

Активационные функции

Использование активаций

Активации могут использоваться как через слой `activations`, так и через аргумент `activation`, поддерживаемый всеми предыдущими слоями:

```
from keras.layers import Activation, Dense
model.add(Dense(64))
model.add(Activation('tanh'))
```

Эта запись эквивалентна:

```
model.add(Dense(64, activation='tanh'))
```

В качестве активации можно также передать функцию TensorFlow/Theano/CNTK по элементам:

```
from keras import backend as K
model.add(Dense(64, activation=K.tanh))
```

Доступные активации

elu

```
keras.activations.elu(x, alpha=1.0)
```

Экспоненциальный линейный блок.

Аргументы

- **x**: Входной тензор.
- **alpha**: Скаляр, наклон отрицательного сечения.

Возвращает

Экспоненциальную линейную активацию: x если $x > 0$ и $\alpha * (\exp(x)-1)$ если $x < 0$.

Ссылки

- Быстрое и точное глубокое сетевое обучение с помощью экспоненциальных линейных блоков (ELU)

softmax

```
keras.activations.softmax(x, axis=-1)
```

Функция активации Softmax.

Аргументы

- **x**: Входной тензор.
- **axis**: Integer, ось, по которой применяется нормализация софтмакса.

Возвращает

Тензор, выход софтмакс-трансформации.

Исключение

- **ValueError**: если $\dim(x) == 1$.

selu

```
keras.activations.selu(x)
```

Масштабируемая экспоненциальная линейная единица (SELU).

SELU равна: $\text{scale} * \text{elu}(x, \alpha)$, где α и scale являются предопределенными константами. Значения α и scale выбираются таким образом, чтобы среднее и дисперсия входов сохранялись между двумя последовательными слоями до тех пор, пока веса правильно инициализированы (см. `lecun_normal initialization`), а количество входов «large enough» (см. Ссылки для дополнительной информации).

Аргументы

- x : Тензор или переменная для расчета функции активации.

Возвращает

Масштабированную экспоненциальную активацию: $\text{scale} * \text{elu}(x, \alpha)$.

Примечание

- Используется вместе с инициализацией «`lecun_normal`».
- Используется вместе с вариантом исключения «`AlphaDropout`».

Ссылки

- Самономализованные нейронные сети <https://arxiv.org/abs/1706.02515>

softplus

```
keras.activations.softplus(x)
```

Функция активации Softplus.

Аргументы

- x : Входной тензор.

Возвращает

активация Softplus: $\log(\exp(x) + 1)$.

softsign

```
keras.activations.softsign(x)
```

Функция активации softsign.

Аргументы

- **x**: Входной тензор.

Возвращает

Активация softsign: $x / (|x| + 1)$.

relu

```
keras.activations.relu(x, alpha=0.0, max_value=None,  
threshold=0.0)keras.activations.softsign(x)
```

Спряmlенный линейный блок.

При значениях по умолчанию он возвращает по элементам. $\max(x, 0)$.

В противном случае: $f(x) = \text{max_value}$ для $x \geq \text{max_value}$, $f(x) = x$ для $\text{threshold} \leq x < \text{max_value}$, $f(x) = \alpha * (x - \text{threshold})$.

Аргументы

- **x**: Входной тензор.
- **alpha**: float. Наклон отрицательной части. По умолчанию ноль.
- **max_value**: float. Порог насыщения.
- **threshold**: float. Пороговое значение для пороговой активации.

Возвращает

Тензор.

tanh

```
keras.activations.tanh(x)
```

Функция активации гиперболического тангенса.

Аргументы

- `x`: тензор ввода.

Возвращает

Гиперболическая активация: $\tanh(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x))$

sigmoid

```
keras.activations.sigmoid(x)
```

Функция активации сигмоида.

Аргументы

- `x`: тензор ввода.

Возвращает

Сигмоидальная активация: $1 / (1 + \exp(-x))$.

hard_sigmoid

```
keras.activations.hard_sigmoid(x)
```

Функция активации жесткая сигмоида.

Быстрее вычисляет, чем активацию сигмоида.

Аргументы

- `x`: тензор ввода.

Возвращает

Активация жесткая сигмоида:

- 0 если $x < -2.5$
- 1 если $x > 2.5$
- $0.2 * x + 0.5$ если $-2.5 \leq x \leq 2.5$.

exponential

```
keras.activations.exponential(x)
```

Экспоненциальная (базовая) функция активации.

Аргументы

- x : тензор ввода.

Возвращает

Экспоненциальная активация: $\exp(x)$.

linear

```
keras.activations.linear(x)
```

Функция линейной (т.е. идентификационная) активации.

Аргументы

- x : тензор ввода.

Возвращает

Входной тензор, без изменений.

По «Расширенным активациям» (Advanced activations)

Активации, более сложные, чем простая функция TensorFlow/Theano/CNTK (например, Обучаемые активации, поддерживающие состояние) доступны в виде уровней расширенной активации, и их можно найти в модуле `keras.layers.advanced_activations`. Они включают PReLU и LeakyReLU.