

ML in Python

Learn Machine Learning with Microsoft Azure **ML Studio**

Microsoft Student Partner

백진헌

CNTK, TensorFlow, Keras, ...

왜 프레임워크를 사용하지 않나요?

1. Linear Regression

(Boston 집값 예측, Boston housing price data set)

2. Logistic Regression

(필기체 숫자 학습 분류 모델, MNIST data set)

3. Neural-network

(필기체 숫자 학습 Deep-learning 모델, MNIST data set)

1. Linear Regression

(Boston 집값 예측, Boston housing price data set)

2. Logistic Regression

(필기체 숫자 학습 분류 모델, MNIST data set)

3. Neural-network

(필기체 숫자 학습 Deep-learning 모델, MNIST data set)

Linear Regression

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n)$$

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n) + b$$

x_1, x_2, \dots, x_n : Y 를 예측할 때 필요한 유의미한 피쳐

w_1, w_2, \dots, w_n, b : Y 를 예측할 때 사용되는 각 피쳐에 대한 무게 값

Y : 학습된 모델을 통해 예측한 값

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n) + b$$

그럼 우리가 학습을 통해 계산해야 하는 것은?

Linear Regression

$$Y_{\text{(predict)}} = (w1*x1) + (w2*x2) + \dots + (wn*xn) + b$$

Linear Regression

$$Y_{\text{(predict)}} = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n) + b$$

$$(Y - Y_{\text{(predict)}})^2$$

$$Y_{\text{(predict)}} = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n) + b$$

$$\text{Cost Function} : (Y - Y_{\text{(predict)}})^2$$

$$Y_{\text{(predict)}} = (w1*x1) + (w2*x2) + \dots + (wn*xn) + b$$

Cost Function – Low : 더 적게 학습

Cost Function – High : 더 많이 학습

**우리가 도움말을 보지 않고 스마트폰을 사용하듯이,
Machine Learning 잘 모르겠지만 일단 해보자.**

Linear Regression

<https://github.com/JinheonBaek/ML-Evangelism>

MSP Machine Learning Evangelism [Add topics](#) [Edit](#)

8 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

JinheonBaek Change ML-Studio image filename-extension (from png to PNG) Latest commit 77003fa 5 days ago

img	Add ML-Studio for predicting my number	5 days ago
ML-Studio.ipynb	Change ML-Studio image filename-extension (from png to PNG)	5 days ago
README.md	Add Azure ML Sudio Connecting Comment in README file	5 days ago
linear-regression.ipynb	Change linear-regression error print epoch size	13 days ago
logistic-regression.ipynb	Push all jupyter notebook files used for evangelism	13 days ago
neural-network.ipynb	Push all jupyter notebook files used for evangelism	13 days ago

README.md

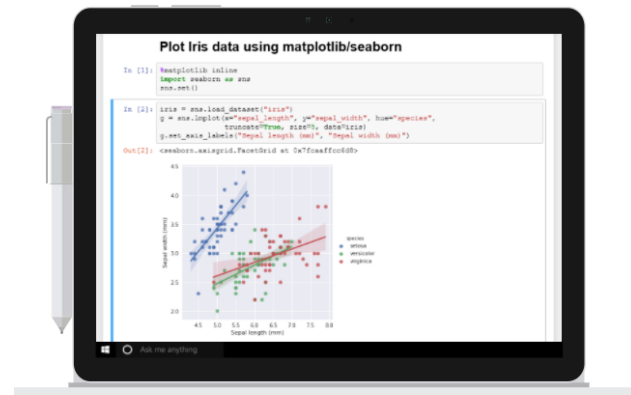
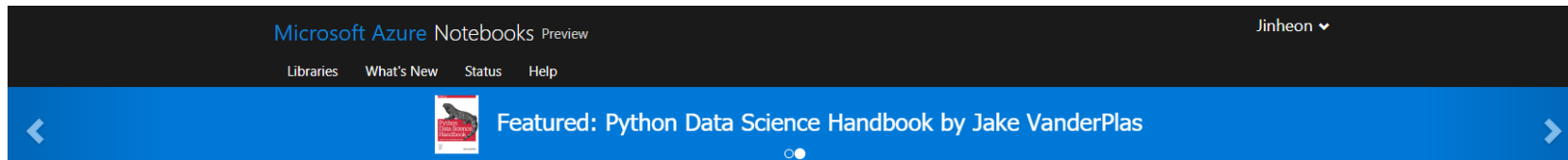
ML-Evangelism

MSP Machine Learning Evangelism

1. Linear Regression (Boston Dataset)
2. Logistic Regression (MNIST Dataset)
3. Neural Network (MNIST Dataset)
4. Azure ML Studio (Connected for MNIST Dataset predicting)

Linear Regression

<https://notebooks.azure.com/>



Interactive coding
in your browser

Free, in the cloud,
powered by [Jupyter](#)

[Get Started](#)

Powerful
Languages

Numerous Charting
Libraries

Built for
Sharing

Use the languages of Data Science

Azure Notebooks provides execution environments for Python 2, Python 3, F#, and R.

Linear Regression

1. Login
2. Libraries 메뉴
3. New Library
4. From Github
5. JinheonBaek/ML-Evangelism (GitHub repository)
6. Import
7. Linear-regression.ipynb

다 함께 해봅시다.

Linear Regression 설명

1. Linear Regression

(Boston 집값 예측, Boston housing price data set)

2. Logistic Regression

(필기체 숫자 학습 분류 모델, MNIST data set)

3. Neural-network

(필기체 숫자 학습 Deep-learning 모델, MNIST data set)

Logistic Regression

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n)$$

개 : 0 / 고양이 : 1

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n)$$

개 : 0 / 고양이 : 1

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n)$$

$Y > 0$: True

$Y < 0$: False

Cost Function ???

개 : 0 / 고양이 : 1

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n)$$

$\text{Sigmoid}(Y) > 0.5 : \text{True}$

$\text{Sigmoid}(Y) < 0.5 : \text{False}$

개 : 0 / 고양이 : 1

$$Y = (w_1 * x_1) + (w_2 * x_2) + \dots + (w_n * x_n)$$

Sigmoid(Y) > 0.5 : True

Sigmoid(Y) < 0.5 : False

Cost Function : 1(고양이) – Sigmoid(Y) (이런느낌)

Logistic Regression

$$Y_1 = (w_{11} * x_1) + (w_{12} * x_2) + \dots + (w_{1n} * x_n)$$

$$Y_2 = (w_{21} * x_1) + (w_{22} * x_2) + \dots + (w_{2n} * x_n)$$

...

$$Y_n = (w_{n1} * x_1) + (w_{n2} * x_2) + \dots + (w_{nn} * x_n)$$

Logistic Regression

$$Y_1 = (w_{11} * x_1) + (w_{12} * x_2) + \dots + (w_{1n} * x_n)$$

$$Y_2 = (w_{21} * x_1) + (w_{22} * x_2) + \dots + (w_{2n} * x_n)$$

...

$$Y_n = (w_{n1} * x_1) + (w_{n2} * x_2) + \dots + (w_{nn} * x_n)$$

$$\text{Max} (y_{(1 \sim n)}) = y_3 ?$$

Logistic Regression

$$Y_1 = (w_{11} * x_1) + (w_{12} * x_2) + \dots + (w_{1n} * x_n)$$

$$Y_2 = (w_{21} * x_1) + (w_{22} * x_2) + \dots + (w_{2n} * x_n)$$

...

$$Y_n = (w_{n1} * x_1) + (w_{n2} * x_2) + \dots + (w_{nn} * x_n)$$

$$\text{Max} (\text{Sigmoid}(y_{(1 \sim n)})) = y_3 ?$$

MNIST data set ?

Logistic Regression



28 x 28, 필기체 숫자 (0 ~ 9)

Logistic Regression

$$Y_0 = (w_{11} * x_1) + (w_{12} * x_2) + \dots + (w_{1n} * x_n)$$

$$Y_1 = (w_{21} * x_1) + (w_{22} * x_2) + \dots + (w_{2n} * x_n)$$

...

$$Y_9 = (w_{n1} * x_1) + (w_{n2} * x_2) + \dots + (w_{nn} * x_n)$$

다 함께 해봅시다.

Logistic Regression 설명

1. Linear Regression

(Boston 집값 예측, Boston housing price data set)

2. Logistic Regression

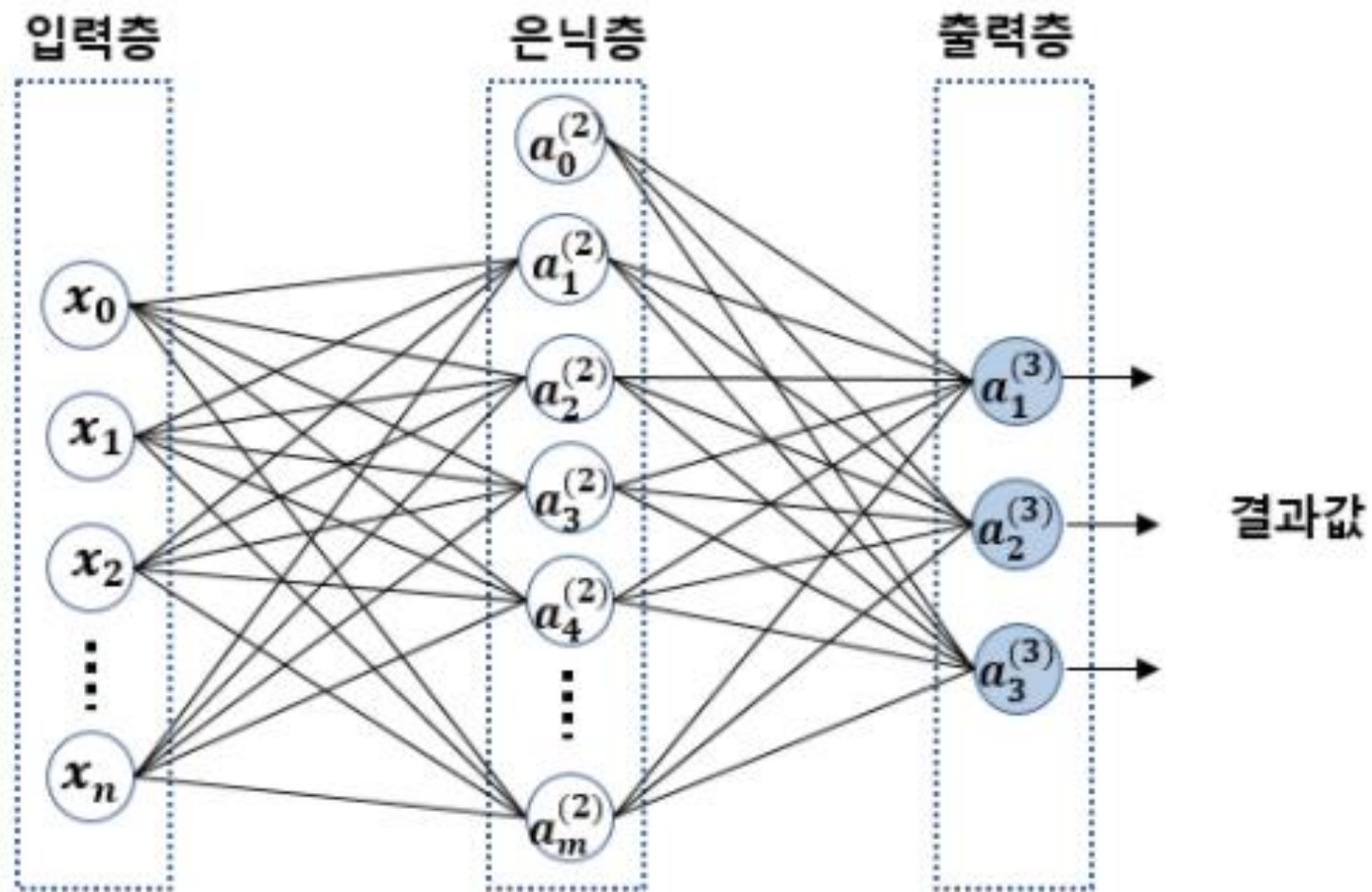
(필기체 숫자 학습 분류 모델, MNIST data set)

3. Neural-network

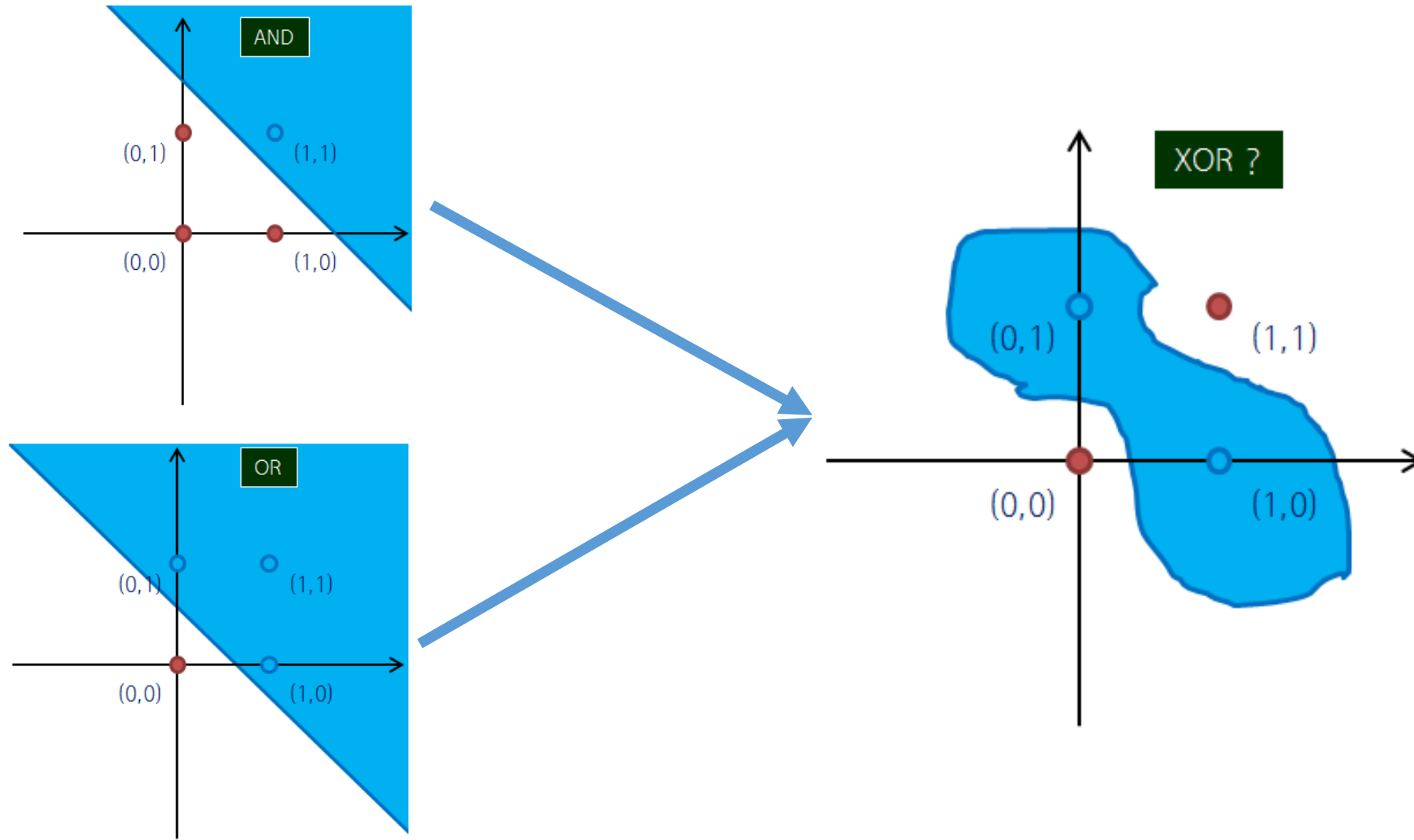
(필기체 숫자 학습 Deep-learning 모델, MNIST data set)

Neural-network 장점

Neural-network

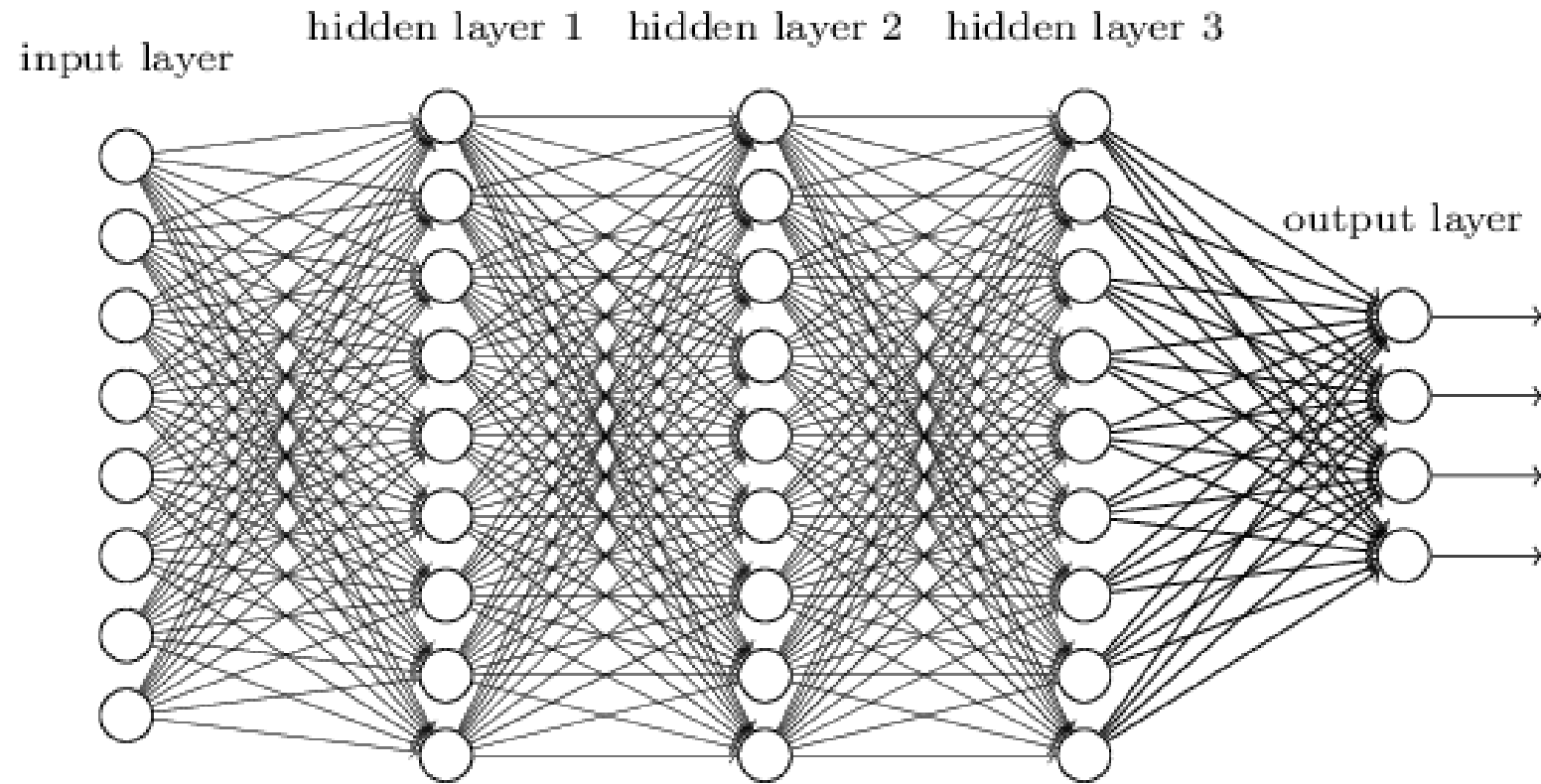


Neural-network

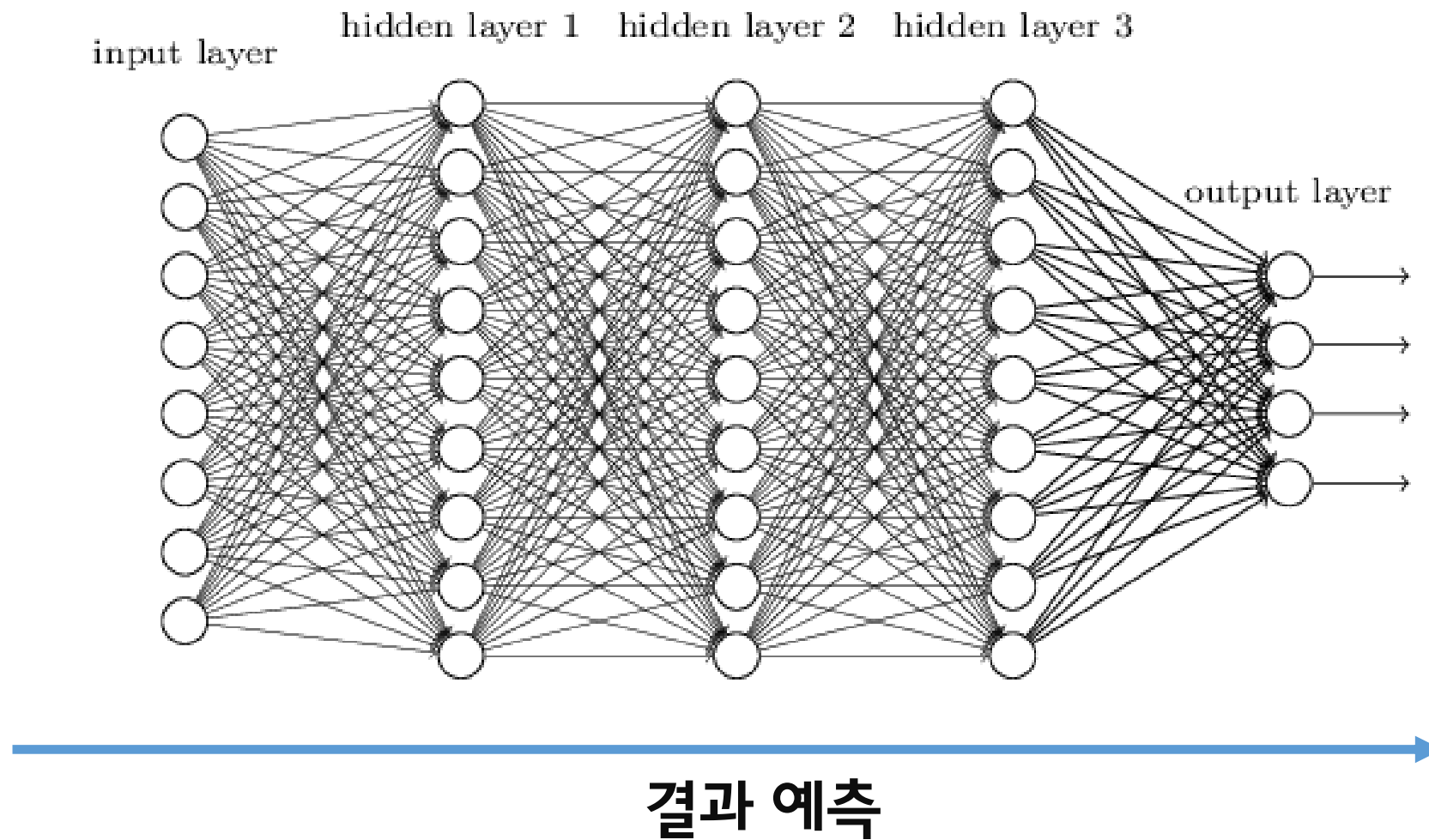


우리가 해야할 일은?

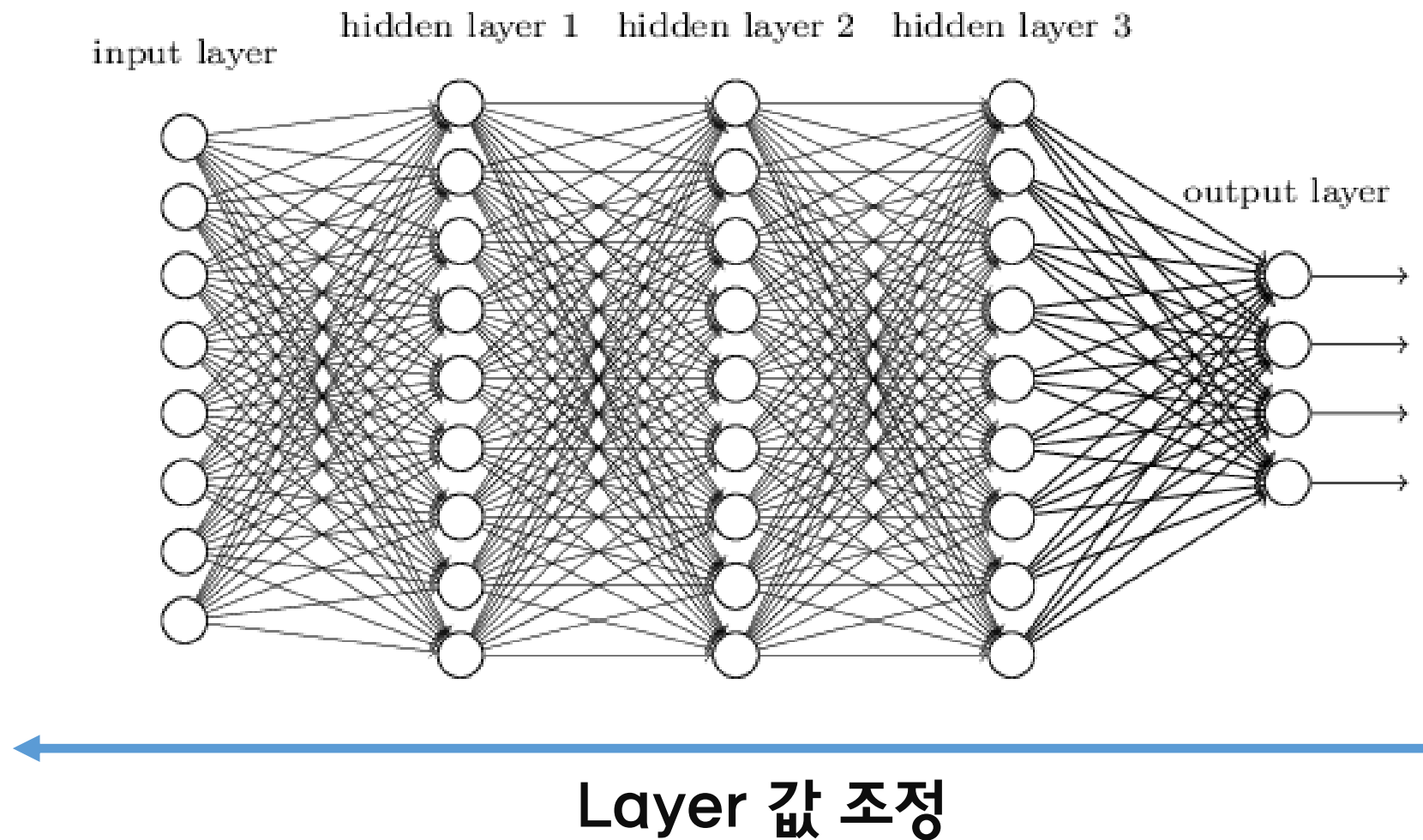
Neural-network



Neural-network



Neural-network



다 함께 해봅시다.

Neural-network 설명

Q & A

감사합니다