

Teme pentru proiecte la ”Calcul neuronal și calcul evolutiv”

2002

Biblioteci de funcții.

B1. Bibliotecă (C++/Java) pentru proiectarea rețelelor feedforward.

Descriere. Se definește o colecție de clase pe baza căreia să se poate proiecta o aplicație concretă. Se pot implementa obiecte pentru unități funcționale, nivele de unități și metode pentru funcții de integrare/transfer, metode de funcționare, metode de învățare (variante ale alg. BackPropagation, alg. de tip ”simulated annealing”).

Biblio: Valluru B. Rao, C++ Neural Networks and Fuzzy Logic (b1/C++...pdf), T. Masters, Advanced algorithms for Neural Networks (BCUT), M. Hagan et al. Design of Neural Network (BCUT

B2. Bibliotecă (C++/Java) pentru proiectarea rețelelor recurente.

Descriere. Se definește o colecție de clase pe baza căreia să se poate proiecta o aplicație concretă. Se pot implementa obiecte pentru unități funcționale, rețea (cel puțin modelul Hopfield) și metode pentru funcții de integrare/transfer (continue, discrete, aleatoare), metode de funcționare, metode de stabilire a parametrilor (pentru probleme de optimizare/ memorare asociativa).

Biblio: Valluru B. Rao, C++ Neural Networks and Fuzzy Logic (b1/C++...pdf), T. Masters, Advanced algorithms for Neural Networks (BCUT), M. Hagan et al. Design of Neural Network (BCUT

B3. Bibliotecă (C++/Java) pentru proiectarea algoritmilor genetici.

Descriere. Se definește o colecție de clase pe baza căreia să se poate proiecta o aplicație concretă. Se vor defini clase pentru clase pentru cromozomi (diverse variante de codificare), populație (de dimensiune constantă) și metode pentru evaluare (calcul fitness), pentru selecție (cel puțin trei variante), încrucișare (cel puțin trei variante) și mutație (cel puțin trei variante).

Biblio. b3/22060092.pdf

B4. Bibliotecă (C++/Java) pentru proiectarea strategiilor evolutive.

Descriere. Se definește o colecție de clase pe baza căreia să se poate proiecta o aplicație concretă. Se vor defini clase pentru clase pentru cromozomi, populație (de dimensiune constantă) și metode pentru evaluare (calcul fitness), pentru selecție (cel puțin trei variante), recombinare (cel puțin trei variante) și mutație (cel puțin trei variante). Se vor implementa și metode de auto-adaptare.

Biblio. b3/22060092.pdf

S1. "Ant systems".

Descriere: Comportarea insectelor ce traiesc în colectivități (furnicile și albinele) reprezintă sursă de inspirație pentru dezvoltarea unor metode de optimizare (pentru rezolvarea unor probleme dificile de optimizare combinatorială - problema comis voiajorului, planificarea activităților etc.) și control distribuit (de exemplu rutarea adaptivă în sistemele de comunicații). Abilitatea acestor comunități de a rezolva probleme se bazează pe comunicarea pe cale chimică (prin intermediul feromonilor): pe traseul parcurs pentru căutarea hranei furnicile lasă o "urme" care permite identificarea ulterioară a traseului care a dus la găsirea hranei. Sistemele artificiale implementează acțiunea feromonilor prin "recompensarea" configurațiilor bune și evaporarea feromonilor prin "uitarea" unor configurații.

Cerința: Implementarea unui algoritm de tip "colonie de furnici" pentru rezolvarea unei probleme de explorare a unui graf (de exemplu TSP).

Biblio: s1/as4.ps, s1/21590011.pdf

S2. "Particle swarm optimization".

Descriere: Tehnica de optimizare de tip "particle swarm optimization - PSO" are ca sursa de inspirație comportarea socială (legată în special de transmiterea și utilizarea în comun a informației) a unor ființe vii cum sunt stolurile de păsări, roiurile de albine sau bancurile de pești. În variantele artificiale procesul de căutare este asigurat de un ansamblu de "particule" a căror mișcare este caracterizată de o "viteză" care se modifică în timp în funcție caracteristicile întregului sistem. Permite găsirea rapidă a optimului însă are dificultăți în evitarea minimelor locale.

Cerința: Implementarea unui algoritm de tip PSO pentru rezolvarea unei probleme de optimizare continuă sau a unei de optimizare discretă.

Biblio: s2/7045.pdf, s2/7128.pdf, s2/7252.pdf

S3. "Immune systems".

Descriere: Sunt inspirate de sistemul imunitar care la apariția unui agent patogen reacționează prin proliferarea (clonarea) unor celule imunitare care "recunosc" patogenul. O parte dintre acestea devin efectoare, producând anticorpi altele au rol de memorare a "amprente" patogenului (astfel încât la atacuri ulterioare ale aceluiași agent sistemul să reacționeze rapid). Sistemele imunitare artificiale sunt utilizate pentru rezolvarea unor probleme de clasificare (pe baza proprietății de afinitate a unei celule imunitare pentru un anumit agent), pentru optimizare și pentru detectarea atacurilor asupra calculatoarelor (de tipul virusilor).

Cerința: Implementarea unui sistem imunitar artificial pentru rezolvarea unei probleme de optimizare.

Biblio: s3/8804.pdf, s3/8814.pdf, s3/7264.pdf, s3/8833.pdf

S4. "Tabu search"

Descriere: Spre deosebire de celelalte metode de căutare în care la generarea unei noi configurații se ține cont doar de configurația (populația) anterioară în "tabu search" se gestionează și folosește o listă de configurații parcurse și care trebuie evitate (configurații interzise).

Cerința: Implementarea unui algoritm de tip tabu search pentru rezolvarea unei probleme de optimizare.

Biblio: s4/hertz92tutorial.ps,

Utilizarea calculului neuronal și evolutiv în aplicații concrete

IN1. Prelucrarea imaginilor cu rețele neuronale

Descriere. Pornind de la o imagine în nuanțe de gri (de exemplu format pgm - portable graymap) se determină un vector de caracteristici ale acesteia (de exemplu caracteristici statistice locale - medie, dispersie etc.). Acest vector va reprezenta data de intrare pentru o rețea feedforward al cărui rol este clasificarea imaginii. Rețeaua va fi antrenată pe baza unui set de imagini pentru care clasificarea este cunoscută.

Cerință: Să se proiecteze o rețea neuronală pentru clasificare de texturi simple (generate artificial sau preluate dintr-o bază de date).

Biblio: in1/1198.pdf, in1/1514.pdf

IE2. Prelucrarea imaginilor cu algoritmi evolutivi (variante: (a), (b), (c))

Descriere. (a) Multe dintre prelucrările de imagini la nivel de scăzut (filtrare, detectare muchii etc.) sunt bazate pe utilizarea unor matrici de convoluție (de exemplu de dimensiune 3*3). Majoritatea lor sunt deduse pe baza unor considerente matematice. Pentru un anumit tip de prelucrare ar putea fi construită în manieră evolutivă o astfel de matrice.

(b) Programarea genetică permite dezvoltarea de "programe" destinate efectuării unor prelucrări de imagini (de exemplu clasificarea texturilor).

(c) Efectul transformărilor evoluteive poate fi ușor ilustrat dacă cromozomii sunt chiar imagini.

Cerință. (a) Se proiectează un algoritm evolutiv (la alegere) care să asigure determinarea unei matrici utilizată pentru netezire (eliminarea zgomot). (b) Se utilizează programarea genetică pentru un tip de prelucrare (la alegere). (c) Se implementează un algoritm care transformă imagini prin mutație și încrucișare.

Biblio (generală): (a) ie2/00887232.pdf, ie2/aguirre01.ps, (b) ie2/7111.pdf, ie2/csrp-96-01.ps, ie2/csrp-97-05.ps (c) ie2/ep95gb.ps

OE3. Optimizare combinatorială cu algoritmi evolutivi

Descriere. Algoritmii evolutivi (în special cei genetici) pot fi instrumente eficiente pentru rezolvarea unor probleme de optimizare combinatorială. Problema dificilă este alegerea adecvată a operatorilor.

Cerință. Se efectuează un studiu practic asupra aplicabilității AE în problemele: parcurgere grafuri (TSP), partiționare grafuri, pb. rucsacului 0/1 (și varianta multiplă), colorarea grafurilor.

Biblio: oe3/csc94.ps, oe3/galinier99hybrid.ps, oe3/p44.ps, oe3/sac94.ps, oe3/two_novel...ps

CE4. Rezolvarea problemelor de clasificare cu algoritmi evolutivi

Descriere. Algoritmii de grupare (cum este algoritmul mediilor) sunt de fapt bazați pe rezolvarea unei probleme de optimizare (se minimizează distanța elementelor dintr-o clasă față de reprezentantul clasei). Această problemă de minimizare poate fi rezolvată în manieră evolutivă.

Cerință. Se studiază practic (pentru o problemă simplă de grupare) aplicabilitatea algoritmilor evolutivi.

Biblio. ce4/00764879.pdf, ce4/00771164.pdf, ce4/20380256.pdf, ce4/22060619.pdf

PE5. Planificarea activităților cu algoritmi evolutivi

Descriere. Planificarea unei activități (de exemplu construirea unui orar) este o problemă cu numeroase restricții și unul sau mai multe criterii de optimizat. Unele restricții sunt puternice (trebuie respectate întotdeauna) iar altele sunt mai slabe (pot fi încălcate însă induc o creștere a costului - vezi tehnica penalizării).

Cerință. Să se genereze o planificare a examenelor unui an de studiu utilizând un algoritm evolutiv.

Biblio. pe5/burke95hybrid.ps, pe5/a-constructive-evol...ps, pe5/burke93university.ps, pe5/corne94evolutionary.ps

RE6. Regresie simbolică cu algoritmi evolutivi

Descriere. Pornind de la perechi de valori numerice obținute de exemplu prin măsurarea a două mărimi se caută o expresie care să descrie relația dintre cele două mărimi. Spre deosebire de regresia numerică unde se presupune că relația de dependență are o anumită formă și se determină doar coeficienții acesteia, în regresia simbolică se "construiește" expresia pornind de la elementele sale atomice: operatori (+, -, *, /), funcții elementare (exp, sin, cos etc), constante și variabile. Arborele asociat expresiei (în care operatorii sunt noduri neterminale iar operanzii sunt noduri terminale) se poate construi prin programare genetică

Cerință. Elaborarea unui algoritm bazat pe programare genetică pentru regresie simbolică.

Biblio. re6/7286.pdf, re6/csrp-98-19.ps, re6/7203.pdf

TE7. Prelucrarea seriilor temporale cu algoritmi evolutivi

Descriere. Seriile temporale sunt valori ale unei mărimi înregistrate la momente de timp succesive. Una dintre problemele cele mai dificile probleme în prelucrarea unei serii temporale (de exemplu efectuarea unei predicții) este determinarea unui model adecvat. Selectarea unui model poate fi realizată în manieră evolutivă.

Cerință. Studiul aplicabilității algoritmilor evolutivi pentru modelarea seriilor temporale.

Biblio. te7/20370335.pdf, te7/20380361.pdf, te7/20700393.pdf, te7/yao_liu....ps

SN8. Securitatea calculatoarelor și rețele neuronale

Descriere. Rețelele neuronale (feedforward sau Kohonen structurate) permit identificarea atacurilor asupra calculatoarelor. Pentru a funcționa on-line o astfel de rețea trebuie să fie cu auto-organizare și eventual incrementală (numărul de unități crește în procesul de învățare).

Cerință. Studiul aplicabilității rețelelor neuronale în detectarea atacurilor asupra calculatoarelor.

Biblio: sn8/...

CA9. Proiectarea evolutivă a automatelor celulare

Descriere. Regulile care descriu dinamica locală a unui automat celular destinat rezolvării unei probleme pot fi construite folosind un algoritm genetic ce determină valoarea fitness-ului pe baza comportării automatului celular în cazul unei probleme concrete.

Cerință. Proiectarea unui algoritm genetic pentru dezvoltarea unui automat celular unidimensional cu stări binare.

Biblio. ca9/evca-review.ps, ca9/life-and-evolution.ps

D10. Implementare paralelă/distribuită a unui algoritm genetic

Descriere. Algoritmii evolutivi fiind costisitori ca urmare a gestiunii unei populații, o variantă de reducere a timpului efectiv de calcul poate fi realizată prin efectuarea în paralel a unor prelucrări (evaluarea elementelor populației, aplicarea operatorilor la nivelul unor subpopulații etc.)

Cerință. Implementarea în paralel a unui algoritm evolutiv.

Biblio. d10/20000602-hiroyasu.pdf, d10/20380064.pdf, d10/20700383.pdf, d10/cantu-paz97survey.ps

V11. Vizualizarea algoritmilor genetici

Descriere. Efectul operatorilor genetici poate fi ilustrat prin "vizualizarea" populațiilor (de exemplu o populație de elemente codificate binar poate fi văzută ca o matrice binară adică o imagine alb-negru).

Cerință. Se va propune o metodă de vizualizare ce va fi exemplificată pentru un algoritm genetic simplu.

Biblio. v11/00942528.pdf

*** Temă la alegere

Cerință. Prezentarea unei probleme ce poate fi rezolvată prin calcul neuronal sau evolutiv și exemplificarea practică a modalității de rezolvare.

Observație. Fiecare proiect presupune:

- Redactarea unui referat privind tematica aleasă care va conține o parte cu caracter bibliografic (bibliografia de pornire poate fi cea furnizată o dată cu tema sau cea selectată de către student) și o contribuție proprie în care se prezintă implementarea (sau încercarea de implementare) practică. Se încurajează prezentarea observațiilor proprii și a dificultăților întâlnite pe parcursul elaborării proiectului.
- Realizarea unei aplicații prin care să se illustreze specificul temei. Interesează în special implementarea metodei (neuronale sau evolute) și mai puțin aspectele legate de interfață. Pentru testare se pot folosi probleme create artificial.