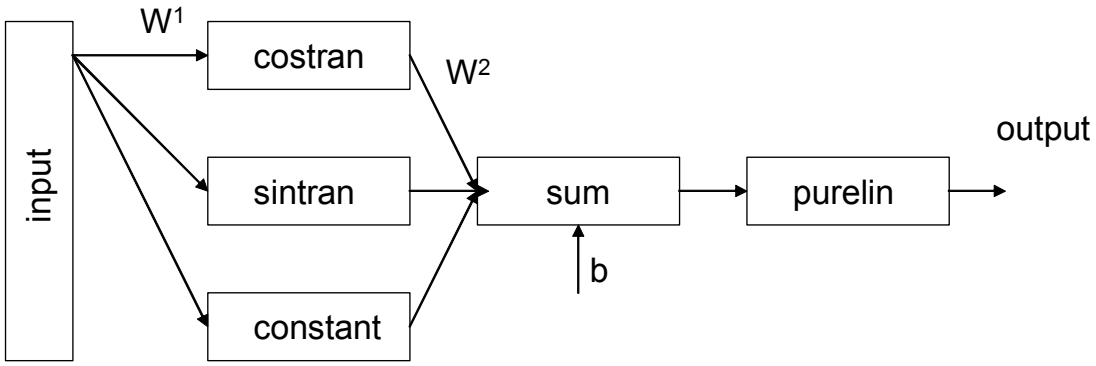
ENEE434 HW#3 solution YJ 02/29/2006

1). a) Matlab function costran.m and sintran.m

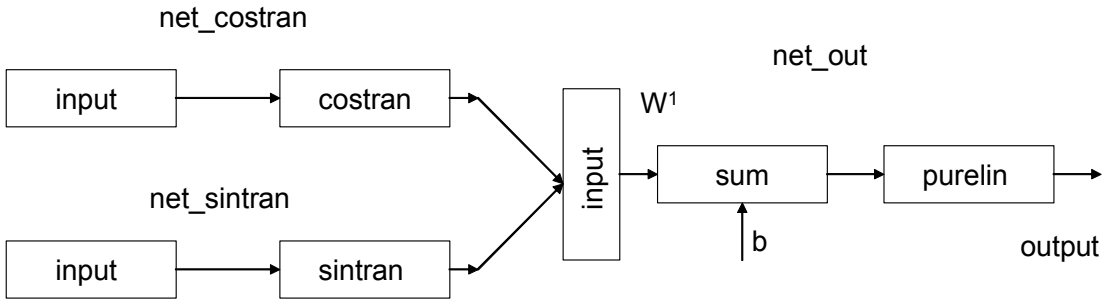
modify tansig.m as following to get costran.m (similar for sintran.m)

```
a = cos(n);
i = find(~finite(a));
a(i) = sign(n(i));
```

b)



c) To train the neural network to give function f(x)=3x+5exp(x/2) as output, with fixed W^1 to give the fouriour series, only W^2 and b^2 are trained. There are various ways to achieve this, the following is one of them:



Three neural networks, net_costran and net_sintran with fixed weight [1 2 3 4] and bias 0. Feed the outputs of net_costran and net_sintran and input to net_out. The weights and bias of net_out is the Fourier series of f(x).

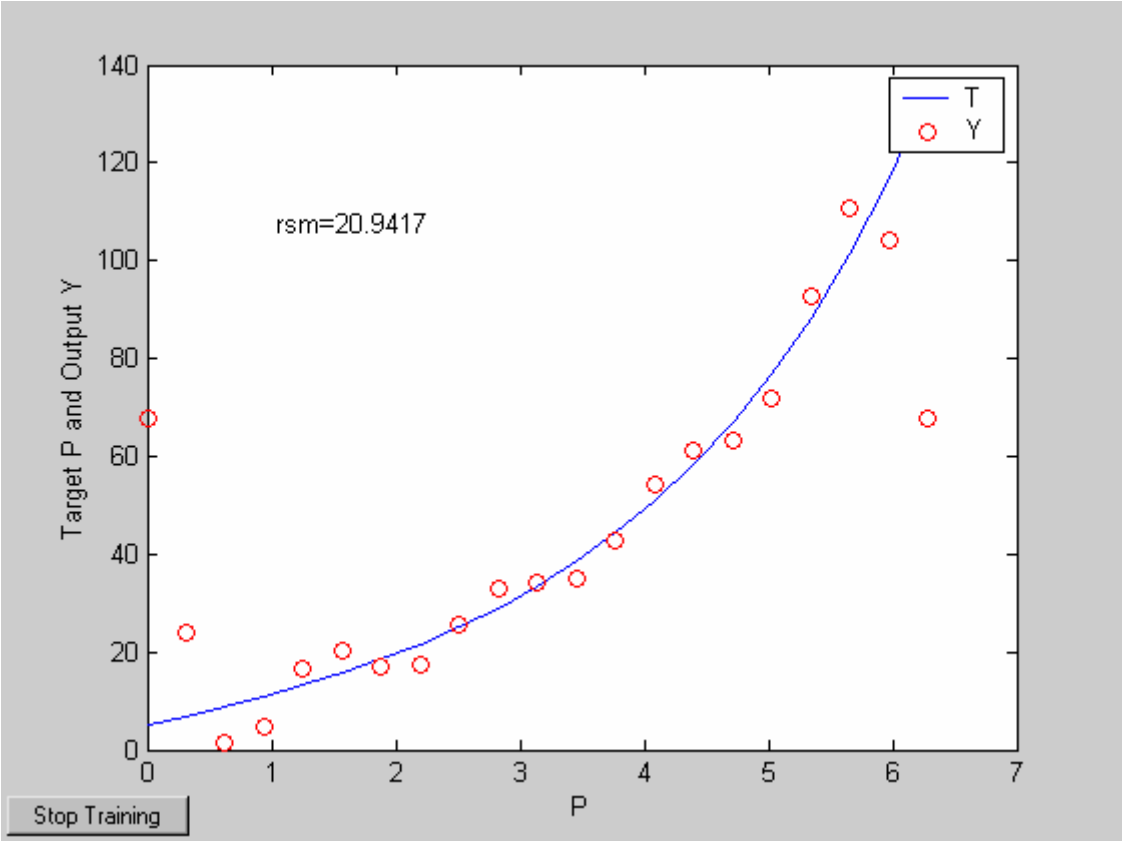
Matlab code for net_costran (similar for net_sintran).

```
P=[0:0.05:1]*2*pi;

%define net_costran
net_costran=newff([-1 8],[4],{'costran'});
net_costran.IW{1,1}=[1;2;3;4];
net_costran.b{1,1}=[0;0;0;0];
Y_cos = sim(net_costran,P);
```

Matlab code for net_out and to find the Fourier series coefficients:

```
%define net_out
P_out=[Y_cos;Y_sin];
net_out=newff([-100 100; -100 100; -100 100; -100 100; -100 100; -100 100; -100 100;
-100 100],[1],{'purelin'});

net_out.trainParam.epochs = 50;
```

```
net_out = train(net_out,P_out,T);

Y=sim(net_out,P_out);

% To find Fourier series coeficients:
% y=f(x)=c0+sum(an*cos(nx))+bn*sin(nx));

co_nn=net_out.b{1,1}
an_nn=net_out.IW{1,1}(1:4)
bn_nn=net_out.IW{1,1}(5:8)
```



Result from matlab simulation: (f(x)=c0+sum(an*cos(nx))+bn*sin(nx))

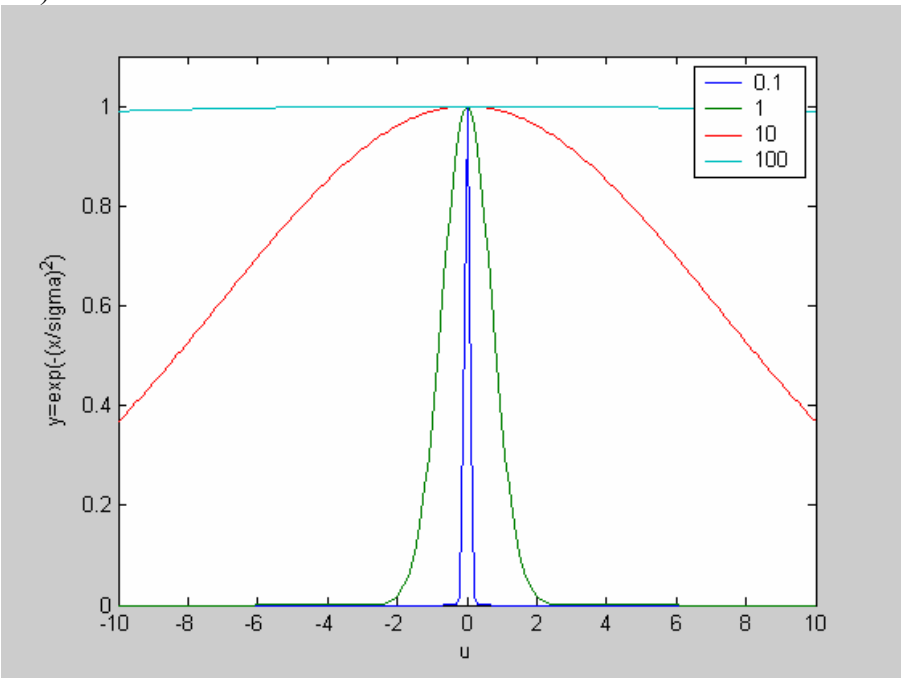
c_0	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4
44.8459	14.4621	4.5148	2.2776	1.4626	-33.8510	-18.9005	-12.3968	-8.7818

d) Hand calculation:

c_0	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4
44.6698	14.0891	4.1348	1.8928	1.0719	-34.2026	-19.5901	-13.4338	-10.1780

If using matlab function fft, notice that fft is a different form of Fourier transform, need to recalculate the coefficients.

2 a)



b) Matlab code for set up RBF neural network

```
netradialbase = newrb(P,T,0.01, 1);
Y_r = sim(netradialbase,P);

rms=sqrt(mse(Y_r-T))
```