

## BENEDEK GÁBOR

### Evolúciós alkalmazások előrejelzési modellekben – II.

---

A tanulmány első részében az evolúciós elméletek általános tulajdonságait tárgyaltuk. A történeti és elméleti ismertetés mellett bemutattunk néhány közgazdasági alkalmazási lehetőséget is. Most egy részletes előrejelzési és közgazdasági döntési probléma segítségével mutatunk példát az evolúciós eljárások alkalmazására. A első részben a genetikus algoritmussal és a neurális hálózatokkal foglalkoztunk, ezért ezeket az algoritmusokat hívjuk segítségül abban a feladatban, amely a részvényárfolyamok előrejelzésével és az ehhez kapcsolódó optimális portfólió-, illetve opcióvásárlási stratégia kialakításával foglalkozik.\*

---

#### Az algoritmusok alkalmazása: az optimális befektetési stratégia modellje

A következő példában tehát olyan előrejelző modellt mutatunk be, amelyben a neurális hálózatot és a genetikus algoritmust együtt alkalmazva alakítható ki egy optimális viselkedési stratégia. A feladat ismertetésénél adottnak vesszük, hogy az Olvasó ismeri a részvényárfolyamokkal és opcióárazással kapcsolatos alapmodelleket (lásd Hull [1993], Benedek [1999b]).

#### *A probléma ismertetése*

Két problémát ismertetünk. Az elsőben adott egy részvény és erre a részvényre szeretnénk egy 30 napos vételi opciót kötni. A kezdeti időpontban – visszamenő elemzések alapján – adottnak vehetjük a részvény *driftjét* ( $\mu$ ) és *volatilitását* ( $\sigma$ ). Ezek, illetve a kezdeti árfolyam ( $S_0$ ) ismeretében könnyen meghatározhatjuk az opció értékét ( $C$ ) a Black–Scholes-formula alapján. (Feltételezzük, hogy az összes Black–Scholes-feltevés teljesül, tehát a formula alapján számított érték megegyezik az opció árával.) A piacon azonban minden szereplő tisztában van azzal, hogy rövid időn belül (10–15 nap) elképzelhető, hogy változás áll be a részvény volatilitásában, mégpedig meghatározott mértékű növekedés várható. Emiatt az opció mai ára magasabb, mint a Black-Scholes-érték. Tegyük fel, hogy jelen pillanatban vásárolunk egy vételi opciót. A vásárláskor kifizetjük az opció értékét, továbbá azt a prémiumot, amely a részvény volatilitásváltozásának

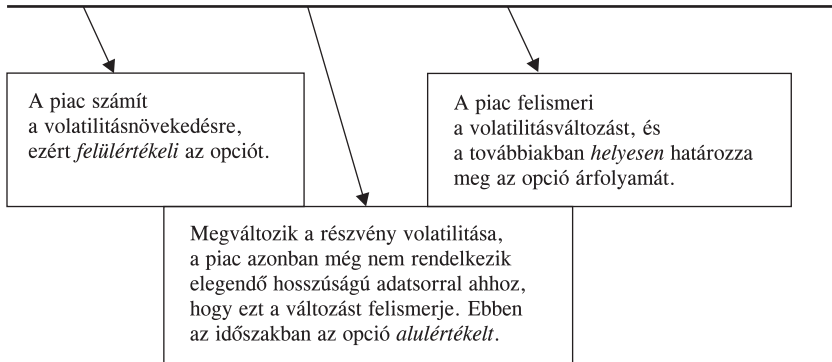
---

\* A szerző köszönettel tartozik *Hideg Évának*, *Lipovszki Orsolyának* és *Pataki Attilának* a tanulmány megírásában nyújtott segítségért és ösztönzésért. A tanulmány a T25372. számú OTKA-kutatás keretében készült.

lehetőségéből származó nyereségből fakad. Ha a futamidő alatt valóban megváltozik a részvény volatilitása, abban az esetben az opció lehívásakor nyereséget, ellenkező esetben veszteséget realizálunk. Természetesen azt, hogy a részvény volatilitása megváltozott, csak utólag tudjuk meghatározni, statisztikai módszerekkel. Az 1. ábra bemutatja, hogyan alakul a piac „véleménye” az opció árfolyamáról.

1. ábra

Az opció árfolyamának meghatározása volatilitásváltozás esetén



Az 1. ábra alapján jól látható, hogy abban az esetben tudunk extraprofitot realizálni, ha hamarabb felismerjük a volatilitásváltozást, mint a piac. Ez csak abban az esetben lehetséges, ha a volatilitás változását megelőző napokban valami olyan „jelenség” figyelhető meg a részvényárfolyam viselkedésében, amely abban az esetben nem jelentkezik, ha a volatilitás változatlan marad. Ráadásul ennek a jelenségnek nagyon kell hasonlítania a megelőző időszak véletlen bolyongásához, továbbá meglehetősen bonyolultnak és statisztikailag felismerhetetlennek kell lennie ahhoz, hogy a piac „ne vegye észre”. (Ilyen lehet például egy kaotikus mozgás.) A befektető számára nem fontos sem az, hogy képlet formájába öntse ezt a jelenséget, sem az, hogy elméletileg alátámassza azt. Egy a fontos: hamarabb felismerni a változást, mint a piac. Első modelünkben ezt a feladatot neurális hálózat segítségével oldjuk meg.

A második feladat továbbviszi az első problémát. Ebben az esetben már a kezdeti időpontban rendelkezünk egy opcióval. Szeretnénk megfelelő fedezeti (*hedge*) stratégiát alkalmazni. Abban az esetben, ha nem változna időközben a részvény volatilitása, a Black–Scholes-féle *delta-fedezeti* (*hedge*) stratégiát kellene alkalmaznunk. Az előző modell neurális hálózata minden periódusban megadja annak a valószínűségét, hogy a részvény volatilitása megváltozott, és így minden periódusban lehetőség van a *delta-fedezeti* stratégia módosítására. Ennek a módosításnak az optimális mértékét genetikusan algoritmus segítségével határozzuk meg.

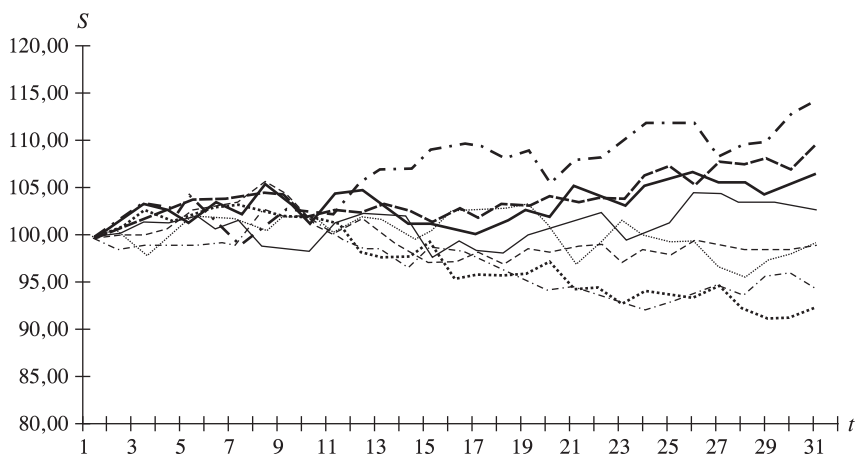
### A volatilitás megváltozásának felismerése statisztikai módszerekkel

A részvényárfolyam-modell alapján a részvények a következő sztochasztikus folyamatot követik:

$$S_{t+1} = S_t \exp \left[ \left( \mu - \frac{\sigma^2}{2} \right) \left( \frac{1}{365} \right) + \sigma \varepsilon \sqrt{\frac{1}{365}} \right], \text{ ahol } \varepsilon \sim N(0,1).$$

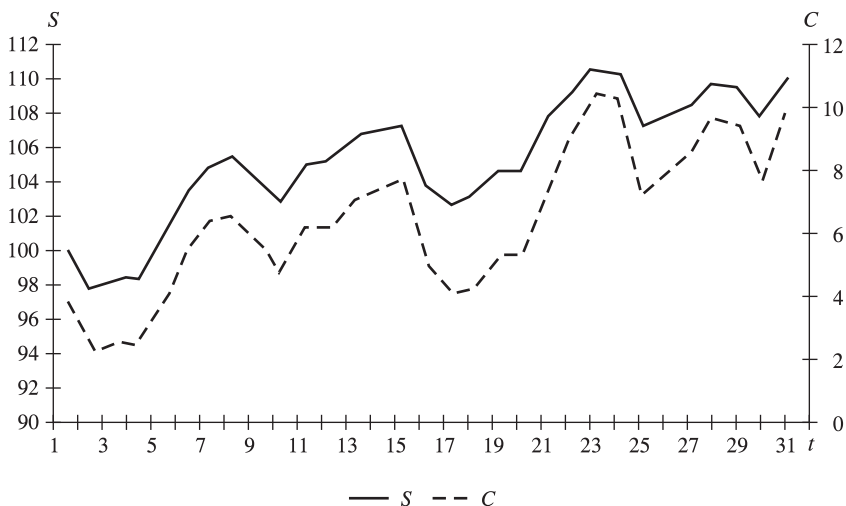
A 2. ábra néhány olyan részvényárfolyamot ábrázol, ahol a volatilitás nem változik a futamidő alatt.

2. ábra  
Részvényárfolyam-lefutások ( $\mu = 0,12$ ,  $\sigma = 0,30$ ,  $S_0 = 100$ )



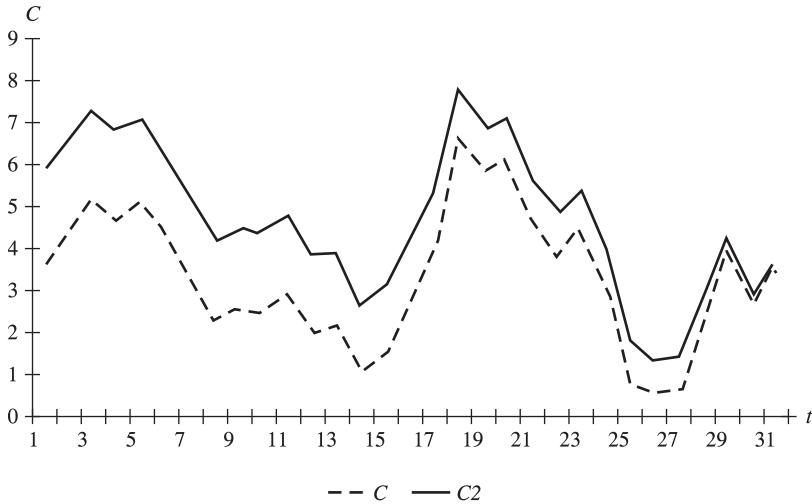
Természetesen a pillanatnyi részvényárfolyamtól függően állandóan változik az opció értéke is. A 3. ábrán egy 30 nap futamidős opció árfolyama és a megfelelő részvényárfolyam látható. A kezdeti pillanatban a részvény árfolyama 100, és a 30 napos opció ára 3,63. Az utolsó periódusban, az opció lejáratakor a részvény 109,80-at ér, az opció értéke tehát 9,80. A köztes periódusokban a Black–Scholes-képlet alapján a hátralévő futamidő és az aktuális részvényárfolyam függvényében ábrázolható az opció árfolyama.

3. ábra  
Részvényárfolyam és opcióárfolyam



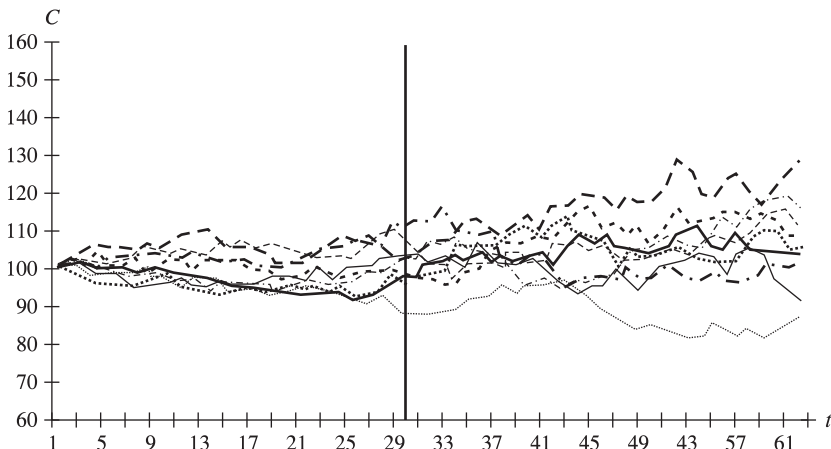
Nézzük mi történik, ha ugyanazt a részvényárfolyam-lefutást két különböző volatilitást feltételezve ábrázoljuk. A  $C$  görbe esetében az előző ( $\sigma = 0,30$ ) volatilitással számoltunk, míg a  $C2$  esetében egy magasabbal ( $\sigma = 0,50$ ). Látható, hogy a futamidő elején  $C2$  jóval meghaladja  $C$ -t, és a futamidő vége felé (ahogy egyre biztosabb az opció várható kifizetése) egymásba simulnak. Az utolsó periódusban, a lehíváskor, nyilvánvalóan egyezik az értékük.

4. ábra  
Opcióárfolyamok különböző volatilitásokat feltételezve



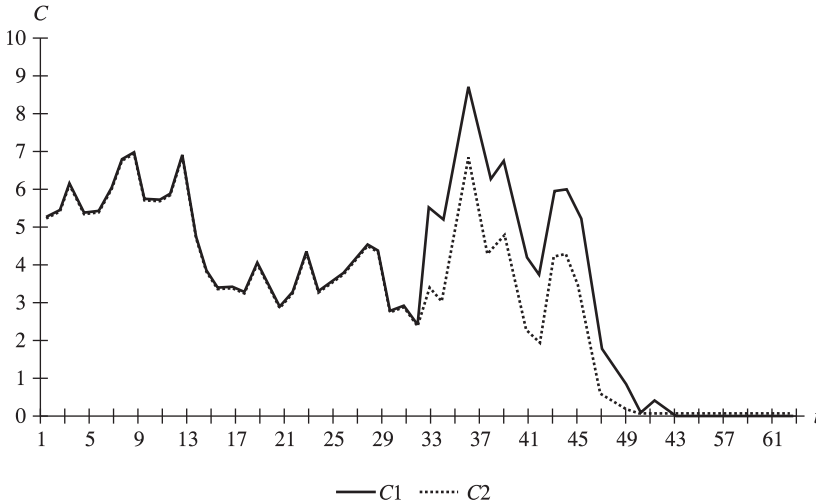
Az 5. ábrán 60 napos időszakon keresztül ábrázolunk néhány részvényárfolyamot, feltételezve, hogy a 30. napon megváltozik a részvény volatilitása. Jól látható, hogy bár a volatilitás közel a duplájára emelkedett, egyszerű ránézéssel igen nehéz eldönteni, hogy valóban megtörtént-e, és ha igen, mikor. Ha pedig a változást megelőző időszakot vizsgáljuk, akkor sem tapasztalunk semmi „rendelleneset”, ami megkönnyítené a változás korai felismerését.

5. ábra  
Részvényárfolyam-lefutások ( $\mu = 0,12$ ,  $\sigma_{0-30} = 0,30$ ,  $\sigma_{31-60} = 0,50$ ,  $S_0 = 100$ )



Nézzünk egy összehasonlító grafikont (6. ábra) arra az esetre, amikor a 60 nap futamidős opciót helytelenül ( $C_1$ ), majd helyesen ( $C_2$ ) árazzuk. Látható, hogy a volatilitásváltozásig egyeznek az árfolyamok, a volatilitásváltozás után a  $C_2$  görbe végig a  $C_1$  görbe felett halad, majd a futamidő vége felé fokozatosan belesimul.

6. ábra  
Különböző opcióárfolyamok a volatilitás megváltozása után



Hogyan lehet eldönteni statisztikai módszerekkel, hogy egy adott periódusban még az alacsony, vagy már a magas volatilitás érvényesül-e. Ehhez először el kell készítenünk a derivált idősort, azaz:  $\Delta S_t = S_{t+1} - S_t$ . Ezek után a derivált idősor átlagát és standard hibáját vizsgáljuk úgy, hogy  $t$  időpontban mindig a  $t, t-1, t-2, \dots, t-n$  időpontbeli mintát vizsgáljuk. A 7. a-c ábrákon ezeket az értékeket ábrázoltuk néhány lefutásra,  $n = 5$  és  $n = 10$  paraméterbeállításokkal.

A 7. a-c ábrák jól szemléltetik, hogy statisztikai módszerekkel csak jóval a volatilitás megváltozása után ismerhető fel a jelenség. A megváltozás időpontját az egyes ábrákon a vastag függőleges vonal jelzi. Optimális esetben, ebben a pillanatban kellene felismerni magát a változást.

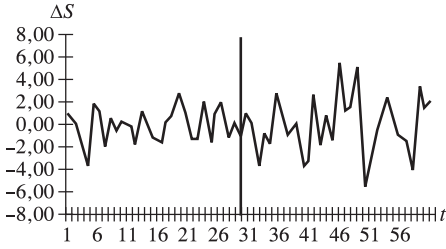
A 8. a-c ábrák egy következő részvényárfolyam lefutást mutatnak. Ebben az esetben hamarabb felismerhető a volatilitásváltozás (körülbelül 6 periódus elteltével, szemben az előző ábrával, ahol 15 periódus után lehetett látványosan elkülöníteni az idősor két szakaszát.)

Végül megjegyezzük, hogy minél rövidebbre választjuk a minta hosszát ( $n$ ), annál hamarabb ismerhető fel a változás, viszont annál bizonytalanabb lesz, hogy valóban a volatilitás változásáról van-e szó, nem pedig csak a véletlen folyamat természetes szóródásáról.

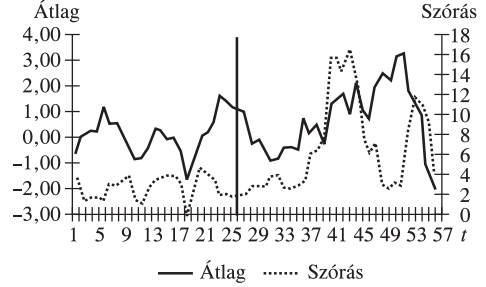
#### *A volatilitás megváltozásának felismerése neurális hálózattal*

Már önmagában annak eldöntése, hogy mekkora  $n$  értéket érdemes választani, illetve milyen küszöbértéket válasszunk a két különböző volatilitás megkülönböztetésére is, nehéz feladat, melynek megoldására eredményesen alkalmazható a neurális háló. Ebben az esetben a neurális háló inputja több különböző  $n$  értékre számolt statisztika,

7. a ábra  
A  $\Delta S$  idősor



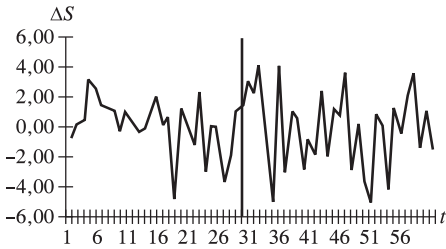
7. b ábra  
Idősori átlag és szórás (n=5)



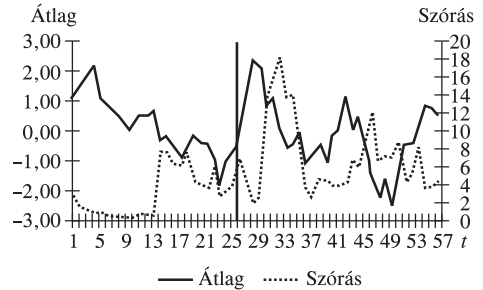
7. c ábra  
Idősori átlag és szórás (n=10)



8. a ábra  
A  $\Delta S$  idősor



8. b ábra  
Idősori átlag és szórás (n=5)



8. c ábra  
Idősori átlag és szórás (n=10)



outputja pedig az, hogy a vizsgált utolsó periódusban magas vagy alacsony volt-e a volatilitás. Mi azonban ennél tovább megyünk. Feltételezzük ugyanis, hogy az időszori adatok között van valamilyen számunkra ismeretlen összefüggés, amely már a volatilitás megváltozása előtt elárulja, hogy változik-e majd a volatilitás, vagy sem. Ezért inputként nemcsak a statisztikákat, hanem a tényleges részvényárfolyamokat is szerepeltetjük. (Megjegyezzük, hogy gyakorlatilag a statisztikákat nem is kellene szerepeltetnünk, hiszen a neurális háló képes ezeket a nem lineáris összefüggéseket automatikusan előállítani, ám ez most felesleges tanulási időt venne igénybe).

A tanuláshoz és a teszteléshez szükséges adatbázist elmúlt időszakokból vettünk. Minden egyes rekord egy időszakot (15 napot) jelent. Ebben az időszakban valamikor megváltozott a volatilitás. Utólagos statisztikai elemzések és más külső információ segítségével<sup>1</sup> megbecsüljük a változás tényleges időpontját. Szükség van olyan idősorokra is, ahol nem volt volatilitásváltozás, hiszen a későbbiekben ezek alapján képes a neurális háló eldönteni egy új adatsorról, hogy abban történt-e változás, vagy sem. Az adatbázis felépítését az 1. táblázat tartalmazza.

1. táblázat

ID	Nap1	Nap2	...	Nap15	Dif1	Dif2	...	Dif14	Átlag	Vari- ancia	Vál- tozás	Idő
001	100,53	98,85	...	105,39	-1,68	-0,57	...	0,83	0,08	9,36	Van	5
002	100,28	99,93	...	86,88	-0,34	0,69	...	-0,76	-1,15	1,85	Nincs	0
003	100,69	100,52	...	107,23	-0,16	1,75	...	-1,71	0,76	5,40	Van	10
004	99,73	100,18	...	108,10	0,45	-2,42	...	1,56	1,12	4,29	Van	11

A neurális hálózat inputváltozói:

Nap1, Nap2, ..., Nap15 – az adott napokon realizált részvényárfolyamok;

Dif1, Dif2, ..., Dif14 – a részvényárfolyamok különbségei;

Átlag ( $n = 3, 5, 10$ ) – az utolsó 3, 75, 10 nap átlagai;

Variancia ( $n = 3, 5, 10$ ) – az utolsó 3, 5, 10 nap varianciája.

Outputváltozó csupán egy van; a változást mutató érték.

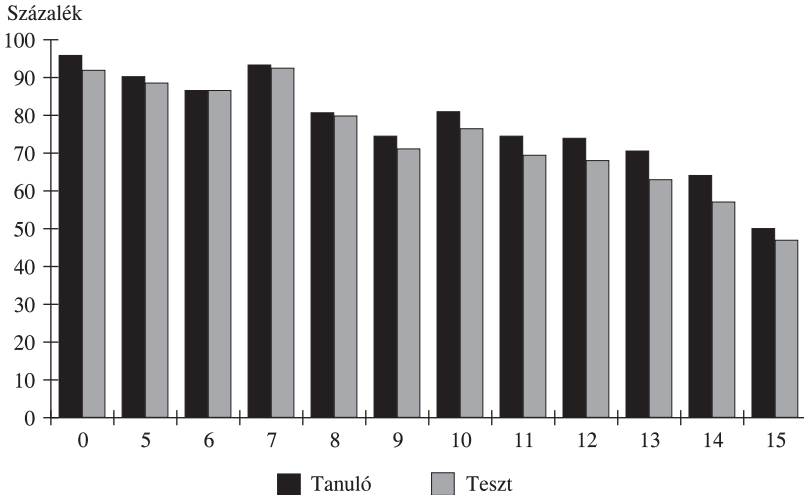
A rendelkezésre álló adatbázist két részre bontottuk; tanuló-adatbázisra (483 rekord) és tesztadatbázisra (483 rekord). Mindkét adatbázisban azonos volt a volatilitásváltozást reprezentáló rekordok száma: 231. Ezek után a tanuló-adatbázison elvégeztük a tanítást és a tesztadatbázison ellenőriztük, hogy a neurális háló valóban az általános összefüggéseket tanulta-e meg. Az eredményeket a 9. ábra mutatja be.

A teljes adatbázisra vonatkozóan a tanuló-adatbázison 85 százalékos, a tesztadatbázison 82 százalékos volt a hálózat teljesítménye. Ez azt jelenti, hogy az esetek 85, illetve 82 százalékában a helyes eredményt állapította meg a hálózat. Az ábrán az is látható, hogy a legsikeresebben azokat a szegmenseket találta el, ahol a volatilitásváltozás régen (15-10) vagy egyáltalán nem (0) történt meg. Jó eredménynek mondható azonban az is, hogy a 3 napnál régebbi volatilitásváltozást legalább 70 százalék eséllyel eltalálja. Ez a – statisztikai eredményeket jóval meghaladó – teljesítmény annak köszönhető, hogy a hálózat felfedezett egy olyan összefüggést, melyet a statisztikai módszerek nem használtak ki.

Ezek alapján látható, hogy a neurális háló segítségével jóval hamarabb és jóval nagyobb biztonsággal tudjuk megállapítani azt, hogy az adott időpontban már magas a volatilitás, vagy még alacsony. A befektető ezek után minden periódusban betáplálja a megfelelő adatokat a neurális háló számára, amely az adatokat kiértékeli és meghatá-

<sup>1</sup> Ilyen lehet például valamilyen, a vállalatra vonatkozó hír megjelenése.

9. ábra  
A neurális hálózat eredményei



rozsa, hogy megtörtént-e a volatilitásváltozás, vagy sem. Abban az esetben, ha igen, akkor a befektető megvásárolja az alulértékelt opciót. Amikor a piac később realizálja a volatilitásnövekedését, az opció ára automatikusan megemelkedik, és a befektető realizálja a nyereséget.

### Fedezeti stratégia genetikus algoritmussal

Tegyük fel, hogy a befektető már az első periódusban rendelkezik egy opcióval, és feladata – az előző példában felvázolt környezetben – olyan fedezeti (*hedge*) stratégia kialakítása, hogy a maximális hozamot érje el a lehető legkisebb kockázattal. Amennyiben a volatilitás nem változna, úgy az optimális stratégia a Black–Scholes-féle dinamikus delta fedezeti stratégia. Jelen esetben azonban a volatilitás megváltozhat. A modellhez azonban felhasználjuk azt a „tudást”, amelyet az előzőekben a neurális háló eredményezett. Minden periódushoz ugyanis hozzárendelhető a neurális háló magas vagy alacsony volatilitást tartalmazó előrejelzése ( $q=0$ , ha még alacsony,  $q=1$ , ha már magas), sőt az is, hogy mekkora ennek a valószínűsége ( $p$ ). Az egyes fedezeti stratégiák egyszerű paraméterekben térnek el egymástól. Az eredeti Black–Scholes-féle delta-fedezeti stratégiában a portfóliókiegészítés mértéke:

$$\text{delta} = \frac{\partial C}{\partial S} = N\left(\frac{\ln(S/E) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}\right),$$

ahol

- $C$  – az opció értékét,
- $S$  – a részvény értékét,
- $E$  – a kötési árfolyamot,
- $r$  – a piaci kamatlábat,
- $T$  – a futamidőt,
- $N(\cdot)$  – a standard normális eloszlásfüggvényt jelenti.

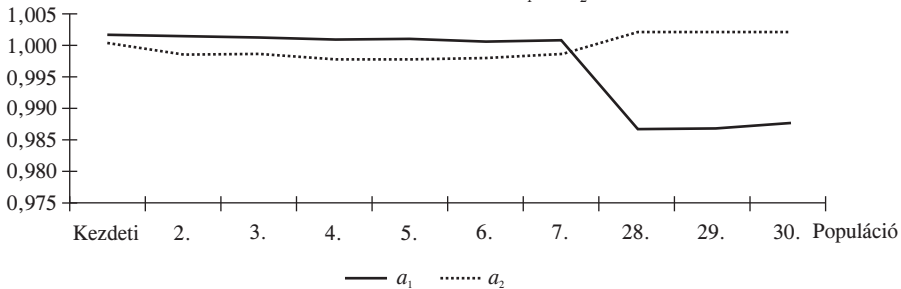


A korrigált modellnek hét különböző paramétere van:  $a_1, a_2, b_1, b_2, c_1, c_2, c_3$ , mégpedig úgy, hogy  $S^* = a_1 S + b_1, t^* = a_2 t + b_2$ , a fedezeti stratégia pedig:

$$\text{delta} = N\left(\frac{\ln(S^*/E) + (r + \sigma^2/2)(T - t^*)}{\sigma\sqrt{T - t^*}}\right) + c_1 q + c_2 p + c_3.$$

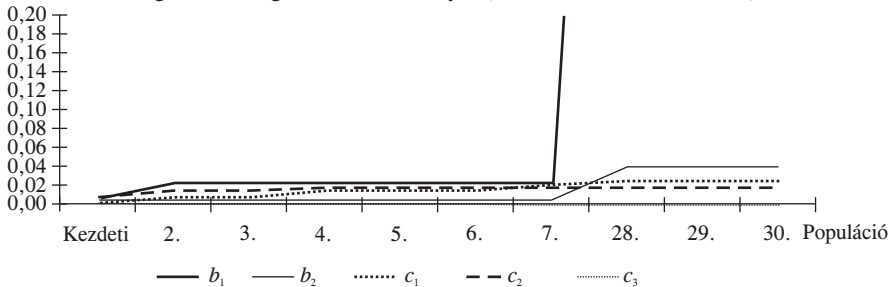
Az értékeléshez a neurális hálózathoz használt tesztadatbázist használjuk. A különböző paraméterbeállítások mellett alkalmazzuk az összes lefutásra a stratégiát, majd kiszámítjuk a kapott nyereségek átlagát és szórásnégyzetét. Ez a számítás gyakorlatilag a genetikus algoritmus célfüggvénye. Mivel az  $a_1 = a_2 = 1, b_1 = b_2 = c_1 = c_2 = c_3 = 0$  paraméterek pontosan a Black–Scholes-féle stratégiát reprezentálják, ezért az algoritmust e paraméterbeállítás környékéről indítottuk. A 10. a–c ábrák a populáció fejlődését mutatják 30 generáción keresztül. Minden generációból egyetlen – a legjobb – egyedet ábrázoltuk (a 7. és a 28. generáció közötti időszakot kihagytuk az ábrázolásból).

10. a ábra

A genetikus algoritmus eredményei (az  $a_1$  és  $a_2$  változók alakulása)

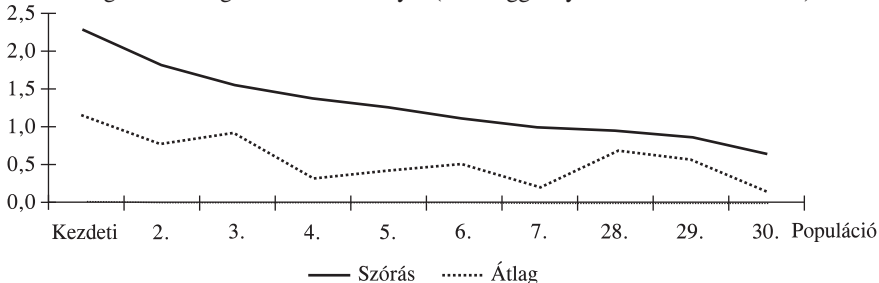
10. b ábra

A genetikus algoritmus eredményei (további változók alakulása)



10. c ábra

A genetikus algoritmus eredményei (a célfüggvények értékeinek alakulása)



Az utolsó populáció legjobb egyedét választva olyan paraméterbeállítást kapunk, amely az eredetileg használt formulánál lényegesen jobb eredményt – magasabb hozamot, kisebb kockázatot – ad.

### Hivatkozások

- AARTS, E.–KORST, J. [1989]: Simulated annealing and Boltzmann machines. Wiley, Chichester, Egyesült Királyság.
- BAKER, J. E. [1985]: Adaptive selection methods for genetic algorithms. Proc. 1<sup>st</sup> Int. Conf. on Genetic Algorithms and their Applications, Lawrence Erlbaum Associates, Hillsdale, NJ.
- BALLA KATALIN–BENEDEK GÁBOR [1999]: Riccati Extended Forms. 6<sup>th</sup> Colloquium on the Qualitative Theory of Differential Equations, augusztus.
- BENEDEK GÁBOR [1999a]: Mesterséges intelligencia az üzleti életben – Marketingakciók hatékonyságának vizsgálata statisztikai és Data Mining módszerekkel. Vezetéstudomány, november.
- BENEDEK GÁBOR [1999b]: Opcióárazás numerikus módszerekkel. Közgazdasági Szemle, 10. sz.
- BIGUS, J. P. [1996]: Data mining with neural networks: solving business problems. McGraw-Hill, New York.
- DAVIS, L. [1991]: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, NY.
- FOGARTY, T. C. [1989]: Varying the probability of mutation in the genetic algorithms. Proc. 3<sup>rd</sup> Int. Conf. on Genetic Algorithms and their Applications, George Mason University.
- FU, L. [1994]: Neural networks in computer intelligence, McGraw-Hill, Inc. Egyesült Államok.
- GREFENSTETTE, J. J. [1986]: Optimisation of control parameters for genetic algorithms. IEEE Trans. On Systems, Man and Cybernetics, Vol. SMC-16, No. 1.
- HIMANEN, V. [1998] Neural networks in transport applications. Megjelent: *Himanen, V. (szerk.):* Ashgate, Aldershot.
- HOLLAND, J. H. [1975]: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI.
- HOLMES, J. N. [1993]: Speech synthesis and recognition. Chapman and Hall, London.
- HOPFIELD, J. J.–TANK, D. W. [1985]: „Neural” computation of decisions in optimization problems. Biol. Cybern., Vol. 52.
- HULL, J. C. [1993]: Options, Futures, and other Derivative Securities. 2. kiadás. Prentice-Hall International, Inc., Engle Wood Cliffs, New Jersey.
- INSTITUTE OF ELECTRICAL AN... [1989]: Neural network based speech recognition systems. (Technical tutorial seminar.) I.E.E.E., Piscataway, N.J.
- KIRKPATRICK, S.–GELATT, C. D. JR.–VECCHI, M. P. [1983]: Optimization by simulated annealing. Science, Vol. 220.
- KOCSIS ÉVA –SZABÓ KATALIN [2000]: A posztmodern vállalat. Oktatási Minisztérium, Budapest.
- KOHONEN, T. [1989]: Self-organisation and associative memory. Springer-Verlag, Berlin.
- MAMMONE, R. J. (szerk.) [1991]: Neural networks: theory and applications. Academic Press, Boston.
- MAREN, A. J.–HARST, C. T. [1990]: Handbook of neural computing applications. Academic Press, San Diego.
- MICHALEWICZ, Z. [1992]: Genetic algorithms + Data structures = Evolution programs. Springer-Verlag, New York, NY.
- PHAM, D. T.–KARABOGA, D. [1998]: Intelligent optimisation Techniques. Springer-Verlag, London.
- SCHAFFER, J. D.–CARUANA, R. A.–ESHELMAN, L. J.–DAS, R. [1989]: A study of control parameters affecting on-line performance of genetic algorithms for function optimisation. Proc. 3<sup>rd</sup> Int. Conf. on Genetic Algorithms and their Applications, George Mason University.
- SIMONOVITS ANDRÁS [1998]: Matematikai módszerek a dinamikus közgazdaságtanban. Közgazdasági és Jogi Könyvkiadó, Budapest.
- WHITELY, D.–HANSON, T. [1989]: Optimising neural networks using faster, more accurate genetic search. Proc. 3<sup>rd</sup> Int. Conf. on Genetic Algorithms and their Applications, George Mason University.
- WINTER, G. (szerk.) [1995]: Genetic algorithms in engineering and computer science. Wiley, Chichester.

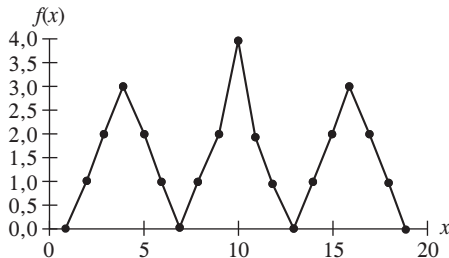
## Függelék

### A genetikus algoritmus konvergenciájának vizsgálata

A következőkben egy rövid és kellően leegyszerűsített feladatot mutatunk be annak illusztrálására, amikor a genetikus algoritmus konvergenciasebessége és minősége kiszámítható. A célfüggvény az *F1. ábrán* látható. A probléma az 1, 2, ..., 19 egész számokon értelmezett. Magát a függvényt azonban nem ismerjük. Az *F1. ábrán előre elárultuk a megoldást* annak érdekében, hogy könnyebb legyen értelmezni az algoritmus működését. A feladat roppant egyszerű, és nyilvánvalóan nem genetikus optimalizálást igénylő probléma, hiszen ábrázolás segítségével (teljes leszámplálással) azonnal meg tudjuk adni az optimális megoldást (10). Mint korábban említettük, az evolúciós eljárásokat – egyéb feltételek mellett – akkor érdemes alkalmazni, ha a lehetséges megoldások száma olyan nagy, hogy lehetetlen a teljes leszámplálás, és/vagy a feladat dimenziója olyan nagy, hogy az ábrázolás lehetetlen. Ennek ellenére a szemléltetéshez maradunk ennél a leegyszerűsített példánál:

1. skalár reprezentációt alkalmazunk, nem binárisat;
2. az induló generáció egyetlen egyedet tartalmaz;
3. az induló generáció meghatározása véletlenszám-generátorral történik, értéke 1, 2, ..., 19 lehet, egyenként  $\frac{1}{19}$ -ed valószínűséggel;
4. a következő generációba kerülő egyedek száma 1;
5. nincs keresztezés (keresztezés ráta: zérus);
6. minden lépésben történik mutáció (mutációs ráta: 1);
7. a mutáció során az adott  $x$  érték 0,5 valószínűséggel növekedik 1-gyel, 0,5 valószínűséggel csökken 1-gyel, továbbá az  $x=1$  esetben biztosan növekedik,  $x=19$  esetben pedig biztosan csökken 1-gyel;
8. a magasabb függvényértékű egyed túlélési valószínűsége 1, az alacsonyabbé 0, emiatt mindegy, hogy melyik szelekciót alkalmazzuk;
9. az algoritmus akkor áll le, ha három lépésen keresztül nincs javulás a függvényértékben.

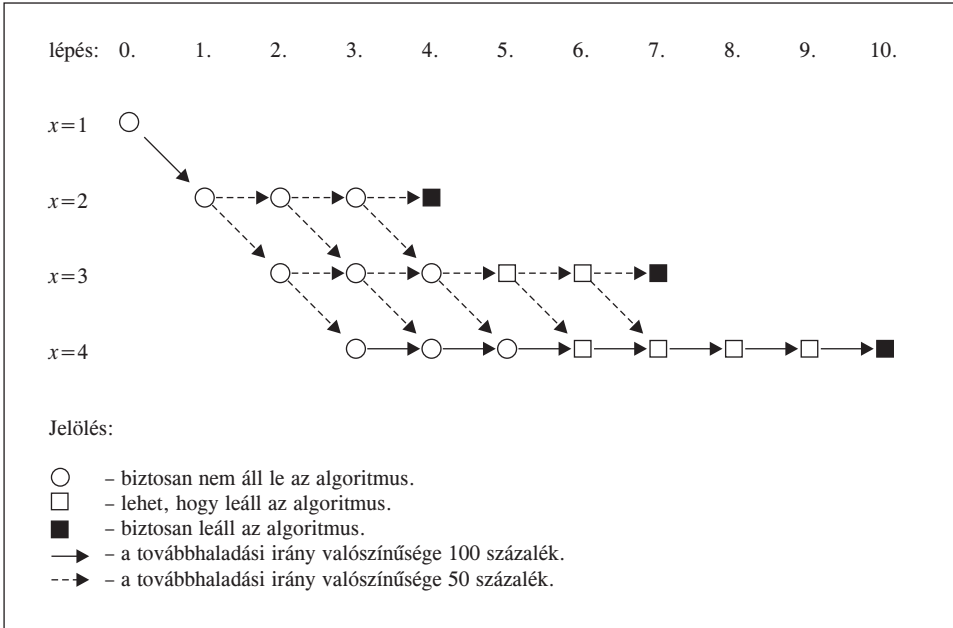
*F1. ábra*  
A célfüggvény



Nézzük, hogyan működik az algoritmus! Tegyük fel, hogy az induló generáció egyedeinek a véletlenszám-generátor a 2-t adja. Az első generáció tehát egyetlen szülőből áll, a 2-esből. A második generáció kialakulása előtt ez az egyed mutációval egy gyereket hoz létre, amely 50 százalékos eséllyel 1, 50 százalékos eséllyel pedig 3. Az előbbi túlélési valószínűsége 0, ezért az első esetben a második generáció továbbra is a 2-es egyedből áll. Az utóbbi túlélési valószínűsége viszont 1, ezért a második esetben a második generáció 2-es egyede kihal, és a 3-as veszi át a szerepét. A harmadik generáció nyilván 2-es, 3-as vagy 4-es lehet. A negyedik generáció szintén, hiszen abban az esetben, ha a mutáció

során a 4-es szülő 5-ös gyermeket eredményezne, a gyermek túlélési valószínűsége 0 lenne, így sohasem juthat tovább az algoritmus 4-nél. Végül leállskor az algoritmus a 2, 3 vagy 4 megoldást adja, mint optimális megoldást. Az *F2. ábra* segítségével könnyen nyomon követhetjük az algoritmus fejlődését az 1-es induló egyedből.

*F2. ábra*  
A genetikus algoritmus fejlődése



Ezek alapján igen könnyű meghatározni az egyes valószínűségeket. Például annak a valószínűsége, hogy az algoritmus a 4. lépésben leáll:  $1 \times 0,5 \times 0,5 \times 0,5 = 0,125$ , annak a valószínűsége, hogy az 5. lépésben áll le:  $1 \times 0,5 \times 0,5 \times 0,5 \times 0,5 = 0,0625$  stb. Az *F1. táblázatban* összefoglaltuk a leállási valószínűségeket.

Az  $f(x) = 0$ -nak megfelelő  $x$  értékeken (1, 7, 13, 19) sohasem áll meg az algoritmus. Az  $f(x) = 1$ -ről, 2-ről és 3-ról – ismervé a megoldást – megállapítható, hogy nem optimális megoldások, ennek ellenére ilyen pontban elképzelhető, hogy megállunk. (Pl.  $x = 1$ -ből indulva 12,5 százalék annak az esélye, hogy az  $x = 2$  – nemoptimális – pontban megáll az algoritmus.) Az  $x = 4$  és  $x = 16$  értékek lokális optimumot képviselnek, míg az  $x = 10$  az egyetlen globálisan optimális megoldás.

Az *F1. táblázat* segítségével az is könnyen kiszámítható, hogy átlagosan hány lépés alatt áll le az algoritmus. Például az  $x = 3$  indulóérték esetén:  $0,125 \times 3 + 0,5 \times 4 + 0,25 \times 5 + 0,125 \times 6 = 4,375$  lépés alatt, továbbá annak a valószínűsége, hogy nemoptimális pontban áll le, 0,125, és annak, hogy lokális optimumban, 0,875. Az *F2. táblázat* összefoglalja az alkalmazott genetikus algoritmus teljesítményét, amelyből kiderül, hogy átlagosan 5 lépés alatt talál megoldást az algoritmus, de ezen megoldásoknak csupán 26,5 százaléka a valóban optimális. A szelekción és/vagy a leállási kritériumon viszonylag kicsit változtatva, természetesen lényegesen jobb eredményeket érhetünk el.

F1. táblázat  
Leállási valószínűségek

$x$	$f(x)$	0.	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	Össze- sen
		lépésben megáll											
1	0	0	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	1
2	1	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	0	1
3	2	0	0	0	0,125	0,5	0,25	0,125	0	0	0	0	1
4	3	0	0	0	1	0	0	0	0	0	0	0	1
5	2	0	0	0	0,125	0,5	0,25	0,125	0	0	0	0	1
6	1	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	0	1
7	0	0	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	1
8	1	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	0	1
9	2	0	0	0	0,125	0,5	0,25	0,125	0	0	0	0	1
10	4	0	0	0	1	0	0	0	0	0	0	0	1
11	2	0	0	0	0,125	0,5	0,25	0,125	0	0	0	0	1
12	1	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	0	1
13	0	0	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	1
14	1	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	0	1
15	2	0	0	0	0,125	0,5	0,25	0,125	0	0	0	0	1
16	3	0	0	0	1	0	0	0	0	0	0	0	1
17	2	0	0	0	0,125	0,5	0,25	0,125	0	0	0	0	1
18	1	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	0	1
19	0	0	0	0	0	0,125	0,0625	0,28125	0,265625	0,1875	0,0625	0,015625	1

F2. táblázat  
Az algoritmus teljesítménye

$x$	$f(x)$	Indítási valószínűség	Átlagos leállás	Nemoptimális leállás	Lokálisan optimális leállás	Globálisan optimális leállás
1	0	0,0526	6,5781	0,2344	0,7656	0,0000
2	1	0,0526	5,5781	0,2344	0,7656	0,0000
3	2	0,0526	4,3750	0,1250	0,8750	0,0000
4	3	0,0526	3,0000	0,0000	1,0000	0,0000
5	2	0,0526	4,3750	0,1250	0,8750	0,0000
6	1	0,0526	5,5781	0,2344	0,7656	0,0000
7	0	0,0526	6,5781	0,2344	0,3828	0,3828
8	1	0,0526	5,5781	0,2344	0,0000	0,7656
9	2	0,0526	4,3750	0,1250	0,0000	0,8750
10	4	0,0526	3,0000	0,0000	0,0000	1,0000
11	2	0,0526	4,3750	0,1250	0,0000	0,8750
12	1	0,0526	5,5781	0,2344	0,0000	0,7656
13	0	0,0526	6,5781	0,2344	0,3828	0,3828
14	1	0,0526	5,5781	0,2344	0,7656	0,0000
15	2	0,0526	4,3750	0,1250	0,8750	0,0000
16	3	0,0526	3,0000	0,0000	1,0000	0,0000
17	2	0,0526	4,3750	0,1250	0,8750	0,0000
18	1	0,0526	5,5781	0,2344	0,7656	0,0000
19	0	0,0526	6,5781	0,2344	0,7656	0,0000
<i>Átlag</i>			<i>5,0016</i>	<i>16,28%</i>	<i>57,15%</i>	<i>26,56%</i>