ModuleDict

class torch.nn.ModuleDict

Holds submodules in a dictionary.

ModuleDict can be indexed like a regular Python dictionary, but modules it contains are properly registered, and will be visible by all Module methods.

ModuleDict is an **ordered** dictionary that respects

- the order of insertion, and
- in update(), the order of the merged OrderedDict, dict (started from Python 3.6) or another ModuleDict (the argument to update()).

Note that update() with other unordered mapping types (e.g., Python's plain dict before Python version 3.6) does not preserve the order of the merged mapping.

Parameters

modules (*iterable, optional*) – a mapping (dictionary) of (string: module) or an iterable of key-value pairs of type (string, module)

Example:

```
class MyModule(nn.Module):
    def __init__(self) -> None:
        super().__init__()
        self.choices = nn.ModuleDict({
                'conv': nn.Conv2d(10, 10, 3),
                'pool': nn.MaxPool2d(3)
        })
        self.activations = nn.ModuleDict([
                ['lrelu', nn.LeakyReLU()],
                ['prelu', nn.PReLU()]
        ])

    def forward(self, x, choice, act):
        x = self.choices[choice](x)
        x = self.activations[act](x)
        return x
```