

- ▶ Creare uno script in python, un file *.py* chiamato «*custom\_ops.py*».

# Esercitazione

- Il file avrà il seguente contenuto.

```
custom_ops.py > Offset > forward
1  import torch
2
3  # Mini-rete che esegue la potenza.
4  class Power(torch.nn.Module):
5
6      def __init__(self):
7          super().__init__()
8
9  > def forward(self, x : torch.Tensor) -> torch.Tensor: ...
11
12 # Mini-rete che esegue la radice.
13 class Sqrt(torch.nn.Module):
14
15     def __init__(self):
16         super().__init__()
17
18 > def forward(self, x : torch.Tensor) -> torch.Tensor: ...
20
21 # Mini-rete che esegue un offset di 100.
22 class Offset(torch.nn.Module):
23
24     def __init__(self, value):
25         super().__init__()
26         self.value = value
27
28 > def forward(self, x : torch.Tensor) -> torch.Tensor: ...
```

# Esercitazione

- ▶ Completare i tre metodi *forward*:

- ▶ *Power*:

- ▶ Restituisce il quadrato del tensore di input.

- ▶ *Sqrt*:

- ▶ Restituisce la radice quadrata del tensore di input.

- ▶ *Offset*:

- ▶ Restituisce il tensore di input più una costante (100)

# Esercitazione

- ▶ Creare uno script in python, un file *.py* chiamato «*net.py*».

# Esercitazione

- Il file avrà il seguente contenuto.

```
net.py > ...
1  import torch
2  from custom_ops import Power, Sqrt, Offset
3
4  class Model(torch.nn.Module):
5
6  >   def __init__(self) -> None: ...
11
12 >   def forward(self, x : torch.Tensor) -> torch.Tensor: ...
17
18  if __name__ == '__main__':
19
20     min_value, max_value = 0, 10
21     shape = (5, 3, 256, 256)
22
23     random_tensor = torch.randint(min_value, max_value + 1, shape)
24
25     net = Model()
26     output = net(random_tensor)
27
28     print(output)
29
```

# Esercitazione

- ▶ Completare i due metodi:

- ▶ `__init__`:

- ▶ Esegue il metodo `__init__` della classe padre.
    - ▶ Crea 3 variabili di classe: `layer_1`, `layer_2`, `layer_3`.
    - ▶ Assegna alle variabili, rispettivamente, l'operazione di potenza, radice ed infine offset.

- ▶ `forward`:

- ▶ Esegue in sequenza le operazioni del `layer_1`, `layer_2`, `layer_3` sul tensore `x` in ingresso.

# Esercitazione

- ▶ Eseguire lo script «*net.py*»:
  - ▶ Utilizzare il debugger di visual studio per seguire il flusso delle operazioni, riga per riga.
  - ▶ Utilizzare *F10* ed *F11*.

# Esercitazione