
Unitatea de învățare 10.

SISTEME NEURO-FUZZY

Cuprins

<i>10.1. Noțiuni introductive</i>	<i>.....</i>
<i>10.2. Arhitecturi neuro-fuzzy.....</i>	<i>.....</i>
<i>10.3. Câteva tipuri de sisteme neuro-fuzzy</i>	<i>.....</i>
<i>10.4. Arhitectura ANFIS</i>	<i>.....</i>



Introducere

Fiecare tehnică inteligentă are proprietăți tipice care îi conferă aplicabilitatea pentru anumite probleme și prezintă dezavantaje în rezolvarea altora. De exemplu, rețelele neuronale au avantajul recunoașterii modelelor, însă nu pot explica modul în care au ajuns la o concluzie. Pe de altă parte, sistemele fuzzy pot raționa cu informații imprecise și își pot explica raționamentul, dar nu pot achiziționa automat regulile pe care le utilizează pentru luarea deciziilor.

Aceste limitări au fost principalul motiv pentru care s-a încercat crearea de sisteme inteligente hibride, în care două sau mai multe tehnici pot fi combinate pentru a depăși dezavantajele tehnicilor individuale.

Modelarea cu reguli fuzzy, de tip dacă-atunci, este în general potrivită pentru modelarea calitativă. Totuși, în multe situații, datorită incompletitudinii cunoașterii, inerente unor situații reale, această abordare se dovedește insuficientă. De aceea se face apel la unele metode conexioniste, cum ar fi rețelele neuronale. Sistemele neuro-fuzzy sunt astfel capabile să învețe din exemple, să generalizeze pe baza cunoștințelor acumulate și, pe baza datelor primite, să sintetizeze cunoștințe în forma regulilor fuzzy.

În ultimii ani, sistemele hibride au suscitat un deosebit interes. Această abordare s-a dovedit încununată de succes în diferite domenii, precum diagnoza echipamentelor industriale, robotica sau controlul proceselor dinamice. Motivul principal al studierii sistemelor neuronale hibride este crearea de sisteme de reprezentare sub-simbolică, în special a rețelelor neuro-fuzzy. Din punctul de vedere al științelor cognitive, reprezentarea neuronală oferă avantajul omogenității, a capacității de învățare prin exemple și a generalizării, precum și efectuarea de sarcini distribuite, în medii cu date incomplete sau afectate de

zgomote. Din punctul de vedere al sistemelor bazate pe cunoștințe, reprezentările simbolice au avantajul interpretării umane, al controlului explicit și al abstractizării cunoașterii.

Cele două abordări pot fi utilizate în mod complementar. Aceasta este premisa creării de sisteme hibride inteligente, care combină trăsături conexioniste (neuronale) și simbolice (fuzzy). În astfel de sisteme, mai întâi se introduc informații simbolice în rețeaua neuronală, apoi se folosesc exemplele de instruire pentru rafinarea cunoștințelor inițiale. În final, se procesează rezultatul pentru un set de intrări și, prin metode specifice, se extrag informații simbolice din rețeaua antrenată.



Competențele unității de învățare

- *Cunoașterea structurii sistemului de inferență neuro-fuzzy adaptiv și a unor aplicații ale acestuia în proiectarea reguletoarelor fuzzy.*



Durata medie de parcurgere a primei unități de învățare este de 2 ore.

10.1. Introducere

Din momentul în care sistemele fuzzy au fost puse în practică în aplicațiile industriale, proiectanții și-au dat seama de dificultățile care surveneau în mod inerent. Problema găsirii funcțiilor membre potrivite este de multe ori o chestiune de generare de valori și îndepărtarea celor eronate. Aceste soluții sunt greu de găsit, mai ales într-un proces industrial dinamic. De aceea s-a ivit ideea aplicării algoritmilor de învățare sistemelor fuzzy, ceea ce a determinat apariția controllerelor fuzzy adaptabile sau cu auto-organizare.

Sarcinile de lucru complexe pot da naștere la minime locale, de aceea paradigma învățării prin exemple se poate folosi mai mult pentru probleme simple. O rezolvare este împărțirea problemei în sub-probleme, rezolvate de diferite module neuro-fuzzy. În timp ce logica fuzzy asigură mecanismele de inferență în cazul incertitudinii, rețelele neuronale oferă avantajele învățării, adaptabilității, toleranței la defecte, paralelismului și generalizării.

Sistemele neuro-fuzzy moderne au caracteristici care le fac adecvate controlului unor procese industriale: sunt sisteme fuzzy instruite printr-un algoritm de învățare derivat în general din rețele neuronale, pot fi reprezentate printr-o arhitectură de rețea neuronală feed-forward multistrat și pot fi interpretate în termeni de reguli dacă-atunci. Totuși, într-un sistem real, pot exista zeci de semnale de intrare. Un număr excesiv de intrări poate deteriora transparența

modelului considerat și crește complexitatea calculului. De aceea, este necesară selectarea intrărilor relevante și considerarea de prioritate.

Din punct de vedere teoretic, sistemele fuzzy și rețelele neuronale sunt echivalente, deoarece sunt inter-convertibile. Hayashi și Buckley au demonstrat că:

- orice sistem fuzzy bazat pe reguli poate fi aproximat de o rețea neuronală;
- orice rețea neuronală (feed-forward multistrat) poate fi aproximată de un sistem fuzzy bazat pe reguli.

În practică însă, fiecare are avantajele și dezavantajele sale:

Tabelul 10.1. Privire comparativă a avantajelor și dezavantajelor sistemelor fuzzy și rețelelor neuronale.

Sisteme fuzzy	Rețele neuronale
avantaje	
<ul style="list-style-type: none"> • Încorporează cunoștințe predefinite bazate pe reguli • Sunt interpretabile (reguli dacă-atunci) • Oferă o implementare simplă • Cunoștințele sunt disponibile în orice moment 	<ul style="list-style-type: none"> • Pot învăța fără nici o informație inițială • Există mai multe paradigme de învățare, potrivite diverselor situații • Există o multitudine de algoritmi de învățare disponibili în literatura de specialitate
dezavantaje	
<ul style="list-style-type: none"> • Nu pot învăța • Trebuie să dispună de reguli stabilite a-priori • Nu există metode formale pentru ajustarea regulilor 	<ul style="list-style-type: none"> • Sunt „cutii negre” sub-simbolice • Nu pot utiliza cunoștințe a-priori • Necesită algoritmi de învățare complecși • Nu există garanția că învățarea converge spre soluție • Prezintă dificultăți în extragerea cunoștințelor structurale

Sistemele fuzzy prezintă în principal avantajul explicării comportamentului pe baza unor reguli și deci performanțele lor pot fi ajustate prin modificarea acestora. Totuși, achiziționarea cunoștințelor este dificilă, iar universul de discurs al fiecărei variabile de intrare trebuie divizat în mai multe intervale și de aceea, aplicațiile sistemelor fuzzy sunt restrânse la domeniile în care sunt disponibile cunoștințele experților iar numărul variabilelor de intrare este relativ mic.

Pentru rețelele neuronale, cunoștințele sunt achiziționate automat printr-un algoritm (de exemplu, back-propagation), dar procesul de învățare este relativ lent iar analiza rețelei antrenate este dificilă (o rețea neuronală este deseori considerată o „cutie neagră”, în care avem acces numai la intrări și la ieșiri, fără a ști precis ce se întâmplă în interior). Nu este

posibilă nici extragerea cunoștințelor structurale (reguli) din rețeaua antrenată, nici integrarea de informații speciale despre problemă, pentru a simplifica procedura de învățare.

Utilizarea sistemelor hibride neuro-fuzzy minimizează dezavantajele și maximizează avantajele celor două tehnici. Rețelele neuronale sunt folosite pentru adaptarea funcțiilor de apartenență ale sistemelor fuzzy în aplicațiile de control automat al proceselor dinamice. Deși logica fuzzy permite codarea cunoștințelor de tip expert prin termeni lingvistici, de obicei proiectarea și reglarea funcțiilor de apartenență care definesc cantitativ acești termeni este un demers greoi. Tehnicile de învățare ale rețelelor neuronale pot automatiza procesul și deci pot reduce substanțial costul și timpul de dezvoltare a aplicațiilor, conducând și la creșterea performanțelor. Pentru a depăși problema achiziției cunoștințelor, rețelele neuronale pot fi proiectate să extragă automat reguli fuzzy din date numerice. Alte abordări presupun folosirea rețelelor neuronale pentru optimizarea unor anumiți parametri ai sistemelor fuzzy sau pentru preprocesarea datelor de intrare în sistemele fuzzy.

10.2. Arhitecturi neuro-fuzzy

În funcție de modul de combinare a modului simbolic fuzzy cu modulul conexiunist sub-simbolic, există mai multe tipuri de *hibridizări*, care se pot clasifica prin rolul sau „poziția” părților conexiunistă (rețea neuronală, antrenare) și simbolică (inferențe fuzzy).

Rețele neuronale fuzzy

O *rețea neuronală fuzzy* (engl. „fuzzy neural network” - FNN) reprezintă o rețea neuronală care dispune de capacitatea de a prelucra informații fuzzy. Într-o astfel de rețea, semnalele de intrare, ponderile conexiunilor și/sau ieșirile sunt mulțimi fuzzy (termeni lingvistici). De obicei, se utilizează ca modele termeni lingvistici precum „mic”, „mediu”, „mare”, numere sau intervale fuzzy. Structura acestui tip de sistem este:

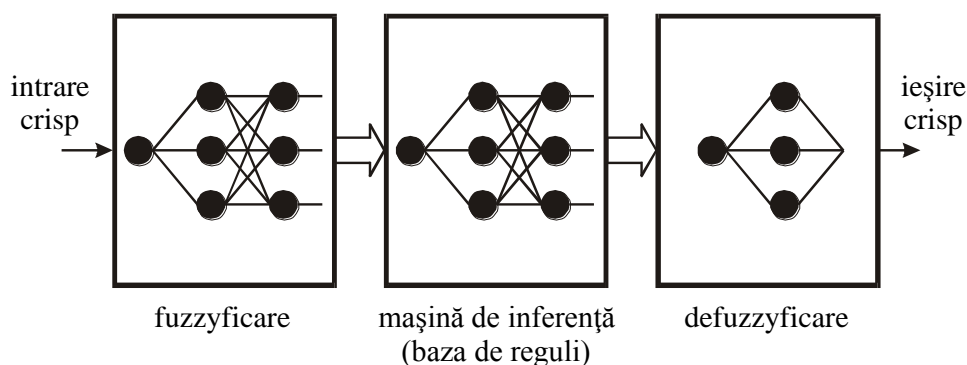


Figura 6.1. Rețea neuronală fuzzy (reprezentare simbolică).

Modulele conexiuniste implementează aici funcțiile unui sistem fuzzy. O rețea efectuează fuzzyficarea datelor stricte de intrare, regulile fuzzy sunt realizate pe baza unei alte rețele, iar operația de defuzzyficare este îndeplinită de o a treia rețea, pentru a furniza datele stricte de ieșire.

Perceptronul fuzzy

În literatura de specialitate există o multitudine de abordări cu privire la implementarea unei rețele neuronale fuzzy, chiar într-o formă foarte simplă, cea a perceptronului multistrat. Detlef Nauck a încercat o sistematizare a acestor propuneri, rezultând o arhitectură generică de perceptron fuzzy, pe care apoi a utilizat-o în sistemul *NEFCON (NEural Fuzzy CONtrol)*. Ideea este identificarea unui model care poate fi inițializat pe baza unor cunoștințe apriori și care poate fi antrenat cu metode specifice rețelelor neuronale.

Perceptronul fuzzy are arhitectura unui perceptron multistrat obișnuit, însă ponderile sunt modelate sub forma unor mulțimi fuzzy. Funcțiile de activare, ieșire și propagare sunt modificate în consecință.

Scopul perceptronului fuzzy nu este fuzzyficarea integrală a unei rețele neuronale de tip perceptron și nici modelarea unui sistem de inferență bazat pe logica fuzzy. Perceptronul fuzzy este asemănător cu perceptronul clasic, utilizat pentru aproximări de funcții. Avantajul său este posibilitatea de a asocia termeni lingvistici cu ponderile sale fuzzy și astfel, rețeaua poate fi creată parțial sau integral pe baza unor reguli fuzzy de tip *dacă-atunci*.

Sisteme fuzzy neuronale

Un *sistem fuzzy neuronal* (engl. „neural fuzzy system”, NFS) este un sistem fuzzy prevăzut cu o rețea neuronală pentru dezvoltarea unor caracteristici precum flexibilitate, viteză și adaptabilitate.

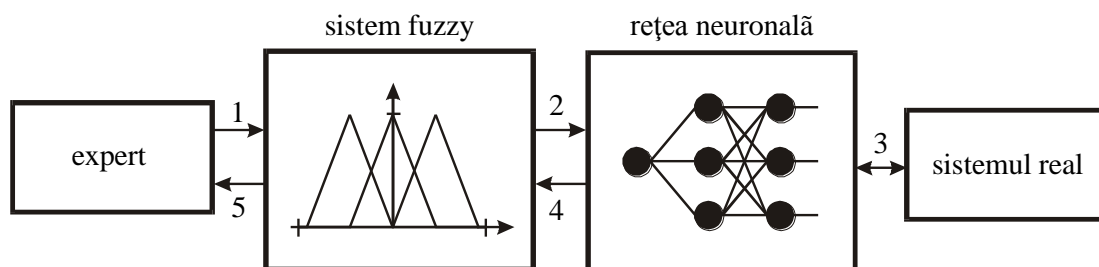


Figura 6.2. Sistem fuzzy neuronal

Cunoașterea de tip expert este convertită mai întâi într-o formă simbolică (1) și apoi utilizată pentru inițializarea rețelei neuronale (2). În continuare, rețeaua este antrenată pe baza intrărilor și ieșirilor unui sistem real (3). Cunoștințele obținute din rețeaua neuronală (4) sunt prelucrate apoi într-o manieră fuzzy pentru determinarea unor reguli de inferență fuzzy, comunicate în final expertului (5).

Aceste sisteme sunt potrivite în special pentru achiziția de cunoștințe și învățare. Ele pot realiza procesul de inferență fuzzy cu ajutorul ponderilor conexiunilor rețelei care corespund parametrilor raționamentului. Folosind algoritmi de învățare cum ar fi back-propagation, aceste sisteme pot identifica reguli fuzzy, pentru ca apoi să învețe funcțiile de apartenență ale inferențelor. Un sistem fuzzy neuronal ar trebui, teoretic, să poată învăța și reguli lingvistice sau să optimizeze regulile deja existente. Există aici trei posibilități:

- Sistemul nu dispune inițial de nici o regulă. El trebuie să creeze noi reguli până când problema de învățare este rezolvată. Crearea unei reguli noi este determinată de un model de antrenare care nu este acoperit suficient de baza curentă de reguli;
- Sistemul pornește cu toate regulile care pot fi create prin partiționarea variabilelor. Pe parcursul antrenării, el elimină regulile inutile din baza de reguli prin evaluarea performanțelor acestora;
- Sistemul are inițial un număr fix de reguli. În timpul învățării, acestea sunt modificate sau înlocuite printr-un proces de optimizare.

Sisteme hibride neuro-fuzzy

Acestea sunt cele mai studiate sisteme hibride la ora actuală. Sistemele hibride neuro-fuzzy sunt o combinație de rețele neuronale și logică fuzzy în care ambele tehnici joacă un rol cheie: logica fuzzy asigură structura în care poate fi exploatată capacitatea de învățare a rețelelor neuronale.

Unii autori consideră că un sistem neuro-fuzzy este definit de următoarele caracteristici:

- este un sistem fuzzy antrenat printr-un algoritm de învățare, derivat din tehnicile specifice rețelelor neuronale;
- poate fi reprezentat printr-o arhitectură de rețea neuronală feed-forward;
- poate fi oricând interpretat în termenii regulilor fuzzy dacă-atunci;
- procedura de antrenare ia în calcul semantica modelului fuzzy pentru a conserva interpretabilitatea sa lingvistică;
- poate realiza cazuri speciale de aproximare funcțională.

În acest domeniu au fost propuse mai multe modalități de combinare a celor două tehnici. În primul rând, rețelele neuronale pot avea un rol de preprocesare. Utilitatea acestui tip de sistem se evidențiază în situațiile în care intrarea în sistemul fuzzy nu poate fi legată direct la o ieșire a unui senzor. Apare deci necesitatea ca ieșirea senzorului să fie preprocesată de rețeaua neuronală și apoi ieșirea rețelei să constituie intrarea pentru sistemul fuzzy.

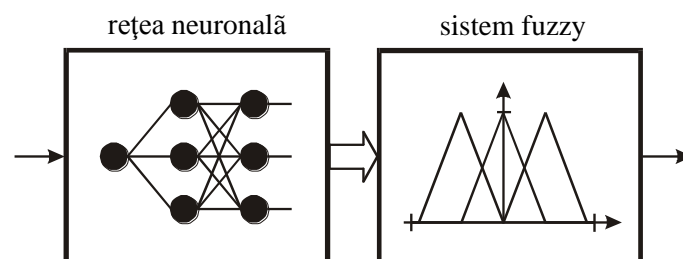


Figura 6.3. Arhitectură de rețea neuro-fuzzy de tip serie (model concurent).

Există de asemenea și sisteme de postprocesare cu arhitectură de tip paralel, atunci când ieșirea sistemului fuzzy nu poate fi conectată direct cu echipamente externe. În acest caz, rețeaua neuronală se constituie într-o interfață care realizează o mapare care ar fi mai greu realizabilă prin tehnici analitice.

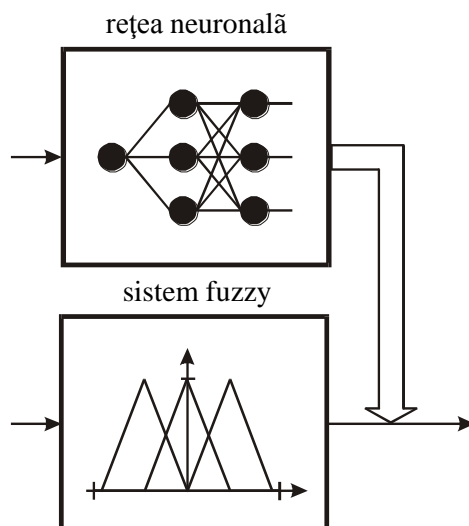


Figura 6.4. Arhitectură de rețea neuro-fuzzy de tip paralel (model cooperant).

Rețeaua neuronală reglează ieșirea sistemului fuzzy pe baza a ceea ce a învățat despre preferințele personale ale utilizatorului din ajustările anterioare pe care acesta le-a efectuat.

O ultimă variantă, poate cea mai importantă, este cea a utilizării rețelei neuronale pentru generarea și reglarea funcțiilor de apartenență ale sistemului fuzzy.

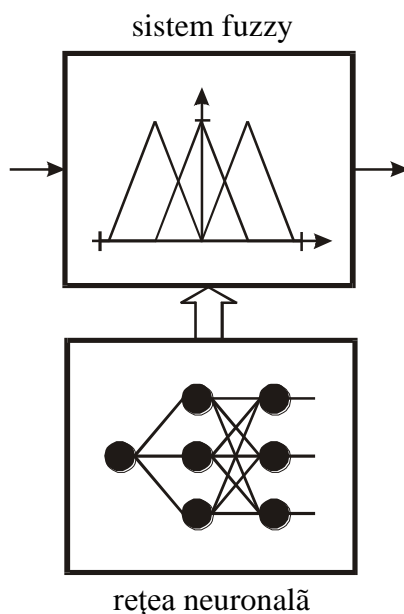


Figura 6.5. Reglarea neuronală a funcțiilor de apartenență (model cooperant).

10.3. Câteva tipuri de sisteme neuro-fuzzy

Într-o arhitectură hibridă neuro-fuzzy, algoritmul de antrenarea este folosit pentru a determina parametrii sistemului de inferențe fuzzy. Algoritmii convenționali (metode de gradient) nu pot

fi aplicați direct unui sistem fuzzy, în principal din cauza faptului că funcțiile de inferență nu sunt, în general, diferențiabile. Soluțiile acestei probleme sunt două: utilizarea unor funcții de apartenență diferențiabile și utilizarea unor sisteme cu metode de antrenare diferite de cele standard. Dintre cele mai cunoscute la ora aceasta se pot aminti:

- NEFCON – Neuro-Fuzzy Control (Control neuro-fuzzy) – Nauck și Kruse;
- FUN – Fuzzy Net (Rețea fuzzy) – Sulzberger;
- ANFIS – Adaptive Neuro-Fuzzy Inference System (Sistem de inferențe neuro-fuzzy adaptiv) – Jang;
- RFNN – Recurrent Fuzzy Neural Network (Rețea neuronală fuzzy recurentă) – Lee și Teng;
- FALCON – Fuzzy Adaptive Learning Control Network (Rețea cu antrenare fuzzy adaptivă) – Lin și Lee;
- SONFIN – Self Constructing Neuro-Fuzzy Inference Network (Sistem de inferențe neuro-fuzzy cu auto-organizare) – Feng și Teng;
- GARIC – Generalized Approximate Reasoning based Intelligent Control (Control inteligent prin raționament aproximativ generalizat) – Bherenji și Khedkar;
- FINEST – Fuzzy Inference Software – Tama, Oyama și Arnould;
- EFuNN – Evolving Fuzzy Neural Network – Kasabov și Song;

Această listă poate fi oricând completată de alte scheme (Abraham și Nath). Scopul este doar de a prezenta pe cele mai importante.

Control neuro-fuzzy (NEFCON)

Schemele de control neuro-fuzzy implementează sisteme de inferențe de tip Mamdani.

Conexiunile sunt ponderate prin valori fuzzy (termeni lingvistici) și reguli cu aceleași propoziții în premise, fapt simbolizat prin elipsele care grupează sinapsele.

- Stratul 1: realizează *fuzzyficarea* datelor de intrare prin convertirea intrărilor stricte (crisp) în valori ale funcțiilor de apartenență corespunzătoare.
- Stratul 2: asigură procesul de inferare, prin funcțiile de activare ale unităților din acest strat.
- Stratul 3: are rolul defuzzyficării.

Procesul de învățare este o combinație între algoritmul cu propagare înapoi a erorii și algoritmul de întărire. NEFCON este utilizat pentru:

- obținerea prin antrenare a unei variante inițiale a bazei de reguli, dacă nu există cunoștințe inițiale despre sistem;
- optimizarea unei baze de reguli alese.

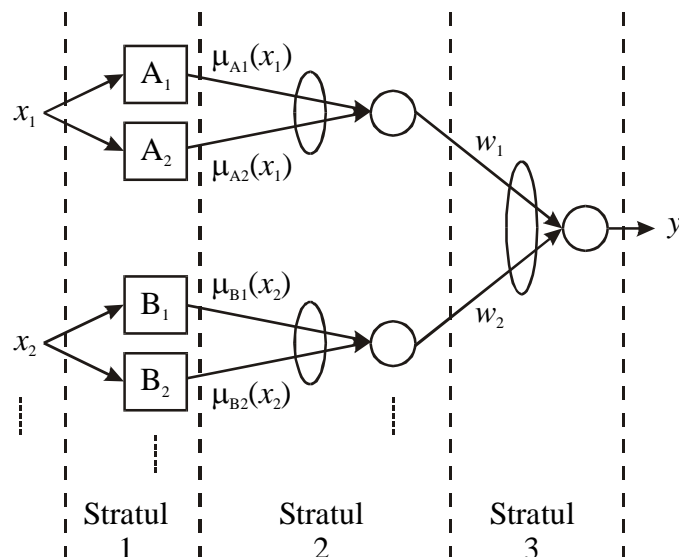


Figura 6.6. Arhitectura NEFCON.

Rețele fuzzy (FUN)

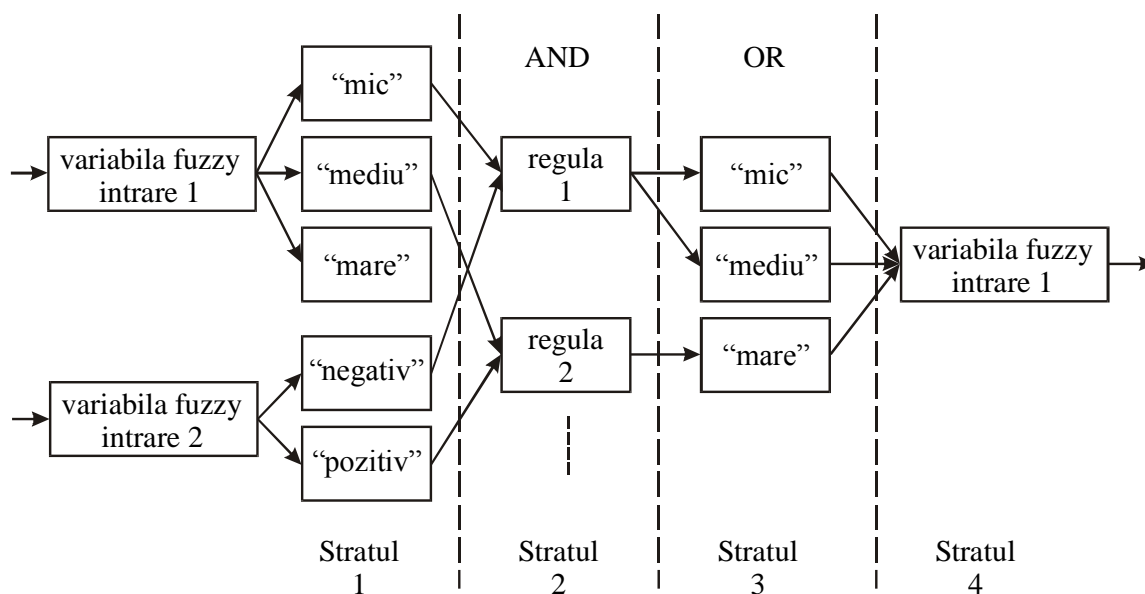


Figura 6.7. Arhitectura FUN.

- Stratul 1: implementează funcțiile de apartenență și realizează fuzzyficarea valorilor de intrare.
- Stratul 2: realizează operația de conjuncție fuzzy (fuzzy-AND) din premisă (propoziția din premisă este compusă).
- Stratul 3: conține funcțiile de apartenență ale variabilei de ieșire. Funcția de activare a neuronilor de aici este o disjuncție fuzzy (fuzzy-OR).
- Stratul 4: realizează defuzzyficarea.

Rețeaua este inițializată printr-o bază de reguli și prin parametrii seturilor de termeni lingvistici, apoi utilizează o metodă de învățare, care modifică aleator funcțiile de apartenență (parametrii). Dacă rezultatul obținut nu este mai bun, se revine la forma anterioară a parametrilor.

Sistem de inferențe neuro-fuzzy adaptiv (ANFIS)

Acesta mai este denumit și sistemul de inferență fuzzy bazat pe rețele adaptive (Adaptive- Network-Based Fuzzy Inference System - ANFIS). A fost propus de J.S.R. Jang și implementează un sistem de inferență fuzzy bazat pe modelul Takagi-Sugeno. Arhitectura ANFIS este structurată pe 5 straturi:

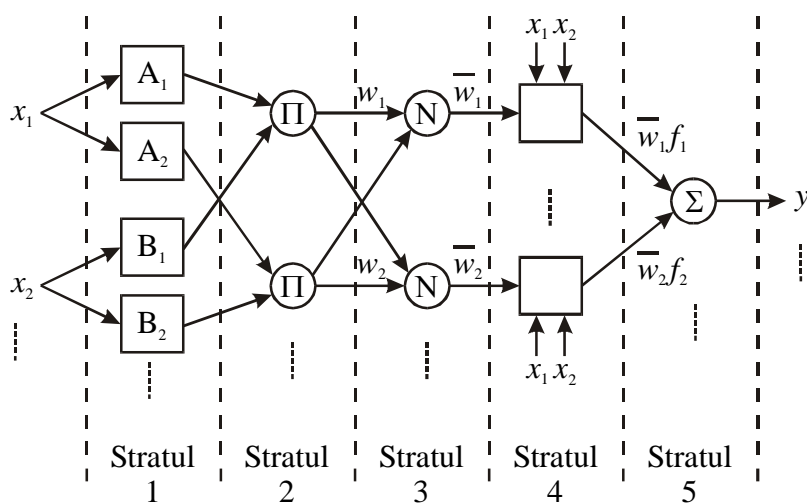


Figura 6.8. Arhitectura ANFIS.

- *Stratul 1*: realizează *fuzzyficarea* datelor de intrare prin convertirea intrărilor stricte (crisp) în valori ale funcțiilor de apartenență corespunzătoare.
- *Stratul 2*: este *stratul de reguli* care multiplică semnalele de intrare. Ieșirile nodurilor din acest strat semnifică *puterile de aprindere* (*firing strengths*) ale regulilor corespunzătoare.
- *Stratul 3*: efectuează *normalizarea* puterilor de aprindere, astfel încât suma acestora să fie 1. Fiecare semnal de intrare se divide cu suma totală a intrărilor.
- *Stratul 4*: este *stratul de defuzzyficare*. Aici se aplică modelul Sugeno-Takagi și se calculează un polinom de ordinul 1 de variabilele de intrare $f_i = p_i x_1 + q_i x_2 + r_i$. Constantele p_i , q_i și r_i sunt *parametrii concluzivi* ai sistemului de inferență. Rezultatul este ponderat cu puterile de aprindere normalizate calculate în stratul anterior.
- *Stratul 5*: este format dintr-un singur *neuron sumator*, care produce ieșirea sistemului ANFIS.

Învățarea într-un sistem ANFIS se bazează pe o schemă hibridă. Mai întâi, utilizatorul decide numărul de reguli și modul lor de interacțiune, construind astfel cunoștințele integrate apriori despre structura soluției. Funcțiile de apartenență trebuie inițializate astfel încât să împartă cât

mai egal universul discursului și să aibă suficiente suprapuneri. În pasul „înainte” al antrenării, se calculează ieșirile neuronilor până în stratul 4, iar parametrii concluzivi (coeficienții funcției liniare) sunt calculați prin metoda celor mai mici pătrate. În pasul „înapoi”, semnalele de eroare sunt retro-propagate și se actualizează parametrii antecedenti (ai funcțiilor de apartenență) prin metoda gradientului descendent.

Baza de reguli trebuie cunoscută apriori, deoarece ANFIS nu poate decât să ajusteze funcțiile de apartenență ale parametrilor antecedenti și concluzivi. Deoarece este ușor de implementat, acest sistem este atractiv pentru aplicații. Totuși, algoritmul de învățare necesită mari resurse de calcul, iar modelul are dificultăți în privința problemelor de mari dimensiuni, care implică un mare număr de partiții, reguli și deci parametri concluzivi, mai ales că fiecărui termen lingvistic îi corespunde o singură mulțime fuzzy.

Rețea neuronală fuzzy recurentă (RFNN)

Rețeaua neuronală fuzzy recurentă este o variantă modificată a unei rețele neuronale recurente care utilizează recurența pentru a realiza procesul de inferență. Este posibilă antrenarea rețelei pe baza experienței exprimată sub forma unei baze de reguli și interpretarea datelor obținute prin antrenare (valorile ponderilor) tot într-o formă lingvistică.

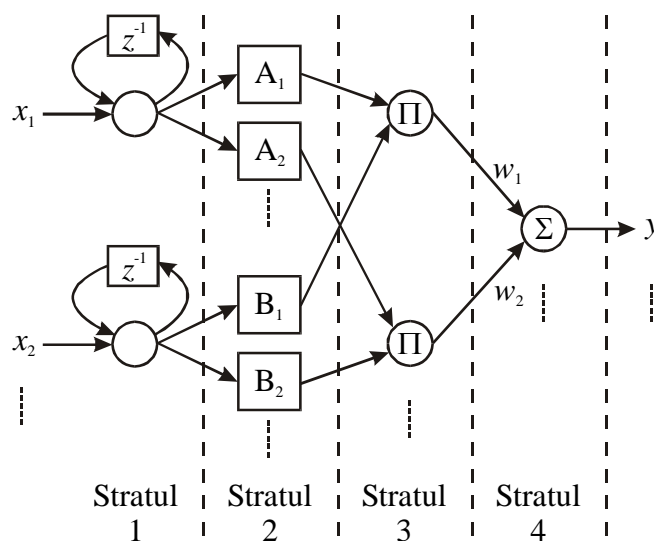


Figura 6.9. Arhitectura RFNN

- *Stratul 1:* primește variabilele de intrare pe care le transmite mai departe. Aici sunt adăugate *conexiunile feedback* pentru a introduce relații temporale în rețea.
- *Stratul 2:* realizează *fuzzyficarea*. Funcțiile de apartenență sunt în cazul de față funcții gaussiene.
- *Stratul 3:* este *stratul de reguli* care realizează procesul de inferență.
- *Stratul 4:* este *stratul de ieșire* care realizează defuzzyficarea.

10.4. Structura și funcționarea ANFIS

Sistemul de inferențe fuzzy neuronal adaptiv (în engleză Adaptive Neural Fuzzy Inference System - ANFIS) este cea mai des întâlnită structură hibridă care combină tehnici specifice rețelelor neuronale cu mecanismele de inferență în logică fuzzy. Varianta cea mai des utilizată în aplicații de control automat este cea propusă de Jang, deși variante apropiate au fost independent prezentate de alți autori.

Varianta Jang realizează un mecanism de inferențe de tip Sugeno-Tagaki, iar adaptarea constă în modificarea parametrilor bazei de reguli printr-o metodă asemănătoare algoritmului cu propagare înapoi a erorii introdus pentru rețelele neuronale unidirecționale.

Pentru o prezentare succintă a structurii, fie cazul unui regulator fuzzy de tip PI sau PD, având ca intrări ale mașinii de inferență eroare de reglare e și derivata acesteia de și care furnizează derivata mărimii de comandă du . Prezentarea este clară pentru o mașină de inferență cu două reguli fuzzy:

dacă e este E1 și de este DE1 atunci $du = f_1(e, de) = p_1e + q_1de + r_1$

dacă e este E2 și de este DE2 atunci $du = f_2(e, de) = p_2e + q_2de + r_2$

Pentru acest caz simplu, structura ANFIS este prezentată în figura 3.7. Structura este alcătuită din 5 straturi, care corespund etapelor în care se realizează evaluarea inferențelor.

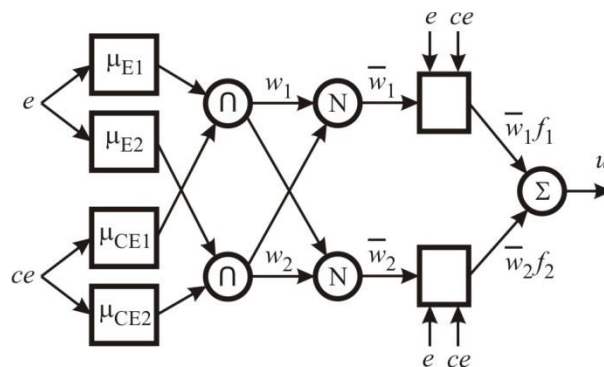


Figura 6.10. Structura ANFIS pentru o aplicație de control automat.

- *Stratul 1:* realizează *fuzzyficarea* datelor de intrare prin convertirea intrărilor stricte (crisp) în valori ale funcțiilor de apartenență corespunzătoare.
- *Stratul 2:* este *stratul de reguli* care multiplică semnalele de intrare. Ieșirile nodurilor din acest strat semnifică *puterile de aprindere (firing strengths)* ale regulilor corespunzătoare.
- *Stratul 3:* efectuează *normalizarea* puterilor de aprindere, astfel încât suma acestora să fie 1. Fiecare semnal de intrare se divide cu suma totală a intrărilor.
- *Stratul 4:* este *stratul de defuzzyficare*. Aici se aplică modelul Sugeno-Takagi și se calculează un polinom de ordinul 1 de variabilele de intrare $f_i = p_i x_1 + q_i x_2 + r_i$. Constantele p_i , q_i și r_i sunt *parametrii concluzivi* ai sistemului de inferență. Rezultatul este ponderat cu puterile de aprindere normalizate calculate în stratul anterior.

- *Stratul 5*: este format dintr-un singur *neuron sumator*, care produce ieșirea sistemului ANFIS.

Generalizând pentru o bază de J reguli fuzzy, calculul realizat poate fi descris printr-o relație de forma:

$$du = f_{ANFIS}(e, de) = \sum_{j=1}^J \bar{w}_j f_j(e, de) = \left(\sum_{j=1}^J w_j f_j(e, de) \right) \left(\sum_{j=1}^J w_j \right)^{-1} \quad (3.54)$$

în care

$$f_j(e, de) = p_j e + q_j de + r_j, \quad (3.55)$$

$$w_j = \mu_{Ej}(e) \cap \mu_{DEj}(de) \quad (3.56)$$

și $p_j, q_j, r_j \in \mathbb{R}$, $\mu_{Ej}, \mu_{DEj} \in [0, 1]$.

Din relațiile (3.49)-(3.51), se poate observa că calculul realizat de structura ANFIS conține un număr mare de parametri. Dacă variabilele de intrare au $N_{TL,e}$, respectiv $N_{TL,de}$ termeni lingvistici modelați prin funcții de apartenență triunghiulare (cu 3 parametrii), iar baza de reguli este completă, atunci numărul complet de parametrii în calculul realizat de ANFIS este

$$3N_{TL,e}N_{TL,de} + 3N_{TL,e} + 3N_{TL,de}.$$

Antrenarea sistemelor ANFIS

Metoda de antrenare cu propagare înapoi a erorii presupune cunoașterea inițială a unor date de antrenare în forma (valori de intrare)~(valori de ieșire). Aceste perechi de date sunt obținute prin orice metodă care furnizează valori corespundente corecte. Pentru cazul regulatorului fuzzy PI, perechile de date de antrenare se prezintă sub forma:

$$\{e_i; de_i\} \sim \{du_i\}, \quad i = \overline{0, I}. \quad (3.57)$$

Prin antrenare, se modifică valorile parametrilor descriși în calculul (3.49)-(3.51) și notați în continuare α_p cu $p = \overline{1, P}$, după o metodă de minim.

Pentru aceasta se introduce eroarea de antrenare obținută la perechea de date i cu valorile curente ale parametrilor α_p :

$$E_i(e_i, de_i; \alpha_{p=\overline{1, P}}) = E_i \Big|_{\alpha_p} = \frac{1}{2} (f_{ANFIS}(e_i, de_i) - du_i)^2 \Big|_{\alpha_p}. \quad (3.58)$$

Modificarea fiecărui parametru se face pe baza informației privind eroarea de antrenare, după o relație de forma:

$$\alpha_{p,i+1} = \alpha_{p,i} - \eta \frac{\partial}{\partial \alpha_p} E_i(e_i, de_i; \alpha_{p=1, \overline{P}}) \quad (3.59)$$

$$\alpha_{p,i+1} = \alpha_{p,i} - \eta (f_{ANFIS}(e_i, de_i; \alpha_{p=1, \overline{P}}) - du_i) \frac{\partial f_{ANFIS}(e_i, de_i; \alpha_{p=1, \overline{P}})}{\partial \alpha_p} \quad (3.60)$$

Relația (3.55) arată faptul că antrenarea necesită descrierea analitică a derivatelor parțiale ale ieșirii bazei de reguli după fiecare parametru. Acest fapt face ca implementarea acestui algoritm este dificilă, mai ales pentru cazul funcționării online a unui astfel de regulator. Pe de altă parte, structurile ANFIS și algoritmi de antrenare a acestora sunt deja implementați în aplicații software, cum ar fi Matlab. Așadar, în locul implementării online a acestei structuri, se poate determina un regulator fuzzy folosind datele de antrenare într-o procedură offline de antrenare a unei structuri ANFIS. Folosind un program Matlab, se realizează o antrenare a unei structuri ANFIS, din care apoi se extrage baza de reguli Sugeno-Tagaki pentru utilizarea acesteia în regulatorul conectat online la proces.

Învățarea într-un sistem ANFIS se bazează pe o schemă hibridă. Mai întâi, utilizatorul decide numărul de reguli și modul lor de interacțiune, construind astfel cunoștințele integrate apriori despre structura soluției. Funcțiile de apartenență trebuie inițializate astfel încât să împartă cât mai egal universul discursului și să aibă suficiente suprapuneri. În pasul „înainte” al antrenării, se calculează ieșirile neuronilor până în stratul 4, iar parametrii concluzivi (coeficienții funcției liniare) sunt calculați prin metoda celor mai mici pătrate. În pasul „înapoi”, semnalele de eroare sunt retro-propagate și se actualizează parametrii antecedenti (ai funcțiilor de apartenență) prin metoda gradientului descendent.

Baza de reguli trebuie cunoscută apriori, deoarece ANFIS nu poate decât să ajusteze funcțiile de apartenență ale parametrilor antecedenti și concluzivi. Deoarece este ușor de implementat, acest sistem este atractiv pentru aplicații. Totuși, algoritmul de învățare necesită mari resurse de calcul, iar modelul are dificultăți în privința problemelor de mari dimensiuni, care implică un mare număr de partiții, reguli și deci parametri concluzivi, mai ales că fiecărui termen lingvistic îi corespunde o singură mulțime fuzzy.

10.5. Aplicații ale sistemelor neuro-fuzzy

Sistemele neuro-fuzzy se bucură de apreciere în mai multe domenii. Un exemplu notabil este cel al controlului automat, în special aplicațiile aeronautice și aerospațiale, în care pilotul automat trebuie să simuleze capacitatea umană de a raționa în diverse situații. Alt exemplu este cel al recunoașterii trăsăturilor faciale. Trăsăturile sunt identificate și fuzzyficate pentru a le scădea sensibilitatea față de variațiile inerente care apar chiar în cazul aceleiași persoane. Aceste caracteristici fuzzyficate sunt apoi introduse într-o rețea neuronală care implementează procesul propriu-zis de recunoaștere.

Practic, mai toate aplicațiile prezentate la sistemele cu logică fuzzy și la rețelele neuronale pot fi și/sau au implementări în variantă neuro-fuzzy.

Avantajul major vine din obținerea în multe situații a unor rezultate mai bune, în condițiile utilizării unor module hardware și software adecvate, capabile de a efectua în timp util calculele necesare.

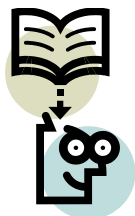
O listă de aplicații specifice

- Clasificare (*clustering*). (Abordarea clasică a problemelor de clasificare, utilizând rețele neuronale, realizează o clasificare dacă clasele sunt cunoscute. Mai mult, problema se complică în cazul clasificării în mai multe clase. O aplicație a sistemelor neuro-fuzzy este cea a clasificării, în condițiile în care nu avem inițial decât setul de obiecte, fără specificații privind numărul claselor și fără a le cunoaște analitic.);
- Recunoașterea textului scris. (Textul scris se împarte în segmente mai mici care apoi sunt supuse unui proces de identificare a literelor scrise. Sunt folosite 2 sisteme: primul pentru a identifica lungimea segmentelor, iar al doilea pentru recunoașterea propriu-zisă a literelor.);
- Recunoașterea vorbirii. (Identificarea vorbitorului și a cuvintelor rostite. Implementarea unui sistem neuro-fuzzy pentru o aplicație de recunoaștere a vorbirii. Motivația vine din faptul că performanțele obținute sunt mai bune decât prin abordarea care utilizează rețele neuronale simple, chiar în condițiile unei structuri cu mai puține unități);
- Regresie fuzzy (descoperirea de relații funcționale între date fuzzy);
- Rezolvarea ecuațiilor matriceale fuzzy (folosite în economie);
- Regulate pentru mașini electrice;
- Controlul mișcării roboților sau a brațului robot;
- Sisteme de identificare al obiectelor în instalații de reciclare;
- Aplicații în instalații de desalinizare;
- Sisteme de control al nivelului în rezervoare;
- Aparatură medicală;
- Control automat în aeronautică;
- Recunoaștere trăsăturilor și expresiilor faciale;
- Recunoașterea direcției de privire și controlul unui dispozitiv (armament) pe direcția privirii.

Să ne reamintim ...

Sistemele fuzzy și rețelele neuronale sunt estimatori numerici și în același timp sisteme dinamice, care prezintă capacitatea de a dezvolta inteligența agenților care acționează în medii nesigure, imprecise și afectate de zgomot. Relația dintre rețelele neuronale și cunoștințele lingvistice este bidirecțională. De aceea, sistemele de clasificare bazate pe rețele neuronale pot fi antrenate cu date numerice și cunoștințe lingvistice, iar sistemele de clasificare fuzzy bazate pe reguli pot fi proiectate cu ajutorul cunoștințelor lingvistice și a regulilor fuzzy extrase din rețele neuronale.

Logica fuzzy și modelele conexiuniste au necesități contrastante din punctul de vedere al aplicațiilor practice. Sistemele fuzzy sunt potrivite atunci când există suficientă cunoaștere de tip expert. Rețelele neuronale sunt adecvate dacă există suficiente date disponibile sau măsurabile. Sistemele fuzzy conduc la o abordare calitativ-simbolică, pe când rețelele neuronale se pretează mai ales la o abordare cantitativ-numerică.



Combinarea sistemelor fuzzy cu rețelele neuronale conduce la reducerea și chiar eliminarea dezavantajelor individuale ale celor două tehnici. În același timp, prin diverse modalități de hibridizare, se pot pune în valoare avantajele incontestabile ale acestora: capacitatea de învățare a rețelelor neuronale și reprezentarea explicită, simbolică a cunoștințelor prin intermediul sistemelor fuzzy.

Rețelele neuronale fuzzy mențin proprietățile de bază ale rețelelor neuronale, la care se adaugă fuzzyficarea unora din elementele lor caracteristice pentru a putea prelucra informații fuzzy. Pentru concretizare, au fost descrise structura și algoritmul de învățare ale unui perceptron fuzzy.

Sistemele fuzzy neuronale încearcă ajustarea regulilor din sistemul fuzzy prin metode tipice rețelelor neuronale, fără a-și pierde caracteristicile esențiale (fuzzyficarea, defuzzyficarea, motorul de inferență și fundamentul logicii fuzzy).

În abordarea hibridă neuro-fuzzy, atât tehnicile fuzzy cât și cele conexiuniste joacă un rol cheie, fiecare îndeplinind funcții diferite în sistem, potrivite naturii sale. În această categorie intră Sistemul de Inferență Fuzzy bazat pe Rețele Adaptive (ANFIS).



Test de evaluare a cunoștințelor

1. Enumerați modurile / arhitecturile de interconectare ale părților simbolice (inferențe fuzzy, mulțimi fuzzy) și conexiuniste (modele neuronale, algoritmi de instruire).
2. Enumerați o serie de aplicații în care sunt folosite sisteme neuro-fuzzy.
3. Descrieți arhitectura ANFIS și subliniați legătura între aceasta și metoda de

inferență fuzzy Sugeno-Takagi.



Test de autoevaluare a cunoștințelor

1. Partea conexiunistă a unui sistem neuro-fuzzy se referă la:
 - a) modul de conectare al sistemului neuro-fuzzy în aplicația în care este utilizat
 - b) mecanismul de inferență fuzzy folosit
 - c) modelul neuronal, arhitectura rețelei neuronale și / sau algoritmul de instruire.
2. Partea simbolică a unui sistem neuro-fuzzy se referă la:
 - a) modelul neuronal și algoritmul de instruire
 - b) aspectele de logică fuzzy
 - c) rolul sistemului de reglare în cadrul aplicației în care este utilizat
3. Sistemele neuro-fuzzy:
 - a) exploatează avantajele fiecărei tehnici, încercând eliminarea dezavantajelor
 - b) au rolul de a înlocui o tehnică ce nu poate fi utilizată într-o anumită aplicație
 - c) au rolul de a crește performanțele unui sistem de reglare fuzzy
4. Între arhitecturile neuro-fuzzy, se pot aminti:
 - a) Rețele neuronale fuzzy
 - b) Sisteme fuzzy neuronale
 - c) Sisteme hibride neuro-fuzzy
5. Între arhitecturile hibride neuro-fuzzy, se regăsesc:
 - a) modelul concurent
 - b) modelul cooperant
 - c) modelul neuronal
6. Sistemele ANFIS realizează un mecanism de inferență:
 - a) Sugeno-Takagi
 - b) Mamdani
 - c) niciuna dintre acestea

7. ANFIS este:

- a) un sistem de inferență neuro-fuzzy adaptiv
- b) un sistem de inferență fuzzy adaptat pentru aplicații numerice
- c) un sistem de inferență fuzzy care adaptează o rețea neuronală

8. Structura ANFIS realizează un calcul în:

- a) 3 etape
- b) 4 etape
- c) 5 etape

9. Algoritmul de antrenare ANFIS este un algoritm de instruire:

- a) supervizată
- b) nesupervizată
- c) depinde de datele de antrenare disponibile

10. Între aplicațiile în care se utilizează sisteme ANFIS se pot aminti:

- a) aplicații de conducere automată
- b) aplicații de identificare experimentală
- c) metode de proiectare ale reguletoarelor fuzzy