

Laborator 2 - RNSF

Extragerea trăsăturilor dintr-un semnal

1. Introducere

Inteligența artificială reprezintă teoria și dezvoltarea sistemelor computaționale capabile să îndeplinească sarcini ce necesită inteligență umană, precum percepția vizuală, recunoașterea vorbirii, luarea deciziilor etc.

Recunoașterea formelor se ocupă cu dezvoltarea algoritmilor și metodelor ce stabilesc o anumită asociere între valorile unui spațiu de intrare și anumite valori ale spațiului de ieșire.

Există trei categorii de probleme:

- a) Probleme de clasificare: acest tip de probleme presupun clasificarea valorilor de intrare conform algoritmului implementat prin etichetarea datelor la ieșire, fiecare etichetă reprezentând o clasă de apartenență
- b) Probleme de regresie (predicție): problemele de regresie asociază o valoare reală pentru fiecare valoare sau set de valori de intrare
- c) Probleme de analiză sintactică: în cazul problemelor de analiză sintactică, pentru fiecare set de date de intrare se asociază la ieșire o structură sintactică, precum: arbore, graf.

2. Structura sistemului de recunoaștere a formelor

Sistemele de recunoaștere a formelor au, în general, o structură ce urmează etapele prezentate în Fig. 1. Datele de intrare sunt preprocesate în vederea îmbunătățirii calității acestora prin operații de filtrare, normalizare sau eliminarea zgomotului. După ce datele de intrare au fost preprocesate, sistemul urmează o etapă de extragere de trăsături. În această etapă semnalul este procesat astfel încât caracteristicile extrase în final să fie cât mai reprezentative pentru datele de intrare, pentru a se obține un rezultat cât mai bun în etapa de clasificare.



Fig. 1 – Structura generală a unui sistem de inteligență artificială

2.1. Etapa de preprocesare

În etapa de preprocesare se urmărește procesarea semnalului pentru îmbunătățirea calității acestuia. În general, această etapă conține filtre de eliminare a zgomotelor, normalizarea datelor, etc.

2.2. Extragere de trăsături

În etapa de extragere de trăsături se urmărește extragerea celor mai relevante informații din semnal în vederea caracterizării acestuia, facilitând învățarea datelor și generalizarea acestora. Astfel, se creează un vector de trăsături ce conține cele mai relevante informații din datele achiziționate, având și o dimensiune redusă.

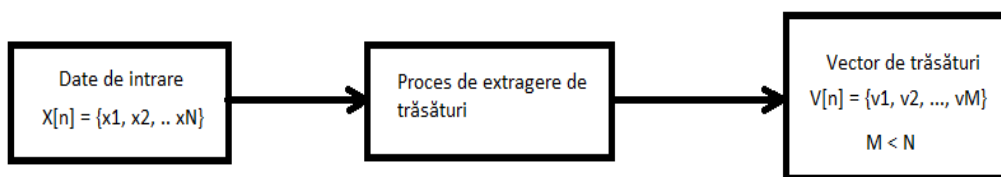


Fig. 2 – Extragere de trăsături

Exemplu:

Se dorește diferențierea vocalelor „a” și „e” analizând semnale achiziționate în timpul mai multor rostiri.

O metodă de extragere de trăsături din semnalul vocal este reprezentată de calcularea puterii pe benzi de frecvență.

Vocalele pot fi diferențiate analizând frecvențele acestora. În tipul rostirii, tractul vocal își modifică poziția pentru a articula sunetele care se doresc a fi rostite, rezultând sunete ce prezintă frecvențe diferite. În Fig. 3 sunt reprezentate spectrele obținute în urma aplicării transformatei Fourier pentru o rostire a fiecărei vocale.

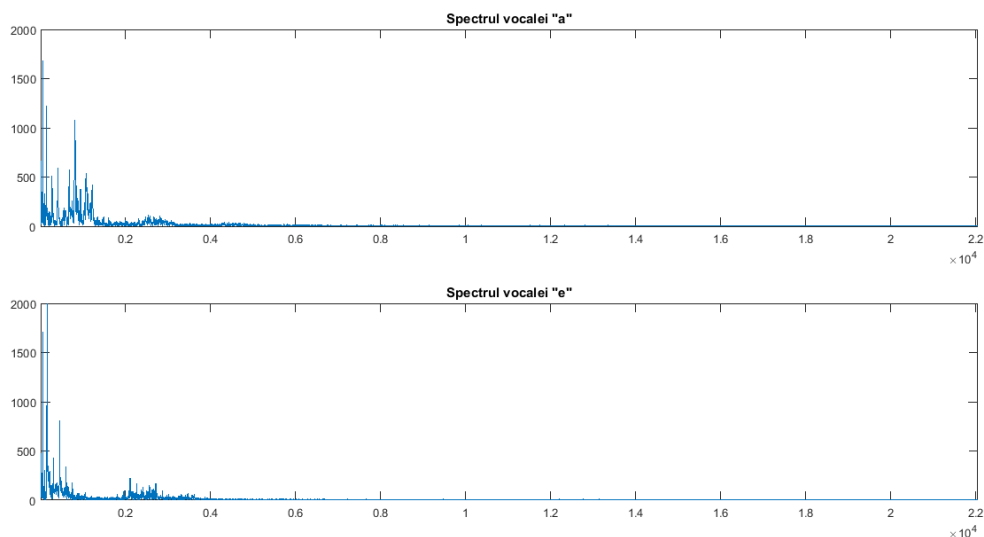


Fig. 3 – Spectrele celor două vocale (sus – vocala a, jos – vocala e)

În această lucrare au fost analizate benzile de frecvență $[0\ 500]$ Hz și $[500\ 1000]$ Hz. Se poate observa în Fig. 4 că puterea este distribuită diferit pe cele două benzi de frecvență pentru cele două vocale.

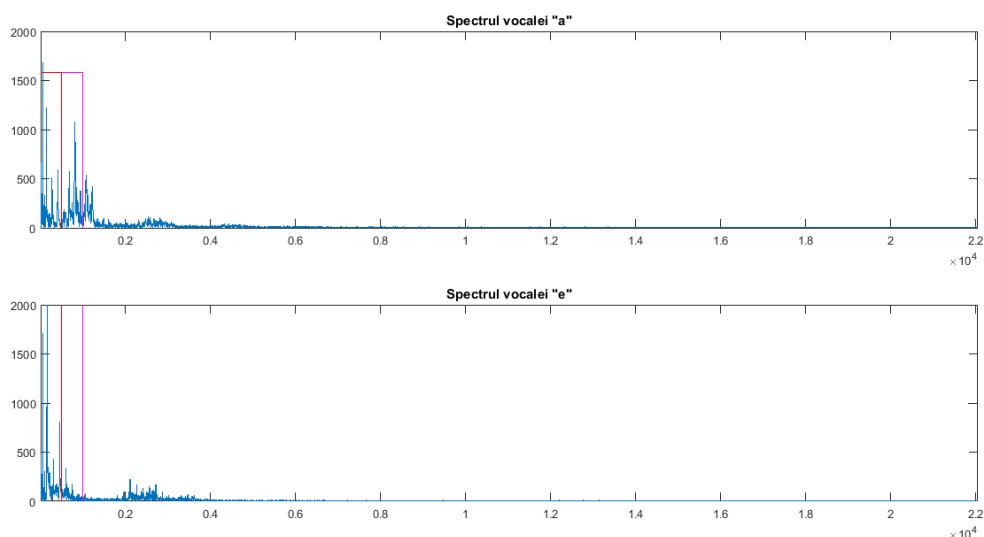


Fig. 4 – Spectrele unei rostiri pentru fiecare vocală împreună cu benzile de frecvență analizate

3. Desfășurarea lucrării

Se dorește diferențierea vocalelor „a” și „e” analizând semnale achiziționate în timpul mai multor rostiri.

O metodă de extragere a trăsăturilor din semnalul vocal este reprezentată de calcularea puterii pe benzi de frecvență. Rostind cele două vocale, se poate observa că acestea prezintă frecvențe diferite. Astfel, calculând energia semnalului pe benzile de frecvență [0 500] Hz și [500 1000] Hz, este suficient pentru diferențierea vocalelor a și e.

Pentru desfășurarea lucrării vor fi parcurse următoarele etape:

1. Stabilirea parametrilor utilizați

În realizarea scopului propus, vor fi inițial stabiliți parametrii necesari calculării puterii pe benzi de frecvență:

- Frecvența de eșantionare a semnalelor este de 44100 Hz
- Prima banda de frecvență este cuprinsă între 0 și 500 Hz
- A doua bandă de frecvență este cuprinsă între 500 și 1000 Hz
- Numărul total de înregistrări pentru fiecare vocală este de 40

2. Calcularea energiei pe cele două benzi de frecvență

Pentru calcularea puterii, se vor încărca pe rând semnalele pentru fiecare vocală. Încărcarea semnalelor se poate realiza utilizând funcția „wavread”.

Odată încărcat semnalul, acesta va fi transformat din semnal stereo în semnal mono, prin eliminarea uneia dintre coloane.

În continuare, va fi realizată transformata Fourier asupra semnalului mono.

Asemănător semnalului, transformata Fourier este reprezentată în eșantioane, nu în frecvență, motiv pentru care trebuie stabilit eșantionul corespunzător frecvențelor celor două benzi. Acest lucru se poate face utilizând formula:

$$idx = \frac{f [Hz] * lungime_spectru}{Fs [Hz]}$$

În continuare, se poate calcula puterea semnalului în banda dorită utilizând formula:

$$P = \frac{1}{N} \sum_{i=1}^N x_i^2$$

Unde:

- N reprezintă numărul de eșantioane
- x_i reprezintă amplitudinea eșantionului i

În final, se salvează cele două puteri calculate pentru fiecare înregistrare a fiecărei vocale. Astfel, având 40 de înregistrări pentru fiecare vocală, în final vor rezulta două matrici de dimensiune 40x2.

3. Reprezentarea trăsăturilor extrase

În această etapă, vor fi reprezentate trăsăturile extrase pentru cele două clase: vocala a și vocala e alegând culori diferite pentru fiecare clasă.

Rezultatul obținut este reprezentat în Fig. 5.

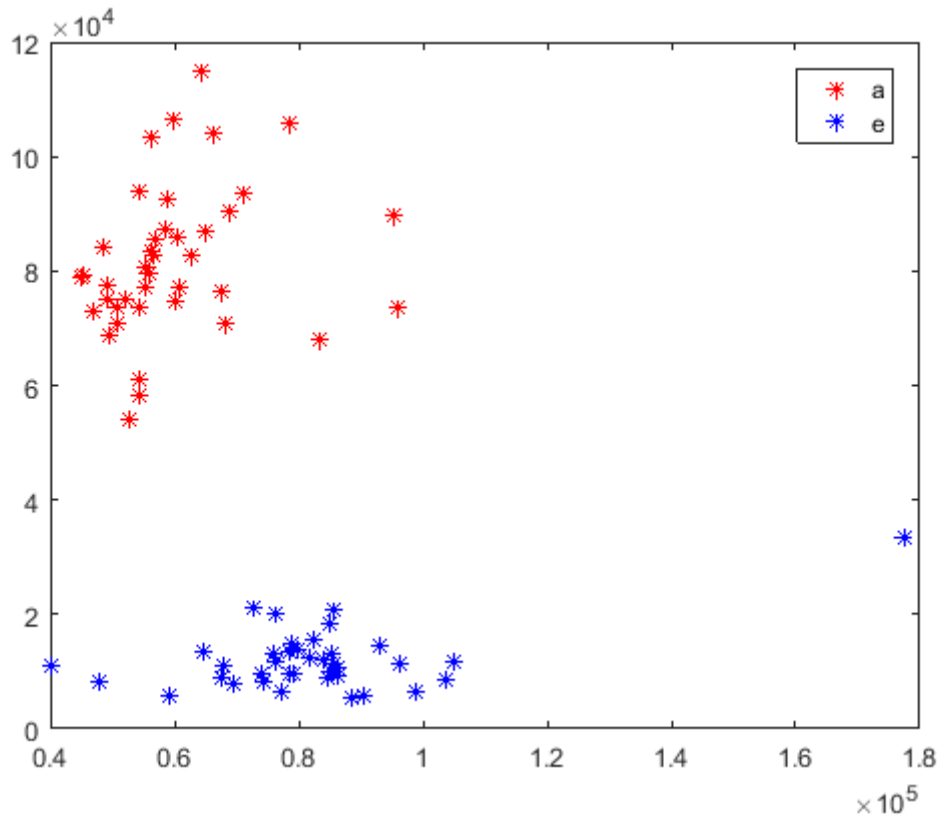


Fig. 5 – Reprezentarea celor două clase în urma extragerii trăsăturilor

4. Clasificare

Analizând graficul din Fig. 5, se poate observa că cele două clase sunt foarte ușor de diferențiat și sunt liniar separabile. Pentru clasele liniar separabile, se poate utiliza un algoritm de clasificare precum Perceptronul simplu.

Perceptronul simplu este cea mai simplă formă a rețelelor neuronale utilizat pentru clasificarea pattern-urilor liniar separabile. Acesta conține un singur neuron cu sinapse ce pot fi ajustate prin modificarea ponderilor și a bias-ului. Prin acest algoritm, s-a demonstrat că dacă două clase sunt liniar separabile, atunci perceptronul converge către o suprafață de decizie de forma unui hiperplan între cele două clase. Deoarece perceptronul are un singur neuron, acesta este limitat la clasificarea vectorilor în doar două clase.

Arhitectura Perceptronului:

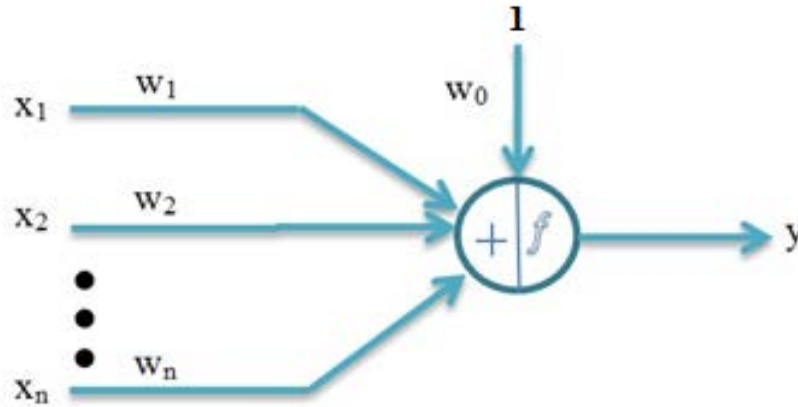


Fig. 6 – Arhitectura perceptronului simplu

În Fig. 6 este reprezentată arhitectura perceptronului simplu, unde $\mathbf{X} = (x_1, x_2, \dots, x_n)$, $x_i \in \mathbb{R}$, reprezintă intrările în neuron, iar $\mathbf{W} = (w_0, w_1, \dots, w_n)$, $w_i \in \mathbb{R}$, reprezintă ponderile neuronului.

Ieșirea neuronului reprezentată în Fig. 6 cu litera y , este dată de ecuația:

$$y = f\left(w_0 + \sum w_i x_i\right) \quad (3)$$

Unde $f(x)$ reprezintă funcția de activare a neuronului și este de forma:

$$f(k) = \begin{cases} 0, & \text{dacă } \left(w_0 + \sum w_i x_i\right) < 0 \\ 1, & \text{dacă } \left(w_0 + \sum w_i x_i\right) > 0 \end{cases} \quad (4)$$

Algoritmul Perceptronului

1. Inițializarea aleatoare a ponderilor la momentul $t=0$ și alegerea ratei de învățare η .
2. Evaluarea ieșirii reale $y_j = f(W^T X_j)$, unde $j=1:N$, N – numărul de vectori de intrare
3. Compararea ieșirii reale y_j cu ieșirea dorită d_j

$$e_j(t) = d_j - y_j(t) \quad (5)$$

În acest caz, există 3 valori posibile pentru eroarea de la ieșire: $\{0, 1, -1\}$

4. Ajustarea ponderilor cu o valoare care micșorează eroarea:

$$\Delta w_i(t) = \eta e_j(t) x_{ij} \quad (6)$$

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \quad (7)$$

Unde η reprezintă rata de învățare.

5. Revenire la pasul 2 până când toate corespondentele $\{(X_j, d_j)\}$ sunt corecte

În urma aplicării algoritmului Perceptron, se poate găsi o suprafață de separare pentru cele două clase, suprafață reprezentată în Fig. 7.

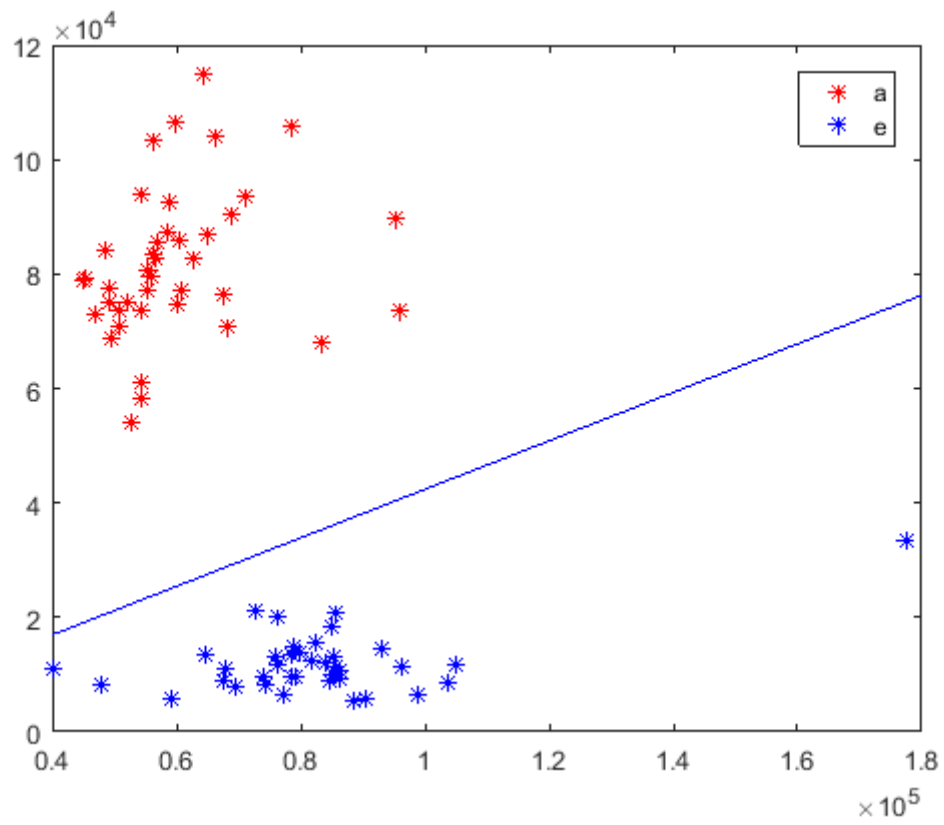


Fig. 7 – Reprezentarea celor două clase, împreună cu suprafața de separație