

02LongShortTermMemory

Scanned Documents

LSTM

Long Short Term Memory

import torch, import torch.nn as nn, import torch.nn.functional as F, ... from torch.nn import

import typing as L, ... from torch.nn.modules import LSTMCell, LSTMCellList

class LSTMCellList(LSTMCellList):

def __init__(self, *args):

super(LSTMCellList, self).__init__()

self.cell_list = LSTMCellList(*args)

self.cell_list = nn.LSTMCellList(*args)

self.cell_list = nn.LSTMCellList(*args)

self.cell_list = nn.LSTMCellList(*args)

self.cell_list = nn.LSTMCellList(*args)

def __init__(self, input_size, hidden_size, num_layers, batch_first):

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

def __init__(self, input_size):

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

def __init__(self, input_size, hidden_size):

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

def __init__(self, input_size, hidden_size):

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

self.input_size = input_size, self.hidden_size = hidden_size, self.num_layers = num_layers, self.batch_first = batch_first

[illegible]