Proiectul are ca scop invatearea supervizata si antrenarea unei retle neuronale cu un strat intermediar (de 16 neuroni), in scopul rezolvarii unei probleme de clasificare binara: va determina dintr un set de imagini cu ciuperci care sunt din specia Agaricum si care sunt din specia Lactarius .

Modelul de predictie al retelei neuronale este:

$\mathcal{M}_{x,X}(a) = g(a^T X)x$, unde a este baza de date (in cazul nostrum A si A_T)

Folosing functia de activare :

27. Funcția SoftPlus parametrizat: $g(z) = a(\ln(e^z + 1) - b)$

Pentru a=0.5 si b=-1

Trebuie sa gasim parametrii optimi X si x prin rezolvarea probemei de minimizare fara constrageri:

$$\min_{x \in \mathbb{R}^m, X \in \mathbb{R}^{(n+1)\times m}} L(e, g\left(\bar{A}X\right)x),$$

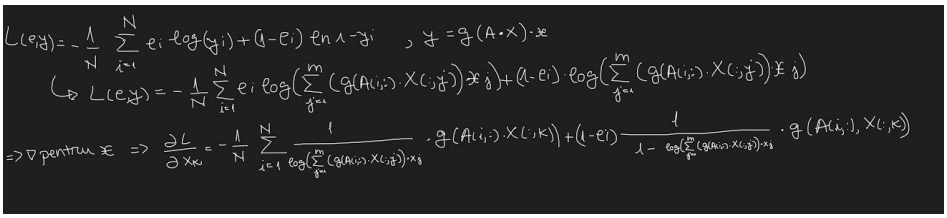
Unde L este functia de pierdere (entropie incrucisata binara):

$$L(e, y) = -\frac{1}{N}\sum_{i=1}^{N} e_i \log y_i + (1 - e_i)\log(1 - y_i),$$

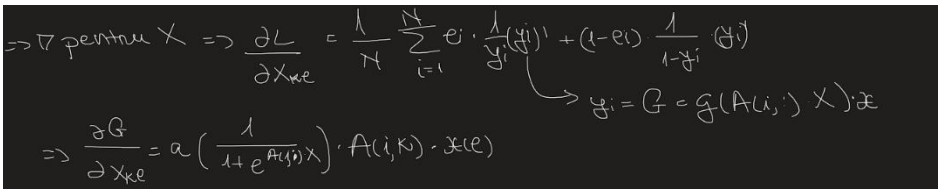
Baza de date:

Voi folosi o baza de date de antrenare de 600 de imagini(300 de imagini cu specia de ciuperca Agaricus si 300 cu specia de Lactarius) si una de testare de 80 de imagini(40 de imagini cu specia de ciuperca Agaricus si 40 cu specia de Lactarius)

Calculul gradientului:



```
function [gradx]=gradx(e,A,X,x)

[N,n]=size(A);
[~,m]=size(X);
gradx=zeros(m,1);

for k=1:m

    sum=0;
    for i=1:N
        sum= sum + e(i)*(1/(SoftPlus_param(A(i,:),X)*x)) * ...
            SoftPlus_param(A(i,:),X(:,k)) + ...
            (1-e(i)) * (1/(1-SoftPlus_param(A(i,:),X)*x)) * ...
            SoftPlus_param(A(i,:),X(:,k));
    end

    gradx(k)=-sum/N;

end

end
```



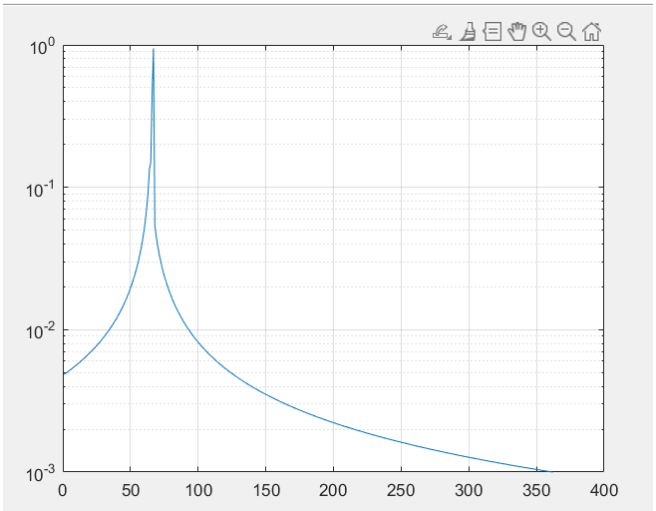
```
function [gradX]=grad_X(e,A,X,x)

a=0.5;

[N,n]= size(A);
[~,m]= size(X);
gradX=zeros(n,m);

for k=1:n

    sum=0;
    for i=1:N
        sum= sum  +e(i)*(1/(SoftPlus_param(A(i,:),X)*x)) *...
            a*(( exp(A(i,:)*X)+ones(1,m)).^(-1) * A(i,k)).*x' +...
        (1-e(i))*(1/(1-SoftPlus_param(A(i,:),X)*x)) *...
            a*(( exp(A(i,:)*X)+ones(1,m)).^(-1) * A(i,k)).*x';
    end

    gradX(k,:)=-sum./N;

end

end
```

Metoda gradient:

Pentru determinarea parametrilor optimi voi folosi metoda de gradul I , Metoda gradient, folosindu-ma de pasul Lipschitz :

```
function [X,x]=MG(e,A,X,x)
er=10e-4;
max_it=1000;

it=0;
f_prev=0;
f=L(e,A,X,x);

ev_crit=[];

while ((abs(f-f_prev)>= er)&& it<=max_it)

    alfax1=1/max(eig(X'*X));
    alfax2=1/max(eig(x'*x));

    gr_X=grad_X(e,A,X,x);
    grad_x=gradx(e,A,X,x);

    X=X+alfax1*gr_X;
    x=x+alfax2*grad_x;

    f_prev=f;
    f=L(e,A,X,x);
    ev_crit=[ev_crit,abs(f-f_prev)];

    it=it+1;

end
        interval=1:it;

        figure();
        semilogy(interval,ev_crit(1:it));
        grid on;
end
```

Evolutia crietriului:



Indicatori de performanta:

```
C =

    35     5
    31     9


precision =

    0.5303


recall =

    0.8750


f1Score =

    0.6604

F1 : 0.66038
```

Metoda Gradient Stocastic

```
p=randperm(N);
A_s=A_1(perm,:);
e_sh=e(perm);
[Xs,xs]=MG(e_s,A_s,X0,x0);
matrice_confuzie(e_s,A_s,Xs,xs);
```