé tevő információ előhívási algoritmussal.

A rendezett topológiában a rétegekbe szervezett neuronok között – akár a rétegen belül is – kapcsolatok vannak. Ezen kapcsolatok segítségével adódnak át az értékek a neuronok között.

A neurális hálózatok alkalmazása a térinformatikában nem újdonság hazánkban sem, példa rá *Barsi Á.* cikke [1], és az, hogy kisebb modulokként megjelentek a térképészeti szoftverekben is (pl. Surfer 7). Feladatuk szerint a neurális hálózatok egyik nagy csoportját képezik azon hálózatok, melyekkel adatok közötti interpolációt hajthatunk végre.

A cikkben egy választott hálózattípuson és egy egyszerű geodéziai példán keresztül megpróbálok bemutatni egy interpolációt végző neurális hálózat előnyeit, hátrányait. A feladatot lényegében skalár terek interpolációjára lehet visszavezetni, ahol a skalárokat a  $\mathbf{z}$  koordináták jelentik. Két behatárolható területen ismert  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$  koordinátájú pontokból kell meghatározniuk tetszőleges vagy ismert  $\mathbf{x}$ ,  $\mathbf{y}$  koordinátákhoz tartozó  $\mathbf{z}$  értékeket. A meghatározandó  $\mathbf{z}$  értékek a behatárolt területen (szelvényben) helyezkednek el.

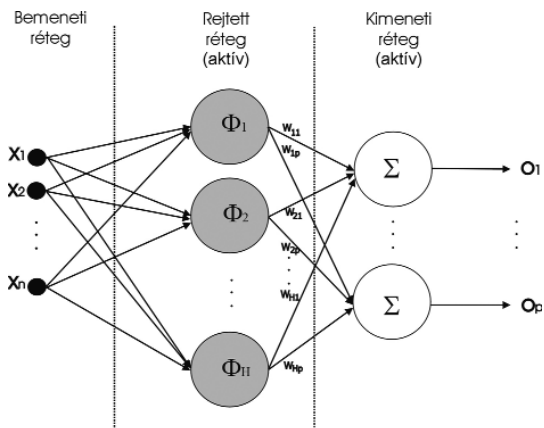
Kulcsszavak: RBF, mesterséges neurális hálózatok, interpoláció.

## Az RBF hálózat felépítése

A radiális bázisfüggvény hálózatoknak (Radial Basis Function networks) egy bemeneti, egy rejtett és egy kimeneti rétege van. Az RBF hálózatok két aktív réteget tartalmaznak, a bemeneti réteget nem szoktuk aktívnek nevezni. Az első aktív rétegbeli aktiváló függvények ( $\Phi_k$ ) általában kör-szimmetrikus függvények, innen kapták nevüket. Egy hálózatban általában egyfajta kör-szimmetrikus függvényt használnak, de azok paraméterei neurononként változhatnak. A második aktív rétegbeli elemek a rájuk jutó értékek lineáris kombi-

<sup>1</sup> A cikk kapcsolódik az OTKA T 030643 számú projekthez

nációból képezik a kimeneti értékeket (1). A bemeneti réteg neuronjainak száma adja meg a bemeneti halmaz dimenzióját, a kimeneti réteg neuronjainak száma a kimeneti halmaz dimenzióját. Általános RBF hálózat felépítését az 1. ábra mutatja.



1. ábra. Általános RBF hálózat

Egy általános RBF hálózat esetén a kimeneti értékek a következőképpen alakulnak:

$$o_i(\mathbf{x}) = \sum_{k=1}^H w_{ik} \Phi_k(\mathbf{x}) \quad i = 1, \dots, p \quad (1)$$

$$\Phi_k(\mathbf{x}) = \Phi(\|\mathbf{x} - \mathbf{x}_k\|) \quad k = 1, \dots, H \quad (2)$$

ahol a  $\Phi_k(\mathbf{x})$  az aktivációs függvény;  $\mathbf{x}_k$  pedig az aktivációs függvény középpont paramétere. A  $\Phi_k(\mathbf{x})$  függvényként alkalmazható minden körszimmetrikus nemlineáris függvény, amelyben a változó a fent látható távolság függvény – az euklideszi távolság:  $\|\mathbf{x} - \mathbf{x}_k\|$ . A neurális hálózatokban lehetőség van további távolság definíciók használatára is. Az RBF hálózatokban elterjedt körszimmetrikus függvény például a multikvadratikus (3) vagy az inverz multikvadratikus (4) függvény, azonban legtöbbször a Gauss függvényeket (5) alkalmazzák [2].

$$\Phi_k(\mathbf{x}) = \sqrt{\mathbf{x}_k^2 + \|\mathbf{x} - \mathbf{x}_k\|^2} \quad (3)$$

$$\Phi_k(\mathbf{x}) = \frac{1}{(\mathbf{x}_k^2 + \|\mathbf{x} - \mathbf{x}_k\|^2)^2} \quad (4)$$

$$\Phi_k(\mathbf{x}) = \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}_k\|^2}{2\sigma_k^2}\right\} \quad (5)$$

A Gauss függvények esetén egy új meghatározandó paraméter is jelentkezik, a szélességparaméter ( $\sigma_k$ ). A szélességparaméterek megválasztására Gauss függvények esetén az interpoláció nem érzékeny, ebből adódóan esetleg rejtett rétegbeli neuronnak ugyanazt a szélességparamétert is adhatjuk [2]. Az RBF hálózat kialakításában alapvető fontosságú az aktiváló függvények számának (a rejtett rétegbeli processzáló elemek) és azok helyzetének megválasztása. A rejtett rétegtől elvárjuk, hogy „helyileg“, azaz egy lehatárolt bemeneti tartományra legyen aktív [3]:

$$\Phi \rightarrow 0 \text{ ha } \|\mathbf{x}\| \rightarrow \infty \quad (6)$$

Az aktiváló függvények helyzetét és szélességét legtöbbször egy klaszterező eljárással határozzuk meg. Itt használhatunk algoritmikus eljárást (például: K-means) vagy akár nem ellenőrzött tanulási eljárásokat (például: versengő tanulás).

A K-means algoritmus olyan klaszter-középpontokat határoz meg, ahol a tanító pontok és a hozzájuk legközelebb eső klaszter-középpontok négyzetes távolságainak összege minimális. Véletlenszerűen meghatározunk K klaszter-középpontot, majd az egyes tanítópontokat besoroljuk a hozzájuk legközelebb eső klaszterbe. Ezek után az egyes klaszterekbe sorolt elemek átlagából új klaszter-középpontokat képezünk, és újra meghatározzuk az elemek klaszterbe tartozását. Mindezt addig folytatjuk, amíg az elemek klaszterbe sorolása nem változik.

A nem ellenőrzött tanítást a neurális hálózatokban széleskörűen alkalmazzák. Ilyen eljárás a versengő tanulás. A nem ellenőrzött tanítás során nem állnak rendelkezésünkre az adott bemenethez tartozó kívánt válaszok [2]. A versengő tanulás során a klaszterekbe sorolás egy súlymátrix meghatározását jelenti. Az első lépésben a véletlenszerűen felvett súlymátrixra kiszámítjuk az egyes kimeneti értékeket. A kimeneti értékek alapján kiválasztjuk a legnagyobb értékkel szolgáló elemet, a győztest. A győztes elem kiválasztása után hajtjuk végre a súlymódosítást. A súlymódosítás csak a győztes értéket meghatározó súlyokat érinti (winner-takes-all), és a bemeneti érték felé módosít:

$$\Delta \mathbf{S}_{k,l}(t) = h(\mathbf{x}_l - \mathbf{S}_{k,l}), \quad (7)$$

ahol  $\mathbf{S}_{k,l}$  a súlymátrix;  $\mathbf{x}_l$  a bemeneti vektor;  $t$  az iteráció száma;  $h$  alkalmasan választott „bátorsági“ tényező, értéke 0 és 1 között van;  $k^*$  a győztes e-

lem indexe. A kezdeti súlymátrix és a szélességi paraméter megválasztásában alkalmazhatunk feltételeket is, pl. szórás és várható érték megadása. A szélességi paraméterek is változhatnak az iteráció során, de használhatunk végig állandó paramétereket is. Kellően nagy számú iteráció után a súlyértékek már csak nagyon kicsit változnak, és egy előre megadott súlymódosító értéket vagy iteráció számot elérve az eljárás befejeződik. A versengő tanulás eljárásai a győztes kiválasztásban különböznek, de mindig az aktuális iteráció kimeneti értékeiből kerülnek meghatározásra. Legegyszerűbb esetben kiválasztjuk a legnagyobb értéket:

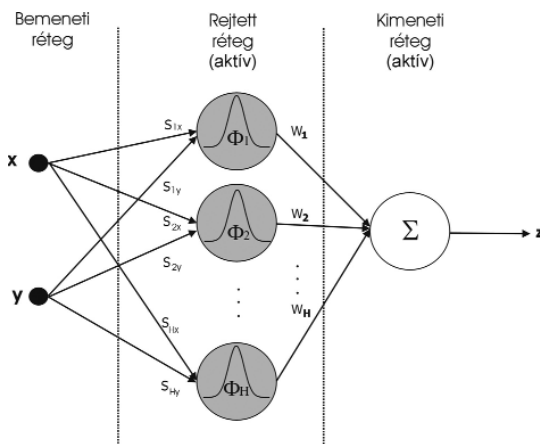
$$k^* = \max(b_k), \quad (8)$$

ahol  $b_k$  az egyes elemekhez tartozó kimeneti érték.

A teljes hálózat működése két vagy három fő lépésre bontható. Jelen alkalmazásban kétlépcsős működést használunk. Az első lépés a hálózat tanítása, vagyis az optimális – vagy annak vélt – súlymátrix megtalálása, a második a konkrét hálózat „használata”, mely után megkapjuk a keresett értékeket. A tanítási fázis nagyságrendekkel számításigényesebb, mint a hálózat alkalmazása.

## Az alkalmazott hálózat

Az alkalmazott hálózat bemeneti rétegében két neuron található, mivel a bemeneti halmaz dimenziója: 2, a kimeneti réteg neuronjainak száma: 1. Az alkalmazott RBF hálózat felépítését a 2. ábra mutatja.



2. ábra. Az alkalmazott RBF hálózat

A 2. ábrán a bemenő  $x, y$  vektorok (az 1. ábrán  $x_1$  és  $x_2$ ) az adatállomány  $x, y$  koordinátáit reprezentálják, a kimenő  $z$  vektor (az 1. ábrán  $o_1$ ) a  $z$  koordinátákat. A bemenő koordináták középpont-

tokhoz való rendelése és a középpontok szélesség-paraméterének meghatározása nem ellenőrzött tanítási módszerrel (versengő tanulás) történt. A maximális iteráció száma 1000 volt, a súlymódosítás minimuma pedig 0,001. A szélesség-paraméterek állandóak (valamennyi neuronra 0,707), a kezdeti súlyokat véletlenszerűen vettem fel. A felügyelt tanítást a második, kimenő réteg végzi. A súlymódosítási módszer (Momentum) kiválasztása után ez két paraméter meghatározását igényli: a tanulási aránytényezőt és a momentum együtthatót. A súlymódosítás a következő összefüggéssel írható le:

$$\mathbf{w}(k+1) = \mu[-\nabla(k)] + \eta \mathbf{w}(k) \quad (9)$$

ahol  $\mathbf{w}(k)$  a  $k$ -adik iteráció súlymátrixa;  $\mu$  a tanulási aránytényező;  $\eta$  a momentum együttható, amelynek értéke 0 és 1 között van;  $\nabla(k)$  a gradiens módszerrel meghatározott súlymódosító érték. A gradiens módszer során a hibafelület minimumát keressük. A szükséges paraméterek megválasztása ( $\mu$ : 2,0 ;  $\eta$ : 0,9) után azok állandóak maradtak valamennyi később alkalmazott hálózat esetén. A  $\mu$  tanulási aránytényező megválasztása fontos pontja a hálózat kialakításának. Szélső esetekben: nagy érték esetén a konvergenciát kockáztatjuk; kis érték esetén a hálózat lassú és számításigényes lesz. A paraméterek kiválasztása a mintater segítségével, tapasztalati úton történt.

## A mintaterek

Az első mintater kiválasztásánál az adathalmaz „kellő” változékonysága volt az elsődleges szempont. Az egyes interpolációs eljárások használhatósága nyilván jobban megmutatkozik egy változékony felszín esetén. Ezért a Sziklás-hegység egy kis szelvényéből indultam ki. A szelvény 31x31, azaz 961 pontot tartalmazott, a pontok egymástól egy 30 m-es négyzetrács rácspontjain helyezkednek el. A legalacsonyabb magasságú pont 2767, a legmagasabb 2922 méter volt, az értékek (torzítatlan) szórása: 38,1. A mintatérből véletlenszerűen kiválasztott 81, 181, 281, 381, 481, 580, 680, 780, 880 pontokat kivéve a halmazból kaptam a 880, 780, 680, 580, 480, 381, 281, 181, 81 tanítópontot tartalmazó halmazokat. A 880 tanítópontot tartalmazó hálózatokhoz a kiválasztott 81 pontot tesztpontként használjuk fel, ugyanígy a 780 tanítópontot tartalmazó hálózatokhoz a kiválasztott 181 pont lesz a teszt alapja.

Az  $x$  és  $y$  koordináták esetén a valós értékeket a halmazban előforduló minimum értékkel csökkenttem, tehát a mintateret a síkban eltoltam.

Ennek célja a felesleges számítások elkerülése volt. A hat és hétjegyű síkkoordináták csak a számítási kapacitást növelik, jelen esetben érdemi jelentésük nincs. Nyilván a keresett érték, a magasság tekintetében már nem tehetjük meg, hogy kiválasztjuk a magasság „értékes és optimális” intervallumát. Alkalmazhatunk durva becslést, azonban az interpoláció „finomítása”, az értékes régió kihasználása ellentétben van azzal, hogy a lehető legkevesebb feltevést szeretnénk tenni a feladat megoldása során. A „z” értékek határok közé szorításával elveszthetjük a hálózat azon tulajdonságát, hogy esetleg a tanítópontok között nem található minimumot vagy maximumot elérje hálózatunk, ami a valóságban is előfordulhat. Az RBF hálózatok alkalmazása esetén célszerű mindig az értékek valamilyen normalizált alakjával (0 és 1 vagy -1 és 1 közé eső számokkal) dolgozni. Ezzel tovább csökkenthető a számítási igény, és az euklideszi távolság definíciójából levezethető hibákat is kiküszöbölhetjük.

## Eredmények

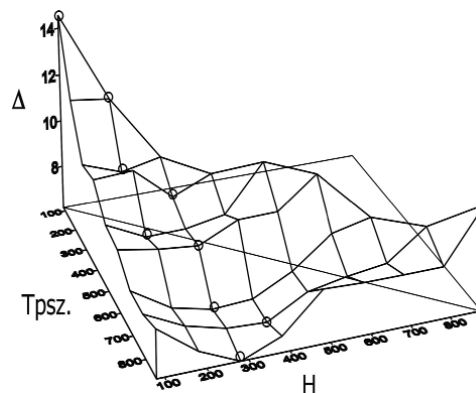
A hálózat használata során a számítási igények tekintetében beigazolódott, hogy a rejtett rétegben található elemek számának növelése alapvetően megnöveli a számítási igényt, míg a 3. ábrán feltüntetett optimum „határt” (a 3. ábrán kis körökkel jelöltem) túllépve a tesztpontok alapján mért eltérések értéke érdemben nem csökken, sőt romlik [2].

A valós és a tesztpontokra kapott „z” értékek eltérését a  $\Delta$  értékkel jellemezve:

$$\Delta = \sqrt{\frac{\sum_{i=L..n} (z_i - z_i^{RBF})^2}{n}} \quad (10)$$

ahol  $n$  a tesztpontok száma az egyes hálózatok esetén, így a 3. ábrán látható hibafelületet kapjuk.

A hibákat ( $\Delta$ ) a tanítópontok számának (Tpsz.) és a rejtett rétegbeli neuronok számának (H) függvényében ábrázoltam. Tanítópont-halmazonként kis körrel jelöltem a legkedvezőbb  $\Delta$  értéket adó hálózatot. Az ábrából azt az alapvető trendet figyelhetjük meg, hogy minél több tanítópontunk van, annál kisebb lesz a rejtett rétegbeli neuron-számtól függő legkedvezőbb  $\Delta$  érték. Érdemes megfigyelni azonban, hogy ez nem mindig igaz (pl. a 680 és a 780 tanítópont-halmazhoz tartozó legkedvezőbb  $\Delta$  értékek: 6,26; 6,53), itt kevesebb adat-



3. ábra. Hibafelület az első mintaterre

ból kiindulva jobb eredményhez jutunk! Ez azzal magyarázható, hogy az egyik bemenő adathalmaz jobban klaszterizálható. Szintén a klaszterizálással függ össze egy tanítópont-halmaz rejtett rétegbeli neuron-számtól függő, felvett  $\Delta$  értékek jelleggörbéje. Az ábrán a 880, 780, 680, 580 tanítópont-halmazhoz tartozó jelleggörbékből hasonlóság figyelhető meg.

A 3. ábrán, ahol a tanítópontok száma megegyezik a rejtett rétegben található neuronok számával, valójában egy lineáris egyenletrendszer megoldásait keressük, ahol az egyenletek száma megegyezik az ismeretlenek számával. Ez a megoldás főleg kevés tanítópont esetén indokolt. Nagy számú tanítópont esetén, ha minden tanítópont egyben egy függvényközpont, akkor a számítási igényt óriásira növelhetjük, tulajdonképpen feleslegesen, ahogy azt nemsokára látjuk. Ilyenkor a hálózat tanítópontok között végez interpolációt, úgy, hogy a tanítópontokban pontosan előállítja a betanított értéket [2].

A rejtett rétegbeli neuronok számának megválasztására bizonyos becsléseket tehetünk, azonban általában próbalgatással, tapasztalati úton határozzák meg a feladatra optimális értéket. A rejtett rétegbeli neuronok számának növelésével – egyéb hálózati paraméter változatlanul hagyásával – a tanítás számítási igénye gép időben mérve jelentősen nőtt, míg a  $\Delta$  érték másfélszeresére emelkedett. A 880, 780 tanítópontú hálózatokra a számítási igényt (t) és a rejtett rétegbeli neuronok növelésével elért  $\Delta$  értékeket az 1. táblázat mutatja. (A számítási igény a következő konfiguráció mellett értendő, gépidőben: PIII, 900MHz, 128 Mb RAM)

A 1. táblázatból megfigyelhetjük, hogy a legkedvezőbb  $\Delta$  értéket szolgáltató rejtett rétegbeli neuronok száma mellett a legrosszabbhoz képest a

Tanítópontok száma	Rejtett rétegbeli neuronok száma (H)	t [m:s]	$\Delta$
880	100	3:02	8,12
880	200	4:30	6,64
880	300	6:15	6,05
880	400	8:18	7,05
880	500	10:26	8,77
880	600	13:15	8,27
880	700	15:46	8,40
880	800	17:30	8,31
<b>880</b>	<b>880</b>	<b>19:24</b>	<b>10,54</b>
780	100	2:43	7,65
780	200	4:08	6,89
780	300	5:35	6,62
780	400	7:27	6,53
780	500	9:10	7,55
780	600	10:52	7,61
780	700	12:46	7,91
<b>780</b>	<b>780</b>	<b>14:19</b>	<b>9,11</b>

1. táblázat

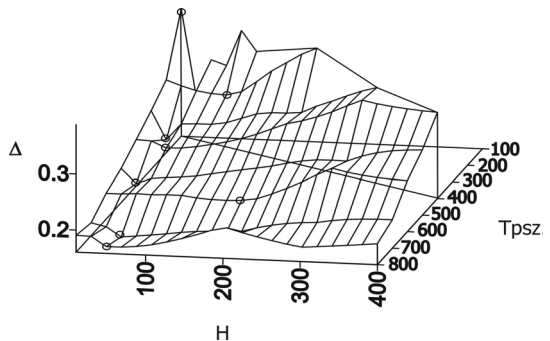
számítási igény mintegy háromszorosára, illetve kétszeresére változik. Hozzá kell tenni, hogy a hálózatok tanítása után a tesztek lefutása (a hálózat alkalmazása) mindig 1 másodperc alatt volt!

Felmerül a kérdés, hogy mennyire „jók” az 1. táblázatban található  $\Delta$  értékek. A kapott eredmény nagyságrendileg nem jobb, mint egyes, már ismert interpolációs eljárások, azonban mindenképpen „életképesek”. A feladatra például krigeléssel kapott eredményeket megvizsgálva tapasztaljuk, hogy egyes esetekben a neurális hálózatokkal kapott értékek kicsivel kedvezőbbek, azonban ez nem általános, sőt szélső esetben a krigeléssel kapott  $\Delta$  érték akár a fele is lehet az RBF hálózat eredményének. Az RBF hálózatra kapott értékek a nagy ismeretlenségű feladat esetén alakulnak inkább kedvezőbben. Míg például a krigelésnél egyértelműen megfogalmazható az eljárás [5], ugyanolyan bemenetnél, ugyanolyan lesz a kimenet, addig a neurális hálózatoknál ez legtöbbször nem igaz, a véletlenszerű kiindulás miatt.

A hálózat alkalmazása a második mintatérre (síknak tekinthető területen) egy másik tulajdonságra hívja fel figyelmünket. A szelvény 30x30, azaz 900 pontot tartalmazott, a pontok egymástól egy 30 m-es négyzetrács rácspontjain helyezked-

nek el. A síknak tekinthető mintatér z értékeinek (torzítatlan) szórása 1,77 volt. Az így vizsgált RBF hálózatok azt mutatták, hogy a tanítópontoknak az egyes középpontokhoz való rendelése jóval gyorsabb és hatékonyabb. A sík mintatérre alkalmazott hálózatok a megadott hibahatárt jóval hamarabb érték el. A 3. ábra analógiájára a sík területre a tanítópontok számának (Tpsz.) és a rejtett rétegbeli neuronok számának (H) függvényében ábrázolt  $\Delta$  értékeket a 4. ábra mutatja. Tanítópont csoportonként a legkedvezőbb  $\Delta$  értékeket itt is kis körök jelölik.

A 4. ábrán látható, hogy az igen kedvező  $\Delta$  ér-



4. ábra. Hibafelület a második mintatérre

tekek kevés középponthez tartoznak, szemben az előző mintatérnél mutatott átlagos 300 középpont számhoz. A 4. ábrán a 400 rejtett rétegbeli neuron szám felett már nem ábrázoltam a  $\Delta$  értékeket. Az egyes hálózatoknak a kívánt pontosság mellett legtöbbször sikerült az értékeket 30-50 középponthez hozzárendelni, ami a számítási igényt is csökkenti. Közel azonos tanítópont és azonos rejtett rétegbeli neuron szám mellett a számítási igény kevesebb, mint hetede, míg a  $\Delta$  érték kevesebb, mint harmincada az előző (hegyvidéki) mintatérben tapasztalt értékeknek.

A bevezetőben említett behatárolható területről érdemes megemlíteni, hogy az interpoláció a területtől csak nagy távolságban (kb. 3000 m) szűnik meg és alakul síkká. Tapasztalatom szerint ezen sík magassága a magasság koordináták számtani átlaga felé közelít.

## Összefoglalás

Az adatok közötti interpoláció megvalósítására a neurális hálózatok új eszközzel szolgáltak. Az egyik ilyen kedvező tulajdonsággal rendelkező mesterséges neurális hálózat az RBF (Radial Ba-



sis Function). A neurális hálózatok adaptív módon oldják meg az interpolációs feladatokat is. A hálózatok kialakítása során számos eljárási lehetőség közül választhatunk. A bemutatott tanított hálózatok paramétereinek száma  $3 \cdot H + 3$ , melyből 4 paramétert kell meghatároznunk, a többi az eljárás során adódik. Az RBF hálózat paramétereinek megválasztására becsléseket tudunk alkalmazni, de azok függnek a bemenő adathalmaztól. A paraméterezés, iteratíván, tapasztalati úton történt. Két eltérő jellegű mintatéren keresztül bemutatam az RBF hálózat főbb tulajdonságait. Megfelelő paraméterezéssel a hálózatok interpolációs és extrapolációs képessége jó. A szükséges pontossági paramétereket néha kevesebb tanítóponttal is el lehet érni, ami a jobb klaszterbesorolás eredménye. Összességében megfogalmazható, hogy az RBF neurális hálózat interpolációs képessége alkalmasá teszi magasság koordináták meghatározására.

#### IRODALOM

- [1] *Barsi Árpád*: Koordinátatranszformáció megoldása neurális hálózattal, Geodézia és Kartográfia, Budapest, LI, No. 10 pp. 12–18., 1999
- [2] *Horváth Gábor* (szerkesztő): Neurális hálózatok és műszaki alkalmazásai, egyetemi jegyzet, Műegyetem Kiadó, Budapest, 1998
- [3] *Michael Bertold, David J. Hand* (editors): Intelligent Data Analysis – An Introduction, Springer-Verlag, 1999
- [4] *Mohamad H. Hassoun*: Fundamentals of Artificial Neural Networks, MIT Press, 1995
- [5] *Steiner Ferenc*: A geostatistika alapjai, Tankönyvkiadó, Budapest, 1990

#### Using RBF neural network for determining altitude coordinates

*G. Veres*  
Summary

For performing interpolation among data, neural networks have developed a new tool. A major feature of Artificial Neural Networks is that they are adaptive by nature, solving problems not by using algorithms but through a learning process using examples. Radial Basis Function is considered to be one of the artificial neural networks having such favourable features. It is a characteristic feature of neural networks to perform tasks of interpolation in an adaptive way. When setting up networks you have the possibility to choose from among several approaches. A major feature of the said RBF network is that it can determine the parameters, – namely the centre of location and spread of the corresponding Gauss function – by competitive learning. In the hidden layer you can find Gauss functions from which you can get the output values by weighting and adding the values obtained. For selecting the parameters for RBF network you can use estimation, but the outcome of such estimations depends, of course, on the input data mass. Parameters are normally determined by experience, in an iterative way.

So, what I really wanted to do in this short description was to demonstrate, through two sample spaces of different feature, the major characteristics of RBF network. By properly selected parameters the interpolation and extrapolation capacities of such networks are considered to be good. Sometimes the required precision parameters can be achieved even by a lower number of teaching points – a result attributable to better clustering.

In summary, it can be concluded that RBF neural network, by its favourable interpolation capacity, can be applied as a suitable method for determining altitude coordinates.

Hatályba lépett a 13.692/2002.

FVM–FTF számú

## Új F2. Szabályzat.

A 2002. március 18-tól érvényes szabályzat  
ingyenesen letölthető a [www.fomi.hu](http://www.fomi.hu) címről,  
illetve beszerezhető a Földmérési és Távérzékelési Intézetnél.