

Deep Neuronal Filter

Generated by Doxygen 1.8.17

1 Deep Neuronal Filter (DNF)	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 DNF Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Constructor & Destructor Documentation	6
3.1.2.1 DNF()	6
3.1.3 Member Function Documentation	6
3.1.3.1 filter()	6
Index	7

Chapter 1

Deep Neuronal Filter (DNF)

A noise reduction filter using deep networks in autoencoder configuration.

github: <https://github.com/berndporr/deepNeuronalFilter>

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

DNF	Main Deep Neuronal Network main class	5
---------------------	---	-------------------

Chapter 3

Class Documentation

3.1 DNF Class Reference

Main Deep Neuronal Network main class.

```
#include <dnf.h>
```

Public Member Functions

- [DNF](#) (const int NLAYERS, const int numTaps, double fs, Neuron::actMethod am=Neuron::Act_Tanh)
Constructor which sets up the delay lines, network layers and also calculates the number of neurons per layer so that the final layer always just has one neuron.
- double [filter](#) (double signal, double noise)
Realtime sample by sample filtering operation.
- Net & [getNet](#) () const
Returns a reference to the whole neural network.
- const int [getSignalDelaySteps](#) () const
Returns the length of the delay line which delays the signal polluted with noise.
- const double [getDelayedSignal](#) () const
Returns the delayed signal by the delay indicated by [getSignalDelaySteps\(\)](#).
- const double [getRemover](#) () const
Returns the remover signal.
- const double [getOutput](#) () const
Returns the output of the [DNF](#): the the noise free signal.
- [~DNF](#) ()
Frees the memory used by the [DNF](#).

3.1.1 Detailed Description

Main Deep Neuronal Network main class.

It's designed to be as simple as possible with only a few parameters as possible.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 DNF()

```
DNF::DNF (
    const int NLAYERS,
    const int numTaps,
    double fs,
    Neuron::actMethod am = Neuron::Act_Tanh ) [inline]
```

Constructor which sets up the delay lines, network layers and also calculates the number of neurons per layer so that the final layer always just has one neuron.

Parameters

<i>NLAYERS</i>	Number of layers
<i>numTaps</i>	Number of taps for the delay line feeding into the 1st layer
<i>fs</i>	Sampling rate of the signals used in Hz.
<i>am</i>	The activation function for the neurons. Default is tanh.

3.1.3 Member Function Documentation

3.1.3.1 filter()

```
double DNF::filter (
    double signal,
    double noise ) [inline]
```

Realtime sample by sample filtering operation.

Parameters

<i>signal</i>	The signal contaminated with noise. Should be less than one.
<i>noise</i>	The reference noise. Should be less than one.

The documentation for this class was generated from the following file:

- dnf.h

Index

DNF, [5](#)
 DNF, [6](#)
 filter, [6](#)

filter
 DNF, [6](#)