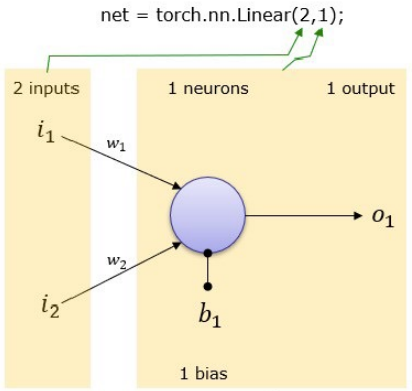
โครงการพัฒนาทักษะการเรียนรู้ของเครื่อง (Machine Learning) ของบัณฑิตเพื่อตอบสนองการพัฒนาประเทศไทย 4.0

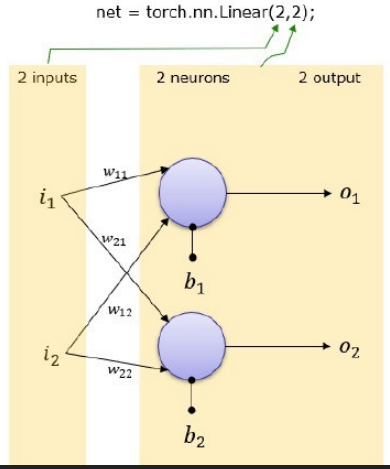
16 และ 23 กุมภาพันธ์ 2566

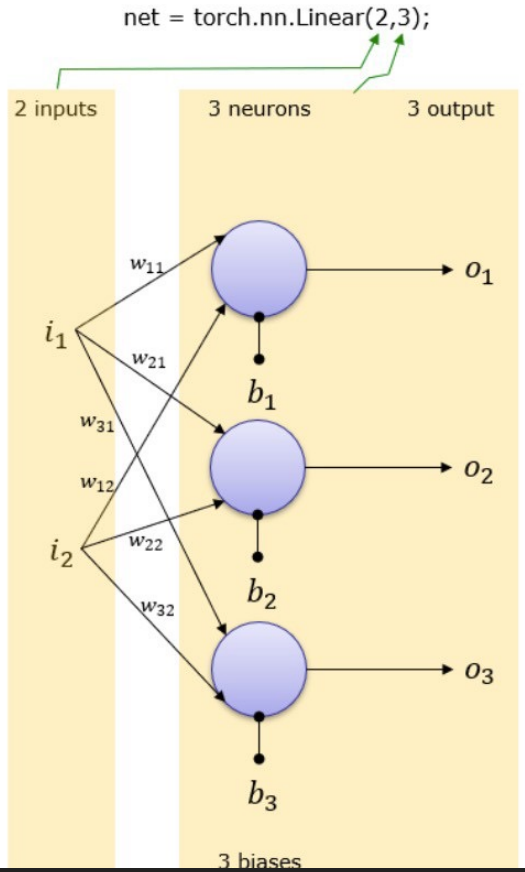
Code

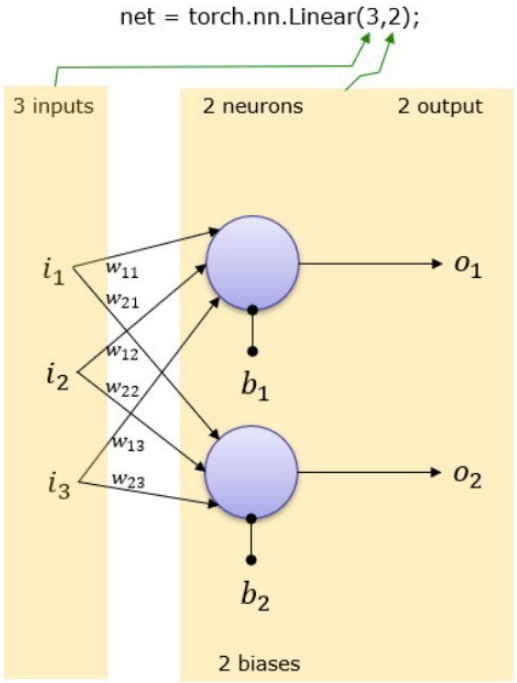
แบบสอบถาม

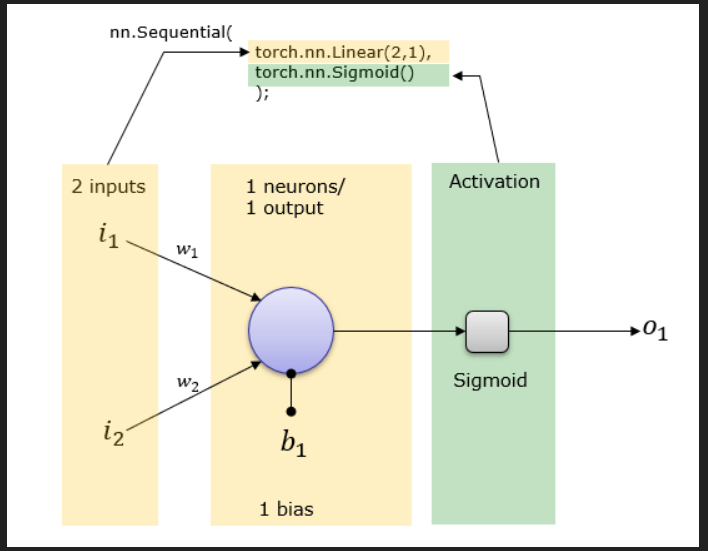
“What are Neural Networks?”

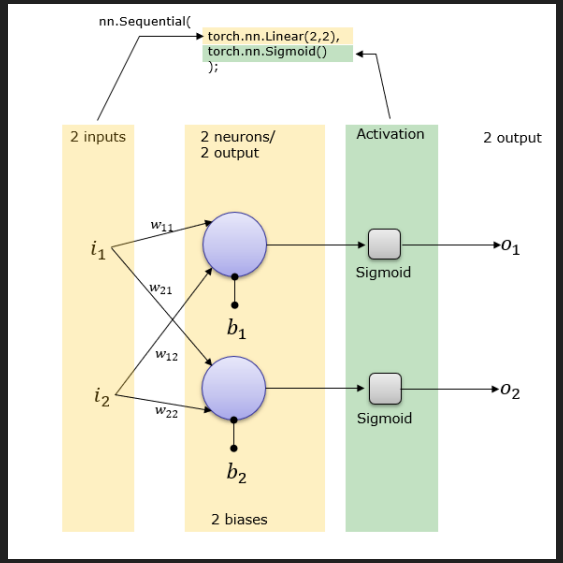

net = torch.nn.Linear(2,1);
2 inputs
1 neurons
1 output
i_1
w_1
o_1
w_2
i_2
b_1
1 bias

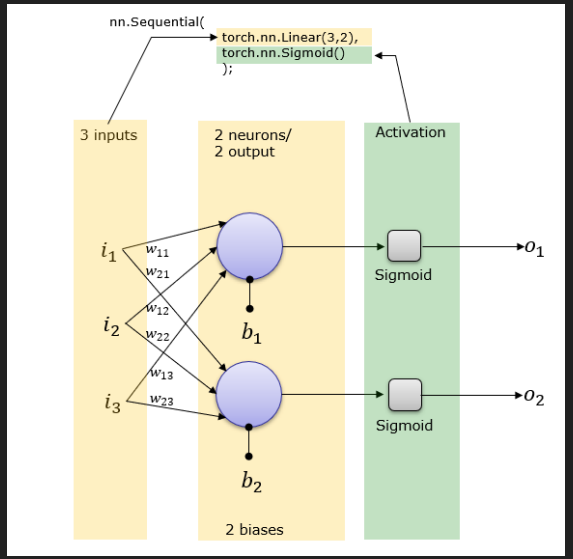

net = torch.nn.Linear(2,2);
2 inputs
2 neurons
2 output
w_{11}
i_1
o_1
w_{21}
b_1
w_{12}
i_2
o_2
w_{22}
b_2

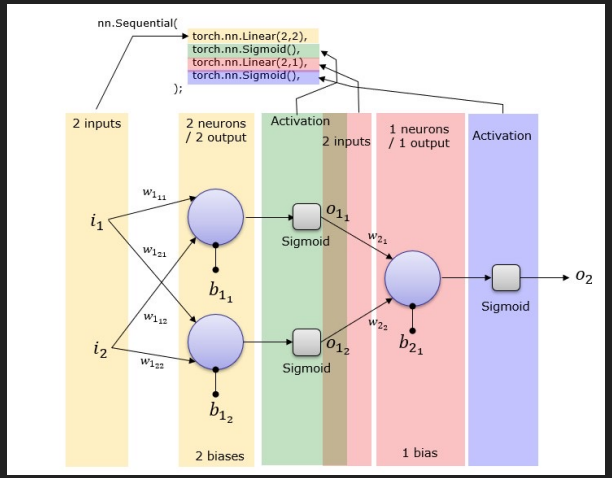

net = torch.nn.Linear(2,3);
2 inputs
3 neurons
3 output
w_{11}
o_1
i_1
w_{21}
b_1
w_{31}
o_2
w_{12}
i_2
w_{22}
b_2
w_{32}
o_3
b_3
3 biases

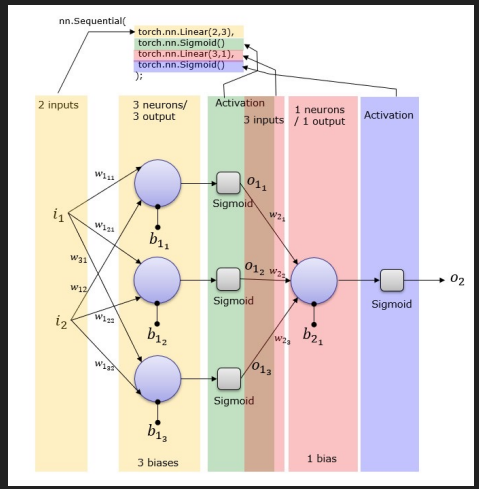

net = torch.nn.Linear(3,2);
3 inputs
2 neurons
2 output
i_1
w_{11}
w_{21}
o_1
i_2
w_{12}
w_{22}
b_1
w_{13}
i_3
w_{23}
o_2
b_2
2 biases


nn.Sequential(
torch.nn.Linear(2,1),
torch.nn.Sigmoid()
);
2 inputs
1 neurons/
1 output
Activation
i_1
i_2
w_1
w_2
b_1
Sigmoid
o_1
1 bias


nn.Sequential(
torch.nn.Linear(2,2),
torch.nn.Sigmoid()
);
2 inputs
2 neurons/
2 output
Activation
2 output
i_1
i_2
w_{11}
w_{21}
w_{12}
w_{22}
b_1
b_2
Sigmoid
Sigmoid
o_1
o_2
2 biases


nn.Sequential(
torch.nn.Linear(3,2),
torch.nn.Sigmoid()
);
3 inputs
2 neurons/
2 output
Activation
i_1
i_2
i_3
w_{11}
w_{21}
w_{12}
w_{22}
w_{13}
w_{23}
b_1
b_2
Sigmoid
Sigmoid
o_1
o_2
2 biases


nn.Sequential(
torch.nn.Linear(2,2),
torch.nn.Sigmoid(),
torch.nn.Linear(2,1),
torch.nn.Sigmoid(),
);
2 inputs
2 neurons
/ 2 output
Activation
2 inputs
1 neurons
/ 1 output
Activation
i_1
i_2
$w_{1_{11}}$
$w_{1_{21}}$
$w_{1_{12}}$
$w_{1_{22}}$
w_{2_1}
w_{2_2}
b_{1_1}
b_{1_2}
b_{2_1}
o_{1_1}
o_{1_2}
o_2
Sigmoid
Sigmoid
Sigmoid
2 biases
1 bias


nn.Sequential(
torch.nn.Linear(2,3),
torch.nn.Sigmoid()
torch.nn.Linear(3,1),
torch.nn.Sigmoid()
);
2 inputs
3 neurons/
3 output
Activation
3 inputs
1 neurons
/ 1 output
Activation
i_1
i_2
$w_{1_{11}}$
$w_{1_{21}}$
$w_{1_{31}}$
$w_{1_{12}}$
$w_{1_{22}}$
$w_{1_{32}}$
w_{2_1}
w_{2_2}
w_{2_3}
b_{1_1}
b_{1_2}
b_{1_3}
b_{2_1}
o_{1_1}
o_{1_2}
o_{1_3}
o_2
Sigmoid
Sigmoid
Sigmoid
Sigmoid
3 biases
1 bias

Three Ways to Build a Neural Network in PyTorch

```python
from torch import nn
from collections import OrderedDict

# define model architecture
model3 = nn.Sequential(OrderedDict([
    ('fc1', nn.Linear(16, 12)),
    ('relu1', nn.ReLU()),
    ('fc2', nn.Linear(12, 10)),
    ('relu2', nn.ReLU()),
    ('fc3', nn.Linear(10, 1)),
    ('sigmoid', nn.Sigmoid())
]))

# print model architecture
print(model3)
```

```
Sequential(
  (fc1): Linear(in_features=16, out_features=12, bias=True)
  (relu1): ReLU()
  (fc2): Linear(in_features=12, out_features=10, bias=True)
  (relu2): ReLU()
  (fc3): Linear(in_features=10, out_features=1, bias=True)
  (sigmoid): Sigmoid()
)
```

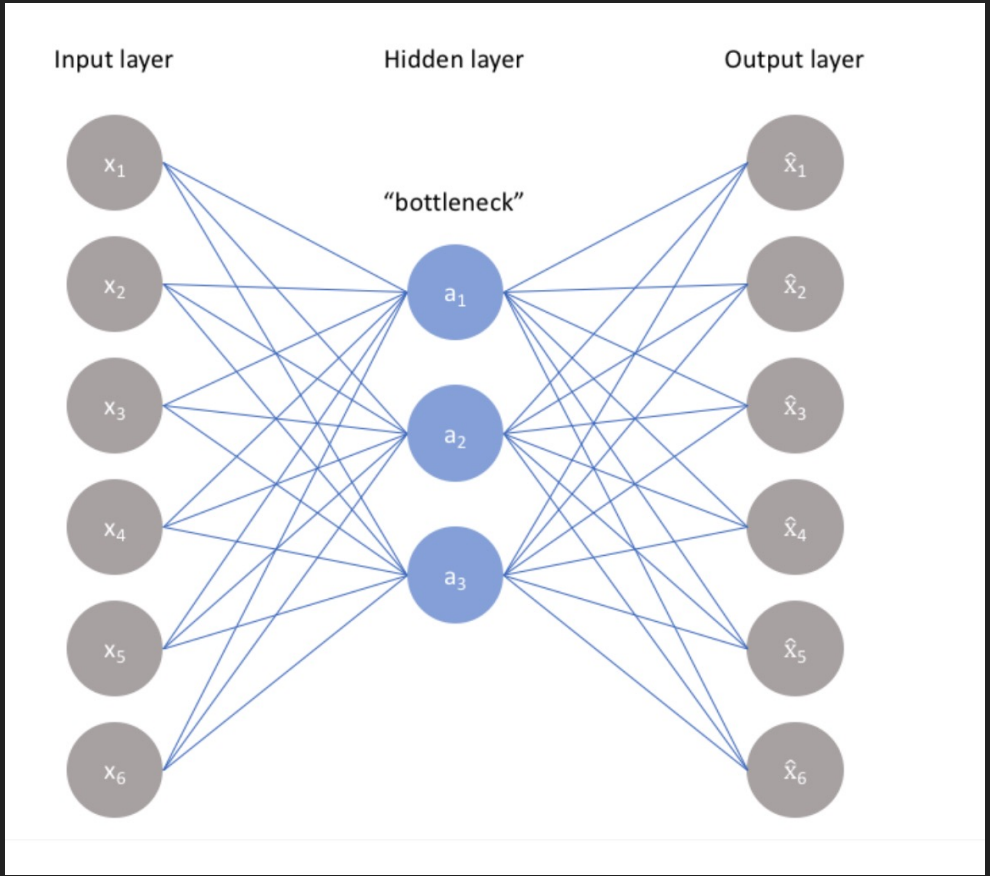

```python
from torch import nn
from collections import OrderedDict

# define model architecture
model3 = nn.Sequential(OrderedDict([
    ('fc1', nn.Linear(16, 12)),
    ('relu1', nn.ReLU()),
    ('fc2', nn.Linear(12, 10)),
    ('relu2', nn.ReLU()),
    ('fc3', nn.Linear(10, 1)),
    ('sigmoid', nn.Sigmoid())
]))

# print model architecture
print(model3)
```

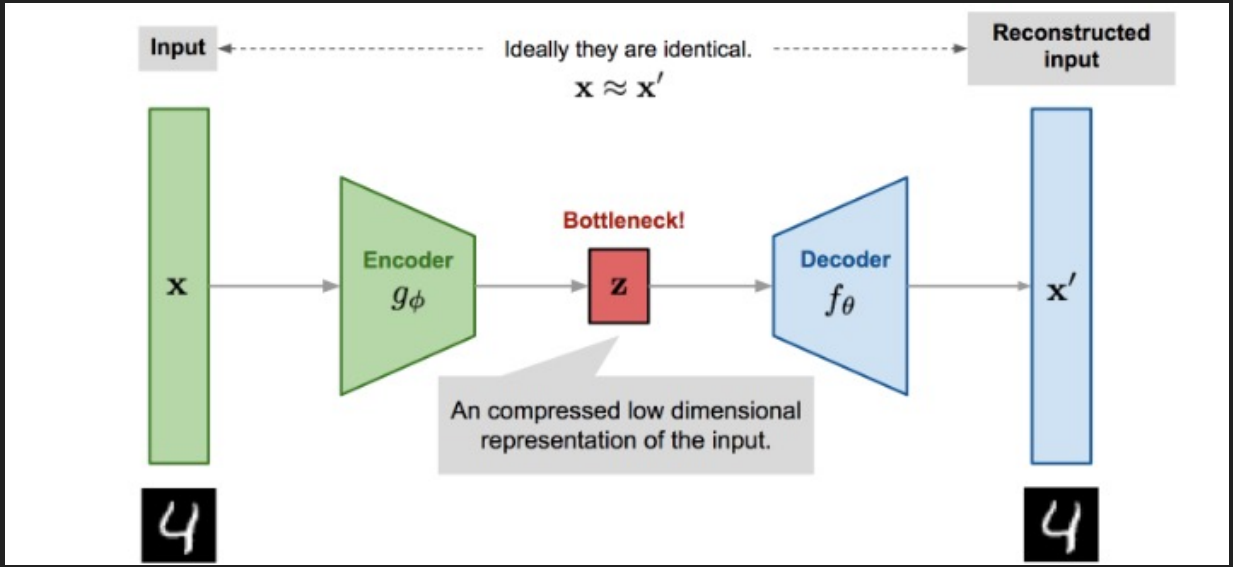
```
Sequential(
  (fc1): Linear(in_features=16, out_features=12, bias=True)
  (relu1): ReLU()
  (fc2): Linear(in_features=12, out_features=10, bias=True)
  (relu2): ReLU()
  (fc3): Linear(in_features=10, out_features=1, bias=True)
  (sigmoid): Sigmoid()
)
```


```python
import torch
import torch.nn.functional as F
from torch import nn

# define the network class
class MyNetwork(nn.Module):
    def __init__(self):
        # call constructor from superclass
        super().__init__()

        # define network layers
        self.fc1 = nn.Linear(16, 12)
        self.fc2 = nn.Linear(12, 10)
        self.fc3 = nn.Linear(10, 1)

    def forward(self, x):
        # define forward pass
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = torch.sigmoid(self.fc3(x))
        return x

# instantiate the model
model1 = MyNetwork()

#

print(model1)
```

```
MyNetwork(
  (fc1): Linear(in_features=16, out_features=12, bias=True)
  (fc2): Linear(in_features=12, out_features=10, bias=True)
  (fc3): Linear(in_features=10, out_features=1, bias=True)
)
```

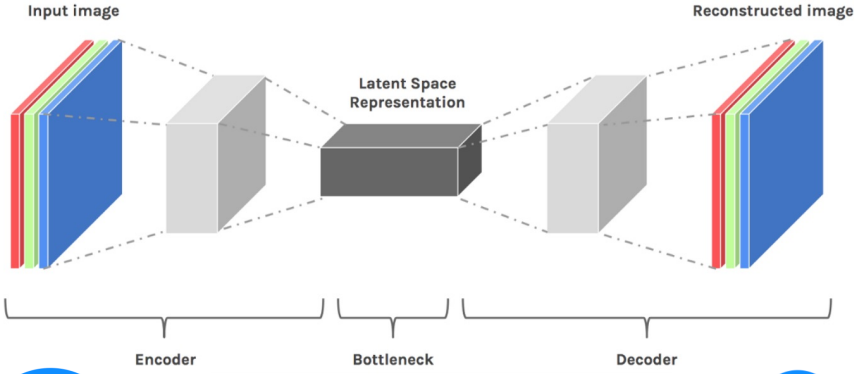
Autoencoders

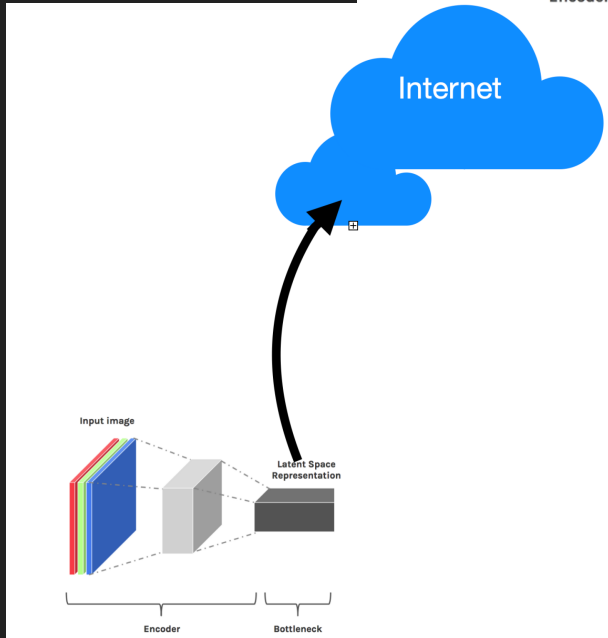

Input layer
Hidden layer
Output layer
"bottleneck"
x_1
x_2
x_3
x_4
x_5
x_6
a_1
a_2
a_3
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$

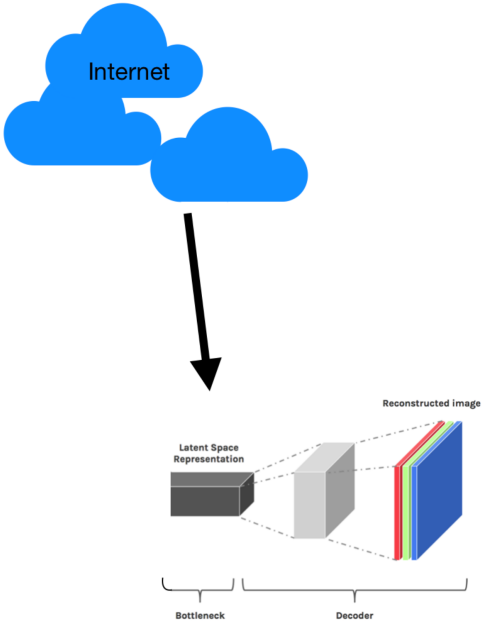


$$L_{\text{AE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$

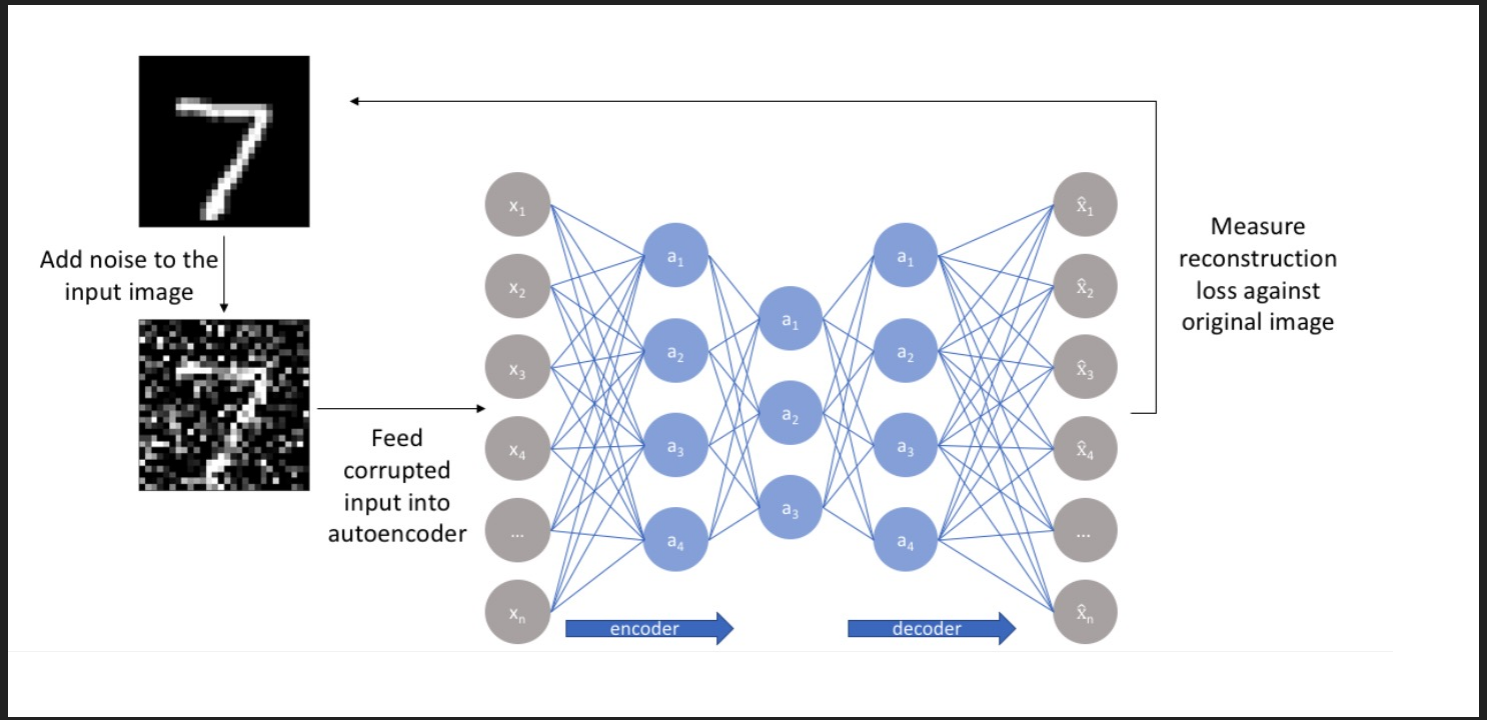
Data compression

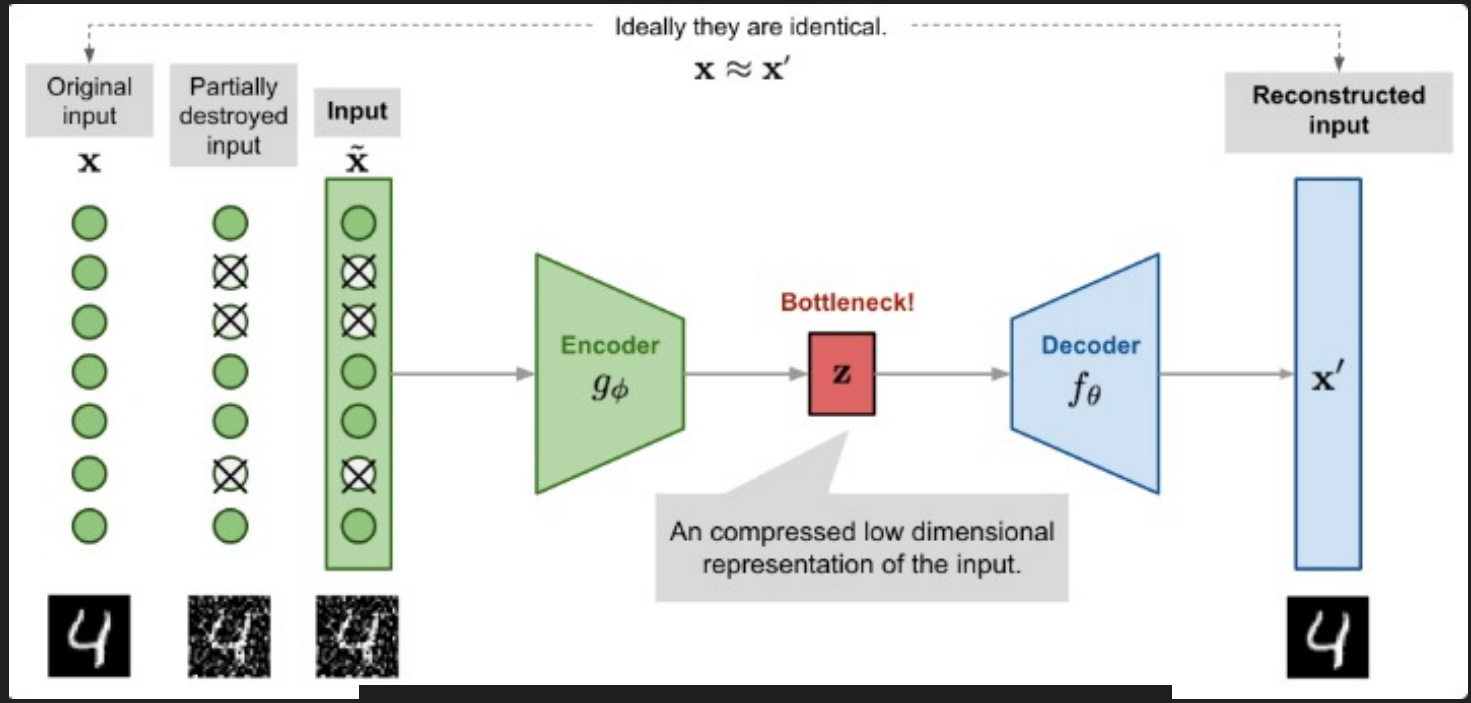






Denoising data





$$\tilde{\mathbf{x}}^{(i)} \sim \mathcal{M}_{\mathcal{D}}(\tilde{\mathbf{x}}^{(i)}|\mathbf{x}^{(i)})$$

$$L_{\text{DAE}}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - f_\theta(g_\phi(\tilde{\mathbf{x}}^{(i)})))^2$$

ดร.ทัศนัย พลอยสุวรรณ

Office: 510-4

tuchsanai@it.kmitl.ac.th

ปริญญาตรี

วศ.บ. (วิศวกรรมไฟฟ้า) มหาวิทยาลัยเกษตรศาสตร์, 2545

ปริญญาโท

วศ.ม. (วิศวกรรมไฟฟ้า) จุฬาลงกรณ์มหาวิทยาลัย, 2547

ปริญญาเอก

วศ.ด. (วิศวกรรมไฟฟ้า) จุฬาลงกรณ์มหาวิทยาลัย, 2552