Activations

BaseActivation

class `paddle.trainer_config_helpers.activations.`**`BaseActivation`**(*name*, *support_hppl*)

A mark for activation class. Each activation inherit BaseActivation, which has two parameters.

Parameters:
- **name** (*basestring*) – activation name in paddle config.
- **support_hppl** (*bool*) – True if supported by hppl. HPPL is a library used by paddle internally. Currently, lstm layer can only use activations supported by hppl.

AbsActivation

class `paddle.trainer_config_helpers.activations.`**`AbsActivation`**

Abs Activation.

Forward: $f(z) = abs(z)$

Derivative:

$$\begin{array}{rcl} 1 & if & z > 0 \\ -1 & if & z < 0 \\ 0 & if & z = 0 \end{array}$$

ExpActivation

class `paddle.trainer_config_helpers.activations.`**`ExpActivation`**

Exponential Activation.

$$f(z) = e^z.$$

IdentityActivation

class `paddle.trainer_config_helpers.activations.`**`IdentityActivation`**

Identity Activation.

Just do nothing for output both forward/backward.

LinearActivation

`paddle.trainer_config_helpers.activations.`**`LinearActivation`**

alias of **`IdentityActivation`**

LogActivation

class `paddle.trainer_config_helpers.activations.`**`LogActivation`**

Logarithm Activation.

$$f(z) = log(z)$$

SquareActivation

class `paddle.trainer_config_helpers.activations.`**`SquareActivation`**

Square Activation.

$$f(z) = z^2.$$

SigmoidActivation

class `paddle.trainer_config_helpers.activations.`**`SigmoidActivation`**

Sigmoid activation.

$$f(z) = \frac{1}{1 + exp(-z)}$$

SoftmaxActivation

class `paddle.trainer_config_helpers.activations.`**`SoftmaxActivation`**

Softmax activation for simple input

$$P(y = j|x) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_j}}$$

SequenceSoftmaxActivation

class `paddle.trainer_config_helpers.activations.`**`SequenceSoftmaxActivation`**

Softmax activation for one sequence. The dimension of input feature must be 1 and a sequence.

```
result = softmax(for each_feature_vector[0] in input_feature)
for i, each_time_step_output in enumerate(output):
    each_time_step_output = result[i]
```

ReluActivation

class `paddle.trainer_config_helpers.activations.`**`ReluActivation`**

Relu activation.

forward. $y = max(0, z)$

derivative:

$$\begin{array}{ll} 1 & if z > 0 \\ 0 & \text{otherwize} \end{array}$$

BReluActivation

class paddle.trainer_config_helpers.activations.**BReluActivation**

BRelu Activation.

forward. $y = min(24, max(0, z))$

derivative:

$$\begin{array}{ll} 1 & if 0 < z < 24 \\ 0 & \text{otherwise} \end{array}$$

SoftReluActivation

class paddle.trainer_config_helpers.activations.**SoftReluActivation**

SoftRelu Activation.

TanhActivation

class paddle.trainer_config_helpers.activations.**TanhActivation**

Tanh activation.

$$f(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

STanhActivation

class paddle.trainer_config_helpers.activations.**STanhActivation**

Scaled Tanh Activation.

$$f(z) = 1.7159 * tanh(2/3 * z)$$