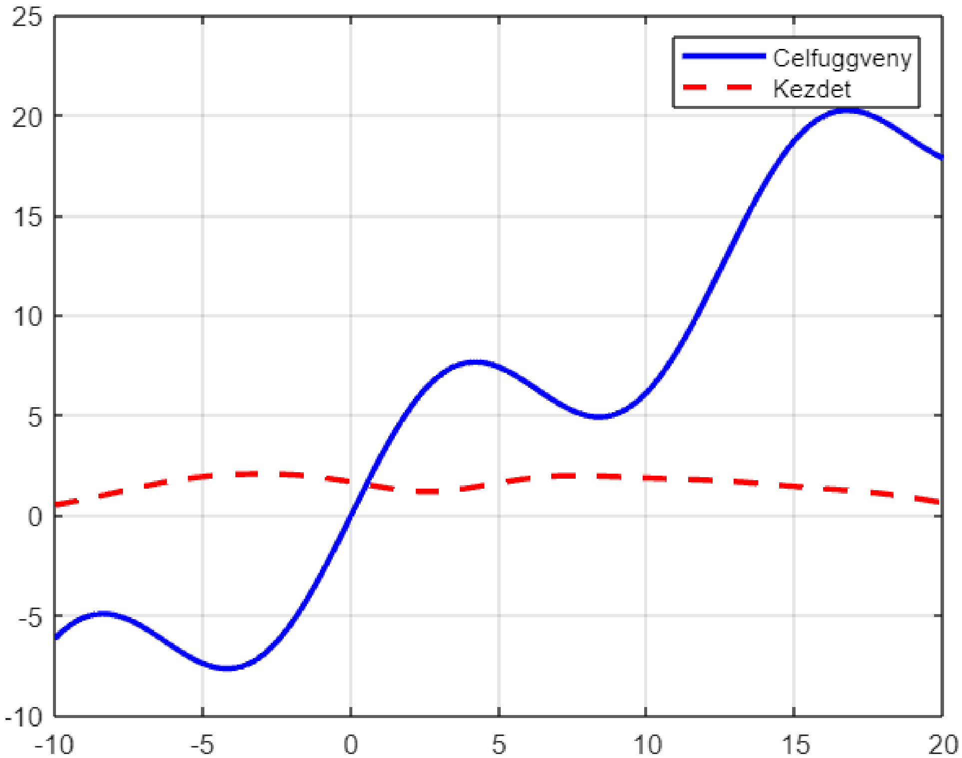
Mesterséges Intelligencia beszámoló

Hamza Balázs

A hatodik labor lényege a RBF (Radiális Bázisfüggvény) hálók használata és tesztelése volt.A keresett célfüggvényt a következöképpen adjuk meg:

```
function [ret] = cel_fuggveny(x , a , b)
ret = x + a * sin(b* x);
```

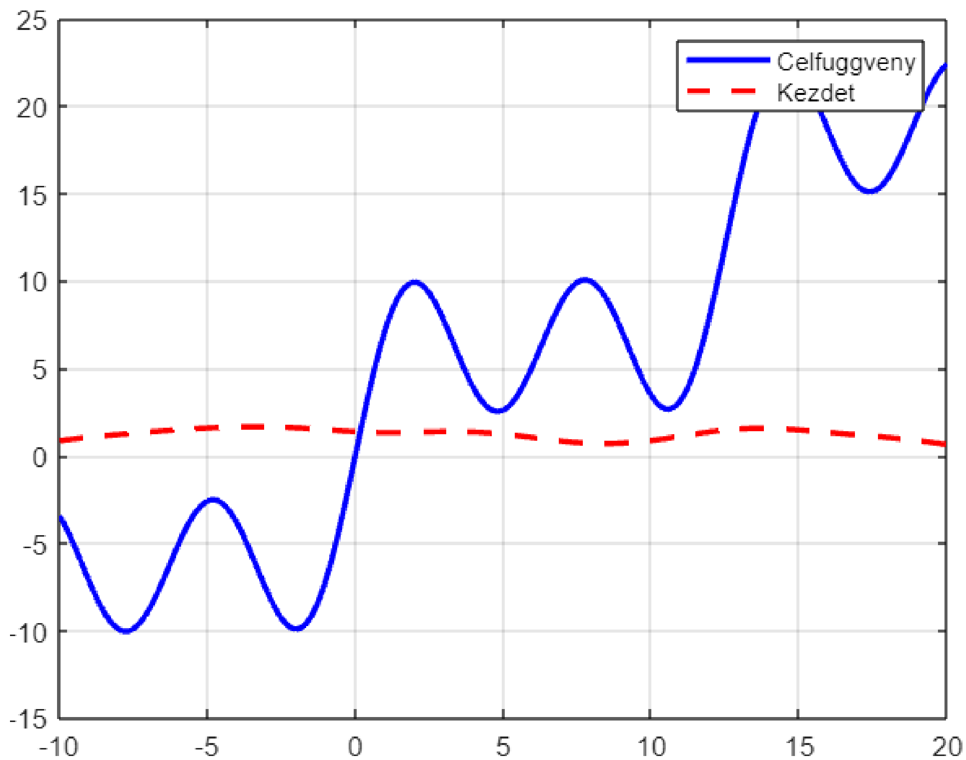
A célfüggvény, ha a = 4, b = 0.5:



A célfüggvényt kibövithetjük még két paraméterrel, ami igy fog kinézni:

```
function [ret] = cel_fuggveny(x , a , b , c ,d)
ret = x + a * sin(b* x) + c * sin(d* x);
```

A célfüggvény, ha a = 4, b =0.5, c = 5, d = 1:



Az RBF hálóhoz szükségünk van egy bázisfüggvényre. Három bázisfüggvényt fogunk használni, az elsö a Gauss bázisfüggvény, a második a köbös bázisfüggvény és a harmadik a multikvadratikus bázisfüggvény. Ezek közül a Gauss bázisfüggvény adja a legjobb eredményeket:

```
%Gauss
function [ret] = bazis(x , c , sigma)
ret = exp(( -(x - c) .^ 2) / (2 * sigma ^ 2));

%köb
%function [ret] = bazis(x)
%ret = x^ 3;

%multikvadratikus
%function [ret] = bazis(x, c)
%ret = ((x.^2 - c.^2) .^1/2);
```

Az RBF háló létrehozásához elsösorban inicializálni kell egy pár adattagot:

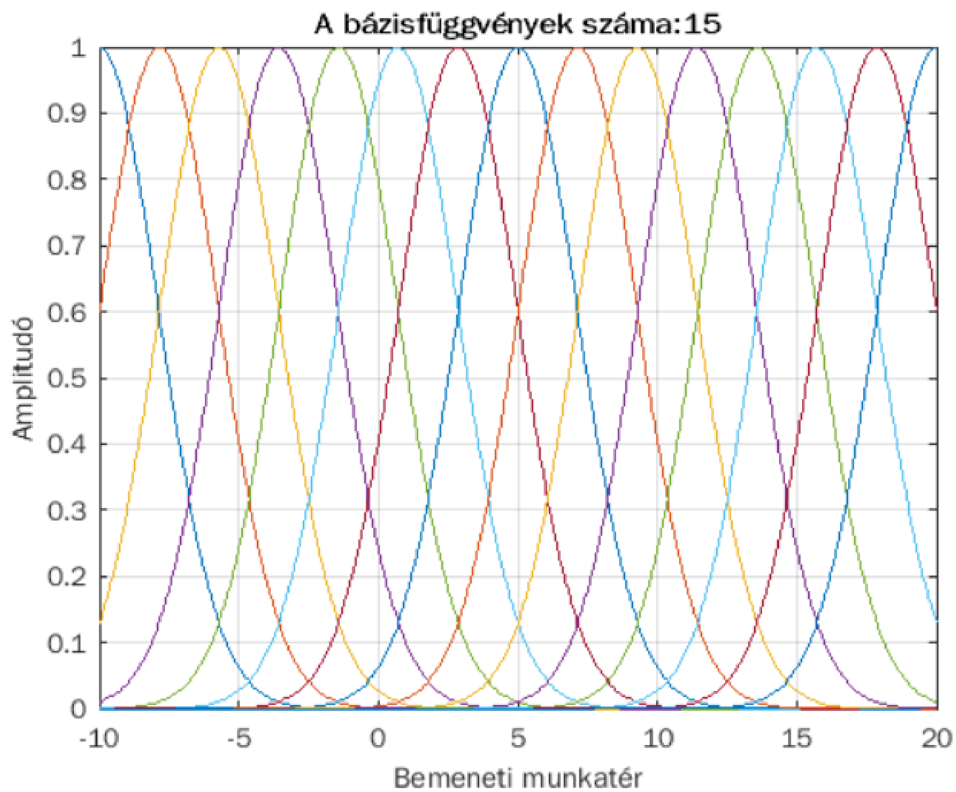
```
min_x=-10; max_x=20; %a bázisfüggvény intervalluma
sigma=2.12; %a bázisfüggvény szélességparamétere
bsz=15;  %bázisfüggvények száma
d_x=0.1  %lépés a bemeneti tér mintavételezésére
mu =[0.0005]; %tanítási együttható
a=4;
b=0.5;
c = 5;
d = 1;
szin = ['g','r','c','m','y','k','g--','r--','c--','m--','y--','k--'];
be=[min_x:d_x:max_x];
ki = cel_fuggveny(be,a,b,c,d); %tanítóhalmaz elõkészítése
n = length(be);
kozeppont = min_x : (max_x - min_x) / (bsz - 1) : max_x; %bázisfüggvények középpont paramétereinek meghatározása
kozeppont_rand = min_x + (max_x - min_x) * rand(1,bsz);
```

(az a, b, c, d együtthatók a célfüggvény létrehozásához szükségesek)

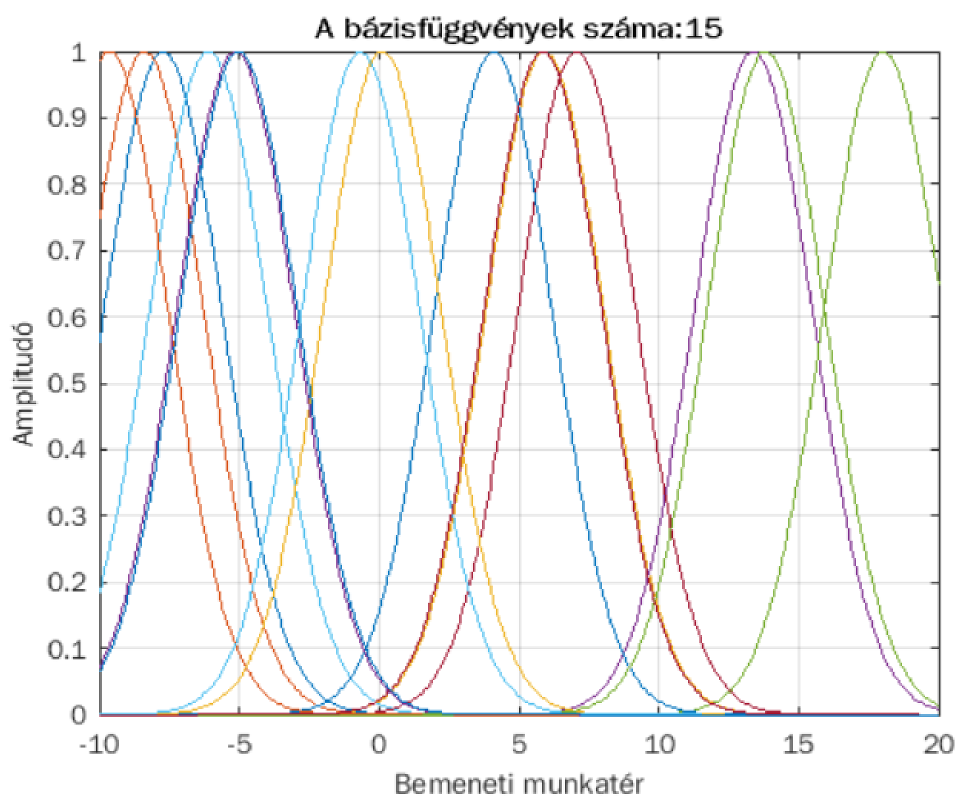
Inicializálás után kirajzoljuk a bázisfüggvényünket:

```
figure(1)
for j=1:bsz
    b=(bazis(be , kozeppont(j) , sigma));
    plot(be,b); grid on;
    hold on;
end
 title(strcat('A bázisfüggvények száma:',num2str(bsz)));
  xlabel('x');
  ylabel('g(x)');

w = rand(bsz , 1);
  xlabel('Bemeneti munkatér');
  ylabel('Amplitudó');
%a tanulás idõbeli alakulása
print -dpng  -r300 bazisfuggvenyek_elhelyezese
```

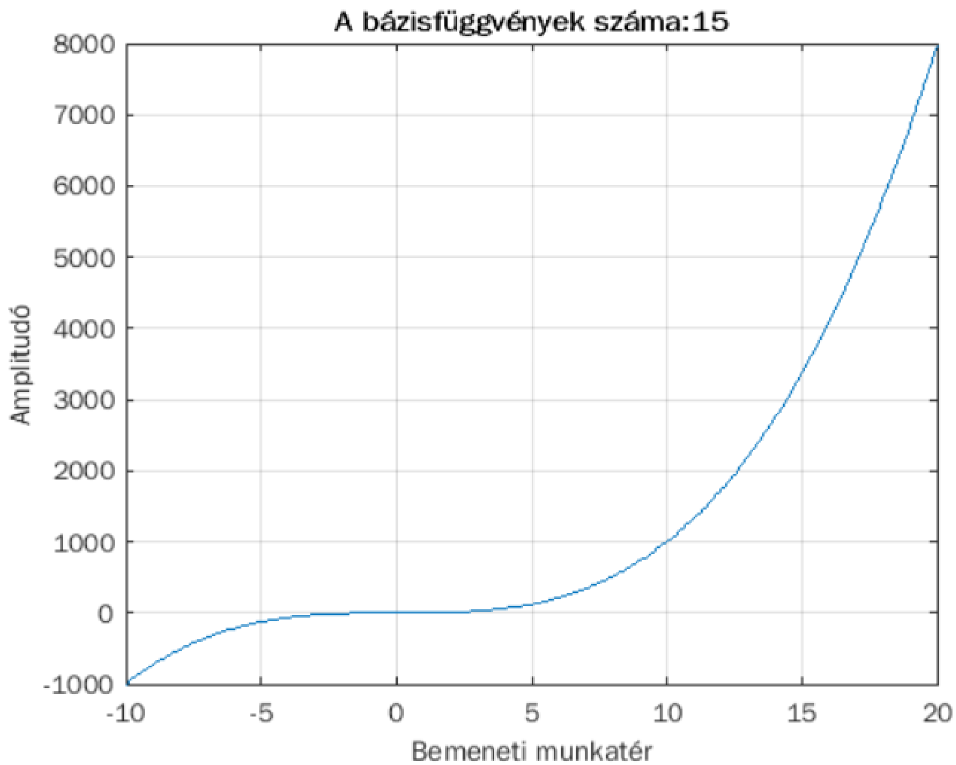
A bázisfüggvény, egyenletes eloszlással (Gauss):



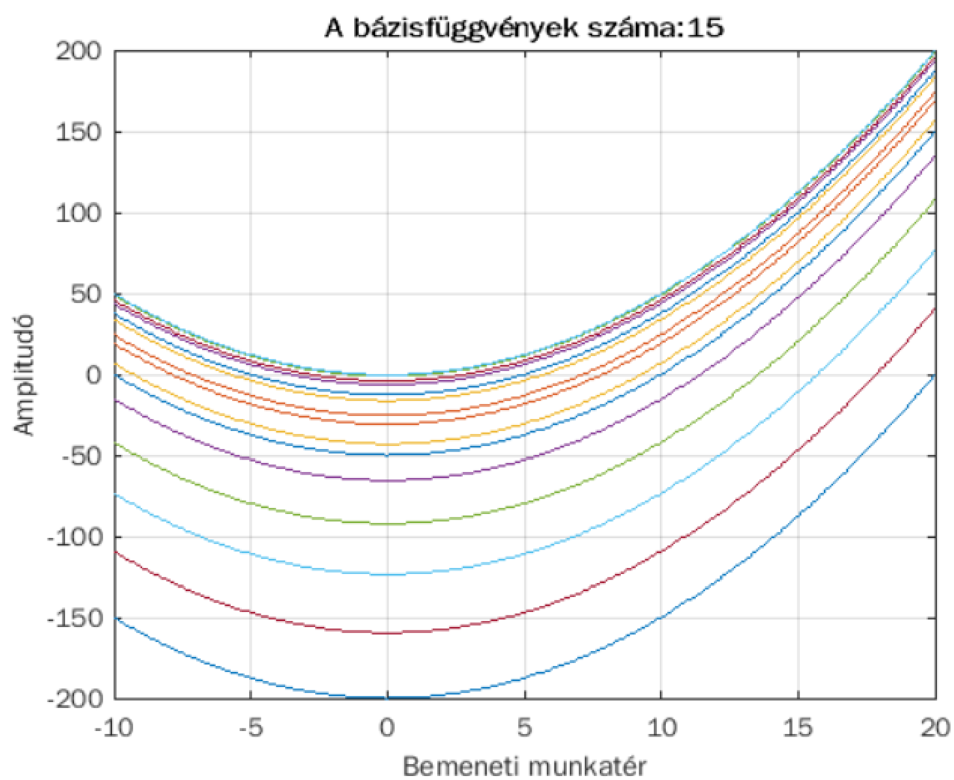
A bázisfüggvény, véletlenszerü eloszlással (Gauss):



A bázisfüggvénny (köbös képlet):



A bázisfüggvény (multikvadratikus):



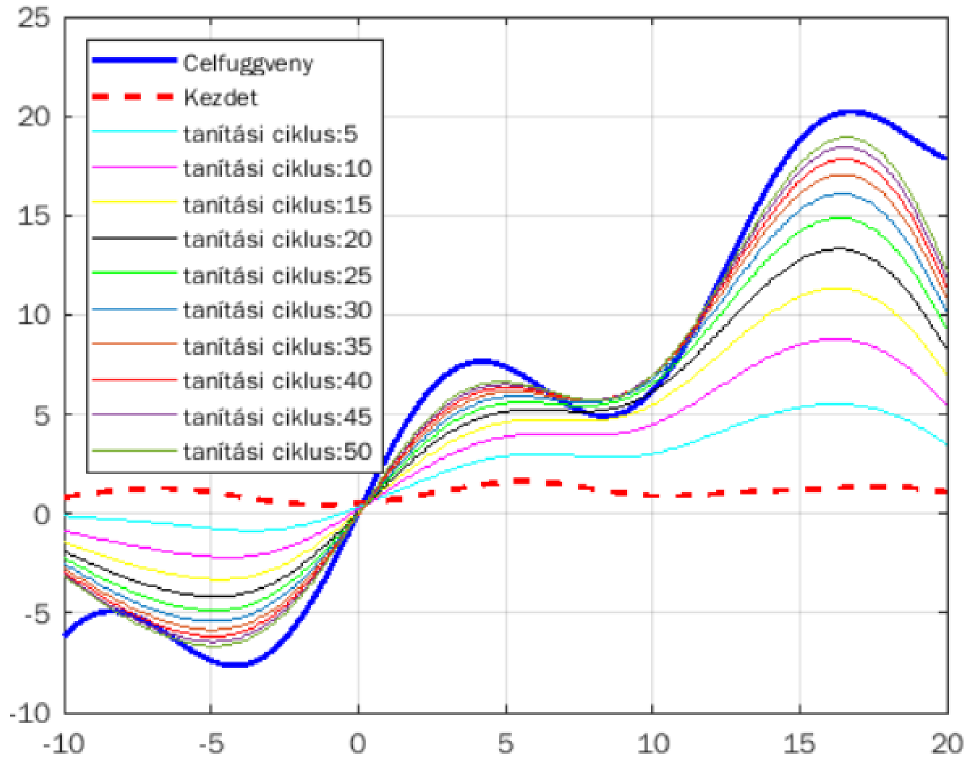
A második figurebe ábrázoljuk a célfüggvényünket és a kezdeti vonalunkat:

```
figure(2);
plot(be,ki,'b','LineWidth',2);grid on;
mlegend{1}=strcat('Celfuggveny');

hold all;
for i = 1 : n,
    yy(i)  = w' * (bazis(be(i) , kozeppont , sigma))';
end
plot(be,yy,'r--','LineWidth',2);
mlegend{2}=strcat('Kezdet');
legend(mlegend);
display('A továbblépéshez nyomd le az ENTER billentyüt')
pause
```

Ha az ENTER billentyüt lenyomjuk, elindul a tanitás:

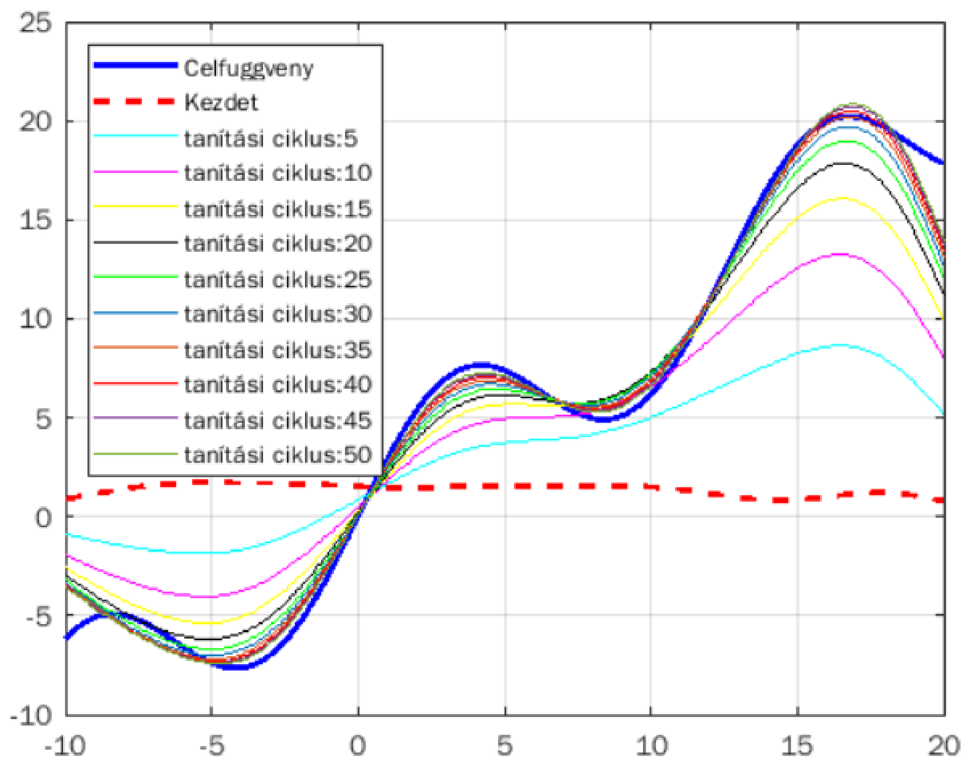
```
l_index=2;
for t = 1 : 50, %tanítási ciklusok ismétlése
    for i = 1 : n, %tanitóminták indexelése
        y  = w' * (bazis(be(i) , kozeppont , sigma))'; %-a háló kiemenetének a kiszámolása
        hiba(i) = ki(i) - y;   % hiba számolás
        w = w + mu(1) * hiba(i) * (bazis(be(i) , kozeppont , sigma))'; %sulyzótanítás
    end;
    negyzetes_hiba(t) = hiba * hiba'; %hibaösszegzés

   % Tanítás alakulása idõben ábrázolva
    if (mod(t,5)==0) %minden ötödik tanítási ciklus után az eredmény ábrázolása
        figure(2);
        hold all;
        for i = 1 : n,
            yy(i)  = w' * (bazis(be(i) , kozeppont , sigma))';
        end
        l_index=l_index+1;
        plot(be,yy,szin(mod(l_index,14)));
        mlegend{l_index}=strcat('tanítási ciklus: ',num2str(t));
        pause(1)
    end;
end;
legend(mlegend,'Location','northwest');
print -dpng  -r300 tanitas_kovetese
```

Minden ötödik tanitási ciklust ábrázoljuk, igy tudjuk követni a változást (tanitási együttható = 0.0005):

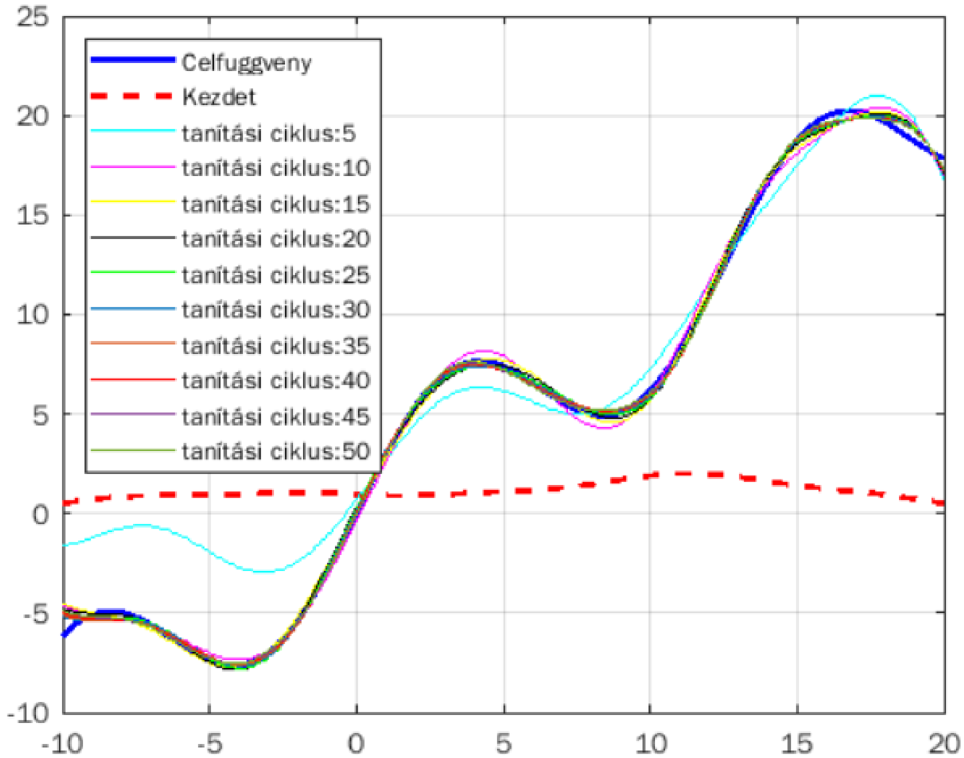


Észre vehetjük, hogy minden ciklussal közeledik a célfüggvény fele a programunk. Ha a tanitási együtthatót növeljük, hamarabb elérjük a célfüggvényt.

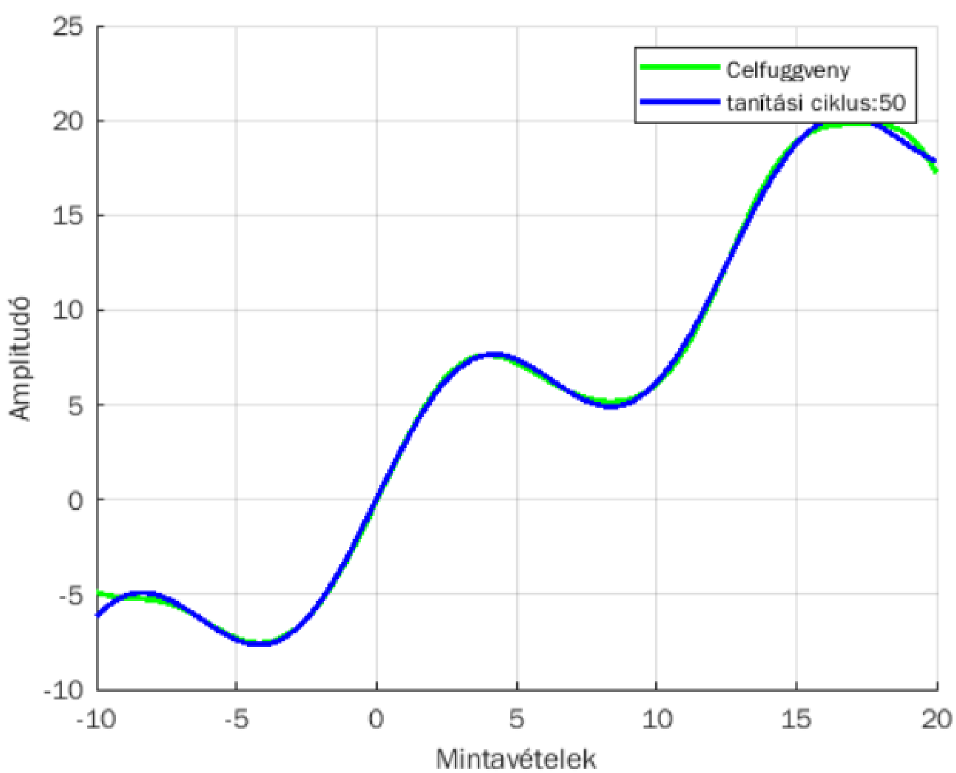
Ha a tanitási együttható = 0.001:



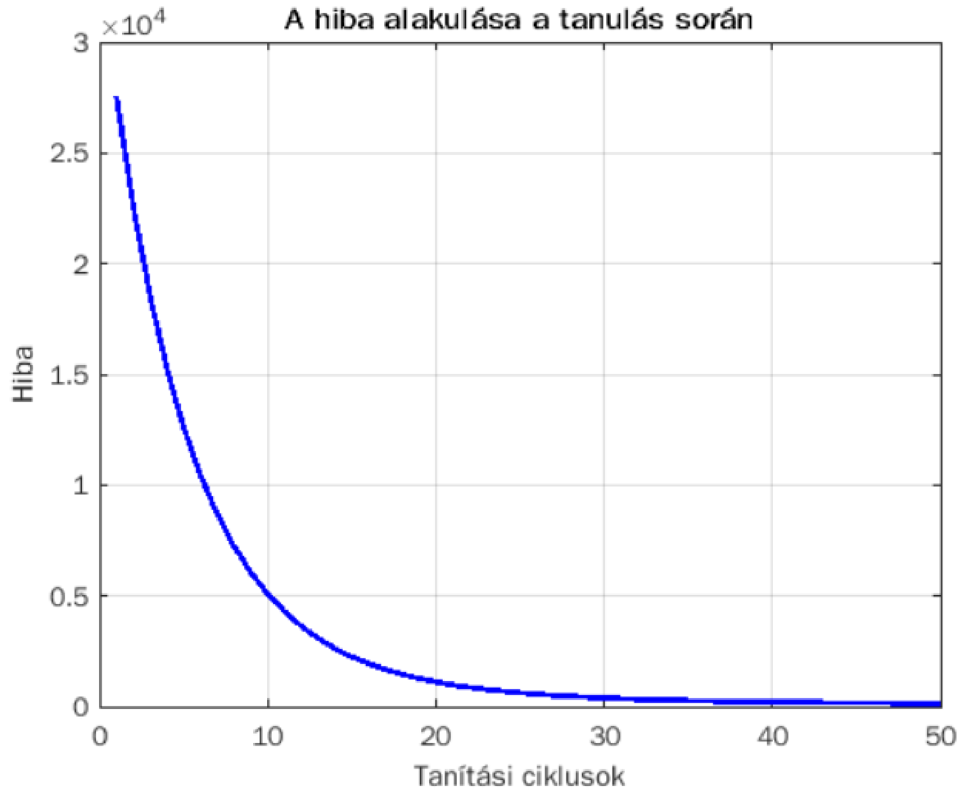
Ha a tanitási együttható = 0.1:



A harmadik figurebe kirajzoljuk a célfüggvényt és a tanitás eredményét:

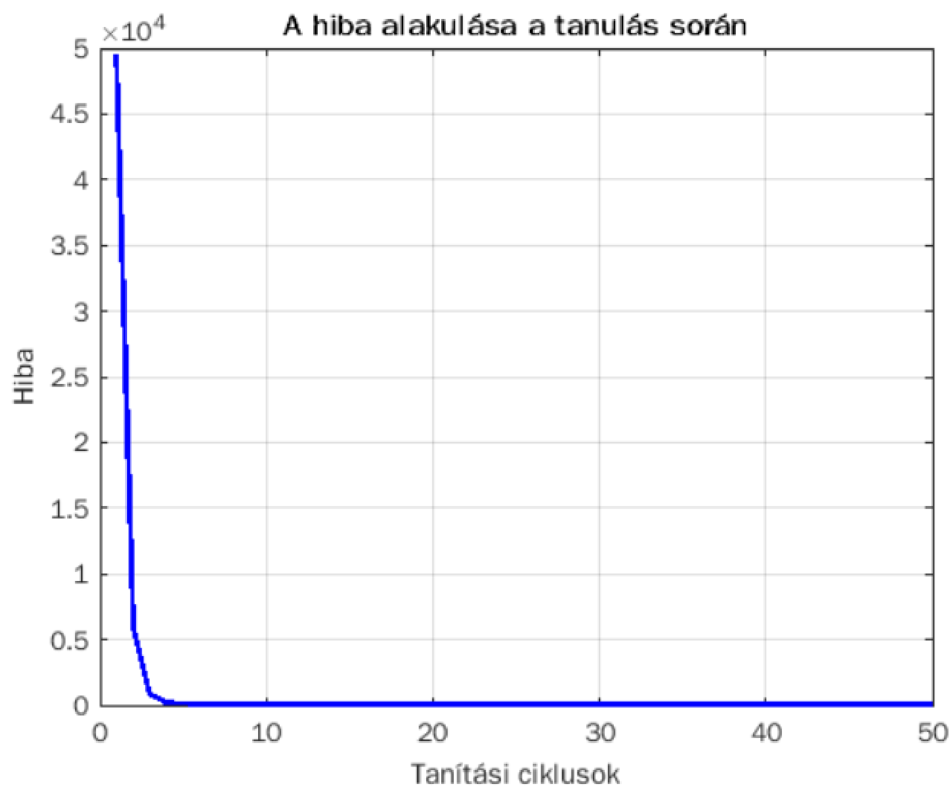


Az utolsó figurebe rajzoljuk ki a hibát, ha a hiba a 0-t közeliti, azt jelenti, hogy jol betanult a programunk:



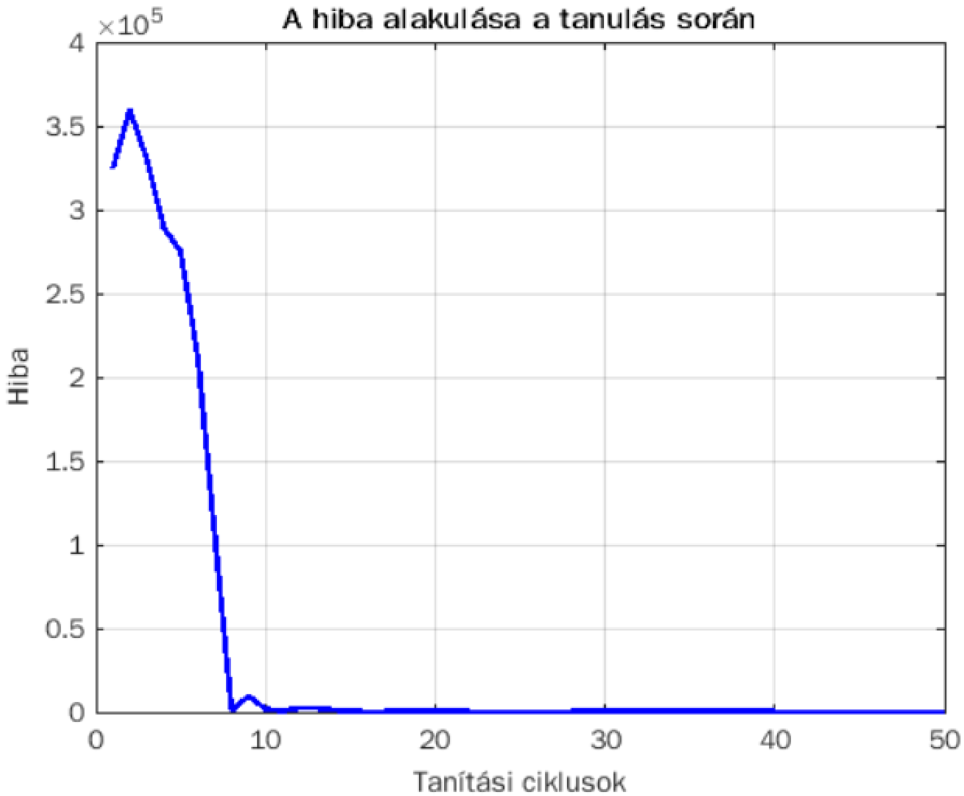
Ha több bázisfüggvényt használunk, hamarabb megoldáshoz jutunk, de létezik olyan eset, hogy tul sok bázisfüggvényt használtunk. Ha tul sok bázisfüggvényt használunk akkor a program leterhelödik, mivel nagyon sok a számolás.

Hiba alakulása ha a bázisfüggvények száma = 500



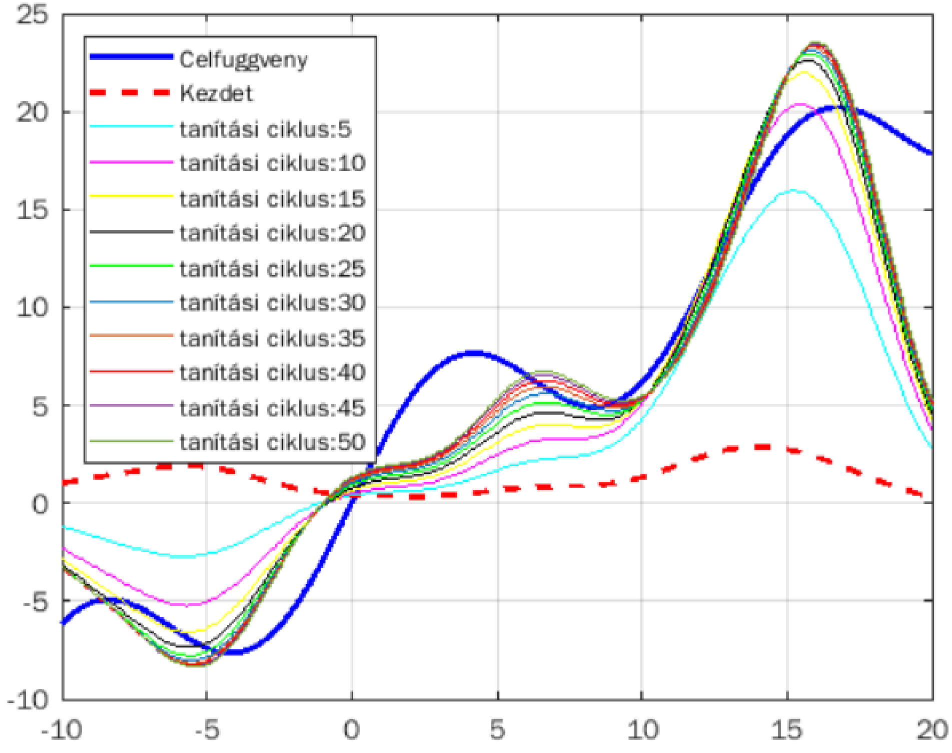
(látható hogy 3-4 ciklus után már betanult a program)

Hiba alakulása, ha a bázisfüggvények száma = 10000



(látható, hogy nem egyenletesen csökken a hibák száma, így ha a bázisfüggvények száma oriási, nem biztos hogy jó megoldáshoz jutunk)

Ha véletlenszerü eloszlású bázisfüggvényt használunk, észrevehetjük hogy bizonyos pontokban nem áll be a program:



Más bázisfüggvényekkel is betanithatjuk a programot:

Gauss bázisfüggvény, bázisfüggvények száma 15, tanitási együttható 0.001