

Idősor-előrejelzés dinamikus neurális hálóval

Égel Zoltán

Házi feladat a Matlab programozás c. tantárgyból
2003/2004 tavaszi félév

A feladat leírása:

Feladatom egy tőzsdei részvény árfolyammozgásának előrejelzése dinamikus neurális hálóval. A feladathoz a MOL Budapesti Értéktőzsdén forgalmazott részvényét választottam, ehhez 1997. január 6-ig visszamenőleg álltak rendelkezésre adataim:

- napi záróárfolyamok: ezek előrejelzését kellene tulajdonképpen megvalósítani
- napi forgalom: mivel a cél egyetlen független idősor előrejelzése volt, ezeket nem használtam fel

Emellett néhány másik idősor előrejelzését is megpróbáltam:

- Egyszerű szinuszos, periodikus idősor
- Mackey-Glass kaotikus idősor
- Napfolttevékenység idősora

A regresszor-vektor, azaz a kialakítandó struktúra megválasztásánál figyelembe kellett venni a feladat néhány specialitását:

1. Gyakorlatilag nem állnak rendelkezésre bemeneti adatok, hiszen csak az idősor megelőző elemeit ismerjük.
2. Az előrejelzés úgy fog történni, hogy a tanított hálót bemenetként nem az idősor tényleges elemeivel, hanem saját kimenetével működtetjük. Emiatt nincs értelme sem a modellezési hibát, sem a háló tanítás alatti kimenetét visszacsatolni a bemenetre, hiszen ezek a fogalmak a tesztelés alatt értelmetlenné válnak: a modellezési hibát gyakorlatilag konstans 0-nak kellene vennünk, így az eredetileg ezzel tanított súlyok nem szólnának bele a kimenet alakulásába, a tanítás alatti kimenet pedig a tesztelés során már megegyezne a kívánt válasszal (mivel ezt is ugyanonnan vesszük).

Így a kialakítandó struktúra az 1. pont miatt nem nevezhető NFIR, a 2. pont miatt pedig NARMAX modellnek. Tanítás alatt leginkább a NARX, tesztelés alatt pedig inkább a NOE modellnek felel meg.

Az implementálás a MATLAB rendszer segítségével történt, elsősorban azért, mert a benne meglévő neurális háló toolbox lehetőséget adott arra, hogy a bonyolult algoritmusokat azok részletekig menő ismerete nélkül is kipróbáljam, és ezáltal ne csak a paraméterek megváltoztatásával alkothassak a problémához esetleg jobban illeszkedő hálót.

A kész program működésének rövid leírása:

A program gyakorlatilag egy FIR-MLP hálót hoz létre, tanítja, majd teszteli. Azért erre a hálótípusra esett a választásom, mert a tapasztalatok szerint erre a feladattípusra jól alkalmazható, struktúrája és működése könnyen érthető.

Az első néhány sor tartalmazza a feladat és a kialakítandó háló paramétereit. Ezek után következik a nyers tanító és tesztelési adatok beolvasása ill. létrehozása. A következő rész létrehozza a tanítandó hálót, inicializálja annak súlyait. Ezután történik a tanítás, végül a kétféle teszt:

- Az első tesztben a továbbra is a tényleges idősor megelőző elemeit adjuk a bemenetre, a háló feladata csupán a következő elem jóslása.
- A második teszt előtt megtörténik a háló átalakítása, azaz a kimenet visszacsatolása bemenetként, a tesztben így a magára hagyott háló előrejelzéseit követhetjük nyomon.

A tesztek indításakor ügyeltem arra, hogy a háló olyan állapotban induljon, mintha előzőleg pontosan a tesztadatok előtti tényleges adatokat dolgozta volna fel, így a FIR-szűrők a teszt szempontjából a lehető legelőnyösebb kezdeti értékekkel indulnak.

Eredmények:

A kialakított hálók viszonylag jól előre tudták jelezni a szinuszos és a Mackey-Glass idősorokat. A „tüskésebb” sorok – a napfolttevékenység és a részvényárfolyam sora – előrejelzése már sokkal kevésbé volt sikeres, bár a napfolttevékenységnél legalább a trend jóslása kielégítő.

A tanítást többnyire a Levenberg-Marquardt módszerrel végeztem annak gyorsasága miatt. Veszélye a túltanulás, ami a **mac_lm3.net** esetében valószínűleg fel is lépett. A momentumokkal kiegészített gradiens módszeres tanítással néhányszor próbálkoztam, sikertelenül. Még valamivel nagyobb méretű háló esetén is messze elmaradt a háló tanulópontokra vett teljesítménye a Levenberg-Marquardt módszerrel tanítottétól. Erre példa a **nap_gdm.net**.

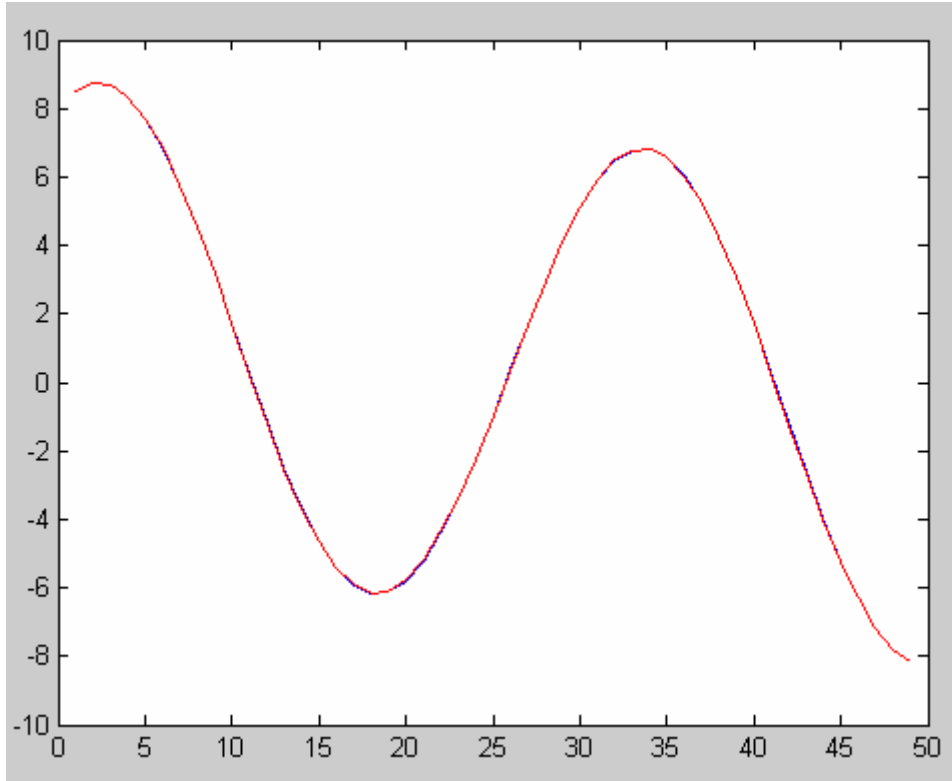
A FIR-ek leírásának értelmezése:

0:x:y azt jelenti, hogy azon réteg FIR-szűrőinek megcsapolásai a 0, x, 2x, ..., y időpillanattal ezelőtti értékeknél találhatók. Ha x nem szerepel, akkor x=1 az alapértelmezés.

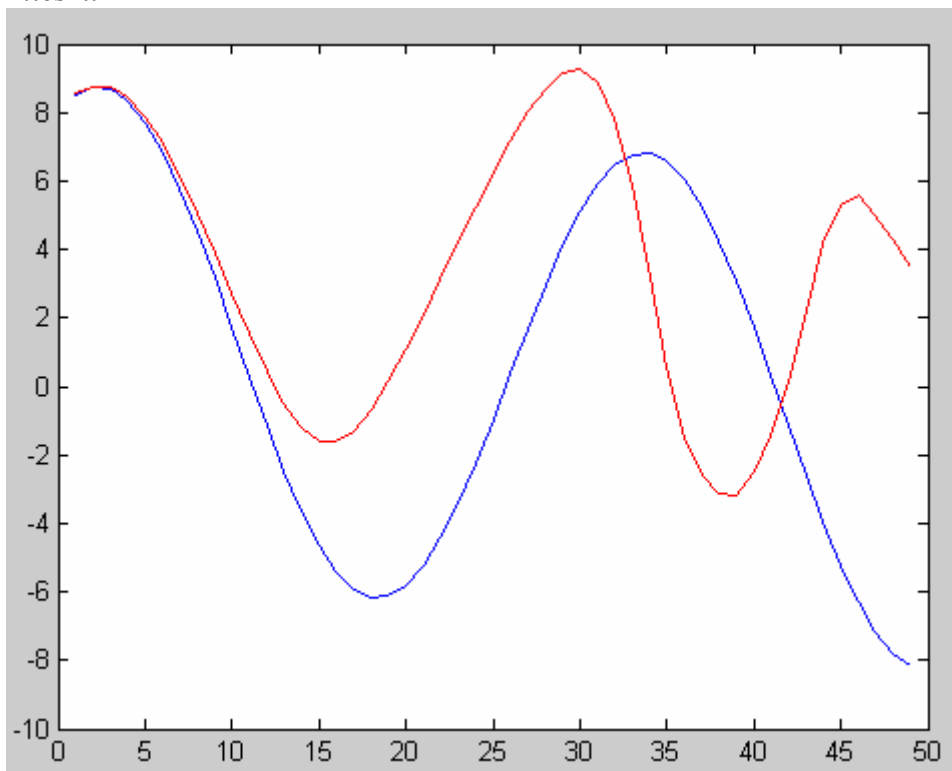
Színuszos sor:

```
sin_lm1.net  
FIRS = {0:5:50 0:10};  
SIZE_OF_LAYERS = [10 1];
```

1. teszt:

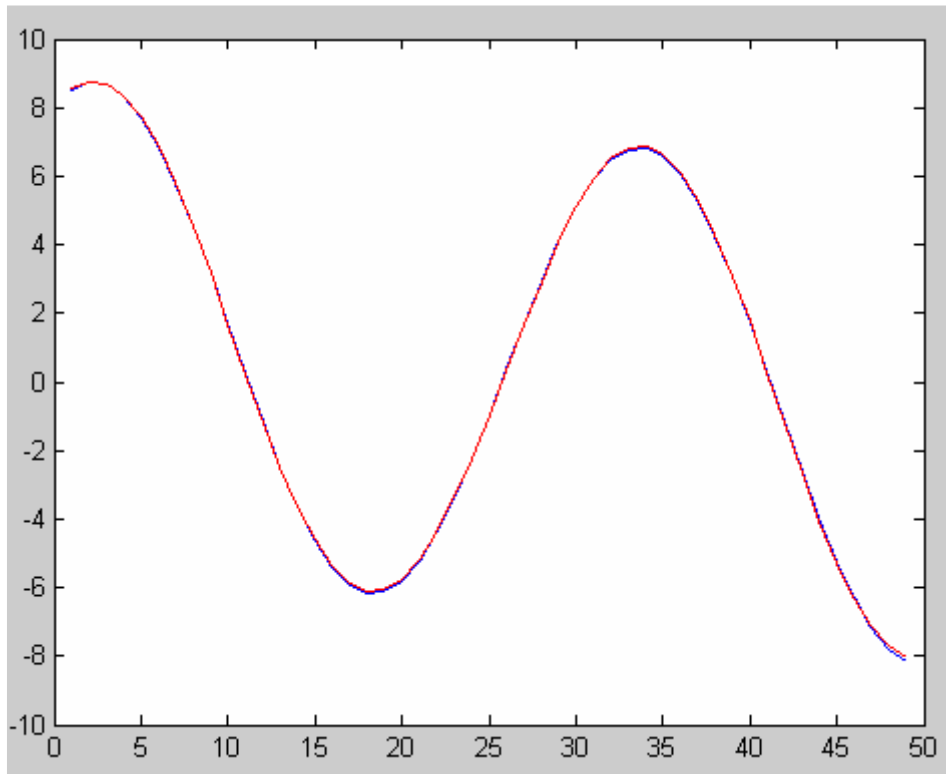


2. teszt:

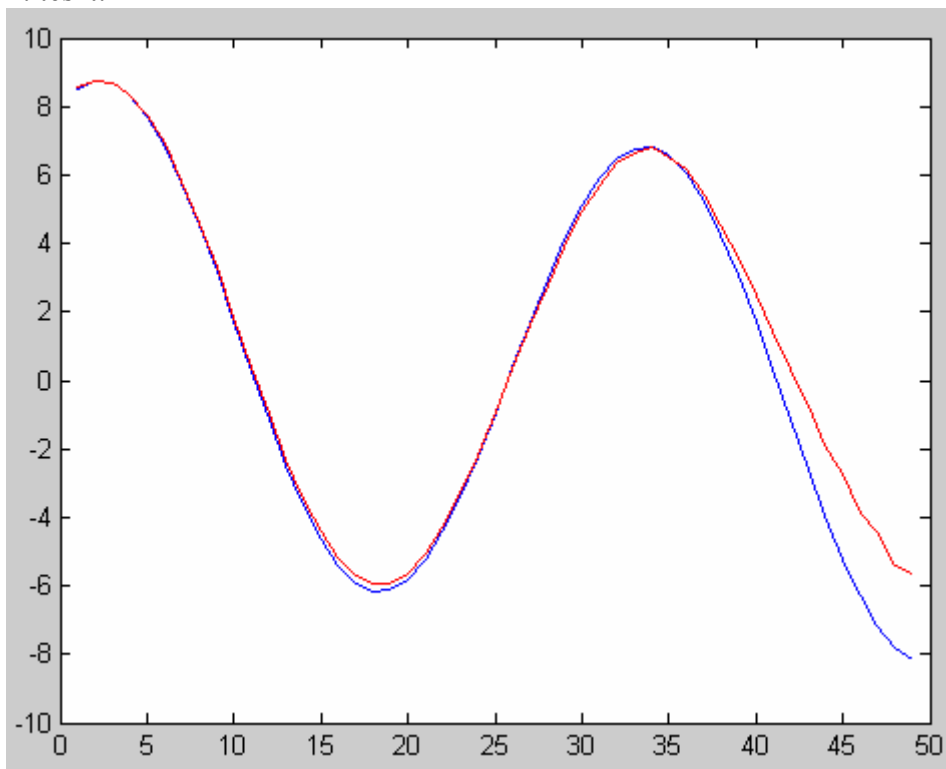


```
sin_lm2.net  
FIRS = {0:30 0:30};  
SIZE_OF_LAYERS = [10 1];
```

1. teszt:



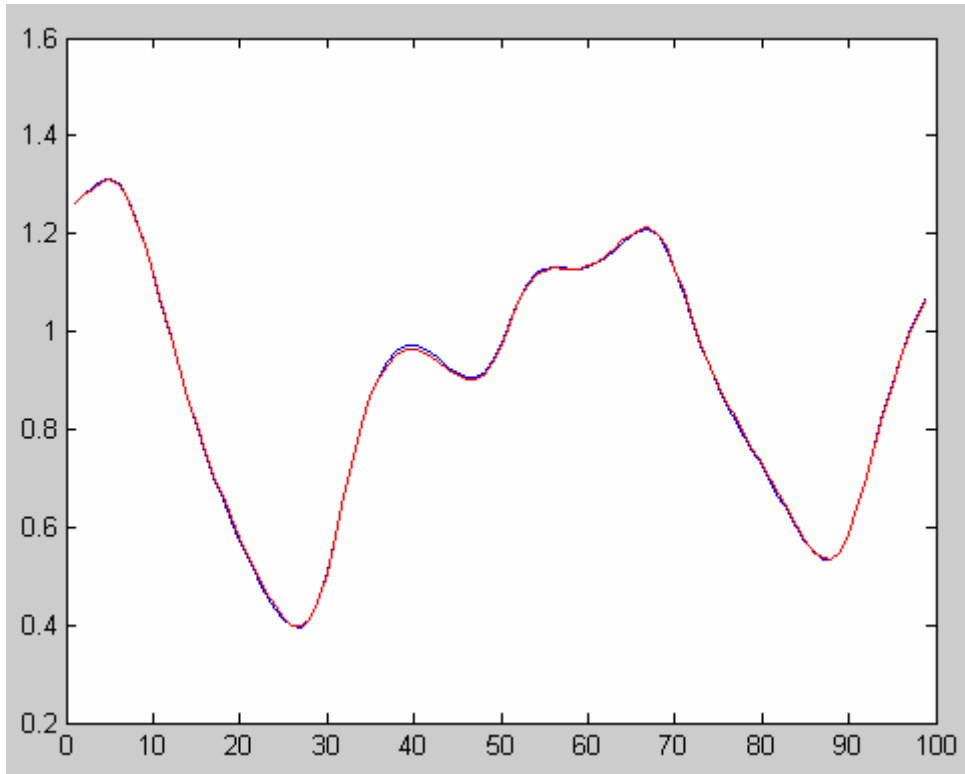
2. teszt:



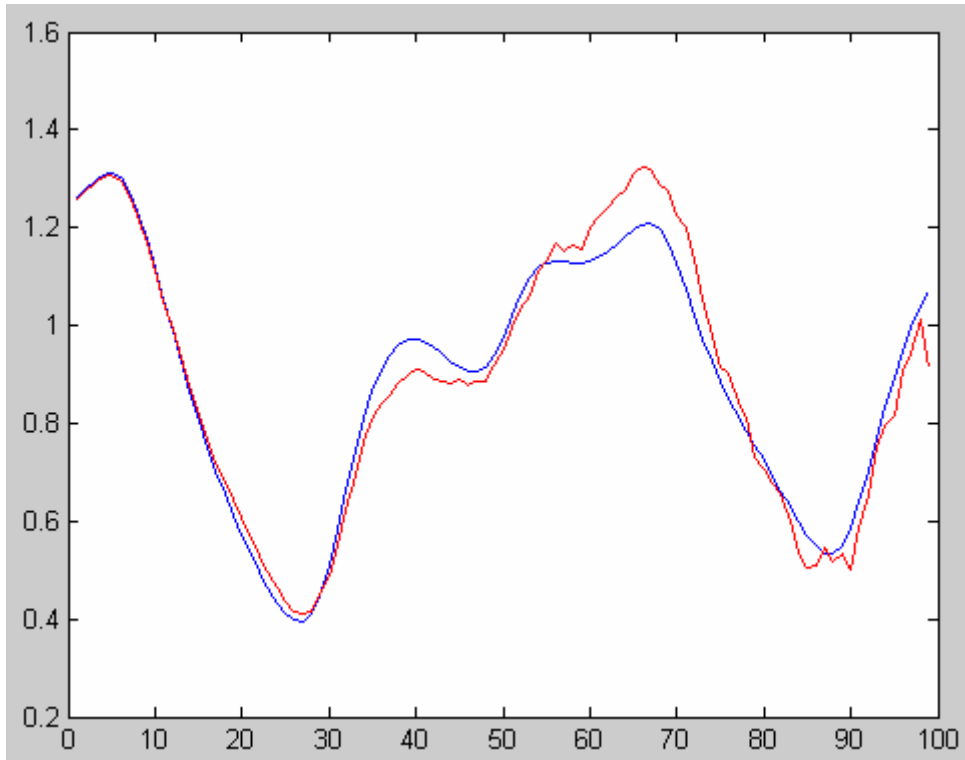
Mackey-Glass idősor:

```
mac_lm1.net  
FIRS = {0:30 0:30};  
SIZE_OF_LAYERS = [10 1];
```

1. teszt:

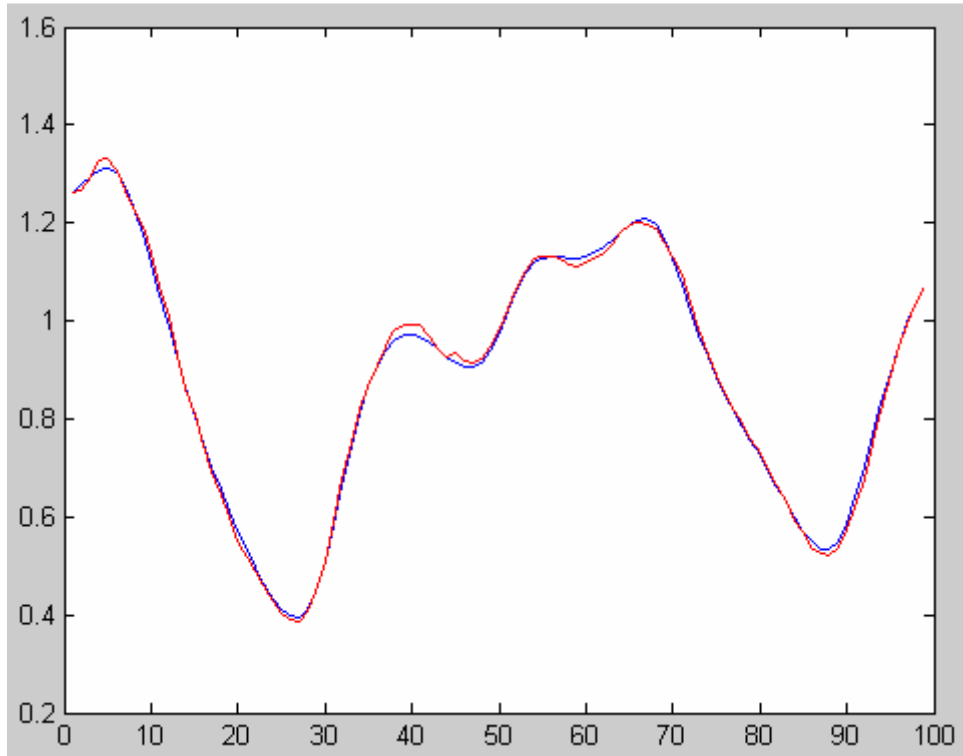


2. teszt:

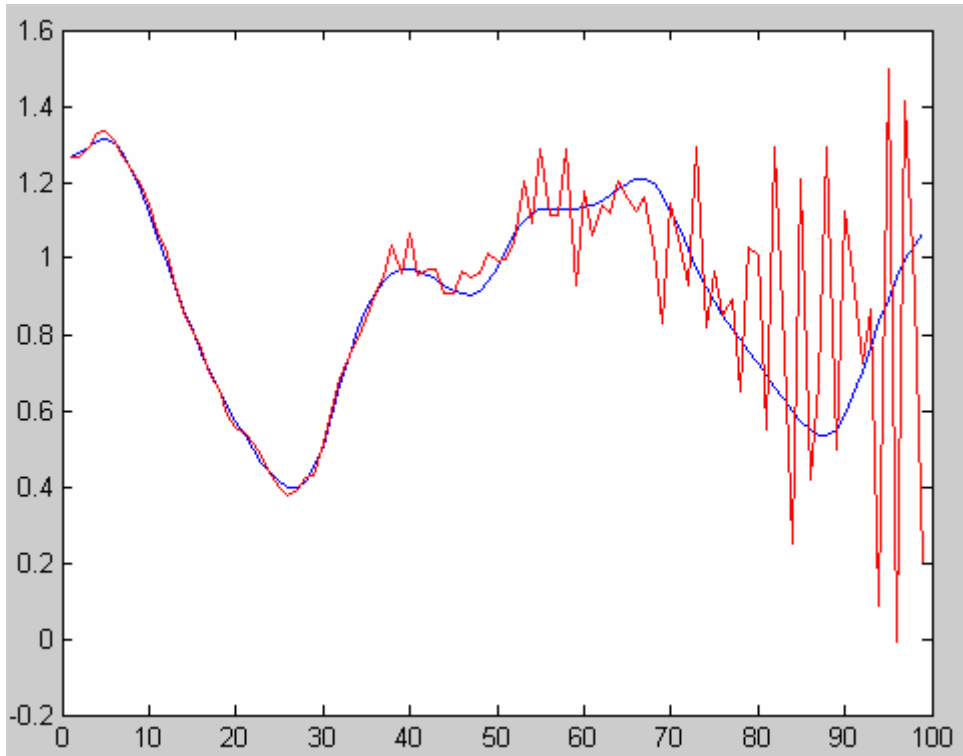


```
mac_lm3.net
FIRS = {0:2:60 0:30};
SIZE_OF_LAYERS = [20 1];
```

1. teszt:



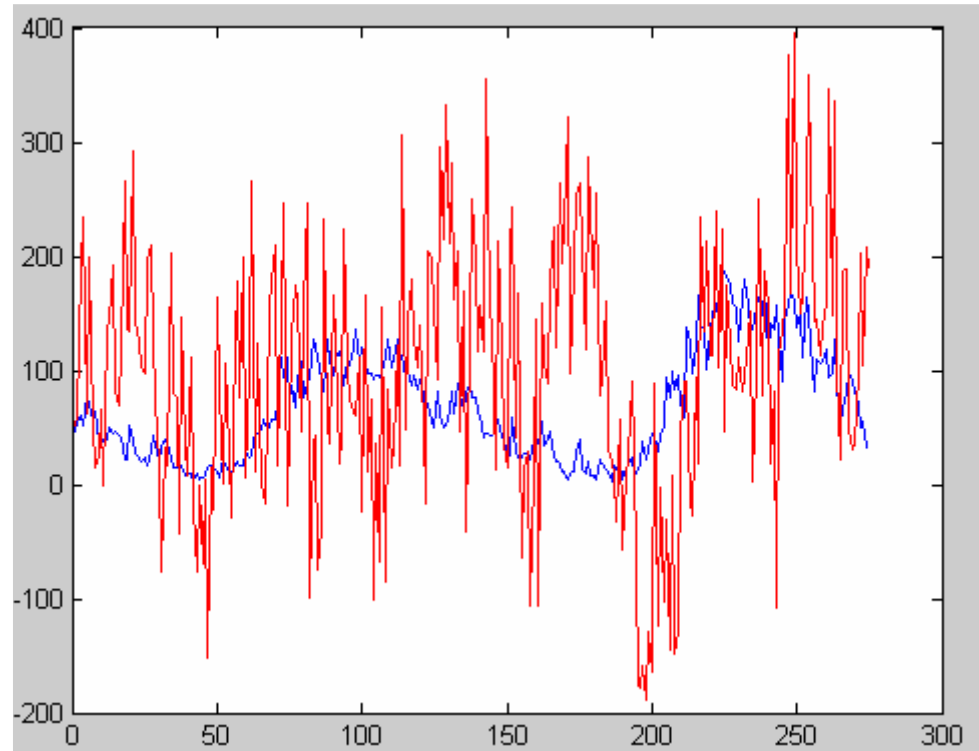
2. teszt:



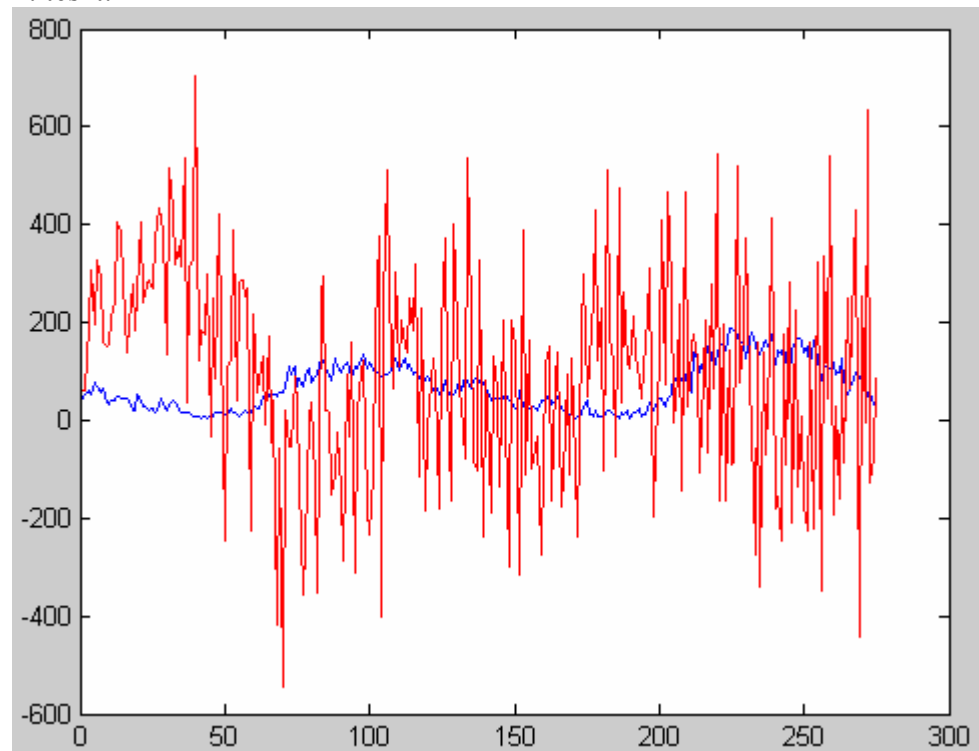
Napfolttevékenység idősor:

```
nap_gdm.net  
FIRS = {0:20:400 0:20};  
SIZE_OF_LAYERS = [70 1];
```

1. teszt:

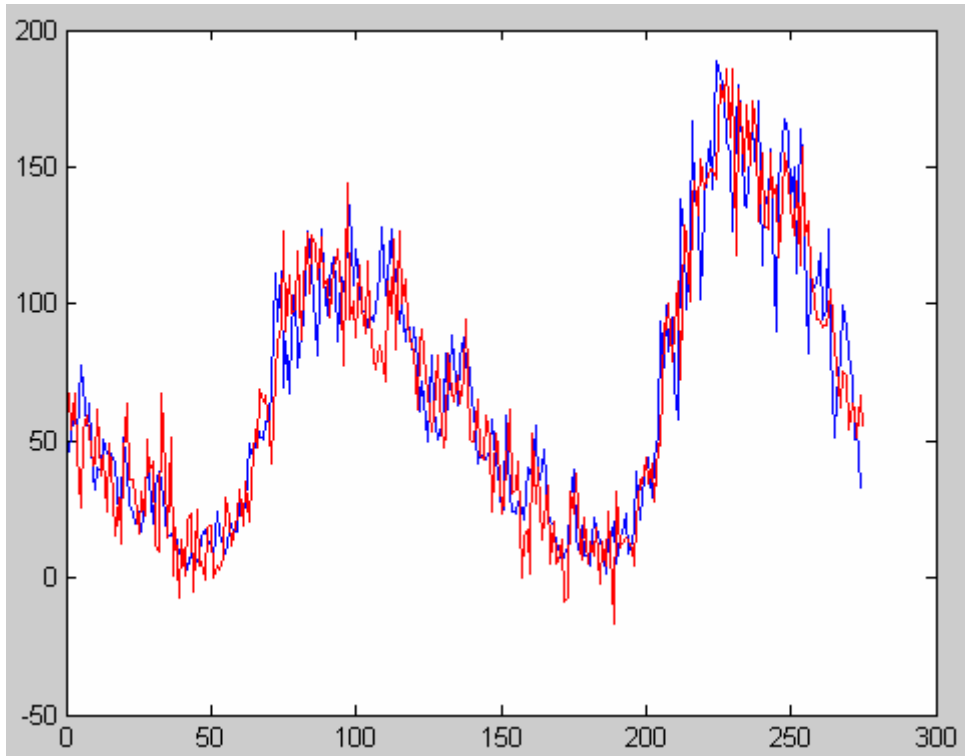


2. teszt:

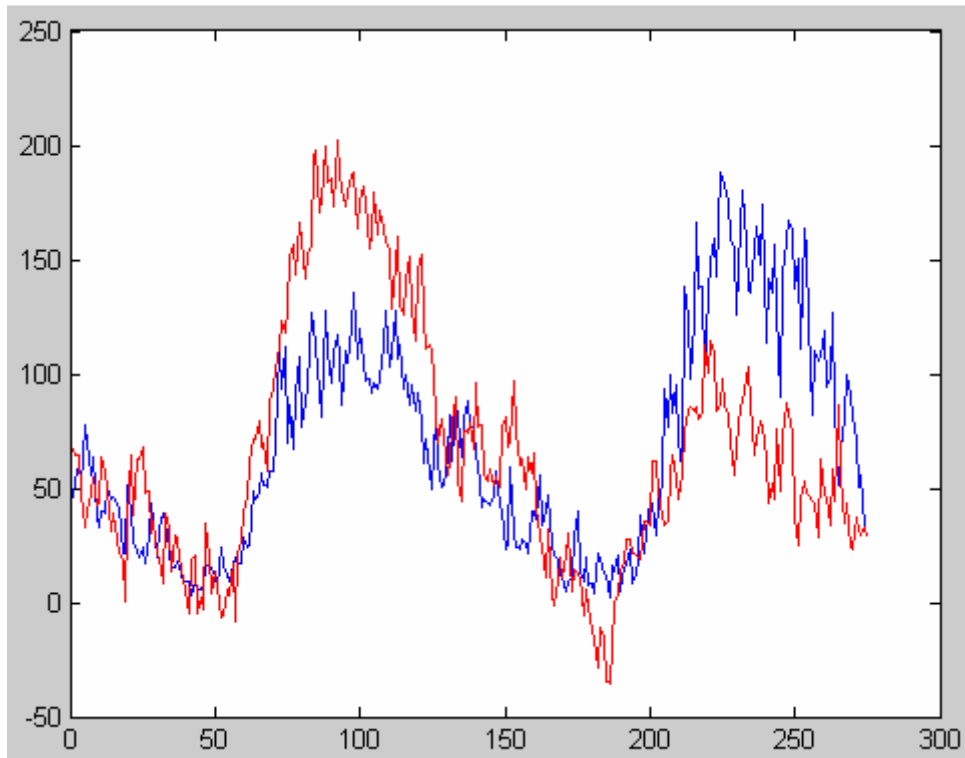



```
nap_lm2.net
FIRS = {0:7:49 0:10};
SIZE_OF_LAYERS = [80 1];
```

1. teszt:



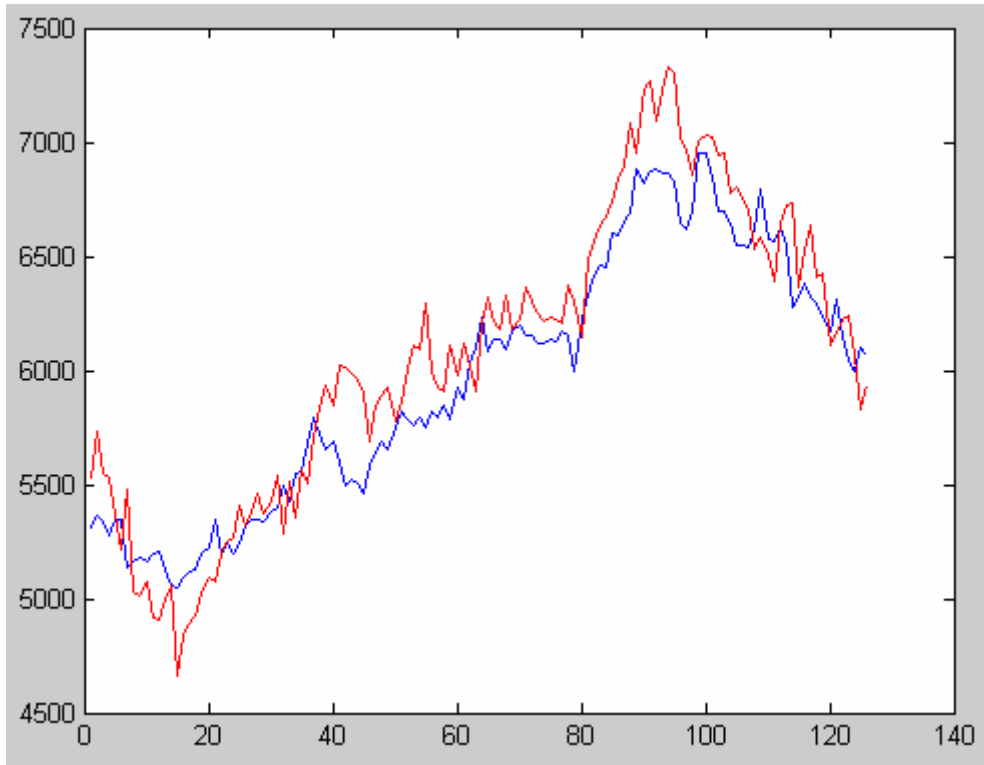
2. teszt:



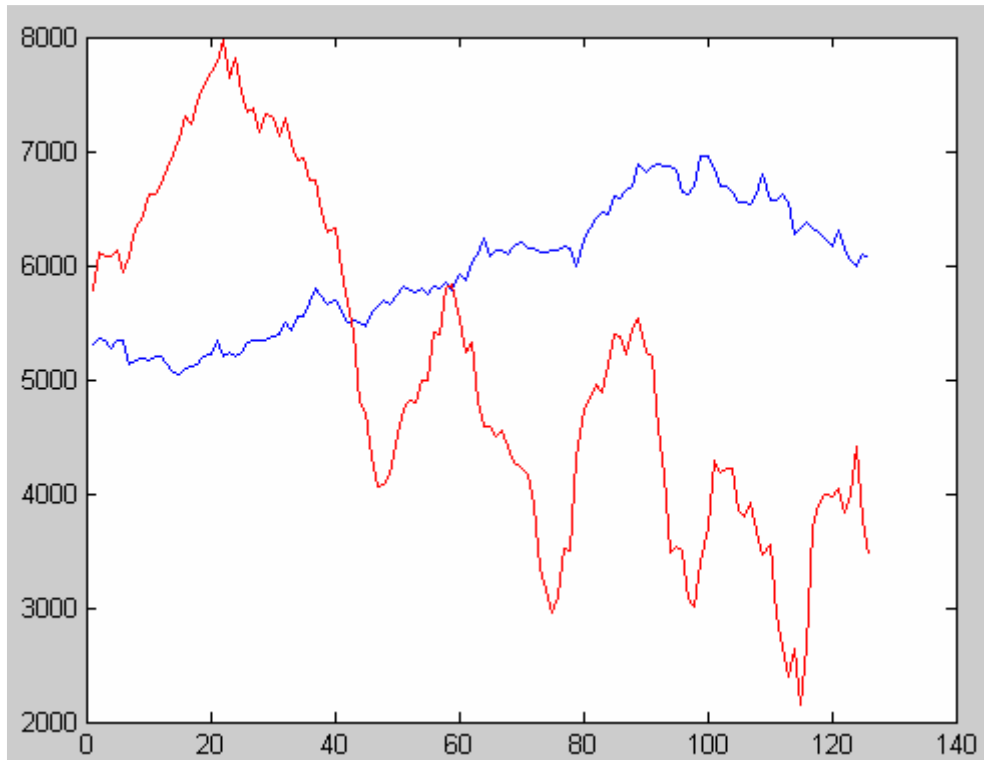
MOL árfolyam:

```
mol_lm1.net  
FIRS = {0:3:90 0:20};  
SIZE_OF_LAYERS = [50 1];
```

1. teszt:



2. teszt:



Annak ellenére kaptam ilyen (és a másik két hálónál is hasonló) eredményt, hogy a tanítópontokat elég jól sikerült megtanítani (`mol_lm1.net`):

