**Keras kernel\_initializer 權重初始化的方法**

 2019-01-27  254

**初始化方法**

初始化方法定義了對Keras層設定初始化權重的方法

不同的層可能使用不同的關鍵字來傳遞初始化方法，一般來說指定初始化方法的關鍵字是kernel\_initializer 和 bias\_initializer，例如：

model.add(Dense(64,

kernel\_initializer='random\_uniform',

bias\_initializer='zeros'))

一個初始化器可以由字串指定（必須是下面的預定義初始化器之一），或一個callable的函式，例如

from keras import initializers

model.add(Dense(64, kernel\_initializer=initializers.random\_normal(stddev=0.01)))

# also works; will use the default parameters.

model.add(Dense(64, kernel\_initializer='random\_normal'))

**Initializer**

Initializer是所有初始化方法的父類，不能直接使用，如果想要定義自己的初始化方法，請繼承此類。

**預定義初始化方法**

**Zeros**

keras.initializers.Zeros()

全零初始化

**Ones**

keras.initializers.Ones()

全1初始化

**Constant**

keras.initializers.Constant(value=0)

初始化為固定值value

**RandomNormal**

keras.initializers.RandomNormal(mean=0.0, stddev=0.05, seed=None))

正態分佈初始化

* mean：均值
* stddev：標準差
* seed：隨機數種子

**RandomUniform**

keras.initializers.RandomUniform(minval=-0.05, maxval=0.05, seed=None)

均勻分佈初始化 *minval：均勻分佈下邊界*maxval：均勻分佈上邊界 \* seed：隨機數種子

**TruncatedNormal**

keras.initializers.TruncatedNormal(mean=0.0, stddev=0.05, seed=None)

截尾高斯分佈初始化，該初始化方法與RandomNormal類似，但位於均值兩個標準差以外的資料將會被丟棄並重新生成，形成截尾分佈。該分佈是神經網路權重和濾波器的推薦初始化方法。

* mean：均值
* stddev：標準差
* seed：隨機數種子

**VarianceScaling**

keras.initializers.VarianceScaling(scale=1.0, mode='fan\_in', distribution='normal', seed=None)

該初始化方法能夠自適應目標張量的shape。

當distribution="normal"時，樣本從0均值，標準差為sqrt(scale / n)的截尾正態分佈中產生。其中：

\* 當```mode = "fan\_in"```時，權重張量的輸入單元數。

\* 當```mode = "fan\_out"```時，權重張量的輸出單元數

\* 當```mode = "fan\_avg"```時，權重張量的輸入輸出單元數的均值

當distribution="uniform"時，權重從[-limit, limit]範圍內均勻取樣，其中limit = limit = sqrt(3 \* scale / n)

* scale: 放縮因子，正浮點數
* mode: 字串，“fan\_in”，“fan\_out”或“fan\_avg”fan\_in", "fan\_out", "fan\_avg".
* distribution: 字串，“normal”或“uniform”.
* seed: 隨機數種子

**Orthogonal**

keras.initializers.Orthogonal(gain=1.0, seed=None)

用隨機正交矩陣初始化

* gain: 正交矩陣的乘性係數
* seed：隨機數種子

**Identiy**

keras.initializers.Identity(gain=1.0)

使用單位矩陣初始化，僅適用於2D方陣

* gain：單位矩陣的乘性係數

**lecun\_uniform**

lecun\_uniform(seed=None)

LeCun均勻分佈初始化方法，引數由[-limit, limit]的區間中均勻取樣獲得，其中limit=sqrt(3 / fan\_in), fin\_in是權重向量的輸入單元數（扇入）

* seed：隨機數種子

**lecun\_normal**

lecun\_normal(seed=None)

LeCun正態分佈初始化方法，引數由0均值，標準差為stddev = sqrt(1 / fan\_in)的正態分佈產生，其中fan\_in和fan\_out是權重張量的扇入扇出（即輸入和輸出單元數目）

* seed：隨機數種子

參考文獻：

**glorot\_normal**

glorot\_normal(seed=None)

Glorot正態分佈初始化方法，也稱作Xavier正態分佈初始化，引數由0均值，標準差為sqrt(2 / (fan\_in + fan\_out))的正態分佈產生，其中fan\_in和fan\_out是權重張量的扇入扇出（即輸入和輸出單元數目）

* seed：隨機數種子

**glorot\_uniform**

glorot\_uniform(seed=None)

Glorot均勻分佈初始化方法，又成Xavier均勻初始化，引數從[-limit, limit]的均勻分佈產生，其中limit為sqrt(6 / (fan\_in + fan\_out))。fan\_in為權值張量的輸入單元數，fan\_out是權重張量的輸出單元數。

* seed：隨機數種子

**he\_normal**

he\_normal(seed=None)

He正態分佈初始化方法，引數由0均值，標準差為sqrt(2 / fan\_in) 的正態分佈產生，其中fan\_in權重張量的扇入

* seed：隨機數種子

**he\_uniform**

he\_normal(seed=None)

LeCun均勻分佈初始化方法，引數由[-limit, limit]的區間中均勻取樣獲得，其中limit=sqrt(6 / fan\_in), fin\_in是權重向量的輸入單元數（扇入）

* seed：隨機數種子

**自定義初始化器**

如果需要傳遞自定義的初始化器，則該初始化器必須是callable的，並且接收shape（將被初始化的張量shape）和dtype（資料型別）兩個引數，並返回符合shape和dtype的張量。

from keras import backend as K

def my\_init(shape, dtype=None):

return K.random\_normal(shape, dtype=dtype)

model.add(Dense(64, init=my\_init))