

---

```

%the steps for evaluation are similar to the back_prop_NN.m function

function[y_out,SSE]=NN_predict(x_train,y,W,k,b)
%the function evaluates the input vector of points x_train using the
%parameters of the neural network W,k and bias b. The value of y is
  taken
%to compute the SSE from the predicted output, y_out
M=max(x_train)-min(x_train);
n=length(x_train);
X =[ones(n,1) x_train'];

H=length(k);
N=1;

%sigma=@(u) (1+exp(-u)).^(-1);
%gosavi improvement of sigmoid function
sigma=@(u) (1+exp(-u/M)).^(-1);

%spall method variables
A=10;
B=100;
%l=2; % increased the ratio, avoiding too far ranges of values->closer
(not too much)
l=1;
C=0.1;
t=1/6;
%t=0.5;
tol=0.001;

v_star=zeros(n,H);
for p = 1:n
    for h=1:H

        for j= 1:N+1
            v_star(p,h)=v_star(p,h)+W(j,h)*X(p,j);
        end
    end
end
v=sigma(v_star);
y_out=b*ones(n,1);
for p=1:n
    for h = 1:H
        y_out(p)=y_out(p)+k(h)*v(p,h);
    end
end

%Predicting the y output
for i=1:length(y)
    y_out(i)=(max(y)-min(y))*y_out(i)+min(y);
end

```

---

---

```
SSE=norm(y_out-y);
```

```
Not enough input arguments.
```

```
Error in NN_predict (line 7)
```

```
M=max(x_train)-min(x_train);
```

*Published with MATLAB® R2021a*