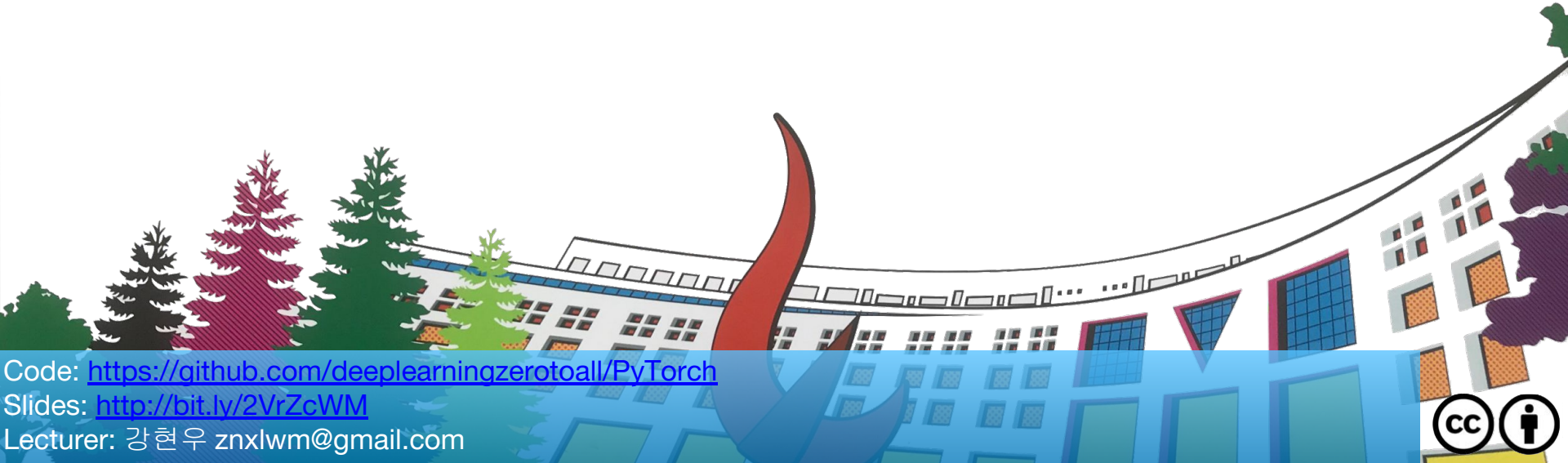
ML/DL for Everyone Season2

Perceptron

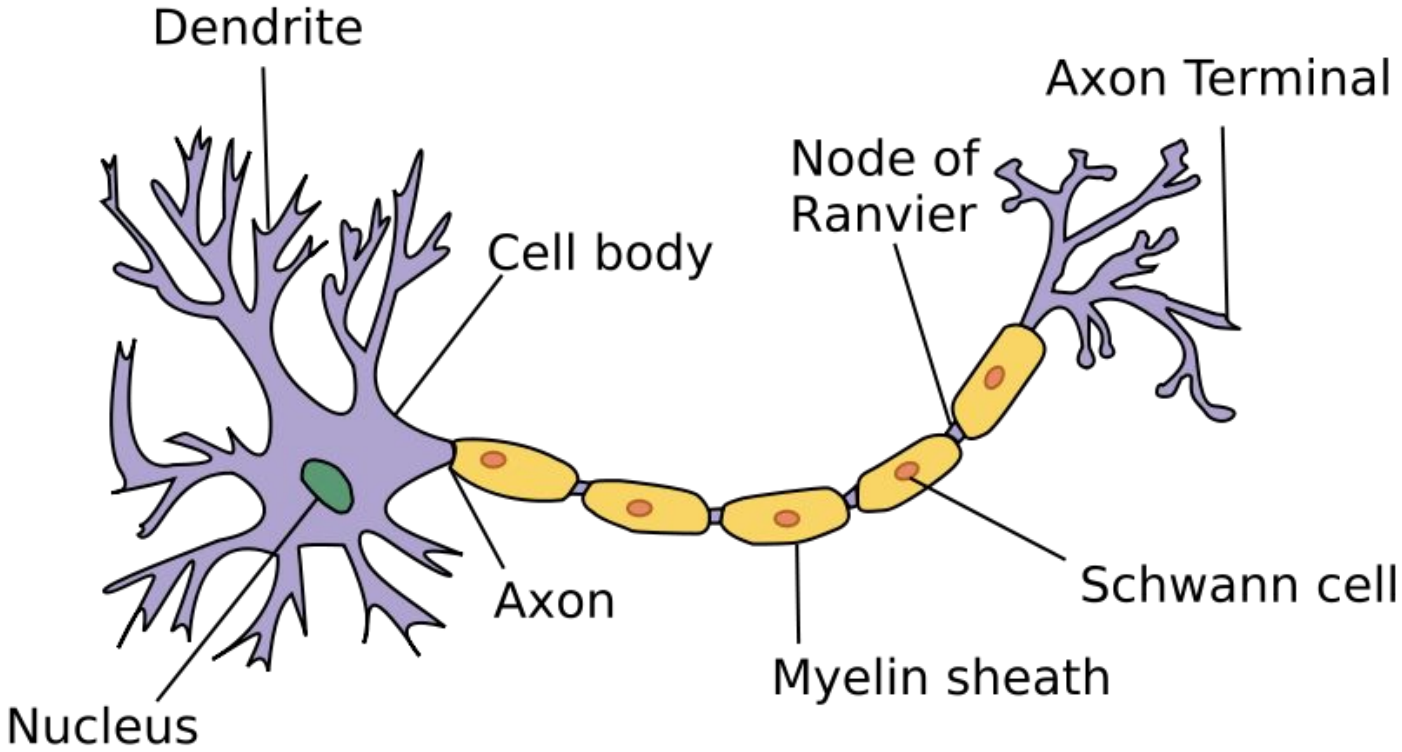


Code: https://github.com/deeplearningzerotoall/PyTorch
Slides: http://bit.ly/2VrZcWM
Lecturer: 강현우 znxlwm@gmail.com

Perceptron

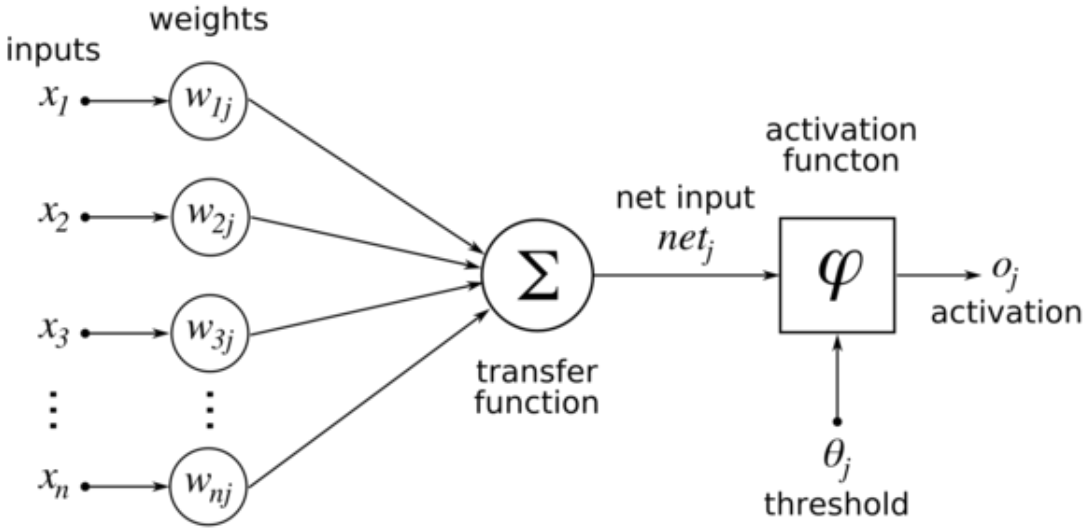
- Perceptron
- AND, OR
- XOR
- Code: xor

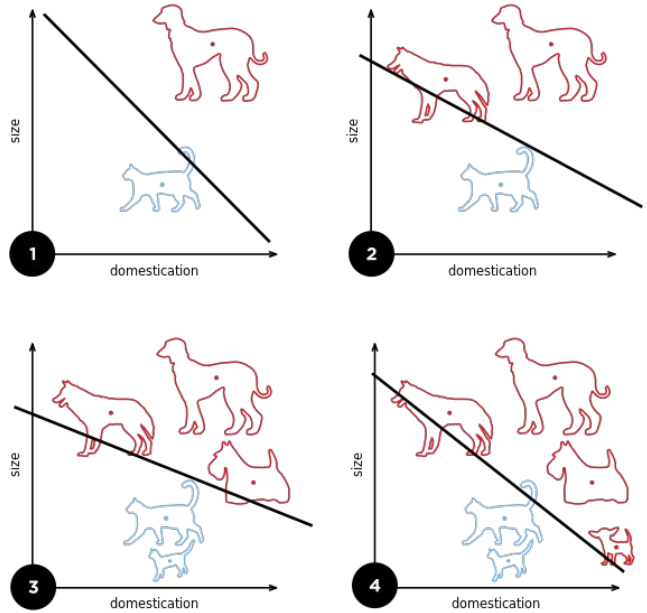
Neuron



https://commons.wikimedia.org/wiki/File:Neuron.svg

Perceptron

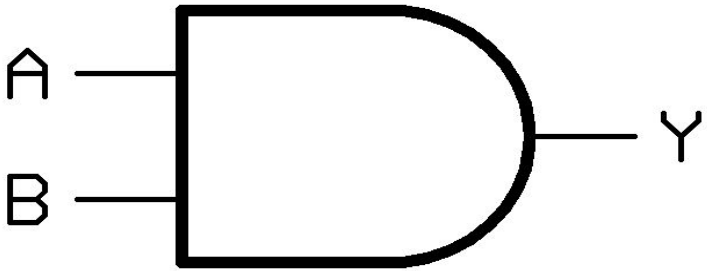


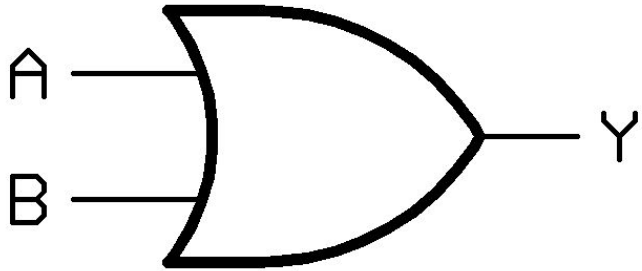


- **Linear classifier**

https://commons.wikimedia.org/wiki/File:Rosenblattperceptron.png
https://commons.wikimedia.org/wiki/File:Perceptron_example.svg

AND, OR

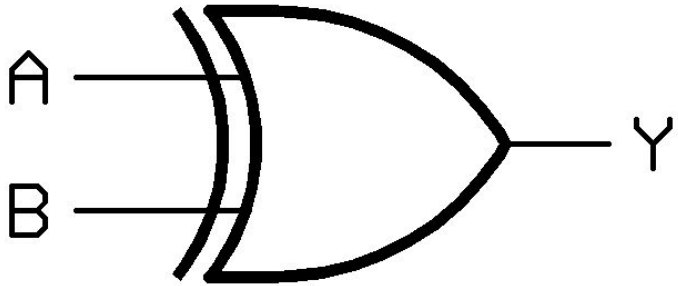




https://commons.wikimedia.org/wiki/File:Logic-gate-and-us.png
https://commons.wikimedia.org/wiki/File:Logic-gate-or-us.png

AND, OR

XOR



https://commons.wikimedia.org/wiki/File:Logic-gate-xor-us.png

xor

```
X = torch.FloatTensor([[0, 0], [0, 1], [1, 0], [1, 1]]).to(device)
Y = torch.FloatTensor([[0], [1], [1], [0]]).to(device)
# nn layers
linear = torch.nn.Linear(2, 1, bias=True)
sigmoid = torch.nn.Sigmoid()
model = torch.nn.Sequential(linear, sigmoid).to(device)
# define cost/loss & optimizer
criterion = torch.nn.BCELoss().to(device)
optimizer = torch.optim.SGD(model.parameters(), lr=1)
for step in range(10001):
    optimizer.zero_grad()
    hypothesis = model(X)
    # cost/loss function
    cost = criterion(hypothesis, Y)
    cost.backward()
    optimizer.step()
    if step % 100 == 0:
        print(step, cost.item())
```

```
0 0.7273974418640137
100 0.6931475400924683
200 0.6931471824645996
300 0.6931471824645996
...
9800 0.6931471824645996
9900 0.6931471824645996
10000 0.6931471824645996
```

```
Hypothesis:  [[0.5]
 [0.5]
 [0.5]
 [0.5]]
Correct:  [[0.]
 [1.]
 [1.]
 [0.]]
Accuracy:  0.5
```

What’s Next?

- Multi Layer Perceptron