

# 초보자를 위한 코딩하기

## - AI의 핵심 구글 텐서플로우

2020년 9월1일

한국과학기술정보연구원 슈퍼컴퓨팅응용센터 이홍석

# 목차

- 인공지능에 대하여 알아보자
  - ✓ 인공지능의 역사
  - ✓ 어떻게 AI 알고리즘을 만드는가?
- 학습 모델로 인공지능 신경망의 이해
- 구글의 텐서플로우 실습해보기

# 이홍석

← → ↻ sites.google.com/view/deepai/home



## DeepAI

- ^ Home
- Research
- Lecture
- Publications
- Contacts Us



이 홍 석 Hong Suk Yi

과학기술연합대학원대학교/교수

Korea Institute of Science and Technology Information

### Work Experiences

Principal Researcher, Korea Institute of Science and Technology Information (KISTI)/Supercomputing Center(1999.12~present)

Adjunct Professor, University of Science and Technology (UST) (2020.2~present)

Adjunct Professor, Korea University (2009.8~2013.12)

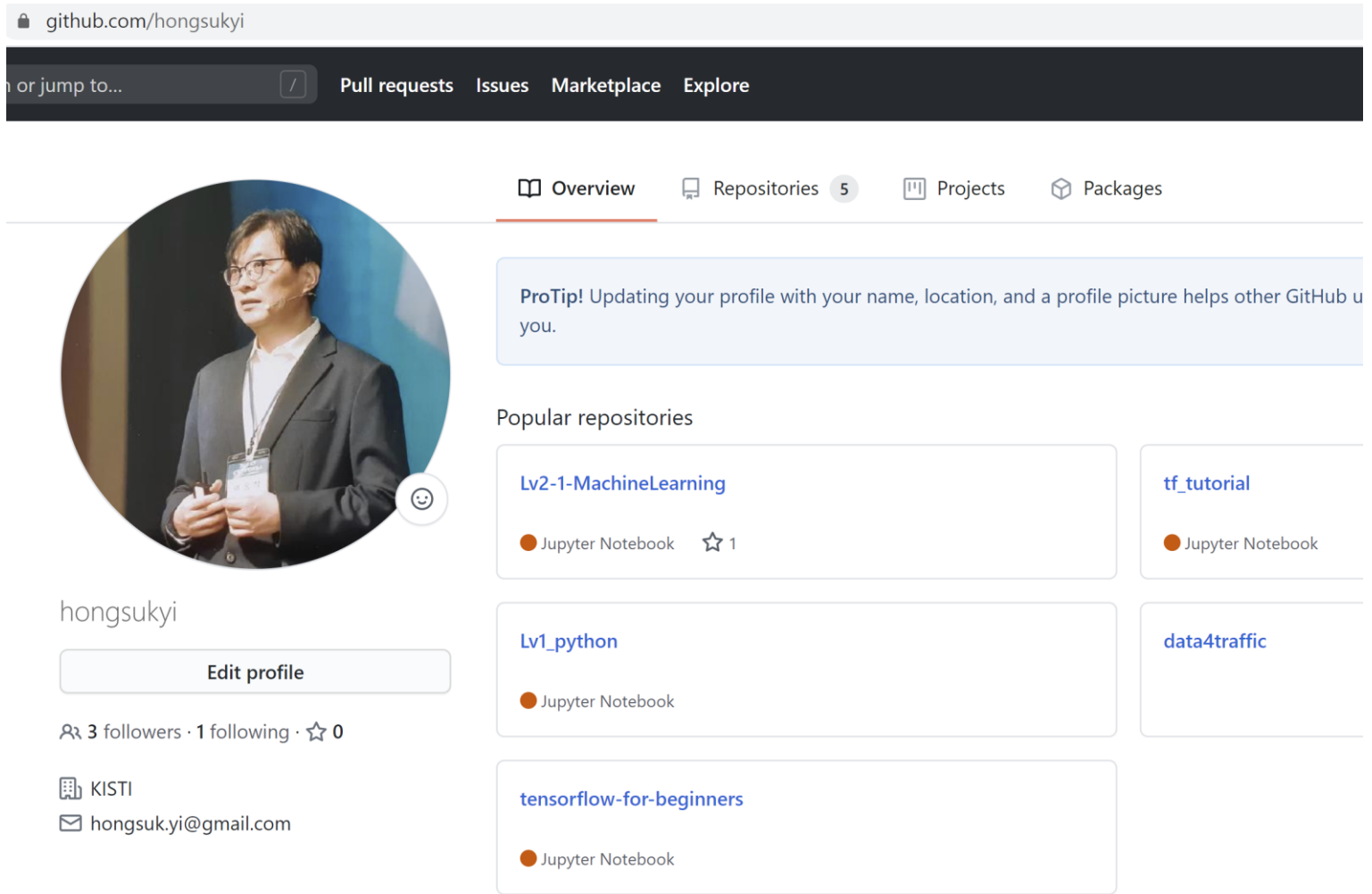
### Research Interests

텐서플로우 2.0 딥러닝 알고리즘 성능 최적화

라지스케일 GPU 분산 병렬 컴퓨팅

CCTV 영상에서 차량/사람/사물을 탐색, 추적, 재인식 기술

# 참고사이트 : <https://github.com/hongsukyi/tensorflow-for-beginners>



The screenshot shows the GitHub profile of a user named 'hongsukyi'. The profile includes a circular profile picture of a man with glasses, a bio section with an 'Edit profile' button, and statistics showing 3 followers and 1 following. The 'Repositories' tab is selected, displaying a list of repositories: 'Lv2-1-MachineLearning' (1 star), 'Lv1\_python', and 'tensorflow-for-beginners'. A 'ProTip!' message is visible at the top right of the repository list.

github.com/hongsukyi

h or jump to... / Pull requests Issues Marketplace Explore

Overview Repositories 5 Projects Packages

**ProTip!** Updating your profile with your name, location, and a profile picture helps other GitHub users find you.

Popular repositories

- Lv2-1-MachineLearning**  
Jupyter Notebook ☆ 1
- tf\_tutorial**  
Jupyter Notebook
- Lv1\_python**  
Jupyter Notebook
- data4traffic**
- tensorflow-for-beginners**  
Jupyter Notebook

hongsukyi

Edit profile

3 followers · 1 following · ☆ 0

KISTI

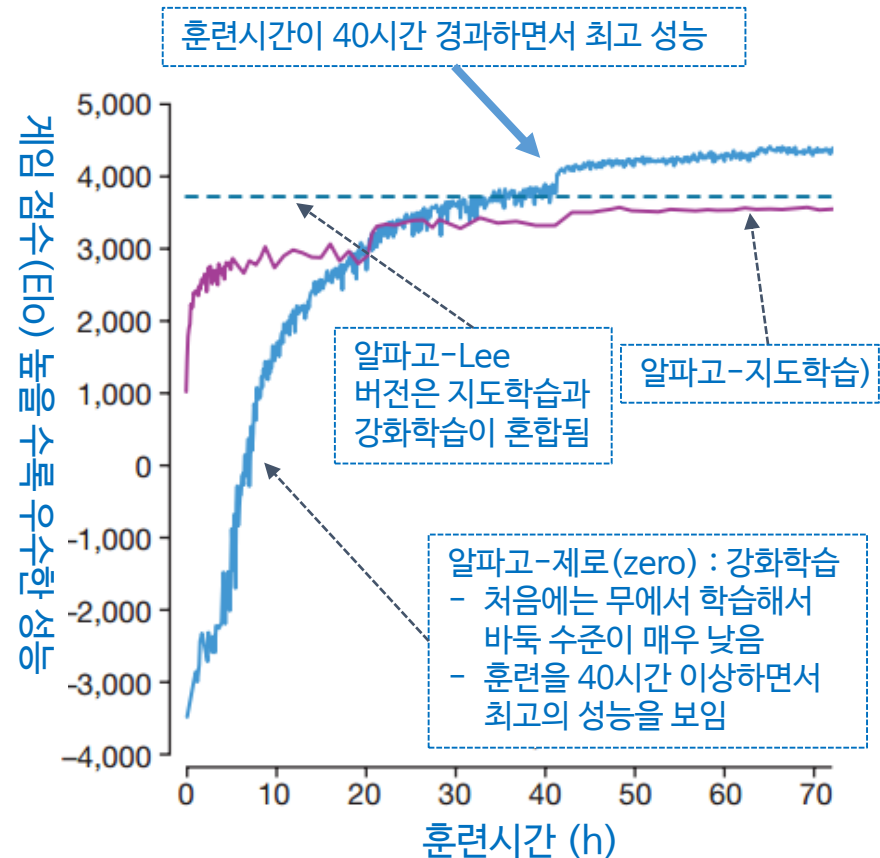
hongsuk.yi@gmail.com

14 contributions in the last year

# ANN의 역사 (14): 알파고 (2016)

알파고 (AlphaGo)는 3개 버전

- ① 알파고-지도학습 (바둑 기보)
- ② 알파고-리 (이세돌 9단)
- ③ 알파고-제로 (강화학습)



자료: Mastering the game of Go without human knowledge, David Silver, et al. Nature(2017)

# 신경망의 역사 1950년 대부터

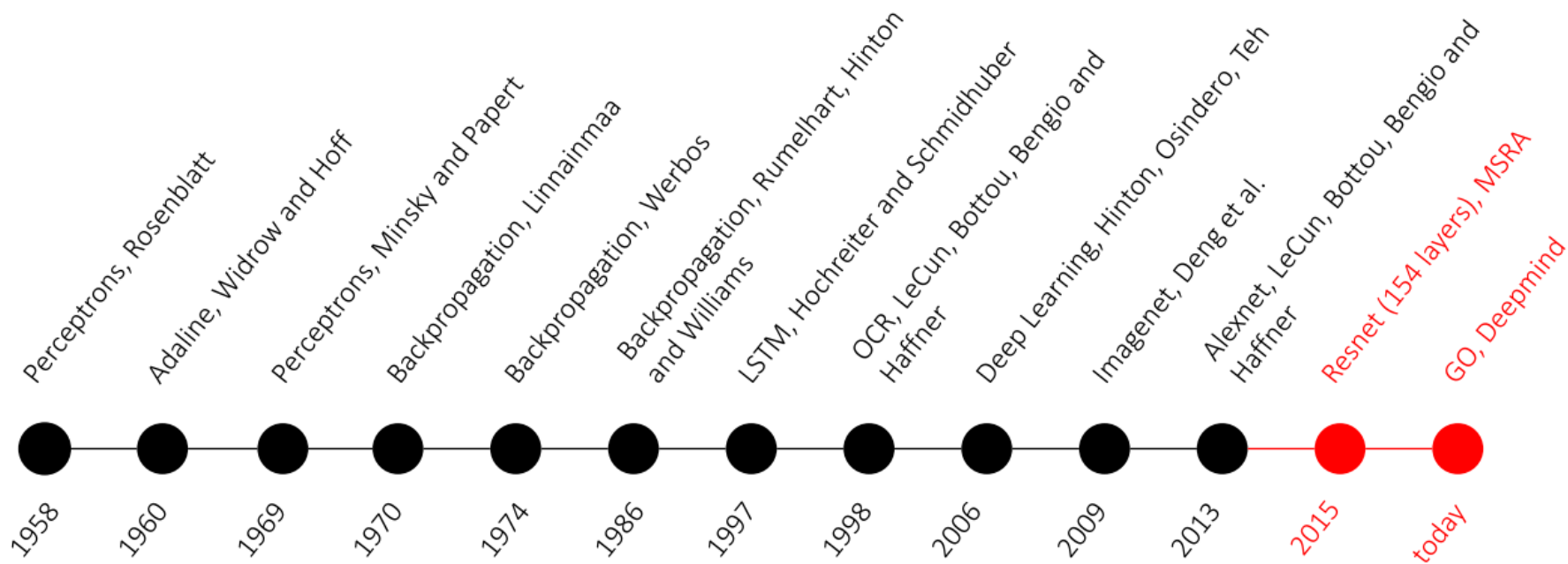
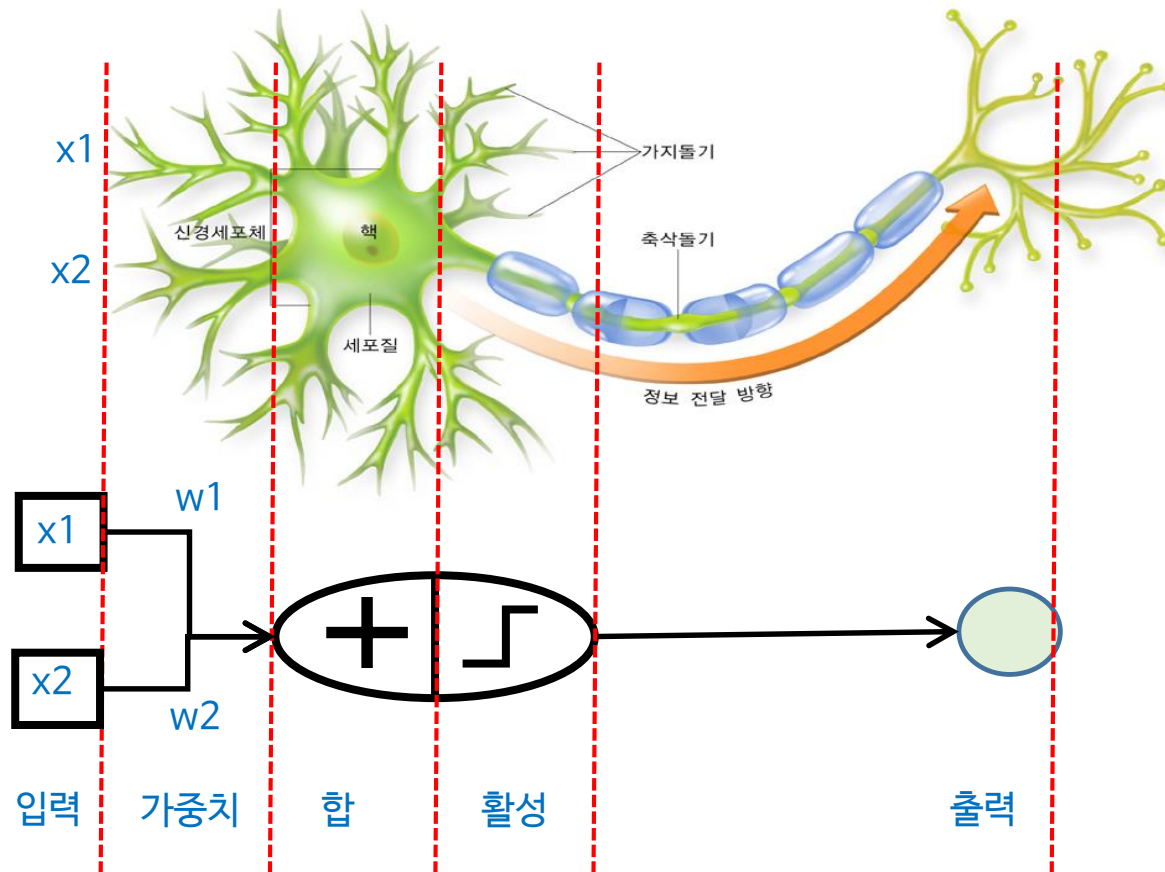


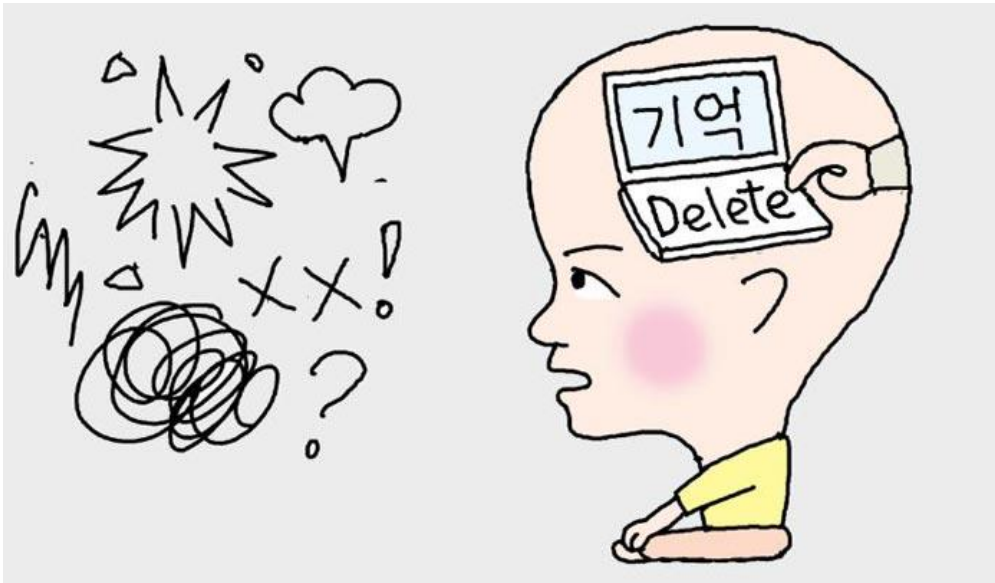
Fig. credit to Efstratios Gavves, Intro. to DL

# 인공신경망 (ANN) 모델



# 기억 (메모리)

두 살 때 기억은 무엇인가?





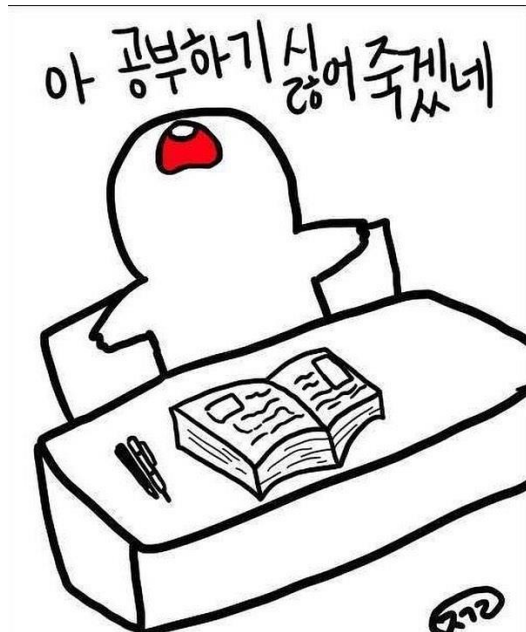
# (뇌)기억 한 것은 대부분 잊어버림

기억량(%)

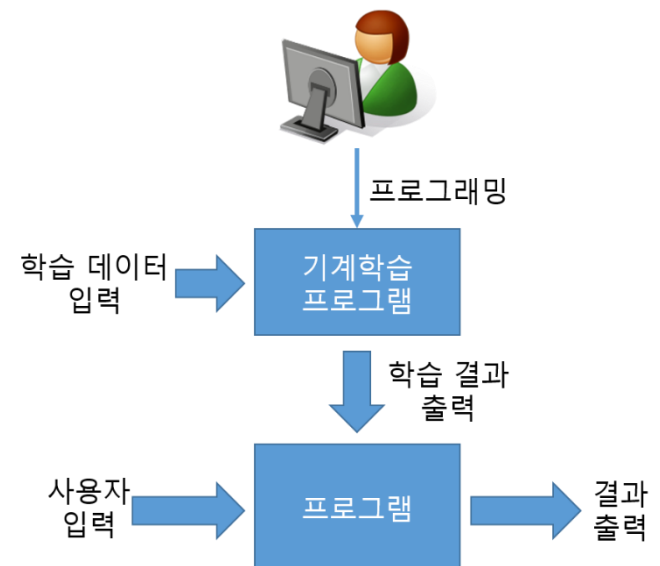


# 반복하는 것은 매우 지루하다!

공부 잘하는 비결은 반복 학습



컴퓨터는 반복을 잘한다



# 좌뇌와 우뇌의 차이는 왜 생길까?



강화학습은 보상!

상도 있고, 벌도 있고

상을 받기 위해서는, 우리의  
뇌는 빨리 빨리 지식을 배워감

## 02. 데이터와 학습

인터넷 기반의 Colab에서 프로그래밍 하기

# 지도학습

지도학습 : 입력, 출력, 라벨, 예측

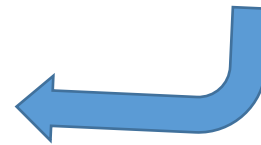
부모

자녀



재활용 통 (라벨)

빅데이터



# MNIST 데이터

label = 5



label = 0



label = 4



label = 1



label = 9



label = 2



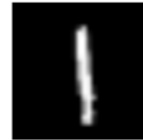
label = 1



label = 3



label = 1



label = 4



label = 3



label = 5



label = 3



label = 6



label = 1



label = 7



label = 2



label = 8



label = 6



label = 9



# MNIST 데이터 구조

MNIST~ Modified [National Institute of Standards and Technology](https://en.wikipedia.org/wiki/MNIST_database)



60,000

train-images-idx3-ubyte.gz

train-labels-idx1-ubyte.gz

10,000

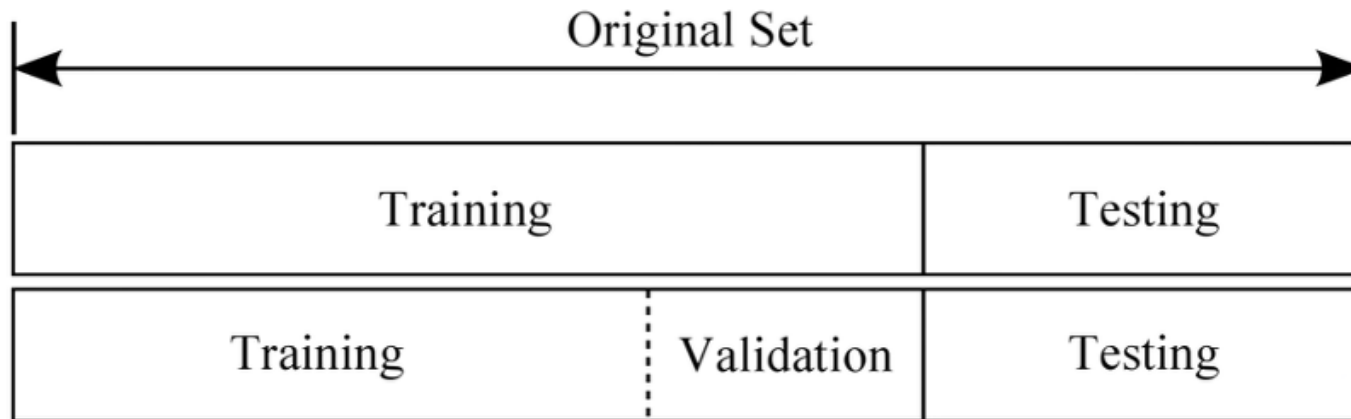
t10k-images-idx3-ubyte.gz

t10k-labels-idx1-ubyte.gz

[https://en.wikipedia.org/wiki/MNIST\\_database](https://en.wikipedia.org/wiki/MNIST_database)

# 기계학습/인공지능을 위한 데이터 나누기

- 학습은 70% (10%는 검증을 포함)
- 테스트는 20%





## 패션 NMINST 데이터(3)

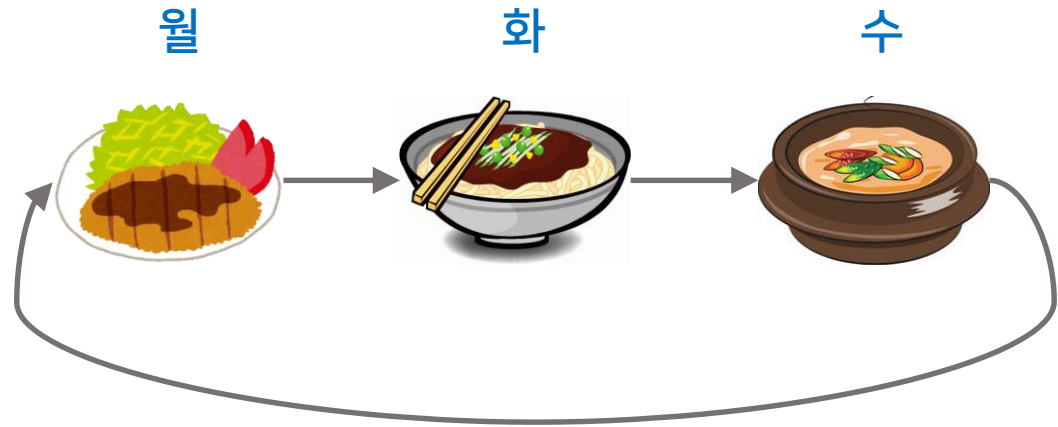
```
for row in range(n_rows):  
    for col in range(n_cols):  
        index = n_cols * row + col  
        plt.subplot(n_rows, n_cols, index + 1)  
        plt.imshow(X_train[index], cmap="binary", interpolation="nearest")  
        plt.axis('off')  
        plt.title(class_names[y_train[index]], fontsize=12)
```



# 학습 방법에 따른 분류

- Supervised Learning (지도학습)
  - ✓ Input 과 labels을 이용한 학습
  - ✓ 분류(classification), 회귀(regression)
- Unsupervised Learning (비지도학습)
  - ✓ Input만을 이용한 학습
  - ✓ 군집화(clustering), 압축(compression)
- Reinforcement Learning (강화학습)
  - ✓ Label 대신 reward가 주어짐
  - ✓ Action selection, policy learning

# 데이터 속의 패턴 예측: 간단한 순환신경망(RNN)



RNN = Recurrent Neural Network

# 날씨에 따라 패턴이 벗어난다

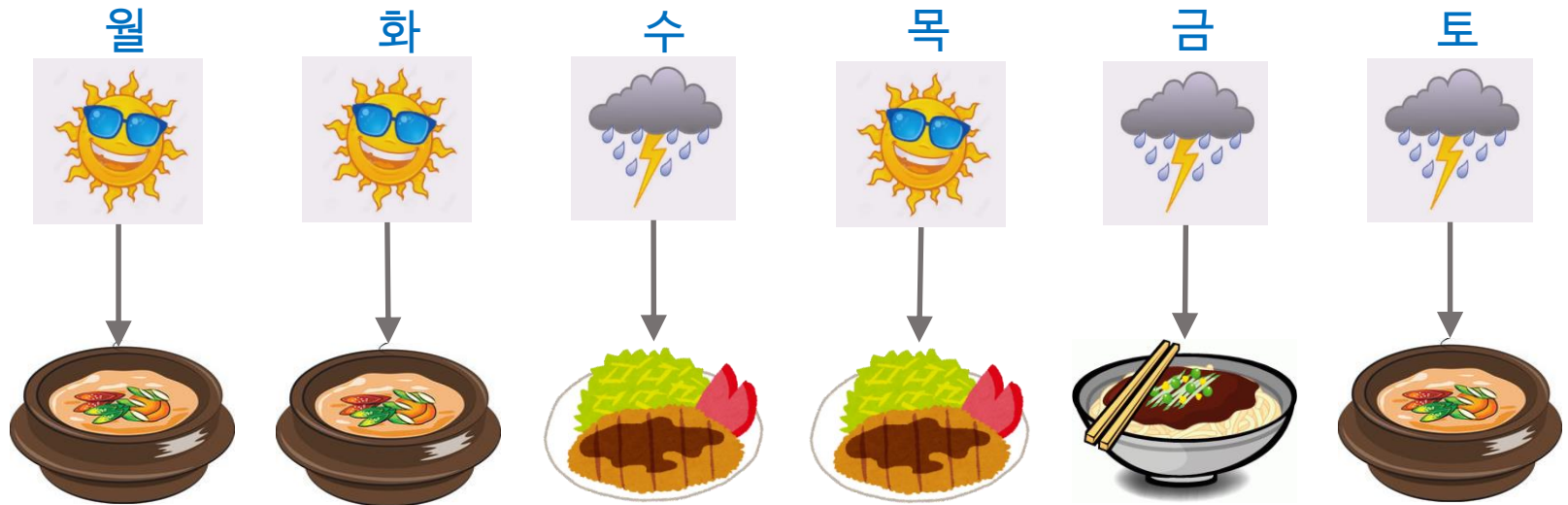


어제 먹었던 음식



원래 RNN 패턴

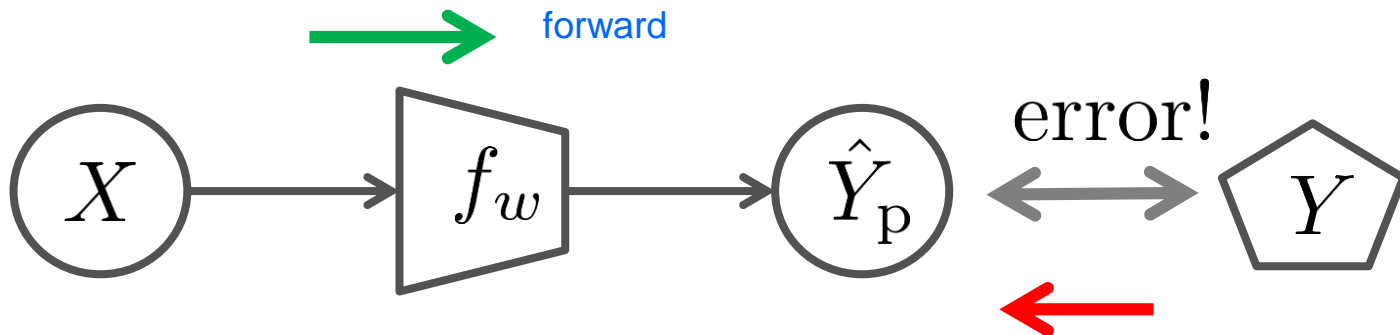
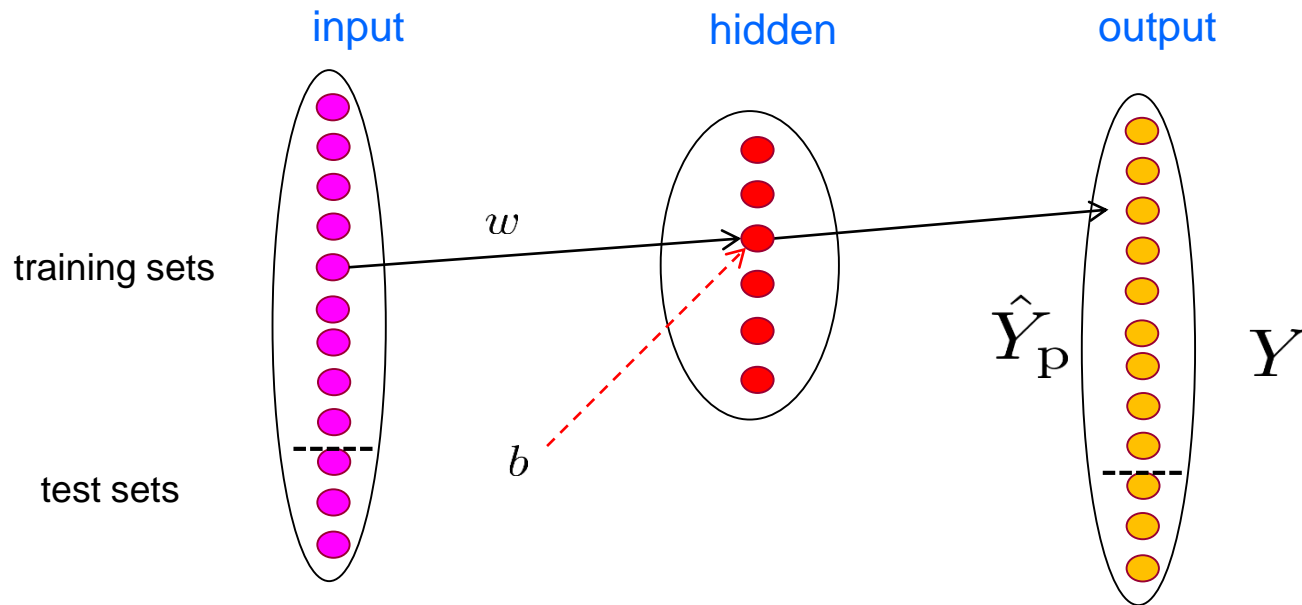
# 날씨에 따른 순환신경망(RNN)



## 03. 인공신경망 모델은 어떻게 작동하는가?

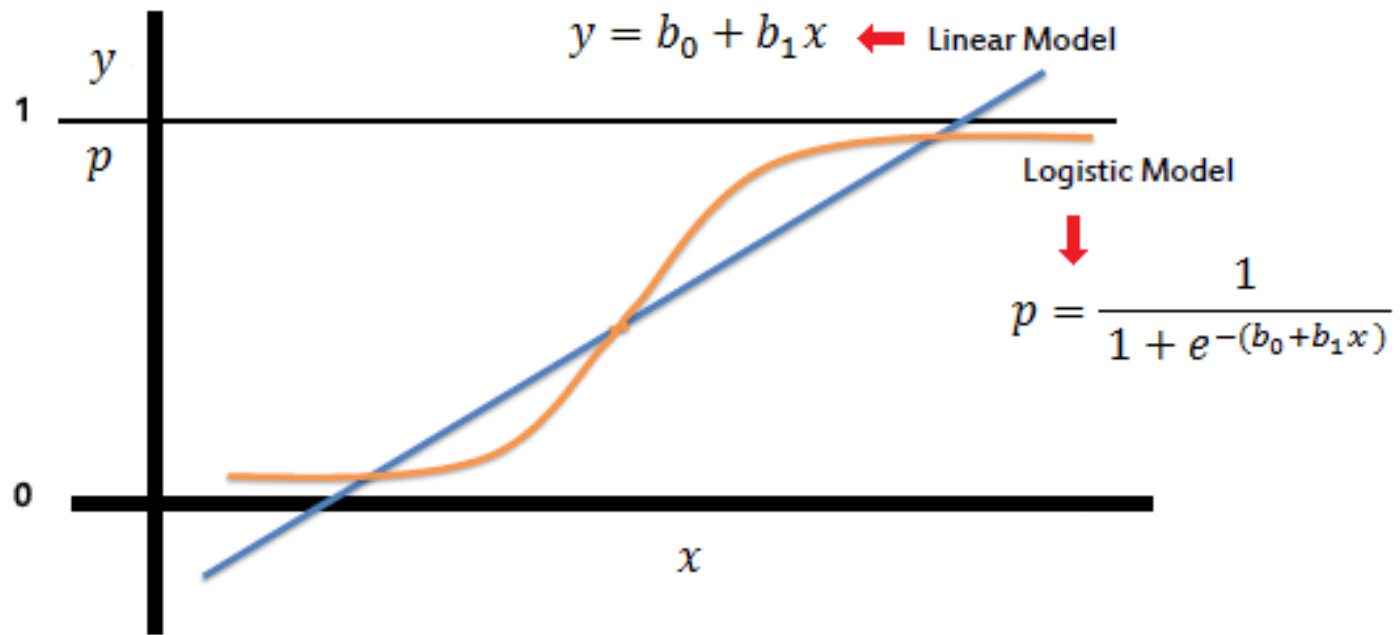
데이터, 학습과정, 결과 예측하기

# 인공신경망 (Artificial Neural Network) 학습



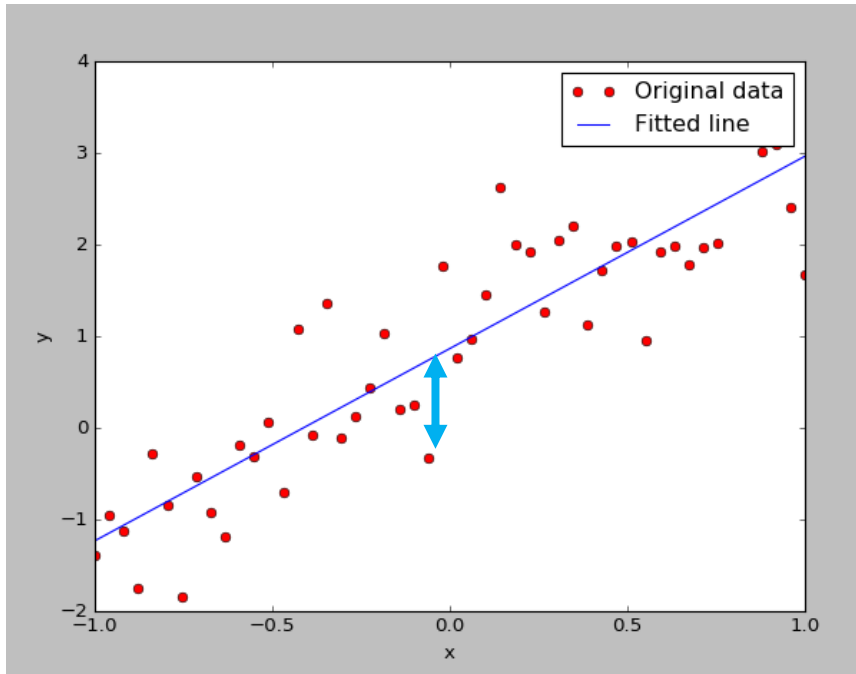
backpropagation of the error over the network using derivative function

# linear models





# 인공신경망의 기본 모델 : 선형회귀



가정:

$$H(x) = Wx + b$$

비용 (손실)

$$\text{cost}(w, b) = \sum_{i=0}^{M-1} (f(h(x^i)) - y^i)^2$$

기울기 구하기

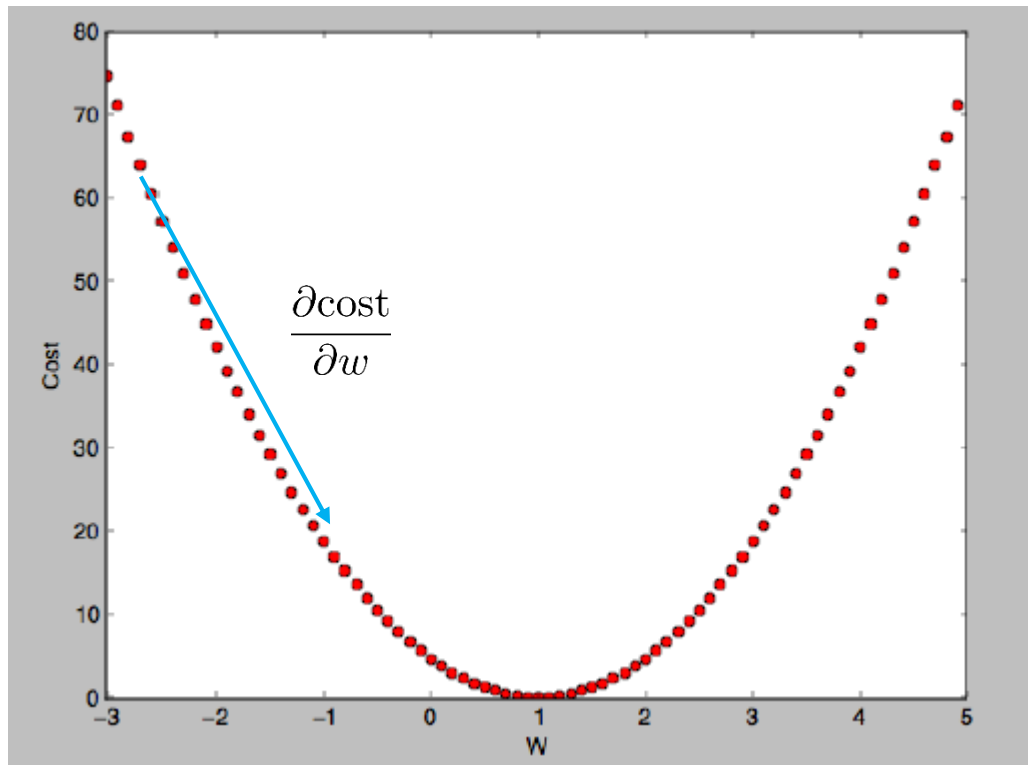
$$w_{\text{updated}} \approx w - \alpha \cdot \frac{\partial \text{cost}}{\partial w}$$

비용은 예측과 실제값과의 차이

비용은 최소화하는 가중치를 구하는 문제 : 신경망으로 근사치 구함.

## 비용함수의 모양은 2차 함수.

$$\text{cost}(w, b) = \sum_{i=0}^{M-1} (f(h(x^i)) - y_-^i)^2 \quad \frac{\partial \text{cost}}{\partial w} = \frac{\partial \text{cost}}{\partial y} \cdot \frac{\partial y}{\partial f} \cdot \frac{\partial f}{\partial h} \cdot \frac{\partial h}{\partial w}$$



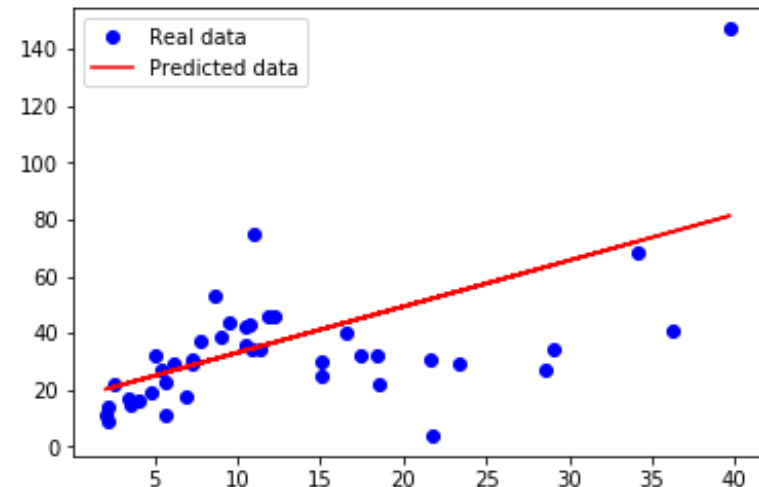
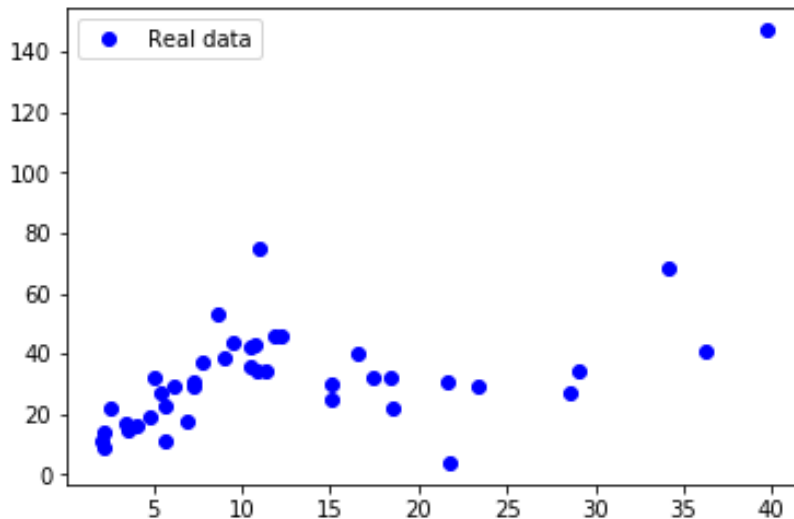
$$w_{\text{updated}} \approx w - \alpha \cdot \frac{\partial \text{cost}}{\partial w}$$

$\alpha$  is the learning rate.

## 모델을 설정하기

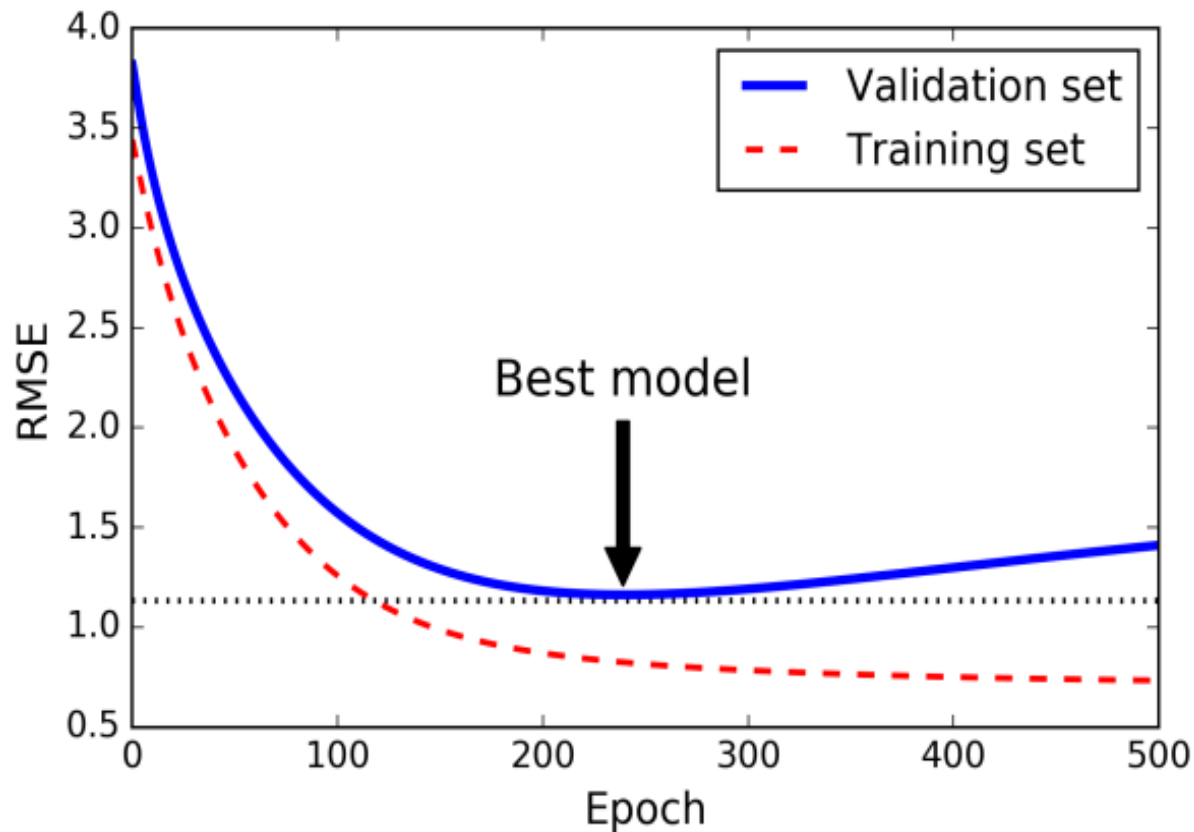
$$Y_{\text{predicted}} = X * w + b$$

$$\text{loss} = (Y - Y_{\text{predicted}})^2$$



# 인공 신경망 반복 계산으로 학습하기.

비용함수=RMSE = 평균 제곱근 오차



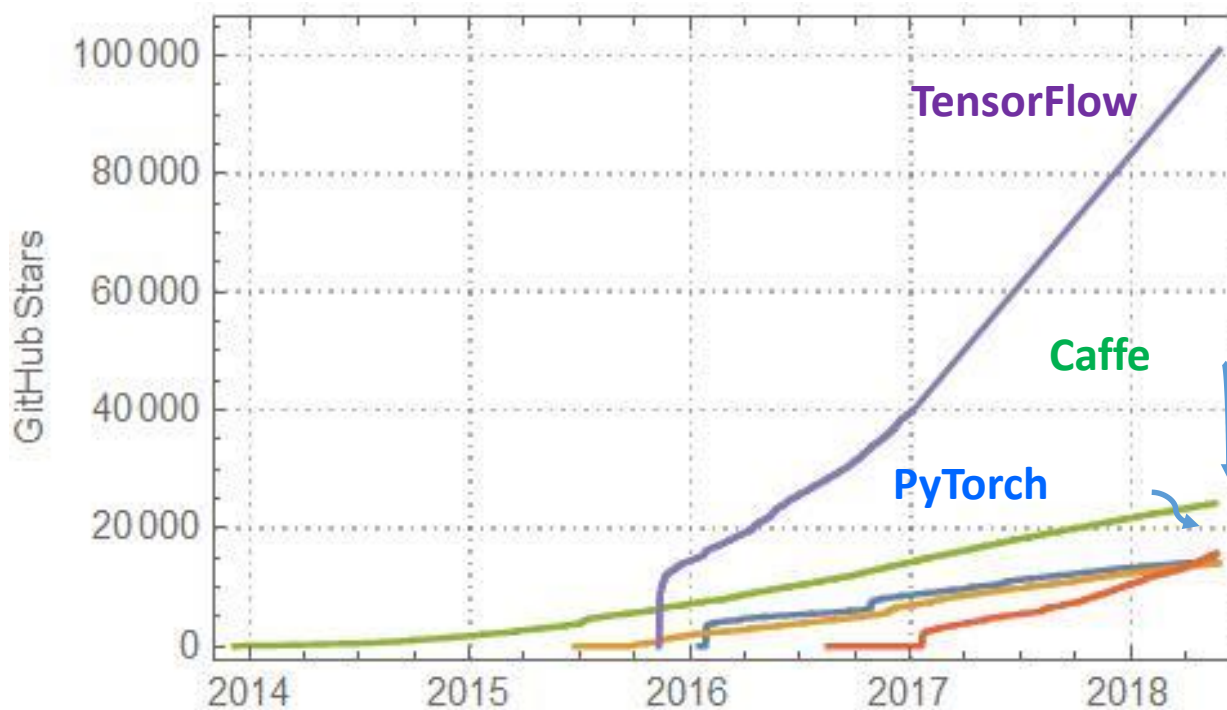
## 04. 구글 텐서플로우에서 첫 프로그램 해보기

인터넷 기반의 Colab에서 프로그래밍 하기

# Deep Learning Framework

구글 텐서플로우 페이스북 파이토치 앰비디아 까페

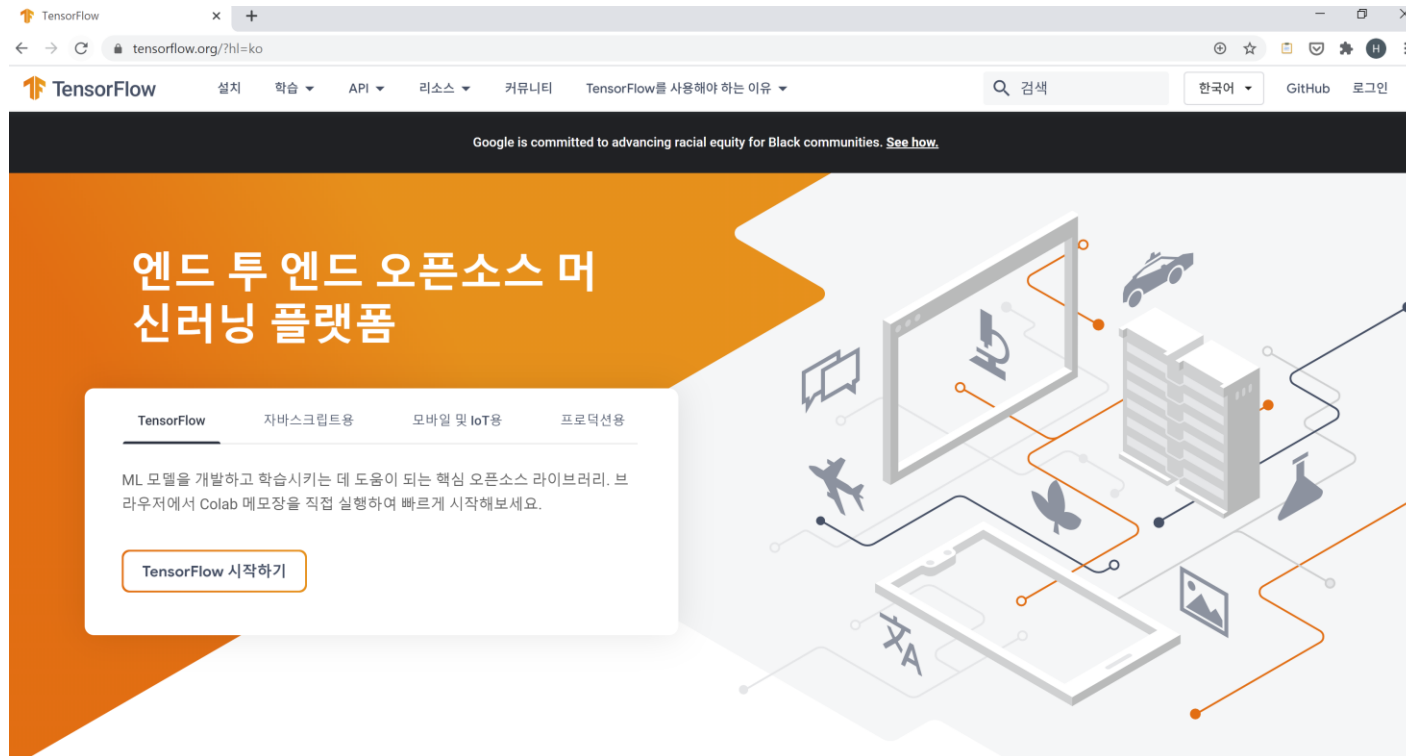
Open Source Deep Learning framework GitHub Stars



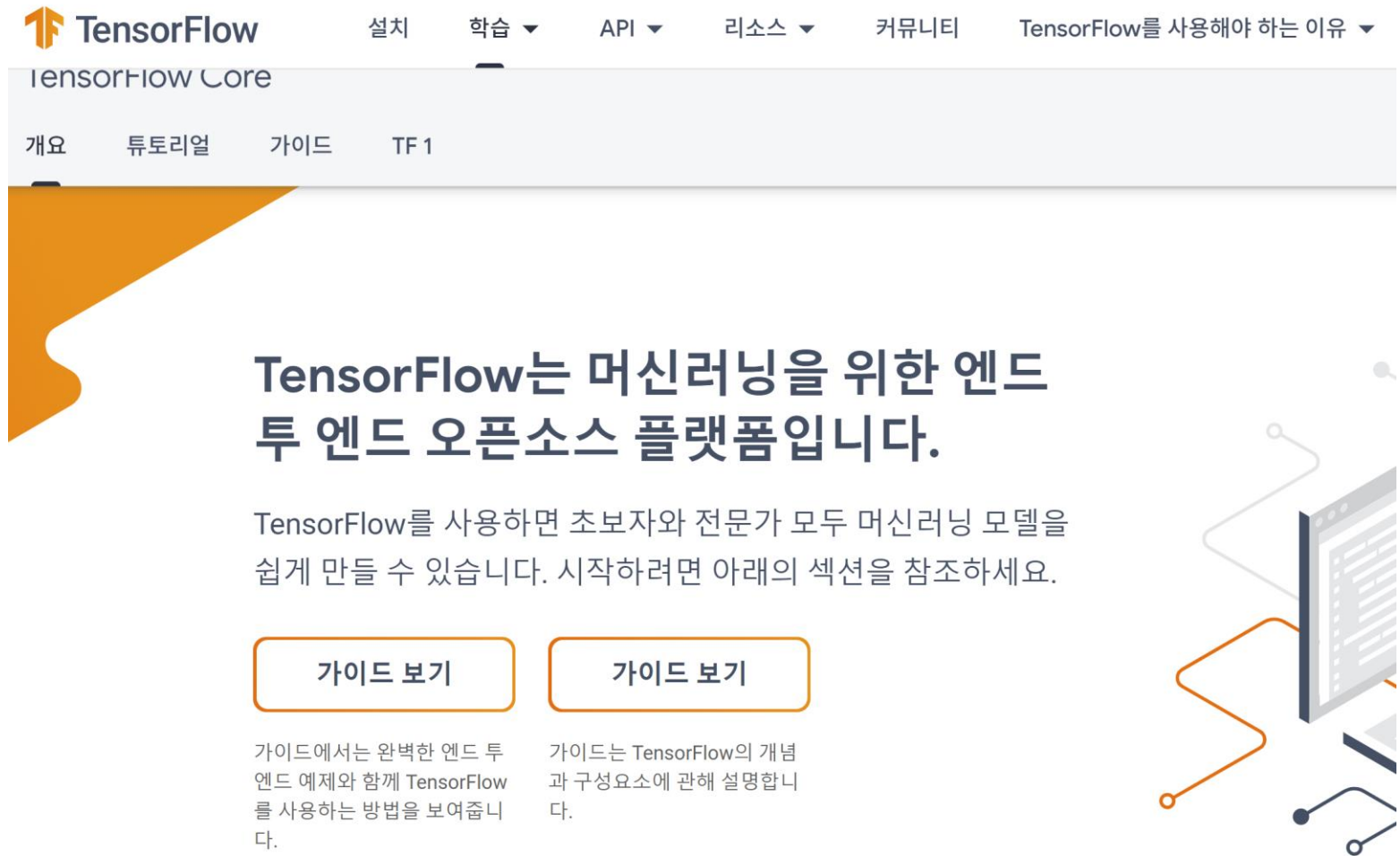
<http://community.wolfram.com/groups/-/m/t/1345642>

# Colab 실습

- Google Tensorflow 검색하기



# 텐서플로우 실습



The image is a screenshot of the TensorFlow website's homepage. At the top, there is a navigation bar with the TensorFlow logo on the left and several menu items: '설치' (Install), '학습' (Learn) with a dropdown arrow, 'API' with a dropdown arrow, '리소스' (Resources) with a dropdown arrow, '커뮤니티' (Community), and 'TensorFlow를 사용해야 하는 이유' (Why use TensorFlow) with a dropdown arrow. Below the navigation bar, there is a sub-header 'TensorFlow Core' and a secondary navigation bar with links: '개요' (Overview), '튜토리얼' (Tutorial), '가이드' (Guide), and 'TF 1'. The main content area features a large orange decorative shape on the left. The headline reads 'TensorFlow는 머신러닝을 위한 엔드 투 엔드 오픈소스 플랫폼입니다.' (TensorFlow is an end-to-end open-source platform for machine learning). Below this, a paragraph states: 'TensorFlow를 사용하면 초보자와 전문가 모두 머신러닝 모델을 쉽게 만들 수 있습니다. 시작하려면 아래의 섹션을 참조하세요.' (Using TensorFlow, both beginners and experts can easily create machine learning models. Refer to the sections below to get started). There are two identical buttons labeled '가이드 보기' (View Guide). Below each button is a short description: the first says '가이드에서는 완벽한 엔드 투 엔드 예제와 함께 TensorFlow를 사용하는 방법을 보여줍니다.' (The guide shows how to use TensorFlow with perfect end-to-end examples), and the second says '가이드는 TensorFlow의 개념과 구성요소에 관해 설명합니다.' (The guide explains the concepts and components of TensorFlow). On the right side of the page, there is a stylized illustration of a computer monitor with orange and grey lines connecting to it, symbolizing a neural network or data flow.

TensorFlow

설치 학습 ▼ API ▼ 리소스 ▼ 커뮤니티 TensorFlow를 사용해야 하는 이유 ▼

TensorFlow Core

개요 튜토리얼 가이드 TF 1

## TensorFlow는 머신러닝을 위한 엔드 투 엔드 오픈소스 플랫폼입니다.

TensorFlow를 사용하면 초보자와 전문가 모두 머신러닝 모델을 쉽게 만들 수 있습니다. 시작하려면 아래의 섹션을 참조하세요.

**가이드 보기**

가이드에서는 완벽한 엔드 투 엔드 예제와 함께 TensorFlow를 사용하는 방법을 보여줍니다.

**가이드 보기**

가이드는 TensorFlow의 개념과 구성요소에 관해 설명합니다.



# 초보자용 텐서플로우 코드

## 초보자용

사용자에게 친숙한 Sequential API로 시작하는 것이 가장 좋습니다. 구성요소를 연결하여 모델을 만들 수 있습니다. 아래의 'Hello World' 예제를 실행한 다음 [가이드](#)를 방문하여 자세한 내용을 알아보세요.

ML에 관해 배워보려면 [교육 페이지](#)를 확인하세요. 엄선된 커리큘럼으로 기본적인 ML 분야의 역량을 키워보세요.

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

지금 코드 실행

Google의 대화형 메모장에서 사용해보기

## 전문가용

Subclassing API는 고급 연구를 위한 define-by-run 인터페이스를 제공합니다. 모델에 대한 클래스를 만든 다음 명령형으로 순방향 패스를 작성합니다. 맞춤형 레이어, 활성화 및 학습 루프를 쉽게 만들 수 있습니다. 아래의 'Hello World' 예제를 실행한 다음 [가이드](#)를 방문하여 자세한 내용을 알아보세요.

```
class MyModel(tf.keras.Model):
    def __init__(self):
        super(MyModel, self).__init__()
        self.conv1 = Conv2D(32, 3, activation='relu')
        self.flatten = Flatten()
        self.d1 = Dense(128, activation='relu')
        self.d2 = Dense(10, activation='softmax')

    def call(self, x):
        x = self.conv1(x)
        x = self.flatten(x)
        x = self.d1(x)
        return self.d2(x)

model = MyModel()

with tf.GradientTape() as tape:
    logits = model(images)
    loss_value = loss(logits, labels)
grads = tape.gradient(loss_value, model.trainable_variables)
optimizer.apply_gradients(zip(grads, model.trainable_variables))
```

지금 코드 실행

Google의 대화형 메모장에서 사용해보기

# 코랩을 이용한 첫 실행

TensorFlow Core | 초보자 및 전 x beginner.ipynb - Colaboratory x +

colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb?hl=ko

beginner.ipynb

파일 수정 보기 삽입 런타임 도구 도움말

목차

<> Copyright 2019 The TensorFlow Authors.  
TensorFlow 2 quickstart for beginners

섹션

+ 코드 + 텍스트 드라이브로 복사

Copyright 2019 The TensorFlow Authors.

숨겨진 셀 1개

TensorFlow 2 quickstart for beginners

View on TensorFlow.org Run in Google Colab View source on GitHub Download notebook

+ 코드 + 텍스트

This short introduction uses [Keras](#) to:

1. Build a neural network that classifies images.
2. Train this neural network.
3. And, finally, evaluate the accuracy of the model.

This is a [Google Colaboratory](#) notebook file. Python programs are run directly in the browser—a great way to follow this tutorial, run the notebook in Google Colab by clicking the button at the top of this page.

1. In Colab, connect to a Python runtime: At the top-right of the menu bar, select *CONNECT*.
2. Run all the notebook code cells: Select *Runtime* > *Run all*.

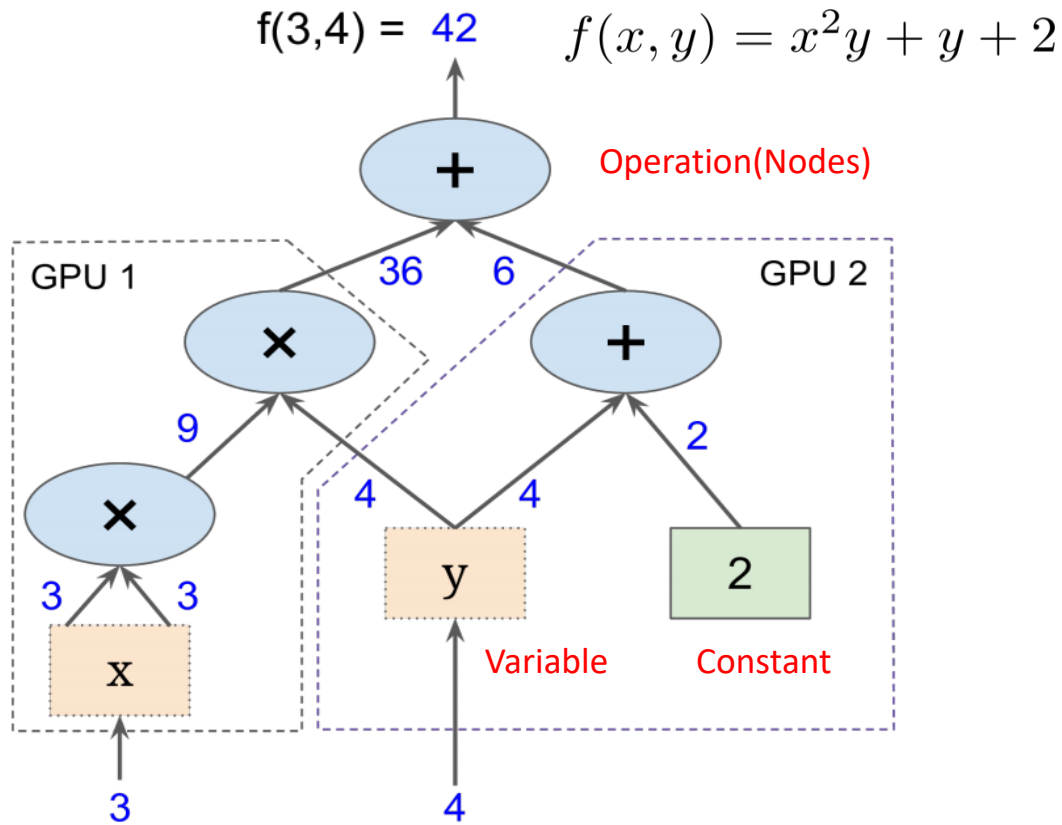
# Thank You!

[www.ust.ac.kr](http://www.ust.ac.kr)

## 04. 텐서플로우 구조 (v1.0)

인터넷 기반의 Colab에서 프로그래밍 하기

# Programming Model



# Your first TensorFlow Program

```
import tensorflow as tf
```

```
x = tf.constant(3)
```

```
y = tf.constant(4)
```

```
b = tf.constant(2)
```

```
c = tf.square(x)
```

```
d = tf.multiply(c,y)
```

```
e = tf.add(y,b)
```

```
f = tf.add(d,e)
```

```
with tf.Session() as sess:
```

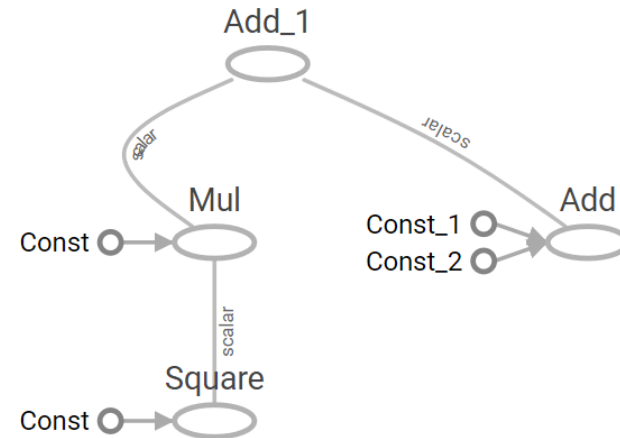
```
# add this line to use TensorBoard.
```

```
    writer = tf.summary.FileWriter('./log2', sess.graph)
```

```
    print(sess.run(f))
```

```
writer.close()
```

```
# 42
```



# Run Tensorboard

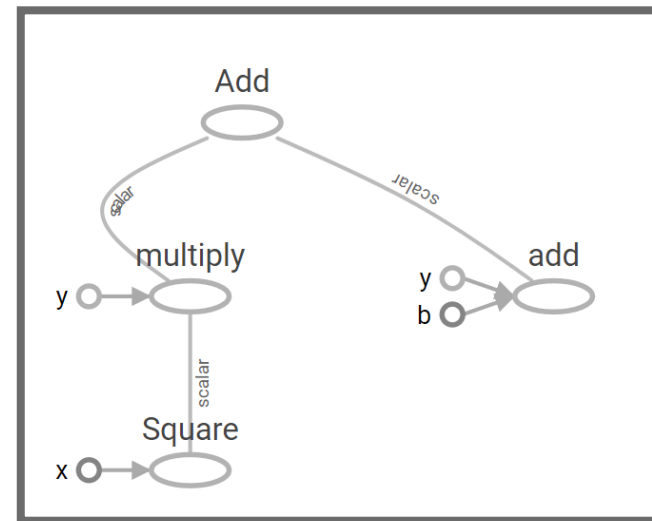
Go to terminal, run:

```
# $ python lab01.py
```

```
$ tensorboard --logdir="./log2" --port 6006
```

Then open your browser and go to: <http://localhost:6006/>

```
>tensorboard --logdir=./log2 --port=6006  
Starting TensorBoard b'41' on port 6006  
<You can navigate to http://192.168.56.1:6006>
```



# Tensorboard

The screenshot displays the TensorBoard web interface in a browser. The address bar shows the URL `192.168.56.1:6006`, which is circled in red. The navigation bar at the top includes links for SCALARS, IMAGES, AUDIO, **GRAPHS** (circled in blue), DISTRIBUTIONS, HISTOGRAMS, and EMBEDDINGS. On the left sidebar, there are controls for 'Fit to screen', 'Download PNG', 'Run (1)', 'Session runs (0)', an 'Upload' button with 'Choose File', a 'Trace inputs' toggle, and 'Color' options for 'Structure' (selected) and 'Device'. The main area shows a computational graph with nodes: 'Add', 'multiply', 'add', and 'Square'. Edges are labeled 'scalar'. Inputs 'x', 'y', and 'b' are shown as small circles with arrows pointing to their respective nodes.