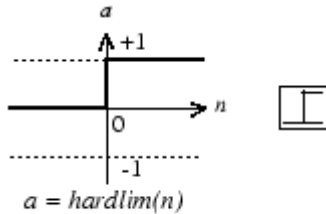


hardlim

Hard-limit transfer function

Graph and Symbol



Hard-Limit Transfer Function

Syntax

```
A = hardlim(N,FP)
dA_dN = hardlim('dn',N,A,FP)
info = hardlim('code')
```

Description

hardlim is a neural transfer function. Transfer functions calculate a layer's output from its net input.

$A = \text{hardlim}(N,FP)$ takes N and optional function parameters,

N	S -by- Q matrix of net input (column) vectors
FP	Struct of function parameters (ignored)

and returns A , the S -by- Q Boolean matrix with 1s where $N \geq 0$.

$dA_dN = \text{hardlim}('dn',N,A,FP)$ returns the S -by- Q derivative of A with respect to N . If A or FP is not supplied or is set to `[]`, FP reverts to the default parameters, and A is calculated from N .

$info = \text{hardlim}('code')$ returns information according to the code string specified:

hardlim('name') returns the name of this function.

hardlim('output',FP) returns the [min max] output range.

hardlim('active',FP) returns the [min max] active input range.

hardlim('fullderiv') returns 1 or 0, depending on whether dA_dN is S -by- S -by- Q or S -by- Q .

hardlim('fpname') returns the names of the function parameters.

hardlim('fpdefaults') returns the default function parameters.

Examples

Here is how to create a plot of the `hardlim` transfer function.

```
n = -5:0.1:5;  
a = hardlim(n);  
plot(n,a)
```

Assign this transfer function to layer `i` of a network.

```
net.layers{i}.transferFcn = 'hardlim';
```

Algorithms

`hardlim`(n) = 1 if $n \geq 0$
0 otherwise

See Also

[hardlims](#) | [sim](#)

Was this topic helpful?

Yes

No