

Laborator 5 - RNSF

Rețeaua neuronală cu Bază Radială (RBF)

1. Introducere

Rețeaua neuronală cu funcții de bază radiale este bazată pe aplicarea unei transformări neliniare asupra spațiului de intrare cu scopul ca în noul spațiu datele să fie (aproape) liniar separabile. Neuronii din stratul ascuns efectuează această transformare asupra spațiului de intrare, iar neuronii de ieșire lucrează pe datele obținute în urma transformării neliniare. Fiecarui neuron din stratul ascuns îi corespunde un centroid, care acționează ca un prototip pentru vectorii din setul de date. Fiecare neuron din stratul ascuns semnalizează nivelul de similaritate al vectorului de intrare cu centroidul asociat acelui neuron.

Arhitectura rețelei RBF

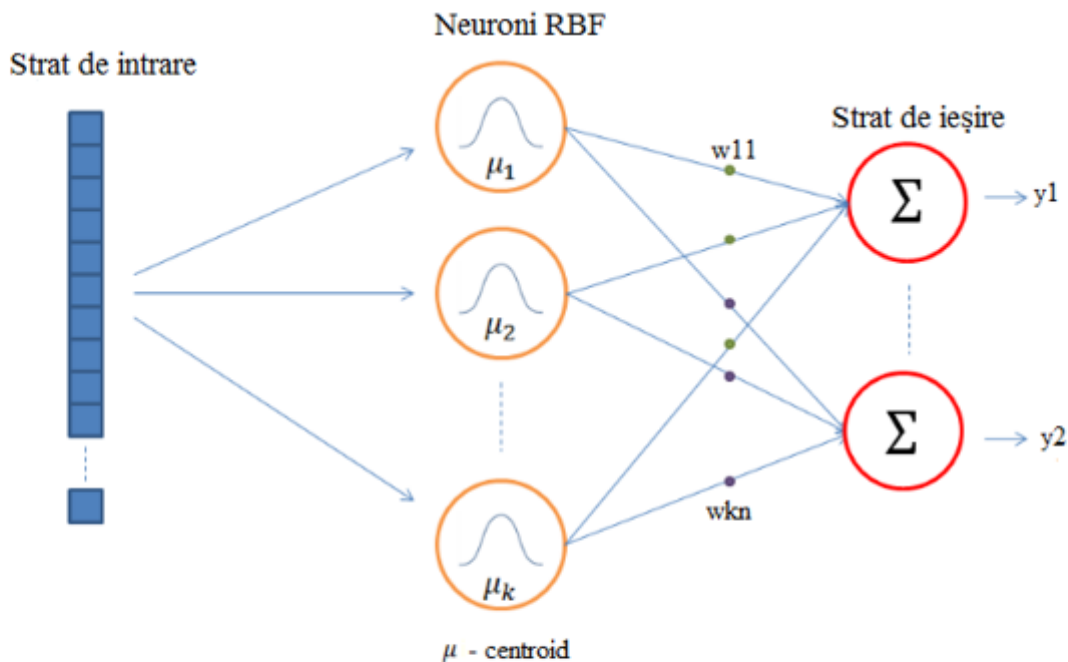


Fig. 1 – Arhitectura rețelei RBF

În Fig. 1 este reprezentată arhitectura tipică a unei rețele neuronale de tip RBF. Aceasta constă dintr-un strat de intrare, un strat de neuroni cu funcție de activare cu bază radială și un strat de ieșire cu un neuron pentru fiecare clasă a setului de date.

Stratul de intrare

Stratul de intrare este reprezentat de un vector n-dimensional care se dorește a fi clasificat.

Stratul ascuns

Fiecare neuron din stratul ascuns conține un centroid, μ_k . Vectorul de intrare este comparat cu fiecare centroid din stratul ascuns, rezultând valori de ieșire cuprinse între 0 și 1 reprezentând similaritatea dintre vectorul de intrare și neuronul respectiv. Dacă intrarea este egală cu centroidul, atunci ieșirea din neuronal RBF va fi 1. Cu cât crește distanța dintre intrare și centroid, răspunsul neuronului scade către 0.

Stratul de ieșire

Ieșirea din rețeaua neuronală constă dintr-un set de neuroni, unul pentru fiecare clasă corespunzătoare setului de date. Fiecare neuron calculează un scor pentru categoria asociată. În general, decizia de clasificare este luată conform scorului maxim.

Scorul este calculat prin însumarea ponderată a valorilor de activare obținute în urma trecerii vectorului de intrare prin fiecare neuron RBF.

Funcția de activare de tip RBF

Fiecare neuron cu bază radială calculează o măsură a similitudinii dintre intrare și centroid. Vectorii de intrare care sunt similari cu prototipul returnează o valoare apropiată de 1. Există mai multe posibilități de calculare a similitudinii, dar cea mai utilizată este funcția Gaussiană. În ecuația 1.1 este reprezentată funcția Gaussiană pentru un vector de intrare unidimensional:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1.1)$$

Unde x reprezintă intrarea, μ reprezintă media, iar σ reprezintă deviația standard (varianța).

Această funcție descrie curba sub formă de clopot reprezentată în Fig. 2, având media=5 și varianța=1.

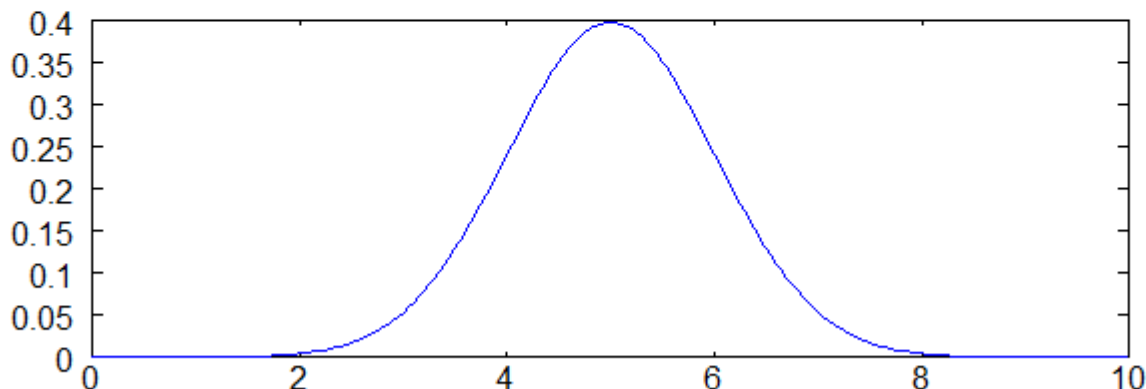


Fig. 2 – Funcție cu bază radială

Funcția de activare în cazul neuronilor cu bază radială este ușor diferită și este, în general, scrisă astfel:

$$\varphi(x) = e^{-\beta \|x-\mu\|^2} \quad (1.2)$$

În cazul funcției Gausiene, μ se referă la media distribuției, însă în cazul funcției de activare a neuronilor, μ se referă la centroid.

Prima modificare realizată în cazul funcției de activare o reprezintă eliminarea coeficientului $\frac{1}{\sigma\sqrt{2\pi}}$. Acest termen controlează înălțimea Gausienei, dar în cazul de față, termenul este redundant datorită ponderilor aplicate la ieșirea din rețea. În timpul antrenării, neuronii de ieșire vor învăța prin modificarea ponderilor conform răspunsului dorit.

O altă schimbare realizată în funcția de activare o reprezintă modificarea coeficientului $\frac{1}{2\sigma^2}$ cu un singur parametru, β . Acest parametru controlează lărgimea clopotului Gaussian. În Fig. este reprezentată funcția de activare având diferite valori ale parametrului β .

2. Algoritmul RBF

Există mai multe variante de antrenare a rețelei neuronale de tip RBF:

1. Antrenarea simultană a tuturor parametrilor (similară algoritmului BackPropagation – doar regulile de ajustare ale centrilor se modifică)
2. Antrenare separată a parametrilor: centri, lărgimi, ponderi.

Antrenare separată a parametrilor

În cazul antrenării separate a parametrilor, se utilizează algoritmul KMeans (Basic Isodata) pentru determinarea centroizilor, după care se calculează varianța pentru fiecare centroid, urmând să se calculeze ponderile pentru stratul de ieșire.

I. Algoritmul KMeans

Algoritmul K-Means este o metodă automată de clusterizare a datelor similare. Cu alte cuvinte, se dă un set de date:

$$\{x^{(1)}, \dots, x^{(m)}\}, \quad (2.1)$$

Unde $x^{(i)}$ este n dimensional.

Și se dorește gruparea datelor într-un număr k de clustere.

Algoritmul K-Means este o procedură iterativă care începe prin alegerea aleatoare a unor centroizi, urmând ca aceștia să fie adaptați prin atribuirea vectorilor către cel mai apropiat centroid și recalcularea centroizilor în funcție de vectorii care le-au fost atribuiți.

Pașii parcurși în realizarea algoritmului sunt:

1. Inițializarea centroizilor
2. Repetă pentru fiecare vector $x^{(i)}$ din set:
 - a) Se calculează distanțele sale la mediile tuturor claselor

$$d_k(x^{(i)}, \mu_k) = \|x_j^{(i)} - \mu_k\|^2 = \sqrt{\sum_{j=1}^m (x_j - \mu_k)^2} \quad (2.2)$$

- b) Se stabilește clasa de apartenență (ce va fi utilizată la iterația viitoare) a vectorului $x^{(i)}$
 - c) Reactualizarea mediilor
3. Dacă (față de iterația anterioară) nici un vector nu a fost mutat în altă clasă (sau media unei clase nu s-a modificat, algoritmul se încheie. Altfel, se reia algoritmul de la pasul 2.

II. Calcularea parametrului β

Odată cunoscuți centroizii și clasele de apartenență a vectorilor de intrare pentru fiecare centroid, se poate determina parametrul β astfel:

$$\beta = \frac{1}{2\sigma^2} \quad (2.3)$$

Unde σ^2 reprezintă varianța și poate fi calculat utilizând următoarea ecuație:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (2.4)$$

III. Calcularea ponderilor pentru stratul de ieșire

În urma aplicării funcției de activare cu bază radială pentru vectorii de intrare, utilizând ecuația 1.2, se obține ieșirea din stratul ascuns. Valorile obținute astfel reprezintă intrări pentru stratul de ieșire.

Funcția de activare pentru neuronii din stratul de ieșire poate fi o funcție liniară de forma:

$$f(x) = x \quad (2.5)$$

3. Desfășurarea lucrării

În această lucrare se urmărește determinarea ieșirii unei rețele neuronale de tip RBF pentru fiecare vector dintr-un set de date de test, utilizând centroizii și ponderile obținute în urma antrenării rețelei cu un alt set de date, numit set de antrenare.

3.1. Prezentarea setului de date

Setul de date de antrenare a fost salvat în matricea \mathbf{X} , unde fiecare linie din matrice reprezintă câte un vector de intrare, iar pe fiecare coloană a matricei este prezentă o caracteristică a vectorului respectiv.

Pentru fiecare vector de intrare, \mathbf{X}_i , există un corespondent d_i care reprezintă clasa de apartenență a vectorului respectiv.

Utilizând acest set de date, a fost antrenată o rețea neuronală de tip RBF, având 20 de neuroni în stratul ascuns. După antrenare, au fost salvați centroizii în variabila \mathbf{C} , valorile parametrului β în variabila \mathbf{B} , iar ponderile din stratul de ieșire în variabila \mathbf{W} .

Matricea $\mathbf{X_t}$ conține vectorii pentru care se dorește determinarea ieșirii din rețeaua RBF. Pentru fiecare vector $\mathbf{X_t}_i$, există un corespondent d_t_i pentru verificarea rezultatelor obținute.

3.2. Determinarea ieșirii rețelei

Pentru încărcarea acestor date de intrare, se va folosi funcția load, sub forma:

```
load('lab05_data01.mat');
```

Obiectiv1: Vizualizarea datelor de antrenare

- 1.1. Să se reprezinte grafic setul de antrenare, \mathbf{X} , în spațiul trăsăturilor, utilizând culori diferite pentru clase diferite.
- 1.2. Pe graficul anterior să se reprezinte și centroizii utilizând un marker diferit de markerul utilizat pentru reprezentarea setului de date.

Observatie: Ne amintim că pentru a reține graficul anterior se utilizează instrucțiunea *hold on*

Obiectiv2: Stabilirea parametrilor utilizați

- dimensiunea spațiului de intrare
- numărul vectorilor de intrare pentru setul de test, $\mathbf{X_t}$
- numărul neuronilor din stratul de ascuns: 20
- numărul neuronilor din stratul de ieșire

Obiectiv3: Stabilirea ieșirii din rețea pentru fiecare vector de test, $\mathbf{X_t}_i$.

- 3.1. Pentru fiecare vector $\mathbf{X_t}_i$ din setul de date se dorește stabilirea ieșirii din stratul ascuns, y_{ih} :
 - se calculează net_{jh} pentru vectorul respectiv conform ecuației:

$$net_{jh} = \|\mathbf{X}_j - \mathbf{c}_h\| = \sqrt{\sum_{i=1}^{Ni} (x_{ji} - c_{hi})^2} \quad (3.1)$$

- se determină ieșirea din fiecare neuron din stratul ascuns, aplicând funcția de activare:

$$f(x) = e^{-\frac{x^2}{2\sigma^2}} = e^{-\beta x^2} \quad (3.2)$$

Observație: Coeficientul $\beta = \frac{1}{2\sigma^2}$ pentru centroidul C_h a fost salvat în vectorul B_h !!!

$$y_{jh} = f(net_{jh}) \quad (3.3)$$

3.2. Pentru vectorul X_{t_i} , se determină valoarea obținută în urma introducerii lui y_{ih} în stratul de ieșire.

- se calculează net_{jo} pentru vectorul de intrare conform ecuației:

$$net_{jo} = w_{0o} + \sum_{h=1}^{N_h} w_{ho} y_{jh} \quad (3.4)$$

- se determină ieșirea din fiecare neuron din stratul de ieșire, aplicând funcția de activare:

$$f(x) = x \quad (3.5)$$

$$y_{jo} = f(net_{jo}) \quad (3.6)$$

Obiectiv4: Stabilirea apartenenței la clasă pentru fiecare vector de test și verificarea rezultatului.

Pentru a determina clasa de apartenență, se alege din stratul de ieșire neuronul care prezintă cel mai mare răspuns pentru vectorul respectiv.

Rezultatul obținut pentru fiecare vector de test va fi verificat utilizând vectorul d_t .