Kolozsvár, 1998. március 20-21.

BACK-PROPAGATION NEURON HÁLÓ LOGIKAI JÁTÉK JÁTSZÓ KÉPESSÉGÉNEK VIZSGÁLATA

Dudás László¹, Ferenczi László²
egyetemi docens¹, doktorandusz²
Miskolci Egyetem

ABSTRACT

The back-propagation learning algorithm for artificial neural networks invented by *Werbos* in 1974 is a famous and frequently used model for training of multi-layered feedforward neural networks. It has a good capability of association of many patterns presented as input and required output pairs. After training the network can remember the used output for a given input and has some more interesting features.

It seems to be a different area the logical game playing. In these cases we can generate the tree of the game having graph points symbolizing the possible states of the game. There are well-known playing tree evaluation methods, the mini-max algorithm, α - β pruning and so on, but there is a problem if the tree is very big. If this happens we have to use a heuristic static evaluation function which can give as good result as good the heuristical evaluation function itself. This problem suggests an another resolution possibility: merge the knowledge represented by the static evaluation function with flexibility and generalisation ability of the back-propagation artificial neural network.

The paper shows the used method in details and gives the experiments we have got.

ÖSSZEFOGLALÁS

A back-propagation tanulási algoritmust a mesterséges neuron hálókhoz *Werbos* találta fel 1974-ben, ez egy híres és gyakran használt modell a többszintű hátrafelé tanuló neuron hálókbán. A tanítás során megmutatott be- és kimeneti minta párok alapján nagyon jól képes asszociálni. A betanítás után a háló emlékszik a kimenetre a bemenet alapján, de emellett más érdekes tulajdonságokat is mutat.

Ez látszólag elkülönülő területet alkot a logikai játékoktól. Ebben az esetben mi generáljuk a játék fát, amelyben egy grafikon pontjai szimbolizálja a játék lehetséges állapotait. A játék fa kiértékelésére több módszert ismerünk, úgymint a mini-max algoritmus, α - β metszés és így tovább, de ezek alkalmazása problematikus ha a játék fa nagy. Ilyen esetben használhatunk statikus heurisztikus próba függvényeket, amelyek akkor adnak jó eredményt ha maga a heurisztikus próba függvény is jól meghatározott. Ezek a problémák sugalltak egy másik döntési lehetőséget: vonjuk össze a tudásbázis reprezentációt, a statikus próba függvényt és a rugalmas és általánosításra képes back-propagation mesterséges neurális hálózattal.

Ebben a cikkben bemutatjuk részletesen a módszert és összegezzük a kísérleteink eredményeit.

BEVEZETÉS

A mesterséges neurális hálózat (MNH) kutatása divatos tendencia napjainkban. Nagy feladat az emberi gondolkodást megértése olyan szinten, hogy szimulálhassuk az agy működését. Érdeklődő olvasó ismerhet több mesterséges neurális hálózatot, olyanokat mint a Perceptron, ADALINE, Neocognitron, ART, Boltzmann machine, BAM, és a többi. Szerencsére rendelkezünk a BrainMaker szoftverrel [1] amely a back-propagation tanítási módszerrel alapszik. Ez egy egyszerű be-, kimeneti minta asszociátor, ami rendelkezik néhány számunkra előnyös

tulajdonsággal: könnyű megérteni a működését, képes zajos bemenetet feldolgozni, elég jó eredményt ad eddig még nem ismert tények esetén. A back-propagation háló tanújelét adta annak, hogy használható az eredményekből való jóslásra, kiértékelésre és extrapolációra, de elviseli a rendezetlen vagy helytelen adatokat is.[2]

A mesterséges intelligencia egyik alkalmazási területe: a logikai játékok. Több jól ismert játéknak (pl: go , amőba, Othello, Tic-Tac-Toe) van procedurális értelmezése Assembly, C , Pascal nyelveken.

Mi is rendelkezünk egy tucat amőba programmal ezek közül. A legjobbak minden embert megtudnak verni. De itt felmerül egy kérdés: képes-e a back-propagation neurális hálózat amőbát játszani? Képes-e kiértékelni egy soha be nem tanított helyzetet az interpolációs képességét használva.

A NEURÁLIS HÁLÓ ALAPÚ AMŐBA KONCEPCIÓJA.

Az amőba szabályai: két játékos O vagy X-et ír egy négyzet hálós papíron felváltva. A győztes az lesz aki öt jelet tesz le egymás mellé függőlegesen, vízszintesen vagy átlósan. Lásd az 1. ábrát a következő oldalon.

	X	X	X	X	X	O
			O	O	X	
			O	X		
			O	O		
			O			
			X			

1. ábra. 1. Az amőba játék.

A mi neuronhálós amőbánkban az egyik játékos az ember a O jellel, míg a másik a neuron háló lesz ahol az X jeleket a számítógép használja. De hogyan tanítunk meg egy egyszerű minta asszociatort, amőbát játszani?

A BrainMaker példái között van egy minta a Tic-Tac-Toe-ra. Ez a játék nagyon hasonlít az amőbára de ezt csak egy 3x3-as területen játsszák. A győztes itt az lesz, aki 3 egyforma jelet egy vonalban helyez el. Az alkalmazott tanítási módszer a következő: a bemeneti kép tartalmaz két játék területet. A második terület egy vagy több jellel többet tartalmaz mint az első. A tanítási kimenete (Valós szám) közelít egyhez, ha a plusz jel a jó lépést jelenti., vagy nullához, ha nem.

	X	X	X	X	X
	O			O	
	O	O		O	O

1.0

Bemeneti minta

Kimeneti érték

2. ábra A Tic-Tac-Toe tanítási mintái

Sajnálatos hogy, a Tic-Tac-Toe-ban a alkalmazott bemenet nem alkalmas nekünk, mert ebben az esetben sok a mintában a lehetséges O vagy X jel. A következő részben elfogjuk magyarázni hogyan lehetséges a tanításhoz bemeneti mintákat készíteni és ezekhez a kimeneti párjukat generálni.

A NEUTONHÁLÓ TANÍTÁSÁNAK ELŐKÉSZÍTÉSE

Bemenet: Képi bemenetet fogunk használni, mivel ez könnyebben megfelel a játéknak

Lehetségesek:

1. Az egész területet megjelenítő kép

Probléma: • a tervezett terület itt $20 \times 20 = 400$ jelnek felel meg, így az eredmény 3^{400} különböző lehetséges bemeneti mintát reprezentál

- a tudás amit a tanított háló reprezentál függeni fog az egész játék méretétől

2. A kép méret kisebb a játék területnél

ezt az ötletet a procedurális program környezetek sugallták, amelyek végig pásztázzák az egész játék mezőt mint egy emberi szem és így képezik a bemeneti mintát. A pásztázott terület lehet:

- 9x9-es kockából álló négyszög, egy üres hely a közepén. Ez a minta megmondaná a jóságát a középső helynek, ami a lehetséges következő lépése a neuron hálónak. Ez látható a 3. ábrán.

	X	O		X				O
	O			X		X		O
		O		O				X
			O		O			
	O			?			O	X
	X		O		X	O		
		O		O	X	O	O	O
	X		O			X	X	
X	O			X				O

3. ábra

- Problémák
- $3^{(81-1)}$ lehetséges különböző bemeneti minta
 - 81 bemeneti neuron szükséges

- Előnyök
- a tudás megjelenítés a tanított hálóban nem függ az egész játék méretétől.
 - a játék terület pásztázott része megjeleníthető a BrainMakerben mint egy bemeneti kép és itt módosítható is. Az eredményül adott kimenetet is vizsgálhatjuk egyben.
 - kihasználhatjuk BrainMaker azon képességét, hogy a bemeneti mintában szimmetria adható meg, így készíthetünk hét vagy több bemenetet egy mintából
 - csillagforma amit 4x8-as bemeneti mintába transzformáltunk. A középső helynek a szerepe a csillagban ugyan mint a 9x9 négyzet esetén.

O					O				X
	O							O	
		X			X				
				O	O	X			
O	X	X	O	?	O	O	X		
			O	X	O				
		X			O		X		
	X				X			X	
X					X				O

O		X	O	X	O	X	X
O	O	X	O	O	X	X	O
O	X	X	O	O	O	X	
X	X	X	O	X	X	O	X

4. ábra

- Probléma:
- a 4x8 bemeneti kép, amelyet a játékban csillag formába transzformálunk különbözik attól ahogyan az valós O-X jelekkel a BrainMakerben megjelenik, de ez különbség nem nagy.

- Előny
- csak 32 bemeneti neuronra van szükség, 3^{32} lehetséges különböző bemeneti minta van csak
 - a csillagforma megjelenés az amőbában, tartalmaz minden olyan jelet amely kapcsolatban lehet a kiértékelt középső helyel
 - felhasználhatjuk a BrainMaker szimmetria függvényét, de itt átlós tükrözés nem használható, így csak 3-mal több bemeneti mintát kapunk egy mintából

Kimenet: Szükségünk van több ezer bemeneti és kimeneti párra a hálózat tanításához. A bemenetet az előzőekben megoldottuk. De maradt még egy kérdés. Hogyan vizsgáljuk meg a bemeneti minta jóságát, abból a szempontból hogy jó-e az a következő lépés ami a középső pozíción megjelenik a pásztázot csillagban? Hogyan generáljunk megfelelő kimenetet könnyen és gyorsan? A megoldást a procedurális amöbák adták: az amöba nagyon jól használható értéket adott vissza a saját kiértékelő függvényével, ez lehet a kimeneti értéke a neuron hálónak. Szerencsére rendelkezünk egy Pascal amöbával amit Dudás László kollegám írt korábban hasonló szándékkal. Néhány módosítás után ez a program generálta a tanuló fájlt a BrainMaker számára. Ezzel a trükkel az kiértékelő függvény eredményét tudjuk használni kimenetként a BrainMakerben. A Pascal amöba kiértékelő függvénye egy pozitív egész számot (0-65535) eredményez, ezt egy 16 neuronból álló csoport fogja megtestesíteni, azért mert a neuron háló rossz számológép, így néha $2*2=5$. Ennek a trükknek a segítségével lehetővé tesszük egy nagy hiba lehetőséget a kimeneti neuronoknak ez éppen a fele lehet a 0-1 kimeneti intervallumnak. Az előállított tanuló fájljainkkal körül pásztázzuk az üres helyeket, és az ismétlődő minták eltávolítása után megőrizzzük az értékes és kicsi tanuló fájlt.

A hálózat definíciós fájl: Most mi rendelkezünk tanító és tesztelő fájlokkal, de ez nem elég.

Nekünk még el kell készítenünk egy definíciós fájlt amely megadja a hálózat tulajdonságait és a tanítási folyamatot. A bemeneti réteg 32 neuront a kimeneti réteg 16 neuront tartalmaz. Mi csak egy rejtett réteget használtunk. Sajnálatosan arra nincs szabály hogy hogyan határozható meg a rejtett réteg neuronjainak száma. Érdeklődő olvasókkal szeretném megosztani néhány tapasztalatunkat a következő táblázatban. A kísérletben egy DX4 100 MHz, 256 kByte cache, 12 Mbyte RAM (4 Mbyte RAM drive), 540 Mbyte winchestert tartalmazó számítógépet használtunk.

1. táblázat A tanítási kísérlet paraméterei

Változat	1	2	3	4	5	6	7	8	9	10	11
Rejtett neuronok	48	64	128	256	512	384	448	416	384	448	450
Tanulási idő	24'	20'	125'	17'	∞	20'	22'	21'	4:01'	3:55'	15:20'
Futások	384	381	144	90	∞	66	67	69	299	268	620
Be- és kimeneti minták	388								841		968

A futás megadja a komplett tanító fájl olvasását

A TANÍTÁS NYÚJTOTTA TAPASZTALATOK

- Ha nagyobb a tanító fájl akkor több rejtett neuron szükséges
- A tanítási tény és tanítási idő függvény erősen progresszív
- Ha a rejtett réteg neuronjainak száma kicsi, akkor a rejtett réteget és a kimeneti réteget összekötő súlyok hisztogramja a következő formát veszi fel: . Ha sok a rejtett réteg neuronjainak száma akkor hasonló a hisztogram a bemeneti és a rejtett réteg között. Egy egészséges tanítási folyamat a következő hisztogramot mutatja: de a második hisztogramon van néhány -8-as súly is.
- A sikeres tanításoknál a tanítási folyamat jellemezhető az átlagos hibával, RMS hibával és a standard eltérés függvényével. Ezek az értékek általában nullává válnak a tanítás végére. Az 5. ábra mutat egy példát

5. ábra A hibák függése a futások számától

- A tanítás tűrése 0,3 a tanítási folyamat kezdetén és 0,1 a végén
- Mi teszteltük a tanított hálózatot egy teszt fájlal. Nagyon nagy számú hibás be- és kimeneti párt mutatott a tanított fájl ha a minták száma kevés volt a tanítás során.

MESTERSÉGES NEURÁLIS HÁLÓVAL REPREZENTÁLT TUDÁSBÁZIS BEÁGAZÁSA

MEGVALÓSÍTOTT AMÓBÁBAN

A tanított hálózatot szükséges beágyazni egy programba amely működteti azt. Ez a program használja a RUNTIME.C modult, amelyet a BrainMaker szoftverhez mellékeltek. A program C program nyelven íródott, tartalmazza az amőba ablakát, felhasználói felületet. Megkeresi a lehetséges helyeket, elvégzi a transzformációt az input vektoron a neuron hálós modul számára majd meghívja azt minden egyes helyen és feldolgozza a kimenetet. Végül a legjobb eredmény helyére a számítógép a játék területen kiírja a jelet.

A javított változat lehetőséget ad neuron háló gondolkodásának tesztelésére is.

ÖSSZEGZÉS

A kísérlet végén bebizonyosodott hogy a back-propagation neuron háló alkalmas logikai játékra, olyan mint az amőba. A játék minőségének színvonalát meghatározta a betanított minták száma és hasonlított egy kezdő emberi szinthez. A kísérlet során a legnagyobb probléma a lassú tanulási folyamat volt, ehhez több tucat óra kellet, még néhány ezer minta esetén is.

A tapasztalat eredménye néhány új ötletet ihletett és ezeket fogjuk felhasználni az amőba logikai játék képességeinek növelésére.

IRODALOM

- [1] California Scientific Software: **The BrainMaker Users Guide and Reference Manual**, Grass Valley:CSS,1990
- [2] Jeanette Lawrence: **Introduction to Neural Networks** California Scientific Software, Grass Valley:CSS,1991.

KÖSZÖNET NYILVÁNÍTÁS

A szerzők megköszönik *Dr. Tóth Tibor professzornak*, hogy használhatták a BrainMaker szoftvert.

Dudás László egyetemi docens E-mail:dudaas@iit.uni-miskolc.hu

Ferenczi László doktorandusz E-mail:ferenczi@iit.uni-miskolc.hu Tel:30/585-580

Miskolci Egyetem Alkalmazott Informatikai Tanszék 3515 Miskolc-Egyetemváros