**hw\_04**

**1. When run this following code, what is the output of print(b.requires\_grad)? (c)**

a = torch.tensor([1, 6, 1, 1, 1], dtype=torch.float)

1. None
2. True
3. False
4. Error

**2. When run this following code, what is the output of print(a.grad)? (d)**

a = torch.tensor([1, 6, 1, 1, 1], dtype=torch.float)

a.requires\_grad\_(**True**)

1. -1
2. 0
3. 1
4. None

**3. When run this following code, what is the output of print(s)? (a)**

a = torch.tensor([1, 6, 1, 1, 1], dtype=torch.float)

a.requires\_grad\_(**True**)

s = sum((b-1)\*\*2)

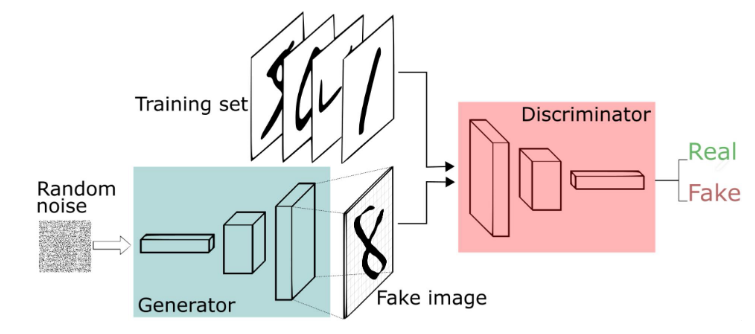
s.backward()

1. tensor(25., grad\_fn=<AddBackward0>)
2. tensor(25., grad\_fn=<MulBackward0>)
3. tensor(10., grad\_fn=<AddBackward0>)
4. tensor(10., grad\_fn=<MulBackward0>)

**4. When do we use retain\_graph function? (b)(e)**

1. When we want to reduce memory usage and speed up computations
2. When we use multi head neural networks
3. When we don’t want the loss backpropagate beyond the retain\_graph point
4. When we are not using the computational tree again
5. When we need to back propagate more than once

**5. When we don’t want the loss backpropagate in Generator, where should we use** detact**? (c)**



1. In the middle of generator
2. In the middle of discriminator
3. Between generator and discriminator
4. Node of sigmoid output (Real, Fake)
5. When entering random noise