

# Testing

For those who want to implement their own modules, we suggest using the `nn.Jacobian` class for testing the derivatives of their class, together with the `torch.Tester` class. The sources of `nn` package contains sufficiently many examples of such tests.

## `nn.Jacobian`

### `testJacobian(module, input, minval, maxval, perturbation)`

Test the jacobian of a module w.r.t. to its input.

`module` takes as its input a random tensor shaped the same as `input`.  
`minval` and `maxval` specify the range of the random tensor `([-2, 2]` by default).  
`perturbation` is used as finite difference (1e-6 by default).

Returns the L-inf distance between the jacobian computed by backpropagation and by finite difference.

### `testJacobianParameters(module, input, param, dparam, minval, maxval, perturbation)`

Test the jacobian of a module w.r.t. its parameters (instead of its input).

The input and parameters of `module` are random tensors shaped the same as `input` and `param`.  
`minval` and `maxval` specify the range of the random tensors `([-2, 2]` by default).  
`dparam` points to the gradient w.r.t. parameters.  
`perturbation` is used as finite difference (1e-6 by default).

Returns the L-inf distance between the jacobian computed by backpropagation and by finite difference.

## testJacobianUpdateParameters(module, input, param, minval, maxval, perturbation)

Test the amount of update of a module to its parameters.

The input and parameters of `module` are random tensors shaped the same as `input` and `param`.

`minval` and `maxval` specify the range of the random tensors  $[-2, 2]$  by default).

`perturbation` is used as finite difference ( $1e-6$  by default).

Returns the L-inf distance between the update computed by backpropagation and by finite difference.

## forward(module, input, param, perturbation)

Compute the jacobian by finite difference.

`module` has parameters `param` and input `input`.

If provided, `param` is regarded as independent variables, otherwise `input` is the independent variables.

`perturbation` is used as finite difference ( $1e-6$  by default).

Returns the jacobian computed by finite difference.

## backward(module, input, param, dparam)

Compute the jacobian by backpropagation.

`module` has parameters `param` and input `input`.

If provided, `param` is regarded as independent variables, otherwise `input` is the independent variables.

`dparam` is the gradient w.r.t. parameters, it must present as long as `param` is present.

Returns the jacobian computed by backpropagation.