

tf.contrib.layers.gdn

```
gdn(  
    inputs,  
    inverse=False,  
    beta_min=1e-06,  
    gamma_init=0.1,  
    reparam_offset=2 ** -18,  
    data_format='channels_last',  
    activity_regularizer=None,  
    trainable=True,  
    name=None,  
    reuse=None  
)
```

Defined in [tensorflow/contrib/layers/python/layers/layers.py](#).

Functional interface for GDN layer.

Based on the papers:

"Density Modeling of Images using a Generalized Normalization Transformation" Johannes Ballé, Valero Laparra, Eero P. Simoncelli <https://arxiv.org/abs/1511.06281>

"End-to-end Optimized Image Compression" Johannes Ballé, Valero Laparra, Eero P. Simoncelli
<https://arxiv.org/abs/1611.01704>

Implements an activation function that is essentially a multivariate generalization of a particular sigmoid-type function:

```
y[i] = x[i] / sqrt(beta[i] + sum_j(gamma[j, i] * x[j]))
```

where **i** and **j** run over channels. This implementation never sums across spatial dimensions. It is similar to local response normalization, but much more flexible, as **beta** and **gamma** are trainable parameters.

Args:

- **inputs**: Tensor input.
- **inverse**: If **False** (default), compute GDN response. If **True**, compute IGDN response (one step of fixed point iteration to invert GDN; the division is replaced by multiplication).
- **beta_min**: Lower bound for beta, to prevent numerical error from causing square root of zero or negative values.
- **gamma_init**: The gamma matrix will be initialized as the identity matrix multiplied with this value. If set to zero, the layer is effectively initialized to the identity operation, since beta is initialized as one. A good default setting is somewhere between 0 and 0.5.
- **reparam_offset**: Offset added to the reparameterization of beta and gamma. The reparameterization of beta and gamma as their square roots lets the training slow down when their values are close to zero, which is desirable as small values in the denominator can lead to a situation where gradient noise on beta/gamma leads to extreme amounts of noise in the GDN activations. However, without the offset, we would get zero gradients if any elements of beta or gamma were exactly zero, and thus the training could get stuck. To prevent this, we add this small constant. The default value was empirically determined as a good starting point. Making it bigger potentially leads to more gradient noise on the activations, making it too small may lead to numerical precision issues.

- `data_format` : Format of input tensor. Currently supports `'channels_first'` and `'channels_last'` .
- `activity_regularizer` : Regularizer function for the output.
- `trainable` : Boolean, if `True` , also add variables to the graph collection `GraphKeys.TRAINABLE_VARIABLES` (see `tf.Variable`).
- `name` : String, the name of the layer. Layers with the same name will share weights, but to avoid mistakes we require `reuse=True` in such cases.
- `reuse` : Boolean, whether to reuse the weights of a previous layer by the same name.

Returns:

Output tensor.

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.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

English

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