

Dezvoltarea sistemelor care învață singure - probleme de clasificare



Obiective

Dezvoltarea sistemelor care învață singure. Algoritmi de învățare supervizată. Modele liniare. Specificarea, proiectarea și implementarea sistemelor care învață singure cum să rezolve probleme de clasificare.



Aspecte teoretice

Proiectarea și dezvoltarea sistemelor care învață singure.

Algoritmi de învățare bazați pe:

- metoda gradientului descendent
- algoritmi evolutivi



Termen de predare

Laborator 5



Cerințe

Specificați, proiectați și implementați o aplicație pentru rezolvarea uneia dintre problemele următoare cu ajutorul unui sistem care învață singur să rezolve problema. Sistemul trebuie să permită alegerea unuia dintre cei 3 algoritmi de învățare:

Tehnica	Cod logic	Interfață	Aplicația overall
Metoda gradientului descendent	130	30	130
Algoritmi evolutivi	240	30	240
Total	370	60	370

Total 800p

Punctaj minim de realizat pentru validarea laboratorului 250p

Bonus

Implementarea unui arbore de decizie pentru probleme de clasificare

Date propuse de student	100
Selectarea atributelor in constructia arborelui conform unor calcule (de ex pe baza entropiei)	100
Date cu attribute categoriale (similare exemplului cu vampirii din curs)	100
Date cu attribute numerice (similare problemelor de mai jos)	200

Aplicația trebuie să permită:

- Încărcarea datelor problemei (probleme cu date deja definite de către programator, probleme cu date definite de utilizator)
- Alegerea și parametrizarea metodei de rezolvare a problemei
- Prezentarea rezultatelor procesului de învățare (antrenare și testare)

Aplicația poate fi realizată în 2 variante:

varianta 1. Tehnica de învățare se bazează pe un tool deja existent (50% din punctaj)

varianta 2. Tehnica de învățare se bazează pe cod dezvoltat de student (100% din punctaj)

Studentii pot alege care variantă de aplicație doresc să o realizeze.

Sisteme disponibile care implementează algoritmi de învățare:

1. Weka <http://www.cs.waikato.ac.nz/ml/weka/>
2. Matlab <http://www.mathworks.com/products/neural-network/>
3. OpenCV http://docs.opencv.org/modules/ml/doc/neural_networks.html
4. Scikit-learn <http://scikit-learn.org/stable/>
5. GPLAB <http://gplab.sourceforge.net/>
6. ECJ <http://cs.gmu.edu/~eclab/projects/ecj/>

1. Să se clasifice o **cardiotocogramă fetală** ca fiind normală, suspectă sau patologică pe baza diferitelor măsurători înregistrate în decursul sarcinii. Datele de test se vor lua de aici <http://archive.ics.uci.edu/ml/datasets/Cardiotocography>.
2. Să se stabilească dacă **un pacient suferă de hernie de disk, de spondiloză sau dacă este sănătos** pe baza informațiile preluate anterior de la mai mulți pacienți (privind forma și orientarea pelvisului și a părții lombare din coloana vertebrală). Datele de test se vor lua de aici <http://archive.ics.uci.edu/ml/datasets/Vertebral+Column>.
3. Să se stabilească **direcția de deplasare a unui robot** (în față, viraj ușor la dreapta, viraj puternic la dreapta, viraj ușor la stânga) dotat cu senzori pe baza informațiilor colectate anterior (sub forma unor citiri a 24 de senzori de ultrasunete plasați în diferite poziții pe robot – poziția unui senzor fiind dată de unghiul de deviație - 180° (față), -165°, -150°, ..., 15°, 0° (spate), 15°, 30°, ..., 150°, 165°). Datele de test se vor lua de aici <http://archive.ics.uci.edu/ml/datasets/Wall-Following+Robot+Navigation+Data>.