PyTorch

September 1, 2024

Ramasamy Kandasamy

1. Preliminaries

1.1. Torch Tensor

Creating a tensor.

`torch.<method>(<options>)`

`arange(l, r, s)`	$(l, l+s, \cdots, l+ks) \ni l+ks < r$ and k is the largest such integer. Default: $l=0$, $s=1$.
`linspace(l, r, n)`	$(l, l+s, \cdots, r(=l+n\times s))\&s = (r-l)/n$.
`ones(<shape>)`	
`zeros(<shape>)`	
`full(<shape>, k)`	Constant tensor filled with each element $= k$.
`randn`	
`tensor`	From NDArray. Eg: `torch.tensor([1, 2, 3])`.
`from_numpy`	

Other related methods: `zeros_like`, `ones_like`, `empty_like`.
Eg: `y = ones_like(x)`.

Tensor Properties & Operation

`numel()`	Number of elements. Eg, a 2×2 tensor has 4 elements.
`dtype`	
`shape`	
`reshape()`	Eg: `torch.arange(8).reshape(4, 4)` Eg: `torch.arange(8).reshape(4, -1)` Use -1 to automatically infer one of the dimensions.
`numpy()`	
`item()`	Can be applied only to a tensor with single element. Returns the element. Single element tensors can also be coverted as follows: `int(x)`, `float(x)`, etc.
`cat`	Concatenate along a dimension. Specify `dim`.
`sum`	

Operations

Uncategorized