

# AKNAKERESÉS NEURÁLIS HÁLÓZAT SEGÍTSÉGÉVEL

## Landmine localization using neural network

*GYÖRGY András<sup>1</sup>, dr. BARBARICS Tamás<sup>2</sup>, dr. PADÁNYI József<sup>3</sup>*

### Rezümé

A kutatómunka célja egy olyan eljárás kidolgozása, amely segítségével a tűzszerész-alakulatok biztonságosabb módon végezhetik munkájukat, azaz a gyalogsági aknák lokalizációja nem igényelne a továbbiakban emberi közreműködést. A cél egy olyan robot kifejlesztése, amely képes meghatározott típusú aknák geometriai paramétereinek ismeretében lokalizálni az aknákat, az ehhez szükséges tudást pedig a mesterséges neurális hálózatok technikáját felhasználva sajátítja el. A végeredmény egy olyan intelligens robot alakfelismerő algoritmusának bemutatása, amely egy terület aknamentesítésének első lépését, a lokalizációt képes megvalósítani emberi élet veszélyeztetése nélkül.

The purpose of the research is to develop a method that can make minesweeping safer by no longer demanding human assistance for the localization of landmines. The objective is to create a robot that can find specific landmine types by their geometric features. For this, the required knowledge is acquired by methods based on artificial neural networks. The final result is the demonstration of the algorithm of an intelligent robot's pattern recognition module that can achieve the first step of demining an area, which is localizing mines without risking human life.

### **Kulcsszavak:**

Soft computing, mesterséges neurális hálózat, aknakeresés

---

<sup>1</sup> BME hallgató

<sup>2</sup> BME egyetemi docens

<sup>3</sup> ZMNE egyetemi tanár

## Aknák – egy nyugtalanító probléma

Aknákat mind a mai napig telepítenek szerte a világon. A gyalogsági aknák egyik legnagyobb hiányossága, hogy lehetetlen megválasztani, kinél robbanjanak, ezáltal a civil lakosságot, a segélyszervezetek dolgozóit, a békefenntartókat és az élővilágot épp úgy, vagy még jobban veszélyeztetik, mint a harcoló katonákat. A balesetek számáról csupán becslések vannak, mivel nagy részük fejlődő országokban történik, de így is 20.000-re teszik éves számukat, ami óránként 2 újabb tragédiát jelent (a példa kedvéért: csak Kambodzsában 1979. és 2005. között több mint 45.000 ember élt át aknabalesetet, 75 százalékuk civil volt [1.]).

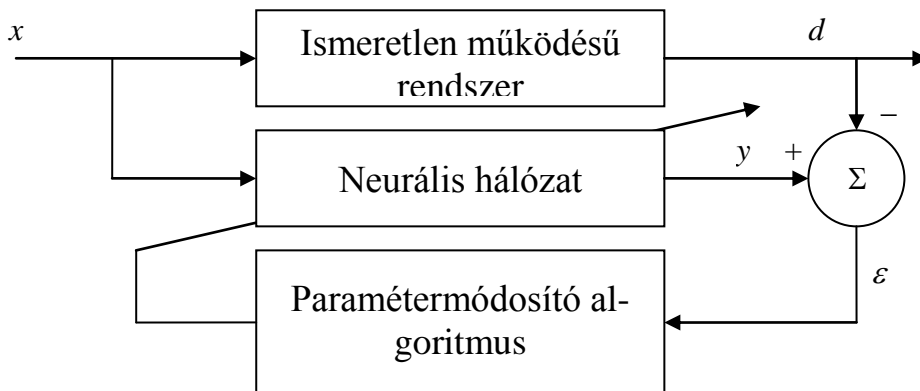
Szerte a világon 80 országnál is többet tartanak nyilván hivatalosan elaknásítottnak, és senki nem tudja pontosan, hogy egy adott területen hány akna van. A legveszélyesebb területek Afganisztán, Angola, Burundi, Bosznia-Hercegovina, Kambodzsa, Csecsenföld, Kolumbia, Irak, Nepál és Srí Lanka. További problémát jelent, hogy néhány ország nem szolgáltat hivatalos és publikus információt a helyzetről, mint például Burma, India és Pakisztán [1].

A múltban többféle érzékelési módszer került kifejlesztésre, ezek közül a legfontosabbak a következők [2.]: manuális keresés fémdetektorral, kutyák bevetése, betanított méhek [3.], rágcsálók [4.], növények [5.], baktériumok [6.], nukleáris és akusztikus érzékelés [7.].

A gyakorlatban alkalmazott módszerek mindegyikének megvan a gyenge pontja: élőlényekkel (kutyák, méhek, rágcsálók) történő keresés során egyrészt egy másik élőlény veszélyeztetése mellett igyekszünk másokat megóvni, ami bár racionálisan indokolható, mégis erkölcsi aggályokat támaszt a módszerrel szemben. Másrészt az adott egyed biológiája miatti befolyásoló hatások bizonytalanra tehetik a keresés eredményét, hiszen akármilyen jól is van kiképezve egy kutya, nem tudja teljes mértékben kiszűrni a környezetéből érkező zavaró hatásokat, ezáltal megbízhatósága és képességei csökkennek.

## A feladat megfogalmazása

Az előzőekben ismertett probléma esetén kézenfekvő választás a mesterséges neurális hálózatok univerzális approximációs képességét felhasználva alakfelismerő algoritmust kifejleszteni.



**1. ábra: A rendszer felépítése**

A következőkben tehát egy olyan algoritmust kívánunk kifejleszteni, amelynek belső működését explicit módon nem mi írjuk elő (hiszen nem is tudjuk, ennek megvalósítása maga a cél), hanem képes alkalmazsan megválasztott adatok, azaz mért bemenetek és ezekhez tartozó kívánt válaszok alapján megtanulni egy ismeretlen rendszer működését (1. ábra). Az ilyen soft computing megoldások a kimenetet ( $y$ ) tipikusan a bemenet ( $x$ ) nemlineáris leképezése segítségével állítják elő. A helyes viselkedés kialakítása tanítóminták segítségével történik: ekkor a rendszer kívánt kimenetét ( $d$ ) ismerjük, és így a hiba ( $\varepsilon$ ) segítségével a nemlineáris leképezés paramétereit (a súlyokat) a helyes irányba módosítjuk. A működés részletes leírása [8.]-ban megtalálható.

Célunk egy ilyen algoritmus létrehozása meghatározott virtuális aknatípusok esetén, azzal a kitételrel, hogy az osztályozás megengedett hibája nem szimmetrikus, hiszen ha néhány esetben fals pozitív jelzést kapunk, az közel sem akkora probléma, mint ha elmulasztanánk egy valódi aknát. A megoldás értékelésekor ez meghatározó szempont lesz.

A gyakorlatban megoldandó feladatok:

1. mintagyűjtés,
2. adatbázis elkészítése (emberi értékelés),
3. neurális hálózat struktúrájának megválasztása (a priori ismeretek alapján),
4. tanítás,
5. tesztelés.

A dolgozat az első két pont esetén kicsit más megoldást választ, hiszen ennek a munkának többek közt az a célja, hogy a Magyar Honvédségnek demonstrálja egy soft computing elven működő aknalokalizátor kifejlesztését, ebből következően ilyen gép még nem létezik, tehát valódi mintákról nem lehet beszélni. Így tehát a mintákat számítógéppel kell generálni, ügyelve arra, hogy azok a lehető legjobban hasonlítsanak a valódiakhoz, tehát meghatározott mértékű zajt kell hozzákeverni, ami által a mintagenerálás szimmetriáját elrontjuk. Ha ezt nem tennénk, akkor a háló alkalmazása ebben a virtuális esetben nem jelentene minőségi újítást, hiszen adott, jól definiált számú és formájú minta felismerése ennél sokkal egyszerűbben is megoldható lenne. Természetesen a humán tudás bevitele is számítógépes algoritmus segítségével történik, de itt is a valóságot próbáljuk követni: a válaszok meghatározása a tanítóhalmaz esetén a „tisztá” minták alapján történik, majd tanítás esetén hozzákeverjük a zajt, tehát a háló bemenetére nem a „szép” minták kerülnek, hanem a „torzítottak”, ezáltal modellezve a valóságot. A számítógépes szimulációt követően a valódi gép létrehozásakor a tanulási bázist nem csak a számítógépes adatbevitel során keletkező adatok, hanem valódi telepített gyakorlóaknak megtaláltatása is szolgáltatja.

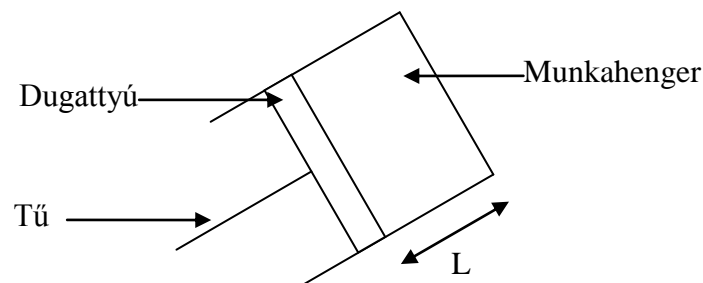
## Érzékelés

A megoldás során kétféle aknatípust vizsgálunk, ezzel jelezve, hogy az elv tetszőleges méretű és formájú akna esetén hatékonyan alkalmazható. Minden aknatípus esetén az előbb megfogalmazott öt lépést kell megismételni, természetesen azonos keretprogrammal. A tanítási ciklus során egy adott aknatípussal telepített aknamezőt végig kell pásztázni az érzékelőkkel, a tűzsereszekkel együttműködve létre kell hozni az adatbázist, majd az erre az aknára specifikus hálót ez alapján megtanítani. Ezt minden aknatípus esetén elegendő egyszer megtenni, így ez a fejlesztési fázis során egyszer jelentkező költség.

Miután megvannak a megtanított aknaspecifikus hálók, az átvizsgálni kívánt aknamezőt egyszer kell végigpásztázni, az így kinyert adatok pedig előfeldolgozásuk után a hálók bemenetére kerülnek. Természetesen a már megtanított aknákat felismerő hálók együtt futnak, nem kell „programot cserélni” minden egyes aknatípushoz. Ilyen módon a keresési terület paramétereinek (keménység, zajosság, mélység stb.) megfelelően néhány beállítást kell csupán módosítani, de ekkor is egy területet csak egyszer kell végigpásztázni. Ezzel elérhető, hogy a háló végső paramétereinek, azaz az algoritmusban szereplő együttthatóknak a tanítása egy adott aknatípus esetén egyszer végzendő el, és az így kapott paraméterek a Balkánon épp úgy alkalmazhatók, mint például Kambodzsában.

Az aknák érzékelése egy, a manuális szűrőbotos érzékeléshez nagyon hasonló robottal történik, ami egymás mellett elhelyezett 20 pneumatikus érzékelőt tartalmaz. Mindegyik érzékelő egy nyomás ellenében mozgó kis dugattyúhoz hasonlítható (2. ábra), amelynek másik oldalán egy tű van, amelyet a földbe szúrunk. A szúrást ferdén kell elvégezni, nehogy az akna detonálódjon. A mélység alapján, ahol a tű fennakad, következtetni lehet a talajviszonyokra. Erre szolgál az L paraméter, ami egyszerűen mérhető. Ezeket az érzékelőket egymástól 5 cm-re elhelyezve egyszerre egy 1 m-es szakasz vonalszerűen átvizsgálható,

majd ezt egymás után 20-szor elvégezve egy 1 négyzetméteres területről megvan a szükséges mélységinformációnk. Természetesen minden érzékelő a többitől függetlenül nyomódik be a földbe, a dugattyú mögött levő nyomástól függő keménységgel hatol előre, ezzel különböző talajtípusok esetén egyetlen paraméter (nyomás) megváltoztatásával alkalmazható a módszer. Tehát különböző területek esetén nem a neurális háló paramétereit, hanem csak a pneumatikus érzékelő egyetlen paraméterét kell átállítani, ezzel univerzálissá és mindenhol könnyen alkalmazhatóvá válik a műszer.



2. ábra: Az érzékelő felépítése

### Felismerendő aknatípusok – alampinták

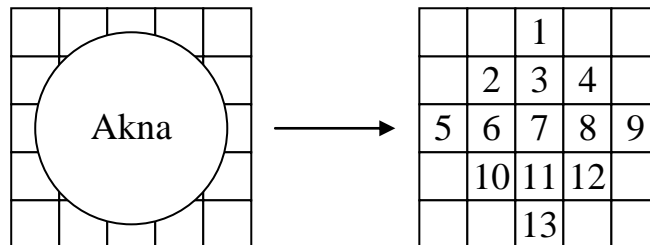
A 3. ábra mutatja az emberek és harckocsik ellen bevetett aknák leggyakoribb típusát: jellegzetes alakja körszimmetrikus, rajta különböző kiálló formákkal



3. ábra: Gyakorlatban alkalmazott aknatípusok

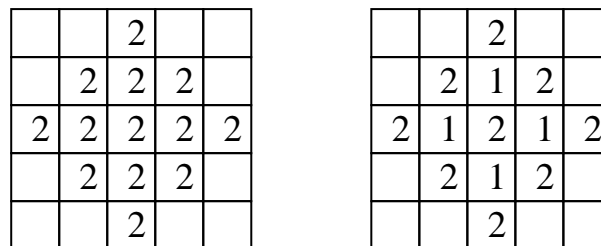
Ha körülbelül 20 cm átmérőjű aknákat vizsgálunk, akkor az akna egy 25 cm × 25 cm-es négyzettel foglalható körbe, tehát a négyzet minden mezőjének középpontjába egy-egy érzékelőt elhelyezve 25 darab szűrési értékből a 4. ábra

által jelölt 13 szolgáltató az adott mezőbe eső aknáról információt. A továbbiakban minden esetben ezt a 13 értéket vezetjük előfeldolgozásuk után a neuronháló bemeneteire (0 – mély, 1 – közepes, 2 – közel van a földfelszínhez).



4. ábra: Akna leképezése

A vizsgált formák: az I. aknatípus egy korongot formáz, míg a II. egy peremmel és közepen kiálló dudorral rendelkezik (5. ábra).



5. ábra: A két keresett aknatípus (I. és II.)

A gyakorlatban az aknákat repülőgépről szórják, illetve földi ütegekkel telepítik, miáltal elhelyezkedésük a földben meglehetősen szabálytalan és véletlenszerű. Vizsgálatunk során tehát figyelembe kell veyük az aknák ferdeségét, valamint az elforgatásra való invarianciát. Így „tisztá”, zajmentes alaptípus az előbbieken kívül 8-8 darab van (6. ábra és 7. ábra).

		2		
	2	2	2	
1	1	1	1	1
	0	0	0	
		0		

		1		
	0	1	2	
0	0	1	2	2
	0	1	2	
		1		

		0		
	0	0	0	
1	1	1	1	1
	2	2	2	
		2		

		1		
	2	1	0	
2	2	1	0	0
	2	1	0	
		1		

		2		
	1	2	2	
0	0	1	2	2
	0	0	1	
		0		

		0		
	0	0	1	
0	0	1	2	2
	1	2	2	
		2		

		0		
	1	0	0	
2	2	1	0	0
	2	2	1	
		2		

		2		
	2	2	1	
2	2	1	0	0
	1	0	0	
		0		

6. ábra: Alaptípusok I. esetén (ferdeség és forgatás)

		2		
	2	1	2	
1	0	1	0	1
	0	0	0	
		0		

		1		
	0	0	2	
0	0	1	1	2
	0	0	2	
		1		

		0		
	0	0	0	
1	0	1	0	1
	2	1	2	
		2		

		1		
	2	0	0	
2	1	1	0	0
	2	0	0	
		1		

		2		
	1	1	2	
0	0	1	1	2
	0	0	1	
		0		

		0		
	0	0	1	
0	0	1	1	2
	1	1	2	
		2		

		0		
	1	0	0	
2	1	1	0	0
	2	1	1	
		2		

		2		
	2	1	1	
2	1	1	0	0
	1	0	0	
		0		

7. ábra: Alaptípusok II. esetén (ferdeség és forgatás)

Ha a neurális háló bementére az érzékelők által szolgáltatott 13 értéket (amelyek a  $[0;2]$  folytonos zárt intervallum elemei, hiszen az  $L$  paramétert ide normálhatjuk) adnánk az eddig még nem részletezett előfeldolgozás nélkül, akkor a háló az elméleti eredmények ([8.]) tükrében meg tudná oldani az osztályozást, azonban a mérete hatalmas kellene legyen, a tanítás rengeteg mintapontot igényelne és túl sokáig tartana.



Mivel a mintagyűjtés szakképzett tűzszerészekkel együttműködve történik, célszerű szakmai tapasztalataikat felhasználni, mintegy tudás formájában implicit módon belekódolni a megoldásba. Ezáltal megkönnyítjük a háló tanulását, hiszen előzetes ismereteinket felhasználjuk, azaz intelligenciát viszünk a rendszerbe, és a mérnöki problémamegoldás egyik legfontosabb eszközével élünk: a bonyolult és túl nagy problémát több egyszerűbb és kisebb problémává redukáljuk, azaz dekompozíciót hajtunk végre.

### Dekompozíció - részminták

A hadseregben dolgozó aknakereső szakemberek az aknákat jellegzetes részformáik alapján ismerik fel, tehát egyszerre csak egy kis részével foglalkoznak, és ha a meghatározott részformák adott struktúra szerint illeszkednek, akkor tudják, hogy aknát találtak. Ezt a módszert alkalmazzuk mi is: részformákat keresünk, és ha ezek megadott geometria szerint helyezkednek el egymáshoz képest, akkor ott akna van. Az alapformákat tanulmányozva (6. és **Hiba! A hivatkozási forrás nem található.** ábra) kézenfekvő (zajmentes) részminták adódnak a két aknatípus esetén, melyek közül néhány a 8. ábrán látható.

		2																	
	2	2	2					2							0	0			
								1	2				2	2					
								2					2	2		0	0		
													2				0		
												1							
								1				0						0	
2	1	2	1	2			1	1				1						0	0
							1	1				0						0	
												1							

8. ábra: Részformák I. és II. esetén

Így az I. aknatípus esetén 22, míg a II. aknatípus esetén 24 részmintát különböztetünk meg. A későbbiek során majd ezeket, illetve ezek kombinációit tanítjuk meg a hálónak, ezáltal leegyszerűsítve a megtanulandó feladatot, hiszen egy alapminta egyértelműen azonosítható a részminták kombinációjaként.

Mindkét aknatípus esetén 9 alapminta van, amit meg kell különböztetni. Azonban a valóságban a földben vannak fadarabok, kövek, csontok, mindenféle „zaj”. Ezáltal nem a „tisztá” alapmintát érzékeljük, hanem valami zajos változatát. Ezt is figyelembe kell venni, tehát az alaptípusokat olyan módon bővítjük, hogy minden rész minta esetén egy érzékelt mélységérték lehet zajos, azaz nagyobb értékű, mint az alaptípus esetén, hiszen ha egy kő van az akna felett, akkor a mechanikus tű előbb fennakad, tehát az érzékelt mélység kisebb lesz.

### **Előfeldolgozás**

Észrevehetjük, hogy rengeteg érzékelt minta van, amit aknának kell jelezni, és eddig csak diszkrét értékekkel foglalkoztunk (0, 1 illetve 2), ami túl durva kvantálás. Folytonos mintahalmaz esetén a felállítandó szabálybázis mérete olyan hatalmas lenne, hogy gyakorlati szempontból megvalósíthatatlan volna még egy ilyen leegyszerűsített példa esetén is, ahol rengeteg előzetes ismeretünk van, ráadásul a valóság ennél sokkal komplikáltabb és rosszabbul definiált. Kijelenthető tehát, hogy a soft computing megközelítés kézenfekvő és célravezető.

Az érzékelők a  $[0;2]$  folytonos intervallum jeleit szolgáltatják, amivel a háló tanítása feleslegesen bonyolult lenne. Ha egyszerűen csak kerekítenénk, akkor megkapnánk a  $\{0;1;2\}$  diszkrét értékekkel jellemezhető mintákat, azonban túl durva lenne a redukció, ezáltal nagy lenne az információvesztés. Éppen ezért egy kisebb folytonos intervallumra képezzük le az érzékelők által szolgáltatott jeleket, ami kellőképpen elkülöníti a különböző mélységeket, de nem mossa el teljesen az árnyalatokat. Kézenfekvő választás a távolság négyzetének valamilyen konstans-szorosára csökkenteni a távolságot, hiszen ekkor három folytonos

intervallumot kapunk, amik jellemzik a mélységet, ami által elkülönül a mély, közepes és földfelszínhez közeli kategória, de megmaradnak a finom különbségek a szinteken belül. Tehát a nemlineáris leképezés a következő alakú, ha az érzékelők által szolgáltatott jel az  $x \in [0;2]$ , a feldolgozott pedig az  $\hat{x}_{preproc} \in [0;2]$ :

$$\hat{x}_{preproc} = \begin{cases} c \cdot x^2 & , \text{ ha } x \in [0;0.5) \\ 1 + \text{sgn}(x-1) \cdot c \cdot (x-1)^2 & , \text{ ha } x \in [0.5;1.5) \\ 2 - c \cdot (x-2)^2 & , \text{ ha } x \in [1.5;2] \end{cases}$$

Ha a neurális hálózat bemeneteire ez a jel kerül, akkor ez implicit módon tartalmazza azt a tudásunkat, hogy a minket érdeklő információ alapvetően az, hogy a három főszint közül melyikhez tartozik, a kisebb információtartalmat hordozó tudás pedig a szinten belüli elhelyezkedés. Ezáltal képesek vagyunk jellemezni a mintákat durván, az intervallum alapján, ezen belül pedig finom disztingválást tesz lehetővé az adott intervallumon belüli elhelyezkedés.

Mivel a neurális hálóban használt aktivációs függvények alapvetően nulla környezetében meredek, azaz itt nagy a gradiensük, célszerű a tanítás gyorsítása érdekében a mintáinkat ide normálni, azaz az előző képletben egy 0.5-es faktort alkalmazni ( $x_{preproc}$ ).

A képletben szereplő  $c \in (0;2]$  konstans beállítandó paraméter, a megoldás során értékét egynek választottuk.

### A neurális hálózat struktúrája

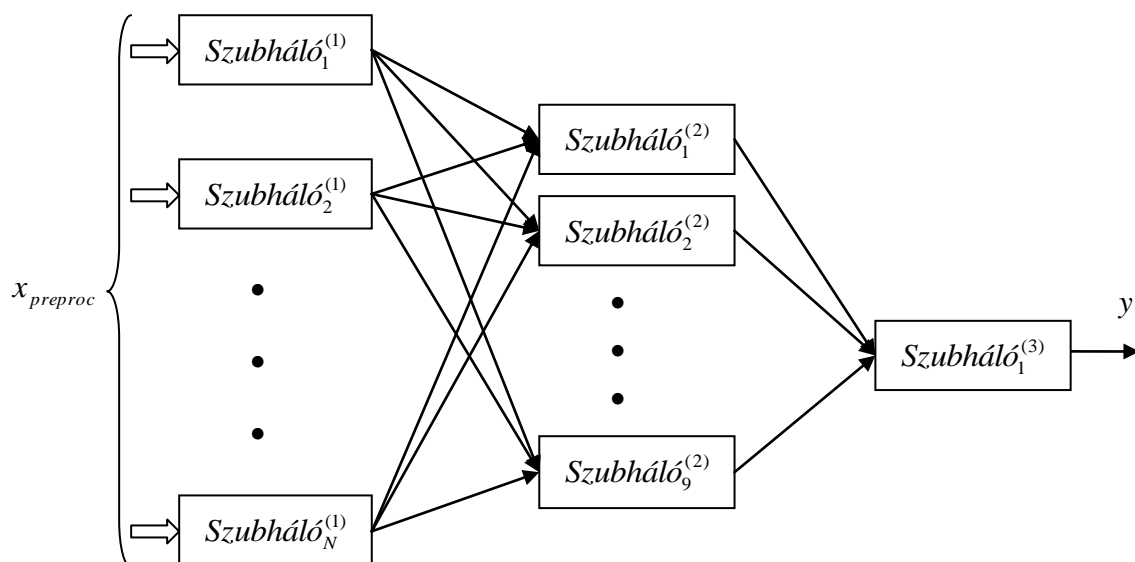
A hatékony tanulás érdekében a priori ismereteinket felhasználjuk a háló struktúrájának kialakítása során, itt visszük be a feladatba az emberi intelligenciát. Ennek fényében moduláris hálózatot hozunk létre (9. ábra), ahol kisebb, úgynevezett szubhálókat alkalmazunk egyes funkciók ellátására, azaz részminták megtanulására. Az elméleti eredmények értelmében a sikeres és gyors tanulás érdekében a szubhálókat két rejtett rétegű MLP (*Multi Layer Perceptron*)

hálókból hozzuk létre, tanításuk pedig az igen hatékony Levenberg-Marquardt-eljárással történik.

Az első modulréteg feladata a részminták felismerése, azaz az I. aknatípus esetén ez  $N=22$ , a II. aknatípus esetén pedig  $N=24$  szubhálót jelent. Minden szubháló csak a számára szükséges előfeldolgozott bemeneti jeleket kapja meg, hiszen a többi csak feleslegesen bonyolítaná a tanulását, mivel minden cél nélkül növelné a bemeneti tér dimenzióját.

A második modulréteg feladata az egyes részminták kompozíciója alapján felismerni az alap aknatípusokat (mindkét esetben 9), azaz ez a modulréteg mindkét esetben 9 szubhálót tartalmaz. Természetesen itt is csak a szükséges részmintákat felismerő első modulrétegbeli szubhálók kimeneteit kapják meg a második rétegbeli szubhálók, a már ismertett okok miatt.

A harmadik modulréteg feladata a végső eredmény szolgáltatása, azaz van-e az adott 5x5-ös mezőben akna. Mivel a kimenet számunkra „igen/nem” (eldöntendő) kell legyen, az ebben a modulrétegben szereplő egyetlen szubháló kimenetére egy lépcsősfüggvényt kell tenni (Döntő), aminek komparálási szintje később kerül meghatározásra.



9. ábra: A moduláris MLP felépítése

## Validációs és teszthalmaz

A rendelkezésünkre álló mintákat tipikusan három részre szokás osztani: tanító, validációs illetve tesztelő szettre. A tanítóhalmazt a neurális hálózat tanítására, a validációs halmazt közelítő értékelésre valamint korai leállításra, a tesztelésre szánt mintákat pedig kiértékelésre használjuk. Esetünkben ez picit módosul: a probléma kritikus volta miatt egy adott mintahalmazt már a folyamat legelején teszthalmaznak különítünk el, hiszen esetünkben szó szerint életbevágó az algoritmus teljesítőképességének biztos ismerete, egyébként alkalmazása feleltelen és értelmetlen lenne. Így a probléma megoldása során a maradék mintahalmazt két részre oszthatjuk a korábbi megfontolásoknak megfelelően: tanító és validációs halmazra.

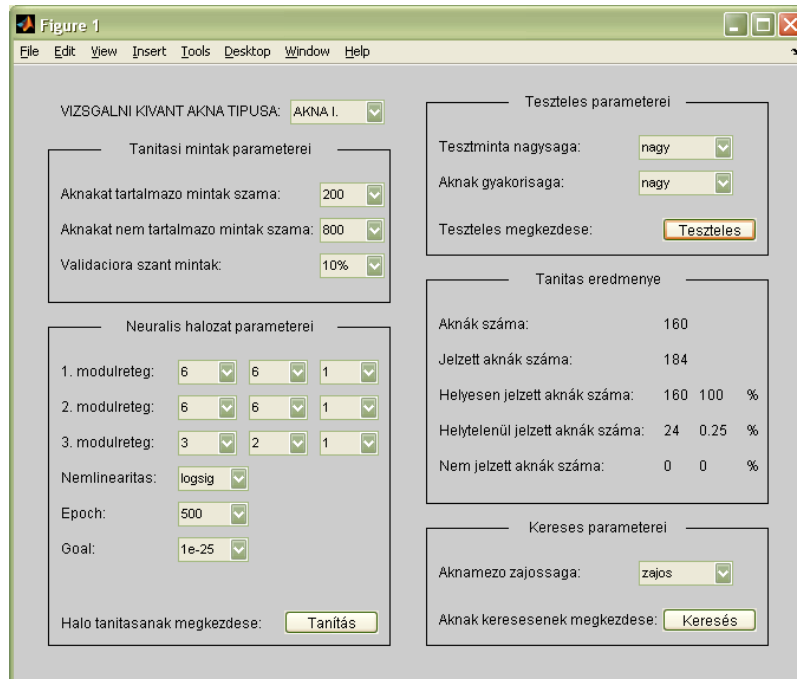
A tanító és a validációs halmaz adott bázisból származik, míg a tesztelés során egy másik adatbázist veszünk elő: itt egy adott, zajjal (kövekkel, sziklákkal, stb.), valamint „majdnem aknákkal” terhelt aknamezővel folyik a tesztelés (minden esetben természetesen más aknamezővel), ami után nyilatkozhatunk arról, hogy a megoldás kielégítő pontosságú-e, azaz eleget tesz-e a kritériumoknak. Tehát a tesztelés nem azon a halmazon történik, amivel tanítottunk, hanem előzetesen letárolt aknamezőn, ezáltal valóban meggyőződhetünk arról, hogy a neurális hálózat a problémát tanulta meg, nem csak a mintapontokat, hiszen a mintákat eltérő módon generáltuk.

## A fejlesztést segítő GUI

A fejlesztési fázis során egy erre a célra készített GUI (*Graphical User Interface*) alkalmazása jelentős mértékben megkönnyíti a feladatot, elkészítéséhez hathatós segítséget nyújt [9.]. A beállítható paraméterek (70. ábra):

- vizsgálni kívánt akna típusa,
- tanítás paraméterei,

- neurális hálózat paraméterei,
- tesztminta nagysága és az aknák száma,
- keresés során az aknamező zajossága.

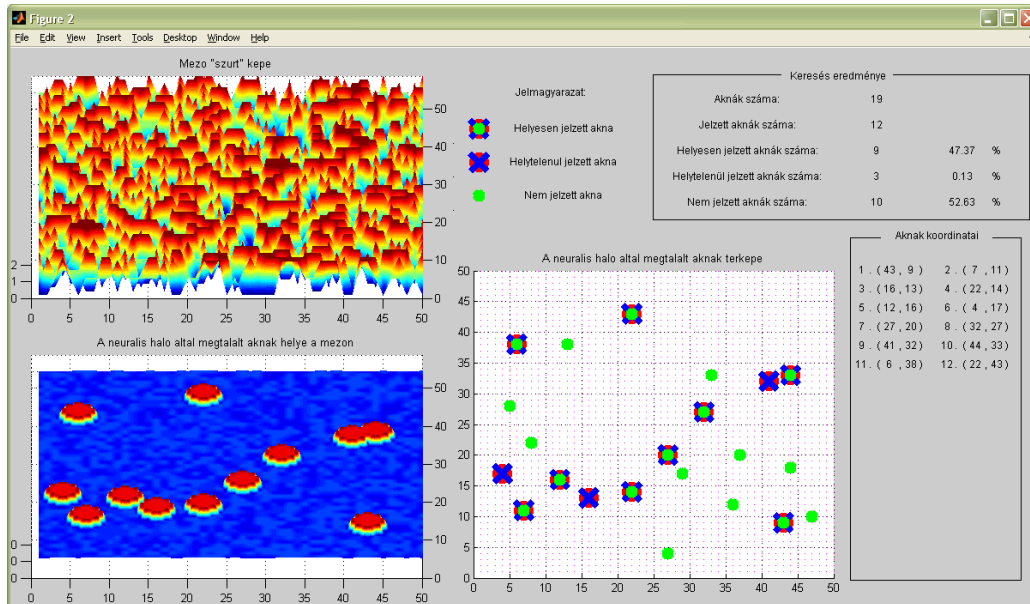


70. ábra: A fejlesztést segítő Graphical User Interface

A Döntő egység komparálási szintjét célszerű úgy megválasztani, hogy minél zajosabb a környezet, annál alacsonyabb legyen, így a megvalósítás során ezzel a megoldással élünk, tehát a zajossággal együtt változik ez az érték. Természetesen, ha a nem jelzett aknák száma túl nagy, akkor csökkenteni kell ezt az értéket, ha pedig túl sok a helytelenül jelzett akna, akkor célszerű növelni.

Keresésnél nagymértékben zajos, sok „majdnem aknával” teleszórt aknamezőt ( $50 \times 50$  szűrásnyi) vizsgálunk, ezzel illusztrálva a valós alkalmazás menetét. A keresési ablak (11. ábra) bal felső sarkában a zajos, feldolgozatlan jeleket látjuk három dimenzióban. Alatta a feldolgozott és szűrt jelek jelennek meg, tehát az itt jelzett aknák középpontjait szolgáltatja számunkra a neurális hálózat. Emellett található az értékelést segítő ábra: itt figyelhetjük meg, hogy a jelzett aknák mennyire egyeznek a valósággal, azaz hol jelzett helyesen, hol jelzett

helytelenül, illetve hol nem jelzett egyáltalán a hálózat. A jobb felső sarokban az előző ábra van számszerűen összefoglalva, alatta pedig a neurális hálózat által szolgáltatott aknák középpontjainak koordinátái vannak felsorolva.



81. ábra: A keresés eredményét megjelenítő ablak

## Tesztelés és a keresés

A tesztelés és a keresés során az aknamezőt felosztjuk négyzetméternyi darabokra, amelyekről mélységi információkat gyűjtünk, ezeket feldolgozzuk, majd kiértékeljük az eredményt. Egy ilyen kisebb egységben egy 25 cm x 25 cm-es ablakot „csúsztatunk”, és megvizsgáljuk, hogy az aktuális pozícióban van-e akna. Amikor ezzel kész vagyunk, megyünk a következő ablakpozícióra és folytatjuk tovább.

Keresés esetén teljesen új körülményekkel találjuk magunkat szemben, hiszen az innen gyűjthető mintákat eddig sehol nem használtuk fel. Ekkor derül ki, hogy a hálónk általánosítóképessége megfelelő-e: képes-e valóban „megtanulni” az akna és a nem akna közti különbséget? Ennek érdekében egy előzőleg letárolt, zajjal terhelt aknamezőt veszünk elő, amire a köveken, sziklákon, fadarab-

okon, stb. kívül még „majdnem-aknákat” is telepítettünk, azaz olyan objektumokat, amelyek alakja részlegesen aknaszerű.

<b>I. típusú akna</b>	Túl kis méretű háló 1: 3-3-1 neuron 2: 3-3-1 neuron 3: 3-3-1 neuron		Elegendő méretű háló 1: 6-6-1 neuron 2: 6-6-1 neuron 3: 3-2-1 neuron		Túl nagy méretű háló 1: 10-10-1 neuron 2: 10-10-1 neuron 3: 5-5-1 neuron	
Helyesen jelzett aknák	<b>90</b>	<b>56.60%</b>	<b>160</b>	<b>100.00%</b>	<b>155</b>	<b>100.00%</b>
Helytelenül jelzett aknák	<b>6</b>	<b>0.06%</b>	<b>24</b>	<b>0.25%</b>	<b>19</b>	<b>0.2%</b>
Nem jelzett aknák	<b>69</b>	<b>43.40%</b>	<b>0</b>	<b>0.00%</b>	<b>0</b>	<b>0.00%</b>

1. Táblázat: Tesztelési eredmények az I. típusú akna esetén

Az I. típusú akna esetén a tesztelési eredményeket (1. táblázat) vizsgálva látható, hogy a túl kis méretű háló esetén az osztályozás nem megfelelő: az aknák megközelítőleg felét ismeri fel a neurális hálózat. Ha a neuronszámot növeljük az első két modulrétegben (ezek a feladat megoldása szempontjából kritikus rétegek), akkor eljutunk egy kielégítő megoldást nyújtó hálózathoz, ahol már nincs nem jelzett akna, ráadásul a helytelenül jelzett aknák száma is 1% alatt van. A hálózatot tovább növelve látható, hogy az osztályozás már alig javul 50%-kal több neuron felhasználása esetén, tehát a feladat megoldása szempontjából elegendő a 6-6-1, 6-6-1 illetve 3-2-1 neuront tartalmazó szubhálókból felépített modulrétegekből álló neurális hálózat. Természetesen a túl kis méretű háló esetén a Döntő komparálási szintjét alacsonyabbra állítva csökkenthető lenne a nem jelzett aknák száma a fals pozitív jelzések gyakoriságának növekedése árán.



<b>I. típusú akna</b>	Túl kis méretű háló 1: 3-3-1 neuron 2: 3-3-1 neuron 3: 3-3-1 neuron		Elegendő méretű háló 1: 6-6-1 neuron 2: 6-6-1 neuron 3: 3-2-1 neuron		Túl nagy méretű háló 1: 10-10-1 neuron 2: 10-10-1 neuron 3: 5-5-1 neuron	
	Helyesen jelzett aknák	<b>9</b>	<b>47.37%</b>	<b>18</b>	<b>100.00%</b>	<b>18</b>
Helytelenül jelzett aknák	<b>3</b>	<b>0.13%</b>	<b>1</b>	<b>0.04%</b>	<b>1</b>	<b>0.04%</b>
Nem jelzett aknák	<b>10</b>	<b>52.63%</b>	<b>0</b>	<b>0.00%</b>	<b>0</b>	<b>0.00%</b>

### 2. Táblázat: Keresési eredmények az I. típusú akna esetén

Vegyük észre, hogy a keresési eredmények (2. táblázat) nagyságrendileg megegyeznek a tesztelés során kapott értékekkel, ami jól mutatja, hogy a háló valóban képes az általánosításra, azaz nem csak a mintapontokat, hanem egy zajosnak generált aknamezőn elhelyezett torzított aknákat is felismeri, a „majdnem aknákat” pedig lényegében teljesen kiszűri, azaz elfogadható mértékűre csökkenti a fals pozitív jelzések számát.

<b>II. típusú akna</b>	Túl kis méretű háló 1: 3-3-1 neuron 2: 3-3-1 neuron 3: 3-3-1 neuron		Elegendő méretű háló 1: 7-7-1 neuron 2: 7-7-1 neuron 3: 4-3-1 neuron		Túl nagy méretű háló 1: 10-10-1 neuron 2: 10-10-1 neuron 3: 5-5-1 neuron	
	Helyesen jelzett aknák	<b>71</b>	<b>47.33%</b>	<b>135</b>	<b>100.00%</b>	<b>151</b>
Helytelenül jelzett aknák	<b>32</b>	<b>0.34%</b>	<b>18</b>	<b>0.19%</b>	<b>15</b>	<b>0.16%</b>
Nem jelzett aknák	<b>79</b>	<b>52.67%</b>	<b>0</b>	<b>0.00%</b>	<b>0</b>	<b>0.00%</b>

### 3. Táblázat: Tesztelési eredmények a II. típusú akna esetén

A II. típusú akna esetén a tesztelési eredményekre (3. táblázat) tekintve elmondható, hogy az előzőhöz nagyon hasonló eredményeket kaptunk, azzal a

kis eltéréssel, hogy a megfelelő teljesítményt nyújtó hálózat ebben az esetben egy kicsit nagyobb, azonban ez összhangban van azzal, hogy a II. aknatípus kismértékben bonyolultabb, mint az I. aknatípus. Észrevehető, hogy a harmadik konfiguráció esetén mintegy 50%-kal növelve a neuronszámot nem értünk el jelentős javulást, tehát a feladat megoldása szempontjából elegendő a 7-7-1, 7-7-1 illetve 4-3-1 neuronokat tartalmazó szubhálókból felépített modulrétegekből álló neurális hálózat.

<b>II. típusú akna</b>	Túl kis méretű háló 1: 3-3-1 neuron 2: 3-3-1 neuron 3: 3-3-1 neuron		Elegendő méretű háló 1: 7-7-1 neuron 2: 7-7-1 neuron 3: 4-3-1 neuron		Túl nagy méretű háló 1: 10-10-1 neuron 2: 10-10-1 neuron 3: 5-5-1 neuron	
	<b>9</b>	<b>45.00%</b>	<b>18</b>	<b>100.00%</b>	<b>18</b>	<b>100.00%</b>
Helyesen jelzett aknák						
Helytelenül jelzett aknák	<b>1</b>	<b>0.04%</b>	<b>5</b>	<b>0.22%</b>	<b>2</b>	<b>0.09%</b>
Nem jelzett aknák	<b>11</b>	<b>55.00%</b>	<b>0</b>	<b>0.00%</b>	<b>0</b>	<b>0.00%</b>

#### 4. Táblázat: Keresési eredmények a II. típusú akna esetén

A keresési eredményeket (4. táblázat) tanulmányozva hasonló mondható el, mint az I. aknatípus esetén: megfelelő méretű háló esetén az általánosítóképesség kielégítő, tehát felismeri a torzított, tanítás során fel nem használt aknákat, valamint a „majdnem aknákat” elfogadható mértékben kiszűri.

Itt kell megjegyeznünk a módszer óriási előnyét, amely a neurális hálók működéséből adódik: ha egzaktul le szeretnénk írni az összes esetet, akkor hatalmas méretű szabálybázisra lenne szükségünk, ráadásul nem tudnánk, hogy minden esetet lefedtünk-e, ami a gyakorlatban tragikus következményekkel járna. Ezzel szemben neurális hálózatokkal a bementi térben hiperbuborékokat hozunk létre az egyes aknatípusoknak megfelelően, amiknél nem szükséges az összes esetet lefedni, elég csak néhányat, a többire ugyanis a válasz az előbbieket

közelében lesz. Itt használjuk ki azt a tényt, hogy az egymáshoz hasonló bemenetekhez hasonló kimenetek tartoznak.

További hatalmas előnyt hordoz az a tulajdonság is, hogy ha egy minta sok alapformára hasonlít, de egyiknél sem szignifikánsan az, tehát a második modulréteg egyik kimenete sem túl nagy, akkor a rendszer nagy valószínűséggel mégis „igen”-t fog kimenetként szolgáltatni, hiszen a „majdnem alaptípusok” mértéke mintegy összeadódik a harmadik modulréteg által, ezáltal gyanús helyzetekben a hálózat kimenete jó irányba mozdul el.

Vegyük észre, hogy nem éles fogalmakkal dolgozunk, hanem elmoszuk ezeket, hasonlatosan az emberi gondolkodáshoz: hogy egy adott folytonos értékkészletű alakzat mennyire egyezik a keresett formával, arra mi, emberek se igent vagy nemet válaszolunk. A klasszikus programozás keretei közt erre nincs lehetőségünk, viszont a neurális hálózatok súlyfilozófiája tökéletesen illeszkedik a gondolatmenethez: minél inkább hasonlít, annál inkább egy a kimenet, ezáltal annál közelebb vagyunk a kívánt válaszhoz.

### **Megvalósítási kérdések**

A kifejlesztett eszköz három fő egységre bontható: a robotra, az érzékelőkre, illetve az információfeldolgozást végző szoftverre, azaz magára a neurális hálóra. A két hardware-jellegű eszköz esetén nem fontos a nagymértékű pontosság, mivel a feladat jellegéből adódóan az adatok elkerülhetetlen módon nagymértékű zajjal terheltek.

Az általunk bemutatott megközelítés, tehát a neurális hálózat alkalmazása jól látható módon igen könnyen algoritmizálható: a tanítás során a backpropagation algoritmus Levenberg-Marquardt-féle változatának kódolása viszonylag egyszerű. A neurális hálózat alkalmazásának második fázisa, az előhívás során pedig még ennél is könnyebb a helyzet: csupán egyszerű aritmetikai műveleteket kell elvégezni. Tehát mind a tanítás, mind az előhívás során egysze-

rú és könnyen megvalósítható algoritmussal van dolgunk, az egyetlen nehézség a lebegőpontos számok használata, hiszen a módszer sok kis művelet egymás után való alkalmazásából épül fel, ahol kritikus lehet a számolás során felhalmozódó eredő hiba. Éppen ezért a szoftvert egy lebegőpontos műveletekre optimalizált mikrokontrolleren célszerű megvalósítani. A mikrokontrolleres megvalósítás további előnye, hogy az algoritmusok megírása például C nyelven történő implementációja esetén a MATLAB teljes mértékben mellőzhető, ezáltal a költségek tovább csökkenthetők.

## Irodalomjegyzék

- [1.] <http://www.icbl.org/>
- [2.] <http://en.wikipedia.org/wiki/Demining>
- [3.] Jerry J. Bromenshenk et al.: *"Bees used in Area Reduction and Mine Detection"*. Journal of Mine Action, Issue 7.3., 2003.
- [4.] <http://www.aegis.com/news/re/2004/RE040956.html>
- [5.] [http://www.acfnewsresource.org/science/mine\\_sniffing\\_plants.html](http://www.acfnewsresource.org/science/mine_sniffing_plants.html)
- [6.] <http://www.mech.uwa.edu.au/jpt/demining/others/ornl/rsb.html>
- [7.] <http://technology.newscientist.com/article/dn10868-vibrations-could-reveal-landmine-locations.html>
- [8.] Altrichter M., dr. Horváth G., dr. Pataki B., dr. Strausz Gy., Takács G., Valyon J.: *Neurális hálózatok*. Panem Kiadó, Budapest, 2006.
- [9.] Berta M., Dunay R., Fazekas I., Gáll J., Gergó L., Gyurkovics É., Jenei A., Kollár I., Kuczmann M., Molnárka Gy., Román Gy., Simon Gy., Gisbert, S.: *MATLAB*. Typotex Kiadó, Budapest, 2005.