



Chapter2

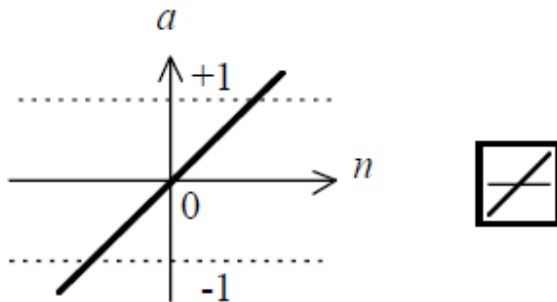
Transfer Functions

List of Transfer Functions:

Transfer Function		
compet	Competitive transfer function.	
hardlim	Hard limit transfer function.	
hardlims	Symmetric hard limit transfer function	
logsig	Log sigmoid transfer function.	
poslin	Positive linear transfer function	
purelin	Linear transfer function.	
radbas	Radial basis transfer function.	
satlin	Saturating linear transfer function.	
satlins	Symmetric saturating linear transfer function	
softmax	Softmax transfer function.	
tansig	Hyperbolic tangent sigmoid transfer function.	
tribas	Triangular basis transfer function.	



Transfer Function Graphs

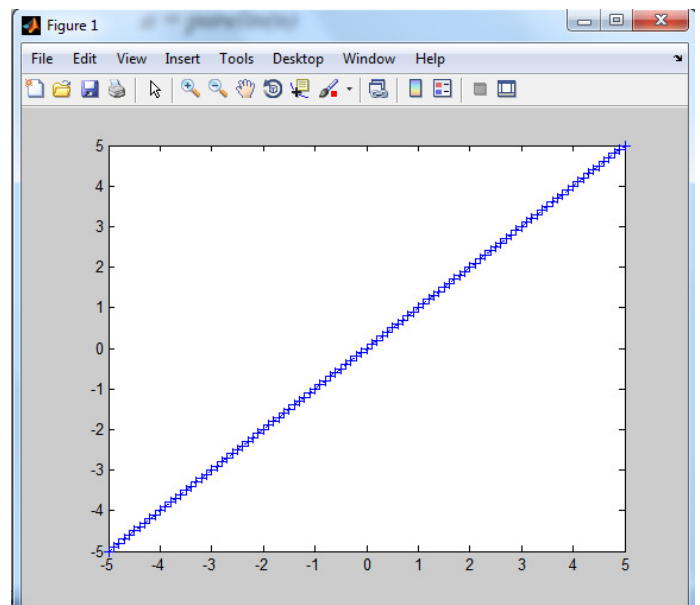


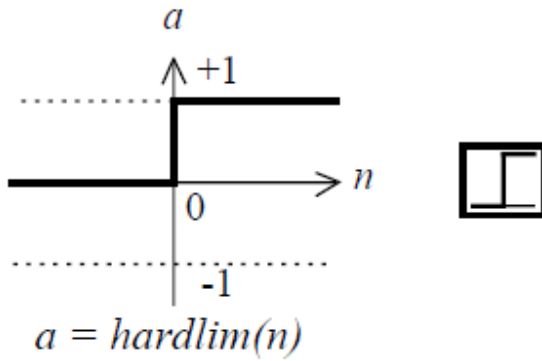
$$a = \text{purelin}(n)$$

Linear Transfer Function

Example1:

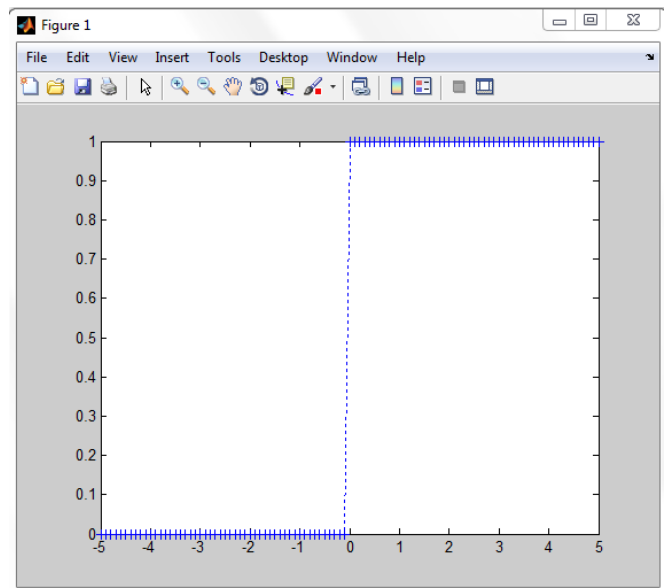
```
%%% Linear Transfer Function %%%%%%%%%%  
n = -5:0.1:5;  
plot(n,purelin(n), 'b+');
```

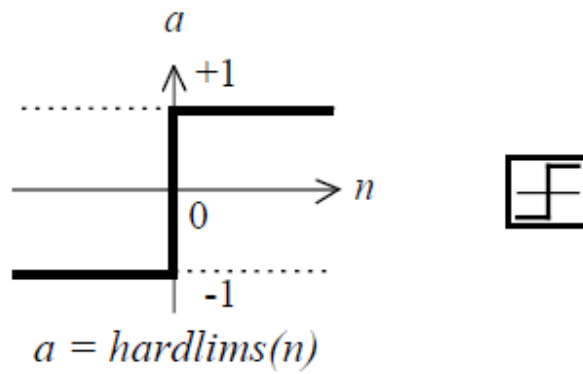




Hard-Limit Transfer Function

```
Ex:%%% Hard-Limit Transfer Function %%%%%%%%%%  
n = -5:0.1:5;  
plot(n,hardlim(n), 'b+');
```

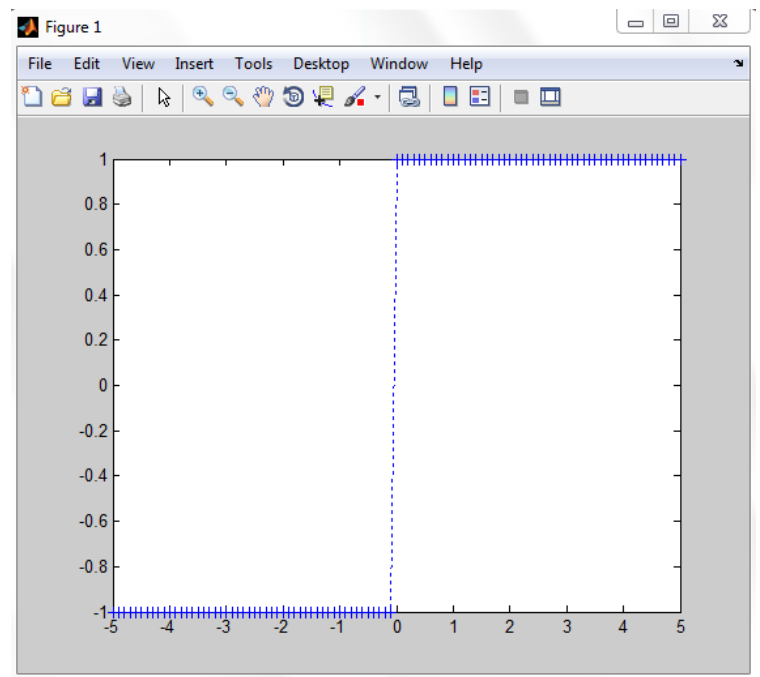


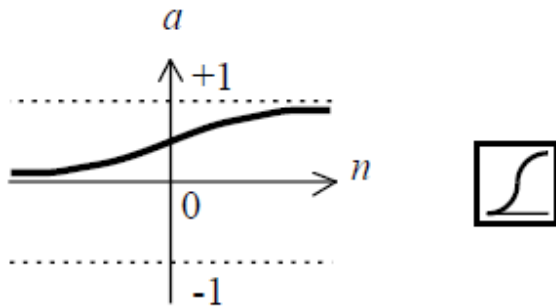


Symmetric Hard-Limit Trans. Funct.

%%% Symmetric Hard-Limit Transfer Function %%%

```
n = -5:0.1:5;  
plot(n,hardlims(n),'b+');
```

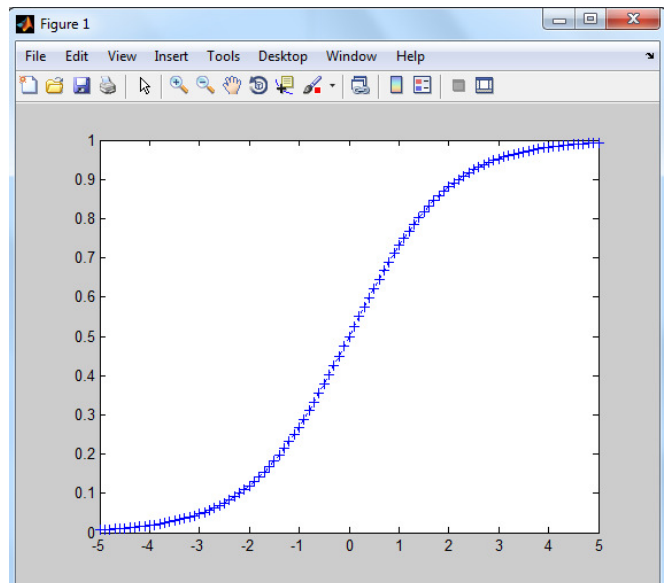


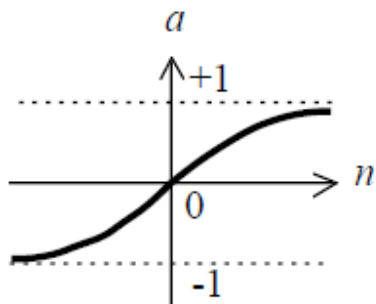


$$a = \text{logsig}(n)$$

Log-Sigmoid Transfer Function

```
Ex: %% Log-Sigmoid Transfer Function %%%  
n = -5:0.1:5;  
plot(n,logsig(n), 'b+');
```

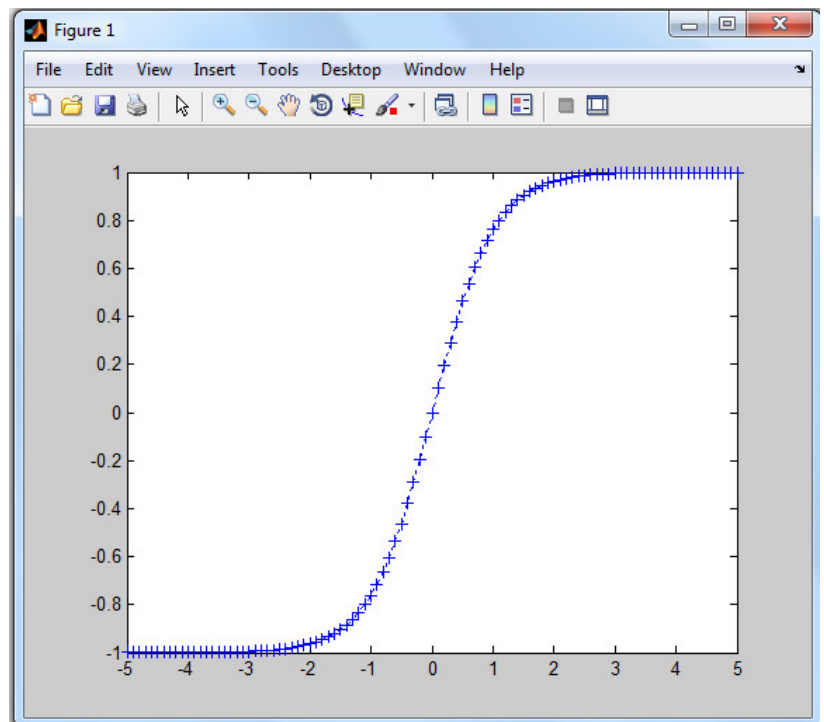




$$a = \text{tansig}(n)$$

Tan-Sigmoid Transfer Function

```
%%% Symmetric Hard-Limit Transfer Function %%%%%%%%%%  
n = -5:0.1:5;  
plot(n,tansig(n), 'b+');
```

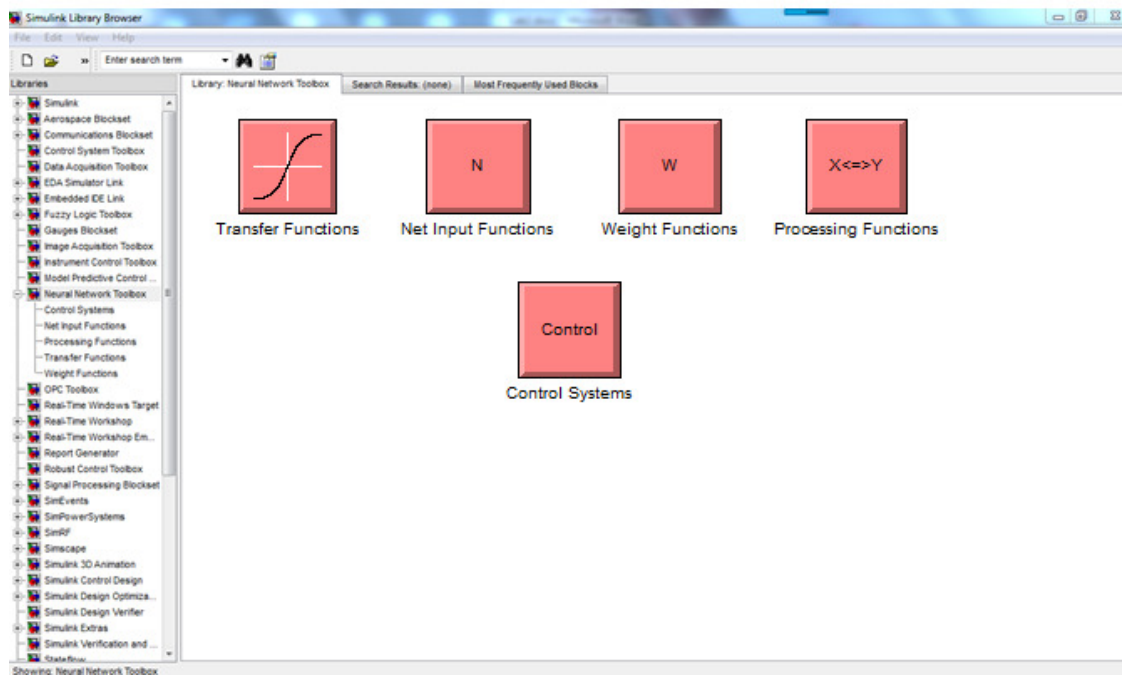




1- `>> simulink` (in command windows in matlab)

2- ENTER

3- Neural Network Toolbox



Double-click on the Transfer Functions block in the **Neural** window to bring up a window containing several transfer function blocks.

