

Module: `tf.contrib.distributions.bijectors`[Contents](#)[Classes](#)Defined in `tensorflow/contrib/distributions/python/ops/bijectors/__init__.py`.

Bijector Ops.

## Classes

**class** `AbsoluteValue`: Computes  $Y = g(X) = \text{Abs}(X)$ , element-wise.**class** `Affine`: Compute  $Y = g(X; \text{shift}, \text{scale}) = \text{scale} @ X + \text{shift}$ .**class** `AffineLinearOperator`: Compute  $Y = g(X; \text{shift}, \text{scale}) = \text{scale} @ X + \text{shift}$ .**class** `Bijector`: Interface for transformations of a `Distribution` sample.**class** `Chain`: Bijector which applies a sequence of bijectors.**class** `CholeskyOuterProduct`: Compute  $g(X) = X @ X.T$ ;  $X$  is lower-triangular, positive-diagonal matrix.**class** `ConditionalBijector`: Conditional Bijector is a Bijector that allows intrinsic conditioning.**class** `Exp`: Compute  $Y = g(X) = \exp(X)$ .**class** `Identity`: Compute  $Y = g(X) = X$ .**class** `Inline`: Bijector constructed from custom callables.**class** `Invert`: Bijector which inverts another Bijector.**class** `PowerTransform`: Compute  $Y = g(X) = (1 + X * c)^{(1 / c)}$ ,  $X \geq -1 / c$ .**class** `Sigmoid`: Bijector which computes  $Y = g(X) = 1 / (1 + \exp(-X))$ .**class** `SigmoidCentered`: Bijector which computes  $Y = g(X) = \exp([X \ 0]) / (1 + \exp(-X))$ .**class** `SinhArcsinh`: Compute  $Y = g(X) = \text{Sinh}(\text{Arcsinh}(X) + \text{skewness}) * \text{tailweight}$ .**class** `SoftmaxCentered`: Bijector which computes  $Y = g(X) = \exp([X \ 0]) / \text{sum}(\exp([X \ 0]))$ .**class** `Softplus`: Bijector which computes  $Y = g(X) = \text{Log}[1 + \exp(X)]$ .

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

[Blog](#)

[GitHub](#)

[Twitter](#)

**Support**

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

**English**

[Terms](#) | [Privacy](#)