
panndas

Charles Frye

Apr 01, 2022

CONTENTS

1	Reference	1
2	Installation	5
3	Usage	7
	Python Module Index	9
	Index	11

REFERENCE

- *panndas.nn*

1.1 panndas.nn

class panndas.nn.**AdditiveSkip**(*block*)

A Module that applies an additive “skip” connection around the provided Module.

forward(*xs*)

Applies the Module to its input.

show()

Displays the Module in a human-friendly format.

class panndas.nn.**AlphaDropout**(*p*, *alpha=0.0*)

A Module that randomly replaces a fraction of its inputs with a fixed value.

(Pseudo-)random values are drawn from the random standard library module.

Parameters

- **p** – The probability that any value is masked on a single call. Sensible values are between 0.0 and 1.0, but this is not checked.
- **alpha** – The value that replaces the masked value. Typically set to the neutral value for a following or preceding non-linearity, e.g. 0.0 for ReLU or sigmoid.

forward(*xs*)

Applies the Module to its input.

show()

Displays the Module in a human-friendly format.

class panndas.nn.**BatchNorm1d**(*eps=1e-05*, *gamma=1.0*, *beta=0.0*)

Standardize each feature across batches and set mean/sd to beta/gamma.

forward(*xs*)

Applies the Module to its input.

class panndas.nn.**Dropout**(*p*)

An AlphaDropout Module with alpha set to 0.0.

See AlphaDropout for details.

show()

Displays the Module in a human-friendly format.

class panndas.nn.Identity

A Module that returns its inputs unaltered.

forward(xs)

Applies the Module to its input.

class panndas.nn.LayerMaxNorm

Normalize across the feature dimension with respect to the infinity norm.

forward(xs)

Applies the Module to its input.

class panndas.nn.Linear(weights_df, bias_series=- 1.0)

A Module that multiplies its inputs by the weights_df and adds the bias_series.

Input 'tensors' can be at most 2-D here: feature (rows) and batch/sequence (columns).

Parameters

- **weights_df** – Weights for the affine transform. Column index is the input feature space and row index is the output feature space.
- **bias_series** – Biases for the affine transform. If not a pd.Series, presumed to be a single element that is promoted to a Series.

Examples

```
>>> import pandas as pd
>>> import panndas.nn as nn
>>> w = pd.DataFrame([[0.0, 1.0],[1.0, 0.0]]) # reflection matrix
>>> w.columns = pd.Index(["left", "right"], name="inputs")
>>> w.index = pd.Index(["right", "left"], name="outputs") # reflection mirrors
↪ inputs
>>> l = nn.Linear(weights_df=w, bias_series=0.0)
>>> s = pd.Series([1.0, 2.0], index=w.columns)
>>> s
inputs
left    1.0
right   2.0
dtype: float64
>>> l(s)
outputs
right   2.0
left    1.0
dtype: float64
```

forward(xs)

Applies the Module to its input.

show()

Displays the Module in a human-friendly format.

class panndas.nn.LinearAttention(queries_df, keys_df, values_df)

The most basic version of an attention layer.

Combines queries, keys, and values linearly.

forward(*xs*)

Applies the Module to its input.

class panndas.nn.**Mish**

Applies the Mish function, element-wise.

For details, see [Mish: A Self-Regularized Non-Monotonic Neural Activation Function](#).

forward(*xs*)

Applies the Module to its input.

class panndas.nn.**ReLU**

Old ReLU-iable.

Applies the rectified linear function, elementwise.

forward(*xs*)

Applies the Module to its input.

class panndas.nn.**Sigmoid**

Applies the sigmoid function, element-wise.

forward(*xs*)

Applies the Module to its input.

class panndas.nn.**Softmax**

Applies softmax function, column-wise.

forward(*xs*)

Applies the Module to its input.

class panndas.nn.**SoftmaxAttention**(*queries_df, keys_df, values_df*)

The best-known version of an attention layer.

Uses a softmax over the sequence dim to select which values to attend to.

forward(*xs*)

Applies the Module to its input.

class panndas.nn.**Softplus**

Applies the softplus function, element-wise.

forward(*xs*)

Applies the Module to its input.

INSTALLATION

```
$ pip install pandas
```

CHAPTER
THREE

USAGE

Simply don't.

PYTHON MODULE INDEX

p

`pandas.nn`, [1](#)

A

AdditiveSkip (*class in panndas.nn*), 1
AlphaDropout (*class in panndas.nn*), 1

B

BatchNorm1d (*class in panndas.nn*), 1

D

Dropout (*class in panndas.nn*), 1

F

forward() (*panndas.nn.AdditiveSkip method*), 1
forward() (*panndas.nn.AlphaDropout method*), 1
forward() (*panndas.nn.BatchNorm1d method*), 1
forward() (*panndas.nn.Identity method*), 2
forward() (*panndas.nn.LayerMaxNorm method*), 2
forward() (*panndas.nn.Linear method*), 2
forward() (*panndas.nn.LinearAttention method*), 2
forward() (*panndas.nn.Mish method*), 3
forward() (*panndas.nn.ReLU method*), 3
forward() (*panndas.nn.Sigmoid method*), 3
forward() (*panndas.nn.Softmax method*), 3
forward() (*panndas.nn.SoftmaxAttention method*), 3
forward() (*panndas.nn.Softplus method*), 3

I

Identity (*class in panndas.nn*), 2

L

LayerMaxNorm (*class in panndas.nn*), 2
Linear (*class in panndas.nn*), 2
LinearAttention (*class in panndas.nn*), 2

M

Mish (*class in panndas.nn*), 3
module
 panndas.nn, 1

P

panndas.nn
 module, 1

R

ReLU (*class in panndas.nn*), 3

S

show() (*panndas.nn.AdditiveSkip method*), 1
show() (*panndas.nn.AlphaDropout method*), 1
show() (*panndas.nn.Dropout method*), 1
show() (*panndas.nn.Linear method*), 2
Sigmoid (*class in panndas.nn*), 3
Softmax (*class in panndas.nn*), 3
SoftmaxAttention (*class in panndas.nn*), 3
Softplus (*class in panndas.nn*), 3