

Soft computing methods in audit

Lágy számítás alkalmazása ellenőrzés területén

Péczka Polett

Óbudai Egyetem

Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar

Budapest, Hungary

peczka.polli@gmail.com

Abstract — The use of soft computing methods is very versatile. It plays a very important role in artificial intelligence, but it can also be used in much simpler ways. My aim was to show the so-called everyday side of this knowledge in engineering, so I will demonstrate the application of neural networks through a verification example.

Keywords: soft computing, neural network, audit

Összefoglalás — Lágy számítási módszerek felhasználása nagyon sokoldalú. A mesterséges intelligenciában nagyon jelentős szerepet tölt be, de ennél sokkal egyszerűbben is fel lehet használni. Céлом volt, hogy a műszaki életben úgymond mindennapi oldalát is bemutassam ennek az ismeretnek, így egy ellenőrzési példán keresztül mutatom be az alkalmazását a neurális hálózatoknak.

Kulcsszavak: Lágy számítás, neurális hálózat, ellenőrzés

1 BEVEZETÉS

A műszaki élet kistestvére szerintem az ellenőrzés. Mint oly sok területen, a műszaki életben is nagyon fontos a minőség megőrzése. A minőség szabvány szerinti definíciója: „egy tevékenység, folyamat, termék, rendszer vagy személy, vagy ezek bármilyen kombinációja, azon jellemzőinek összessége, amely befolyásolja a képességét, hogy meghatározott és elvárt igényeket teljesítsen.”[1] Ennél szerintem sokkal kézzelfoghatóbb ez a kifejezés. A minőség az előre meghatározott követelményekkel szembeni megfelelés, a felhasználói igény kielégítésével. Azonban nem szabad elfelejteni azt sem, hogy a minőség nagyon szubjektív kifejezés, nem lehet konkrétan meghatározni, hogy egy termék mikor „minőségi”, hiszen függ a felhasználók igényeitől, amely mind más és más. [2] A nagy üzemek kevesebb időt és költséget szeretnek erre fordítani, azonban a vásárlók megtartása érdekében elengedhetetlen egyfajta kompromisszumot kötni. Mindenki hallhatott már az ár-érték arányról, azonban a minőségnek semmi köze nincsen az árhoz, mégis én azt gondolom, hogy egy ilyen vizsgálat során nem szabad megfeledkezni arról sem. Ezért érdemes bevezetni a megfelelés fogalmát, hiszen ez objektíven mérhető. Egy termék vagy megfelelő, vagy nem, nincsen köztes részek. A legkisebbtől kezdve a legnagyobbig sorolhatnám a vállalkozásokat, de egy biztos, mind a profit miatt üzemel. Ennek az eljárásnak, amelyet a következőkben felvázolok,

egy finomított, továbbdolgozott formája alkalmas lenne megkeresni azokat a kritikus pontokat az üzemekben, gyárakban, vállalkozásokban, ahol a minőség romlik. A példamunkámban egy 4 állomásos üzemben, 210 napot, körülbelül fél évet fogok vizsgálni. A feladat végén ki lehet értékelni, hogy melyik állomáson mekkora a hatékonyság, milyen gyakran követnek el hibát. Céloom, hogy egy életszerű, de nem túl bonyolult feladaton keresztül bemutassam a neurális hálózat működését. Sajnos ez a rendszer még nem teljesen automatizált, hiszen maguk az ellenőrzési feladatok még humán erőforrást igényelnek, azonban távolabbi céljaim között tartom számon, hogy magát az ellenőrzési feladatokat is automatizálni lehessen.

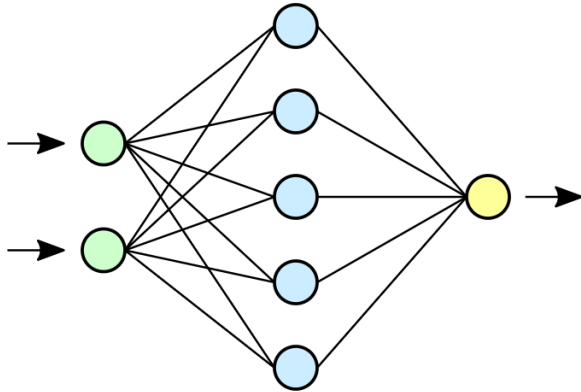
2 LÁGY SZÁMÍTÁSI MÓDSZEREK

A mesterséges intelligencia kutatásában jelentős terület a lágy számítási modell. Alapgondolata a biológiai ismeretek felhasználásán alapuló adatfeldolgozás. A hagyományos számítástechnikával ellentétben közelítő modellekkel dolgozik, tolerálja a pontatlanságot, bizonytalanságot és a részleges igazságot. Elmélete az 1980-as évekre tehető, de mára már számos helyen sikeresen alkalmazzák. [3][4] A számítástechnika fejlődése is segíti a gyors előmenetelét, hiszen az egyre nagyobb teljesítményű processzorok kedvező áron megszerezhetők, nagyon apró mikrochipek is egyre nagyobb számban megtalálhatók a piacon. Az egyre összetettebbé váló rendszereket nagyon nehéz hagyományos vezérlőrendszerekkel irányítani. A lágy számítási módszer kombinálja a Fuzzy logikát, neurális hálózatot és a genetikus algoritmust, így ezek nem versenyeznek, hanem kiegészítik egymást. Kijelenthető, hogy a komplex problémák megoldására célszerű alkalmazni ezen megoldási lehetőséget. [5]

2.1 Neurális hálózat

A fuzzy logikát már több évtizede alkalmazzák komplex szabályozási feladatokra. Ehhez képest a neurális hálók még gyerekcipőben járnak. Ezt a biológiai idegrendszer, az agy inspirálta. A hálózat elemei a neuronok, amelyek össze vannak kötve egymással. Célja, hogy a gépi „gondolkodást” közelítse az emberihez. A neurális hálózatot úgy kell betanítani, hogy a bemenetek ismert halmaza a kívánt kimenetet eredményezze. A rendszernek előre meghatározott tanulási szabályt kell „megtanítani”, ami alapján el tudja végezni a súlyozást is. Előnye, hogy képes kezelni hiányos adathalmazokat is, visszacsatolásoknak köszönhetően minimálisra

csökkenthető a kimeneti jel hibája. Az eredmény attól függ, mennyire pontosak a rendelkezésre álló adatok. Hátránya, hogy esetenként túlzott mértékű képzésre lehet szükség. A neurális hálózatok több rétegből épülnek fel: bemeneti, rejtett és kimeneti. [5]



1. ábra: Neurális hálózatok [6]

2.2 Rekurrens neurális hálózatok

A rekurrens neurális hálózatok memóriája emlékszik, az előző lépésekben kapott eredményekre. Az össze bemenet ugyanazt a feladatot hajtja végre. A független rétegeket azonban átalakítja függő rétegekké. Ekkor az összes rétegre ugyanazt a súlyt adjuk meg, ezzel csökkentve a visszatérő paraméterek és rétegek számát. Ezáltal a neurális hálózat könnyebben memorizálja, tanulja meg az előző kimenetet, hogy a kimenet a következő bemenet lesz. Az összes rejtett réteg összekapcsolható egyetlen ismétlődő rétegbe, hogy a súlyok az összes rétegnél azonosak legyenek. [7]

A rekurrens neurális hálózatoknál fontos a sorrend, annak megfelelően következnek a új bemenetek. Miután lezajlott a belső folyamat a kimeneti jel lesz az új bemeneti jel. A belső folyamatokban játszódhatnak le az előre betáplált szabályok, így vagy a soron következő lesz az új bemenet, vagy ugyanaz, ami volt.

3 FELADAT RÉSZLETEZÉSE

A gépészet területén nagyon gyakran szembe találok magam azzal, hogy nagyon sok plusz költség keletkezik egy-egy tervezési hibából, a rosszul megírt ajánlatból, a gyártáskor, szereléskor pontatlanul, hanyagul elvégzett feladatokból. Sokszor az előttünk rosszul elvégzett munkából adódóan hibázunk, azonban mégis a mi munkánk lesz rosszul elvégezve, ezzel plusz munkát kényszerülünk megcsinálni. Lavinaként működve, egymás hibáiból következnek be a hibák. Egy hibás felmérés magával vonzza a hibás tervezést, ami miatt a gyártásban is bekövetkezhetnek hibák, amelyek majd csak a szereléskor fognak szembe tűnni, például amikor a híddaru beleüt az acélszerkezetbe. A precíz munka nagyon fontos a műszaki területen is, ezért sokszor érdemes lenne több ellenőrzést végrehajtani a munkaállomások között. Egy jó visszaigazolás lehetne arra is, hogy milyen arányokban osszák szét a bónuszt, ami akár motiváló erővel hathatna, hogy jobban működjön a vállalkozás.

Az általam vett példában 4 munkaállomást fogok megvizsgálni, minden nap más állomást. A folyamatot az ajánlatkészítés, felmérés fázissal kezdem el, majd a soron következő a tervezés lesz, utána a gyártás és végezetül a szerelést fogom górcső alá venni. Ha megfelelő a minőség,

a jól működik az a sor, akkor azt egy mátrixban eggyessel jelölöm, azonban, ha hibát találok és nem megfelelőnek minősítem, akkor nulla értéket kap. Ezek segítségével létrehozok egy 210 elemből álló mátrixot, amely egyesekből és nullákból áll. Ez bárhol alakítható, azonban ez fog segíteni abban, hogy el tudjuk dönteni, hogy melyik lépés lesz a következő. Mivel a feladatot rekurrens neurális háló segítségével fogom megoldani, ezért fontos, hogy legyen egy sorrendi mátrix is, amely segítségével fogjuk tudni, hogy melyik munkaállomás a soros. Itt fontos, hogy a második elemmel kell kezdeni a sort, akkor fogjuk kapni az első lépésben a helyes eredményt, tehát ha megfelelő volt, akkor a tervezést, ha nem volt megfelelő, akkor újra az ajánlat, felmérés állomást.

Természetesen ezt manuálisan, kézzel is meg lehet oldani, azonban a lágyszámítási módszerek legfőbb területe a mesterséges intelligencia fejlesztése, így én csak az első pár lépést készítettem el kézi megoldással, hogy szemléletesen be tudjam mutatni, hogy milyen elemekből milyen algoritmus alapján számol a program. A kézi számolás mellett nagyon időigényes, sok hibázási lehetőséget rejt, hiszen egy monoton feladatról beszélhetünk. A MATLAB programot választottam ki, hogy megoldjam a feladatot, hiszen ez a platform remek lehetőségeket nyújt a mátrixműveletek megoldására.

3.1 Mátrixok meghatározása

Mint már fentebb említettem a különböző elemeket mátrixokba rendeztem. Eddig a megfelelőség meghatározására szolgáló naptárról és a sorrendiséget szemléltető mátrixról volt szó. A naptár maga a vizsgálati időszak. A sorrendiségért felelő mátrix (1) első oszlopa a tervezés, második a szerelés, harmadik a gyártás, a negyedik pedig az ajánlat, felmérés.

$$L = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

Ezek mellett szerepel a Kezdeti mátrix, amiben a különböző munkaállomásoknak a „kódját” tanítottam meg a programnak. Mivel 4 különböző állomást kell felprogramozni, így ezek a mátrixok 1 soros és 4 oszloposak. Itt is csak egyeseket és nullákat használtam, az egyesek mindig egy oszloppal el vannak csúsztatva. (2, 3, 4, 5)

$$A = [1 \ 0 \ 0 \ 0] \quad (2)$$

$$T = [0 \ 1 \ 0 \ 0] \quad (3)$$

$$G = [0 \ 0 \ 1 \ 0] \quad (4)$$

$$S = [0 \ 0 \ 0 \ 1] \quad (5)$$

Fő mátrixok soraiba tartoznak a hibamátrixok, amik felelősek azért, hogy a kimeneten melyik értéknek kell majd szerepelnie. Ezeknek már csak két elemük van, ami azért következik be, mert itt két dolgot szeretnénk megfigyelni, megfelelő: nincs hiba (7) vagy nem megfelelő: hibás (6) az adott állomás munkája.

$$H = [1 \ 0] \quad (6)$$

$$N = [0 \ 1] \quad (7)$$

Ezekon kívül, még úgynevezett segédmátrixok fogják előre juttatni a program működését. Az egyik egy egyszerű egység mátrix (8), valamint általam elnevezett igaz (9) és hamis (10) mátrix.

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$I = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \quad (9)$$

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad (10)$$

A mátrixok méretéből is lehet következtetni, hogy mely mátrix melyik másikkal fog az első körben „együttműködni”, hiszen vannak 4 oszlopú és 2 oszlopú mátrixaink. A kiinduláskor a bemenő mátrixok a kezdeti állapot, ebben az esetben az ajánlat készítés valamint a hibamátrix annak megfelelően, hogy volt hiba, vagy sem. A belső rész két részre osztható Először is a bemeneti állapot, egység mátrix és a sorrendi mátrixból keletkezik egy új mátrix, aminek az eredménye egy 1 oszlopú, 8 sorú mátrix, ahol az első 4 elem az aktuális állapot, a következő négy pedig a következő állapot. (11)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \underline{0 & 0 & 0 & 1} & \underline{0} \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (11)$$

Másik része pedig a minőséget, hibát fogja feldolgozni. Mivel a későbbiekben a belső rész összegezve lesz, ezáltal annak is hasonló mátrixnak kell lennie. Itt a nem megfelelő állapothoz tartozó hiba mátrixot és az igaz/hamis mátrixot fogjuk felhasználni. (12)

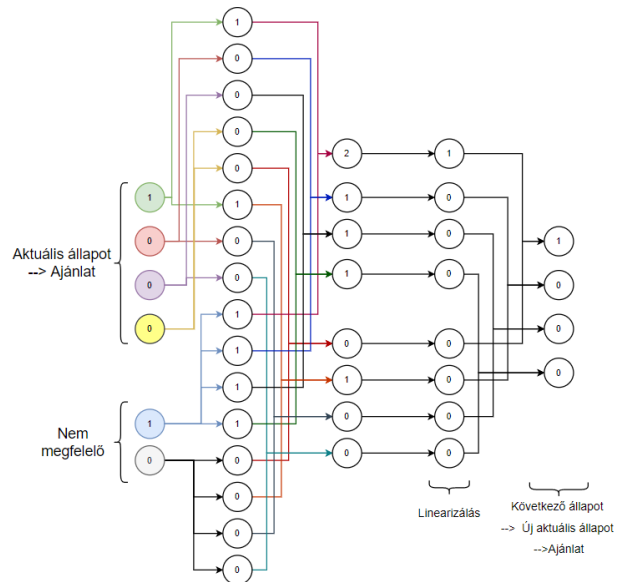
$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ \underline{1 & 0} & \underline{1} \\ \underline{0 & 1} & \underline{0} \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (12)$$

Ezután összegezve a belső rész elemeit, megkapjuk a megoldást. Itt még el kell végezni egy egyszerű linearizálás, hiszen egy helyen az egytől és nullától különböző értéket kapunk. Tehát a maximum érték helyére egyet kell írni, minden más érték benne nulla lesz. Végezetül az egyszerűség jegyében az első négy értéket össze kell adni a mátrix második négy értékével. Itt már összetéveszthetetlenül megkapjuk, hogy mely állomás következnek. Az érthetőség jegyében a következő két ábrán (13):

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ \underline{0} & \underline{1} & \underline{1} & \underline{0} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (13)$$

3.2 Manuálisan elvégzett RNN

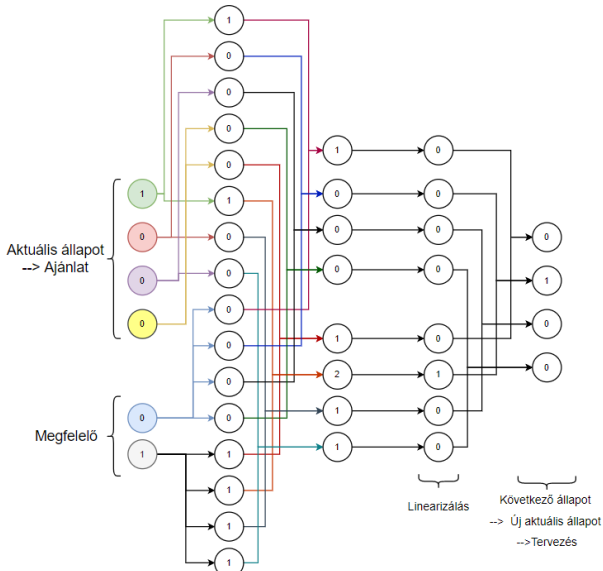
Manuálisan elvégezve is ezt a gondolatmenetet követtem. Ezáltal az első két lépést elvégeztem a MATLAB-ban elkészített program nélkül.



2. ábra: A második állomás megállapítása neurális hálózattal

Már a 2. ábrán is jól megfigyelhető a neurális hálózatoknak a felépítése. Látszik, hogy a kimenet lesz az új bemenet, azzal kell elvégezni a következő napi rutin ellenőrzést. Mivel a második bemenet az mondja, hogy nem megfelelő, ezért a következő bemenetet meg kell

ismételni. Ebből következik, hogy szintén az ajánlatadás ellenőrzése kerül sorra.



3. ábra: A harmadik állomás megállapítása neurális hálózattal

A 2. ábrától eltérően a 3. ábrán a bemenő jelek az ajánlat, hiszen ezt a lépést meg kellett ismételni, és a megfelelő hibamátrix. Az előzőekhez hasonlóan megismételtem a lépéseket. Mivel a hiba jelet megváltoztattam, hogy megfelelt a munkaállomás működése, ezért a kimenet „ugrott”, a tervezés lett, így a harmadik lépésben már a bemeneti jel a tervezés. A MATLAB programom 210 napra kiszámította, hogy mely napokon milyen ellenőrzések voltak. A programozási felületen szintén ezeket a műveleteket végeztem el. Egy for ciklusba számolva másodpercek alatt elvégzi a műveleteket. Ehhez csak a kezdeti értékeket, mátrixokat kell betáplálni, valamint azt, hogy az aktuális napon megfelelő vagy nem megfelelő a működés. A megírt programál már csak a naptár értékeket kell változtatni, és már működik, a többi adat állandó.

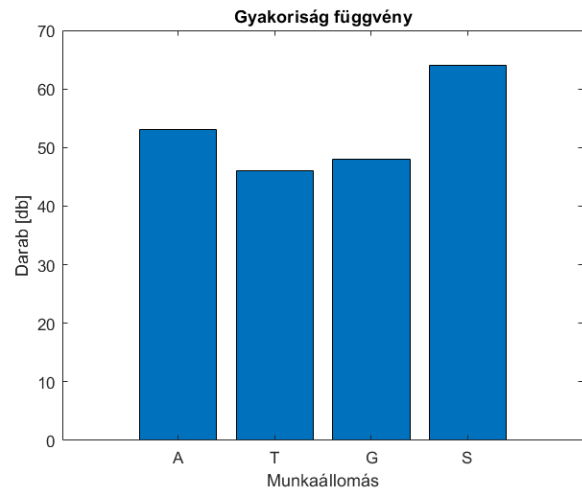
3.3 MATLAB program használata

A feladathoz különböző kiegészítő funkciókat nem használtam, csak a mátrix műveleteket végeztem el, amelyre tökéletesen alkalmas. Mindenképpen a program használatának előnye, hogy hatalmas adathalmaz esetén is nagyon gyors választ kapunk, kevesebb a hibázási lehetőség, egyszer jól megírt program már magától végzi a feladatot, hiszen a szükséges „tananyagot” megtanítottuk neki. Nem mellesleg a mestersége intelligencia ugródeszkája, ezért is fontos, hogy bár nagyon szép ábrák készülhetnek kézi elkészítési módszerrel, szükséges a gépi megoldás. Egyéb előnyei közé sorolható, hogy nem csak magát a hálózatot lehet leprogramozni, de ezeket az adatokat fel lehet könnyedén használni, például diagramok elkészítésére, gyakoriságot számítani, vagy összegezni, hogy mennyi olyan alkalom volt, amikor ténylegesen történt hiba. Az adatokat ki lehet exportálni excel formátumba is, így szélesebb körben fel lehet.

A program első felét lefuttatva csupán megkapjuk azt az adatsort, amely megmutatja, hogy mikor melyik munkaállomás volt ellenőrizve. Tehát a neurális hálózatot lehet lefuttatni. Ezzel az adatsorral tovább lehet dolgozni, egyéb fontos adatokat is ki lehet szűrni belőle.

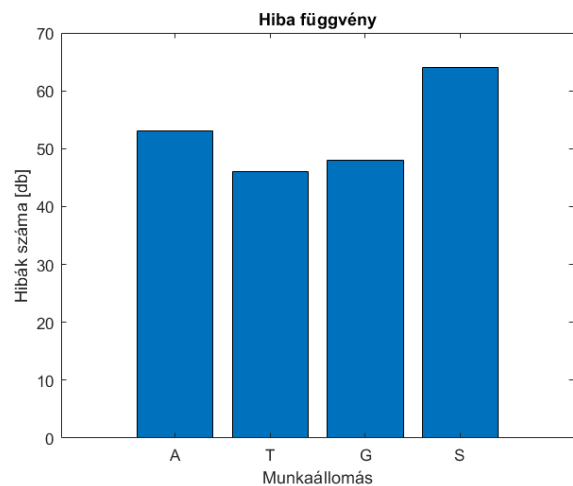
4 KIÉRTÉKELÉS

Az lehet tudni, hogyha 210 napon keresztül zajlanak az ellenőrzések, akkor ideális esetben ajánlat, felmérés és tervezés állomásokra 53 alkalommal esik az ellenőrzés, míg a gyártásra és szerelésre csupán 52 alkalommal. Az ellenőrzéseket egymás után, 3 alkalommal végeztem el, hogy azt is ki lehessen értékelni, hogy mekkora hatással voltak az ellenőrzések a munkaállomások működésére.



4. ábra: Gyakoriság függvény

A 4. ábrán a gyakoriság függvény látható. Ebből látható, hogy a bizonyos állomásokon mennyiszor zajlottak ellenőrzések. Már itt is megfigyelhető, hogy biztosan voltak hibák. Az ajánlat állomáson 45, tervezéskor 48, gyártásnál 48, míg a szerelésnél 64 alkalommal fordultak meg az ellenőrző szerv tagjai. Azonban sajnos ez még nem elégséges információ, hiszen alacsony számok alapja lehet az is, hogy hogy valamelyik állomáson kimagaslóan sok hiba volt, ritkán feleltek meg az elvárásoknak. Természetesen szűrhetünk le belőle adatot, de az csak sejtésként kezelhetjük. Ami ténylegesen megadja az információt, az az, hogy a neurális hálózatban mennyi olyan alkalom volt, amikor a bemenet megegyezett a kimenettel. Ezt le kellett szűkíteni arra, hogy ez külön kiértékelhető legyen az állomásokra is, ezzel megfigyelve, mennyire működnek hatásosan. Ezt hibafüggvénynek neveztem el és az 5. ábrán szemléltetem a kapott eredményeket.

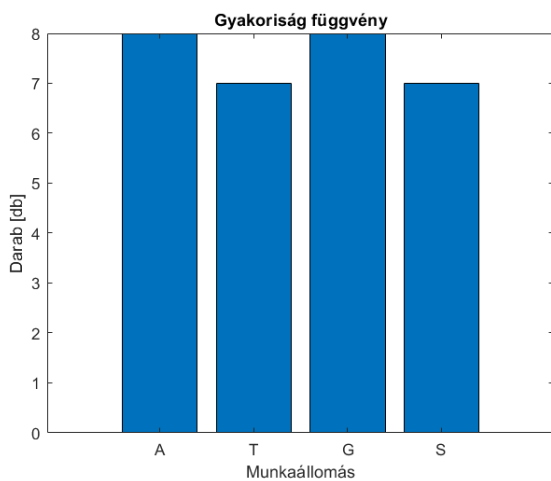


5. ábra: Hiba függvény

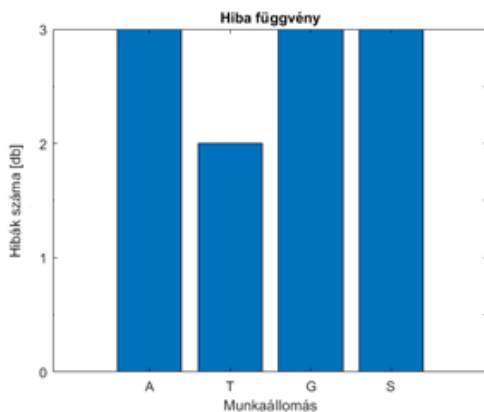
A sejtés bebizonyosodni látszik, hiszen a legtöbb hiba, 38, a szerelési soron történt. Második helyen a nem megfelelési listán az ajánlatkészítés áll, 26-szor kellett megismételni az ellenőrzést, a tervezést egy hajszállal előzte meg a gyártás, a kettő között összesen 3 ellenőrzés volt. Gyártás 22, a tervezés 19 újra ellenőrzéssel végzett ebben a vizsgálati időszakban.

Levonható ebből az a következtetés, hogy összesen 105-ször kellett megismételni valamelyik munkaterepen a folyamatot, tehát 105 alkalommal egyezett meg a bemeneti a kimeneti jellel. A vállalkozás ez által 50%-os megelégedettséggel üzemel. Ha szeretnénk növelni a megelégedettséget, akkor legfőképpen az ajánlat készítést, felmérést és a szerelést kell javítani. A hibáknak a 36%-a szerelésből, közel 25%-a pedig az ajánlatokból származik.

Szeretném megvizsgálni, hogy a következő időszakban, harminc napban sikerült-e fejlődést elérni és javítani a megelégedettséget.



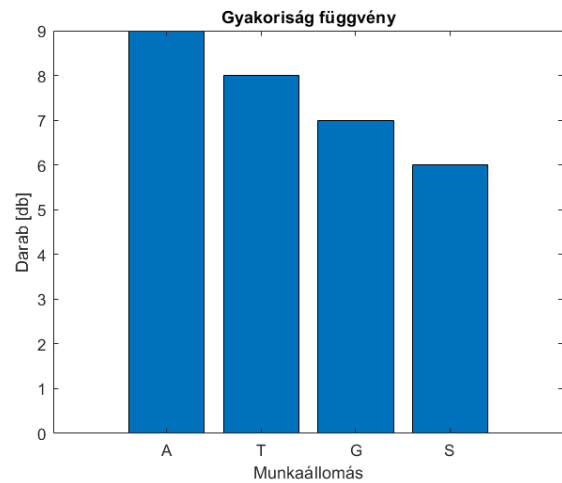
6. ábra: Gyakoriság függvény



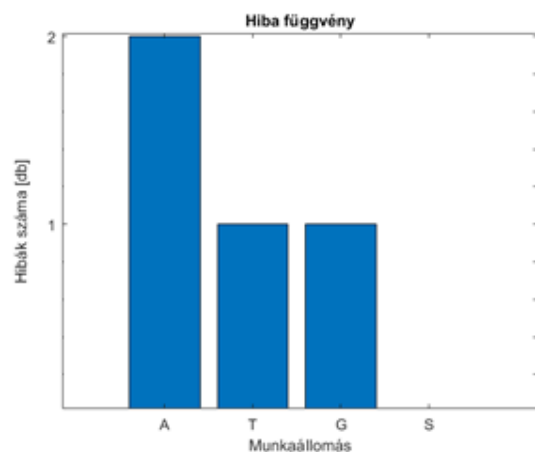
7. ábra: Hiba függvény

Az idő hiányában a különbség az előző ellenőrzéshez képest, hogy csak 30 napot vizsgáltam, valamint a mérésnél is ugyan az volt a vizsgálati tartomány. A 6. ábrán megfigyelhető, hogy voltak hibák, hiszen az első két munkaállomáson 8, míg a maradék állomásokon 7 ellenőrzésnek kellett volna hibátlan működés esetén végbe mennie. A 7. ábra többet mond, hiszen megfigyelhető, hogy a három olyan állomás van, ahol háromszor zajlott le ismételt ellenőrzés, tervezési fázisban pedig csak kétszer. Ez azt jelenti, hogy összesen 11 „hibás nap” volt, így

kijelenthető, hogy ezekben a napokban javult az üzem megfelelése, 36%-os a hibarátá.



8. ábra: Gyakoriság függvény



9. ábra: Hiba függvény

E vizsgálati időpontban is sokat javult a vállalkozás, egyre nagyobb a megfelelés. Itt már körülbelül 15%-os a hibarátá, ami már kifejezetten jónak mondható.

Ez a csökkenés betudható a jutalmak növekedésének, de akár a rövidebb ellenőrzési időszaknak is betudható. Sajnos az idő szűke miatt csökkentenem kellett a vizsgálati időt, azonban szerintem így is látványos, a téma szempontjából fontos, neurális hálózatok gyakorlati alkalmazása az ellenőrzés területén.

5 ÖSSZEZÉS

Célom az volt, hogy az üzemhez tartozó minőségbeli gondok javuljanak, ezzel növelve a megelégedettséget is. A minőség romlása vezethet a profit csökkenéséhez, így az üzemekben nagyon fontos a minőséget ellenőrizni. Különböző eljárásokkal meghatározhatók a gyenge pontok, amiket célszerű vizsgálni. A program fejleszhető lenne még a feladat különböző súlyzó értékekkel, például mennyi plusz költség van egy-egy rossz ajánlat elkészítésekor, vagy mekkora plusz költséggel jár egy nem megfelelő, sok selejtet termelő gyártás. Ezzel a fejlesztett eljárással pontosabb értéket kaphatunk, ezzel párhuzamban növelve a profitot is. Nagy előny lenne, ha az ellenőrzési feladatokban is nagyobb szerepet kapjon az automatizálás. Az automatizálható részeit elkészítettem, azonban szeretnék

a továbbiakban még foglalkozni ezzel a témával, hiszen nagyon sok rejtett potenciál van még benne. Véleményem szerint a neurális hálózatok alkalmazásának területét tovább lehet bővíteni az ellenőrzési feladatok vizsgálatával.

Végezetül szeretném megjegyezni, hogy a neurális hálózatoknak a felhasználási módja nagyon változatos. Nem véletlen ez a mesterséges intelligencia alapja, hiszen az emberi gondolkodáshoz olya hasonlóan működik. Napról napra több lehetőség van, amelyre felhasználható, akár olyan hétköznapi dolgokra, amelyet ebben a cikkben levezettem.

6 KÖSZÖNETNYILVÁNÍTÁS

Ez a cikk az Óbudai Egyetem Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar és a Magyar Fuzzy Társaság támogatásával készült. Szeretném megköszönni Dr. Lukács Judit Tanárnő segítségét és a téma vezetését.

7 HIVATKOZÁSOK

- [1] MSZ EN ISO 8402: Minőségmenedzsment és minőségbiztosítás
- [2] Pietila, G., Lim, T. C. (2012). Intelligent systems approaches to product sound quality evaluations–A review. *Applied Acoustics*, 73(10), 987-1002. DOI: 10.1016/j.apacoust.2012.04.012
- [3] Kóczy T. László, Tikk Domonkos, Botzheim János: *Intelligens rendszerek*, Széchenyi István Egyetem, Győr, 2017
- [4] Fazekas István: *Neurális hálózatok*, Debreceni Egyetem, Debrecen, 2013
- [5] Ibrahim, D. (2016). An overview of soft computing. *Procedia Computer Science*, 102, 34-38.
- [6] https://upload.wikimedia.org/wikipedia/commons/3/3d/Neural_net_work.svg
- [7] <https://hu.photo-555.com/4082704-recurrent-neural-networks-rnn>