

Tutorial Example

Programming Tutorials and Examples for Beginners

Home

Understand TensorDataset with Examples in PyTorch – PyTorch Tutorial

```
torch.utils.data.TensorDataset(torch.utils.data.TensorDataset, ...):  
    Dataset wrapping Tensors.  
  
    Samples will be retrieved by indexing tensors along the first dimension.  
  
    Tensors (Tensor): Tensors that have the same size of the first dimension.  
    repr: Tuple(Tensor, ...)  
  
    __getitem__(self, indexes: Tensor) -> None  
    insert_all(tensors[0].size(0) + tensor.size(0) for tensor in tensors), "  
    self.tensors = tensors  
  
    __getitem__(self, indexes):  
        return tuple(tensors[index] for tensor in self.tensors)  
  
    __len__(self):  
        return self.tensors[0].size(0)
```

In pytorch, TensorDataset allows us to zip serial of pytorch tensors as python zip() function. In this tutorial, we will use some examples to show you how to use.

Category: PyTorch

Understand itertools.chain() with Examples – Python Tutorial



itertools.chain() can take a series of iterables and returns one iterable. In this tutorial, we will use some examples to show you how to use it.

Category: Python

Mixed Precision Training for Beginners – Deep Learning Tutorial

1: ILSVRC12 classification top-1 accuracy

Baseline	Mixed Precision	
56.77%	56.93%	(Kr
65.40%	65.43%	(Simon
68.33%	68.43%	(S
70.03%	70.02%	(Iofl
73.85%	74.13%	(S
75.92%	76.04%	

In this tutorial, we will introduce what is mixed precision training, how about the effect of it and how to use it.

Category: Machine Learning

Implement Mixed Precision Training with GradScaler in PyTorch – PyTorch Tutorial

```

# torch.cuda.amp.GradScaler()

def train_step(): # 8 weeks, this section is for illustration only
    input, target = zip(data_loader, target_loader)
    with torch.cuda.device(device_type):
        output = net(input)
        loss = loss_fn(output, target)

    # Backward pass: calls backward() on scalar loss to create scalar gradient
    scaler.scale(loss).backward()

    # scaler.step() first unscales the gradients of the optimizer's assigned
    # If these gradients do not contain data on both, optimizer.step() is 0
    # otherwise, optimizer.step() is skipped.
    scaler.step(optimizer)

    # Updates the scale for next iteration.
    scaler.update()

optimizer.zero_grad() # set to zero: zeroing out can moderately improve performance

```

In this tutorial, we will introduce how to implement mixed precision training with torch.cuda.amp.GradScaler in PyTorch, which can speed up our training.

Category: PyTorch

Understand PyTorch Tensor.repeat() with Examples – PyTorch Tutorial

H.TENSOR.REPEAT

at("sizes") → Tensor

tensor along the specified dimensions.

id(), this function copies the tensor's data

PyTorch tensor.repeat() function can repeat a tensor along the specified dimensions. In this tutorial, we will use some examples to show you how to use it.

Category: PyTorch

Understand torch.split(): Split Tensor into Chunks – PyTorch Tutorial

PLIT

tensor.split(split_size_or_sections, dim=0) [SOURCE]

tensor into chunks. Each chunk is a view of the original tensor.

split_size_or_sections is an integer type, then tensor will be split into len(split_size_or_sections) equal parts. If the tensor size along the given dimension dim is not divisible by split_size_or_sections, an error is raised.

split_size_or_sections is a list, then tensor will be split into len(split_size_or_sections) chunks of the specified sizes.

PyTorch torch.split() function can split a tensor into some chunks. In this tutorial, we will create some examples to show how to use it.

Category: PyTorch

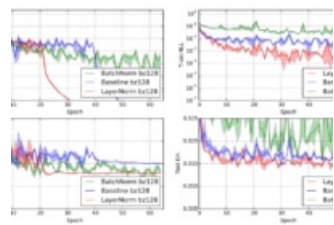
Understand model.zero_grad() and optimizer.zero_grad() – PyTorch Tutorial



In this tutorial, we will discuss the difference between model.zero_grad() and optimizer.zero_grad() when we are training an model.

Category: PyTorch

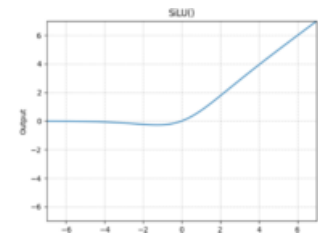
Layer Normalization Effect in RNN, CNN and Feed-Forward Networks – Deep Learning Tutorial



In this tutorial, we will discuss the different effect of layer normalization in RNN, CNN and Feed-Forward Networks.

Category: Machine Learning

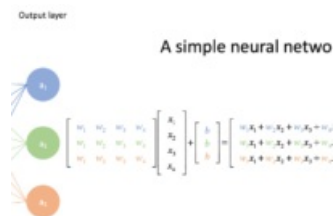
Implement Swish (SiLU) Activation in PyTorch – PyTorch Example



Swish activation function is proposed in comformer model. In this tutorial, we will introduce how to implement in pytorch.

Category: PyTorch

Implement Wrapper Class of torch.nn.Linear() – PyTorch Tutorial



In this tutorial, we will create an example to show you how to create a wrapper class of torch.nn.Linear().

Category: PyTorch



Fix Use `tf.global_variables_initializer` instead in TensorFlow – TensorFlow Tutorial



Fix Command `python setup.py egg_info` failed with error code 1 – Python Tutorial



Best Write, Run and Debug Python Script Online Websites: A Beginner's Guide – Python Tutorial



Matplotlib Plot a Bar with a Dictionary `{x:y}`: A Step Guide – Matplotlib Tutorial



Python Create Float Range for Beginners – Python Tutorial



Bundle a Python Application to an EXE with PyInstaller: A Beginner Guide – Python Tutorial



Why Your Custom LSTM or BiLSTM is Worse than `tf.nn.dynamic_rnn()` and `tf.nn.bidirectional_dynamic_rnn()`



WordPress Display Popular Posts by Category ID Dynamically: A Step Guide – WordPress Tutorial



Understand the Shape or Dimension of a Scalar in TensorFlow – TensorFlow Tutorial



Fix PyInstaller Missing `api-ms-win-crt-process-l1-1-0.dll` Error – Python Tutorial

Meta

Log in

Entries RSS

Comments RSS