

# [Live] 실전! TensorFlow로 배우는 딥러닝

Day1

- Deeplearning
- Tensorflow

# 목차

## 1. Deep learning 개요

- ✓ 딥러닝이란
- ✓ 딥러닝의 역사
- ✓ 딥러닝의 활용

## 2. DNN 이론

- ✓ Feed-forward Network Functions
- ✓ Network Training
- ✓ Error Backpropagation

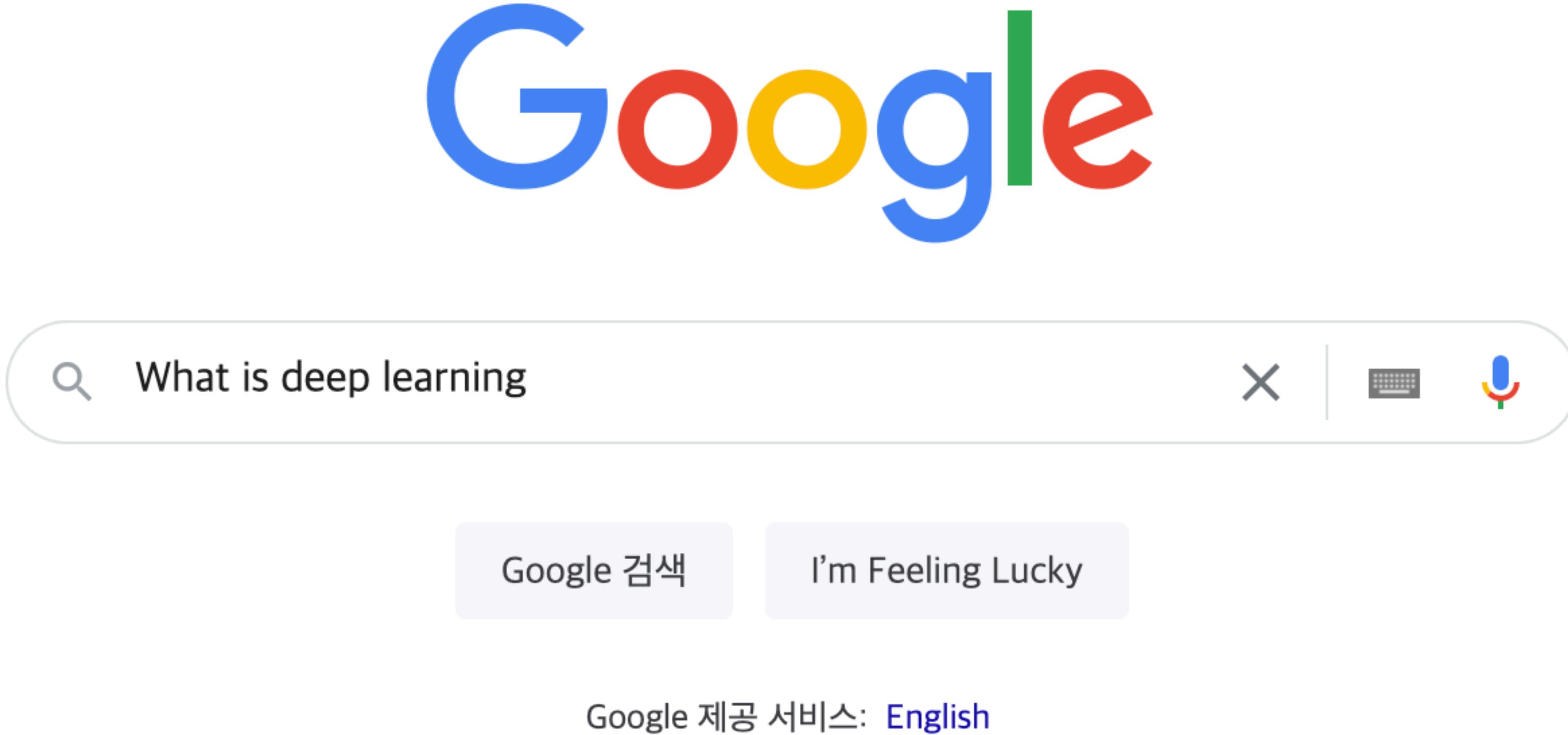
## 3. Numpy, Matplotlib 실습

## 4. Tensorflow 소개

## 5. Tensorflow로 Regression 구현

# 1. Deep Learning 개요

## 딥러닝이란?

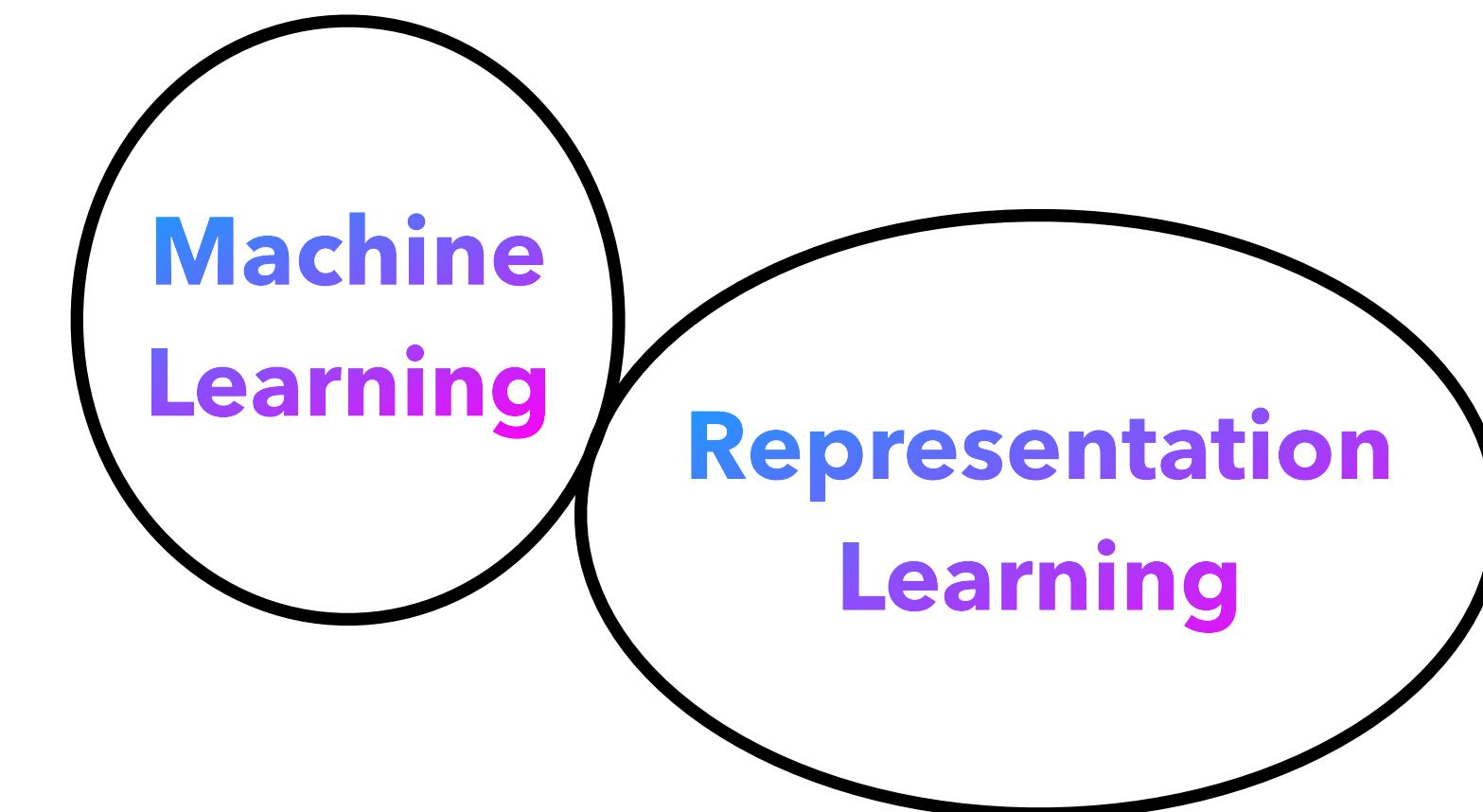


# 1. Deep Learning 개요

## 딥러닝이란?

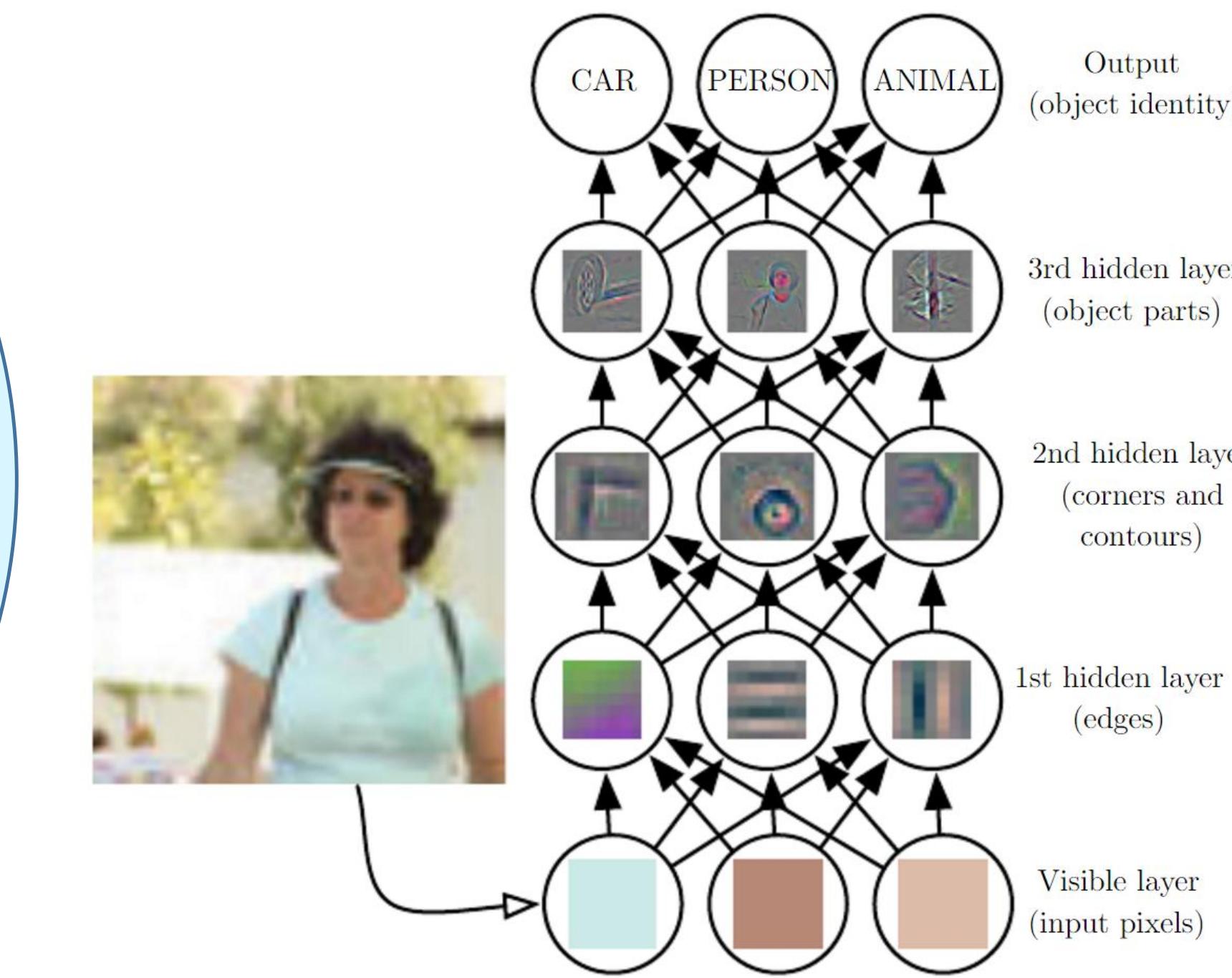
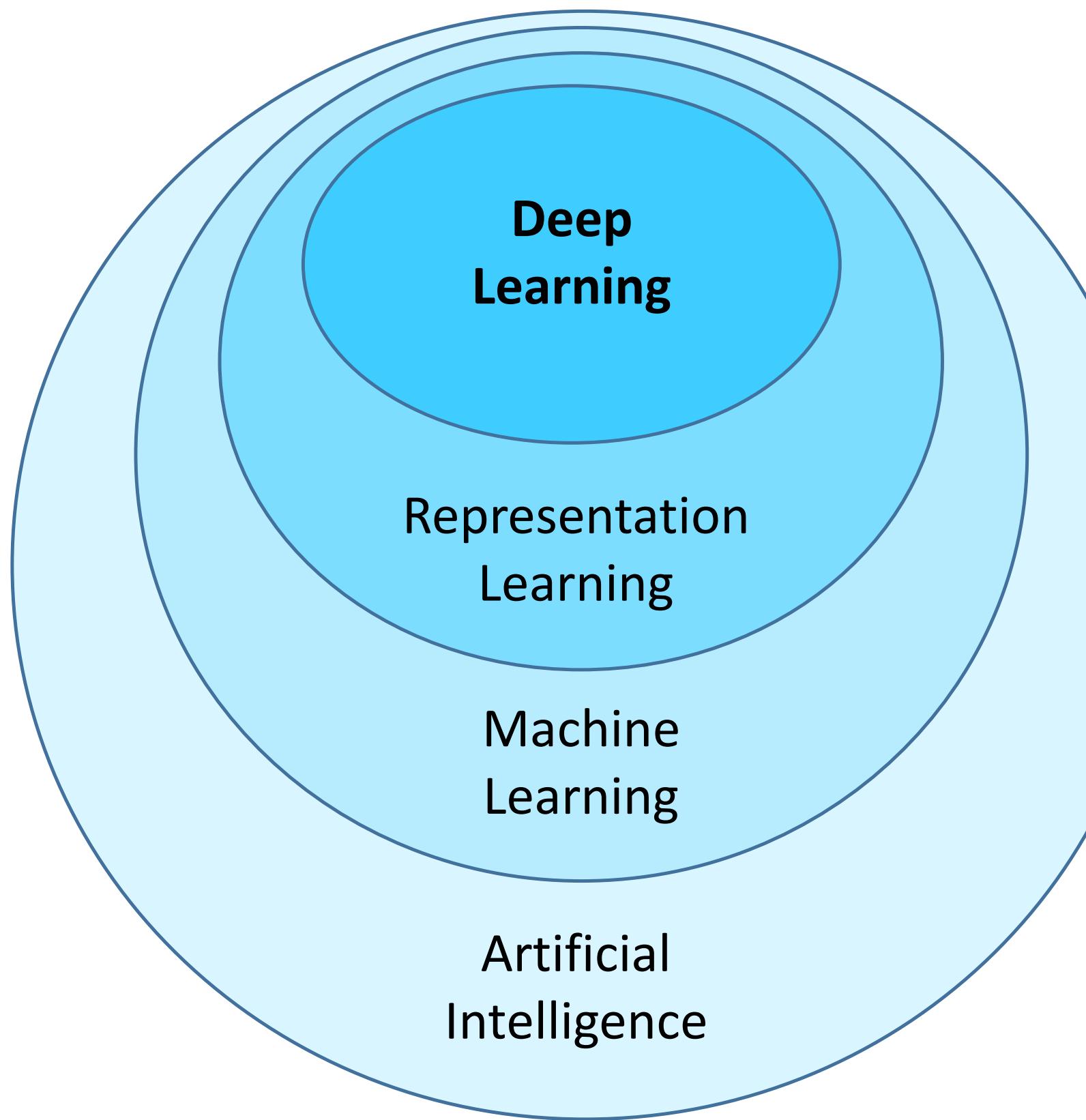
딥러닝

# 1. Deep Learning 개요 딥러닝이란?



# 1. Deep Learning 개요 딥러닝이란?

## Deep Learning is Representation Learning (aka Feature Learning)

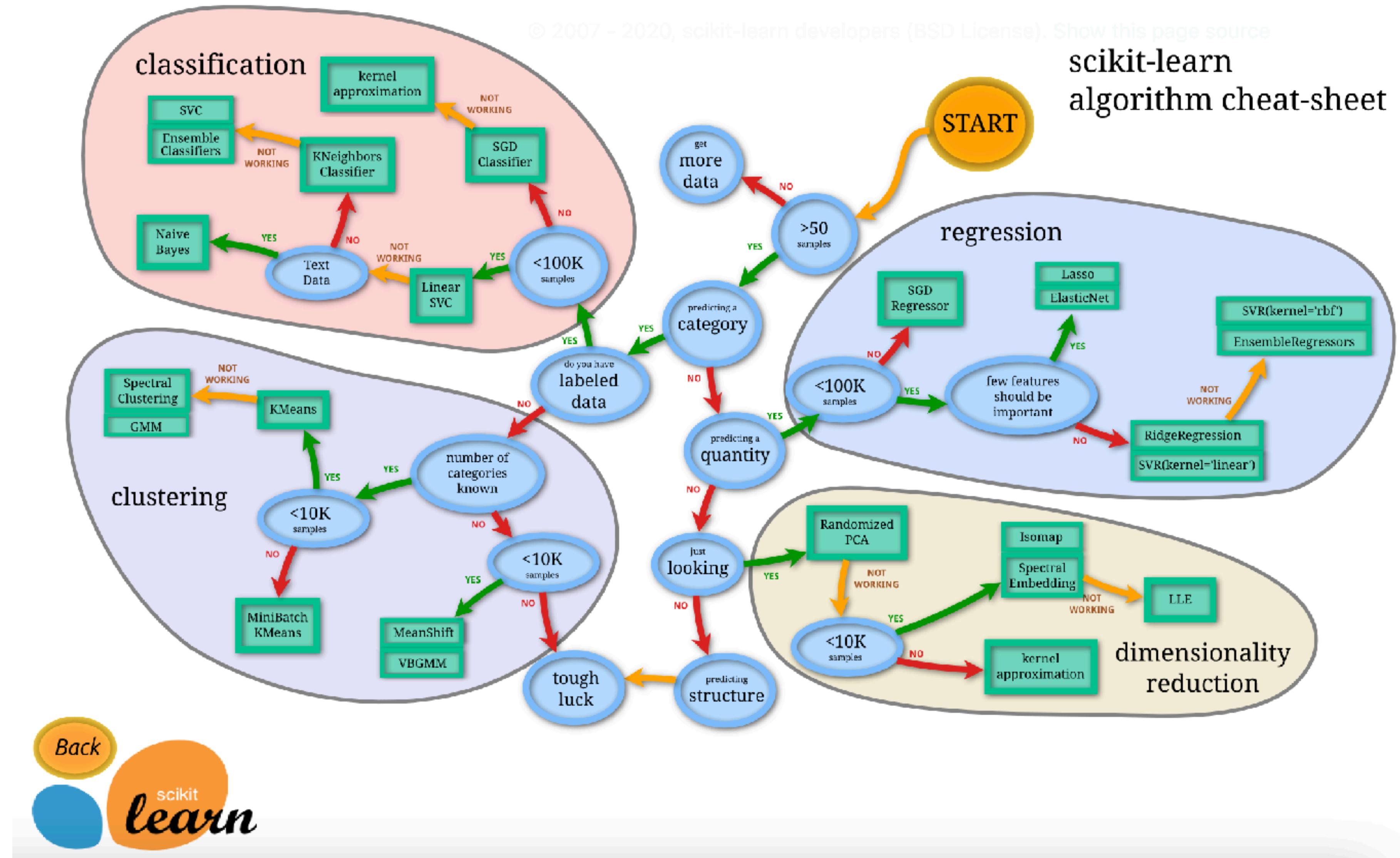


1. Deep Learning 개요  
딥러닝이란?

# Machine Learning

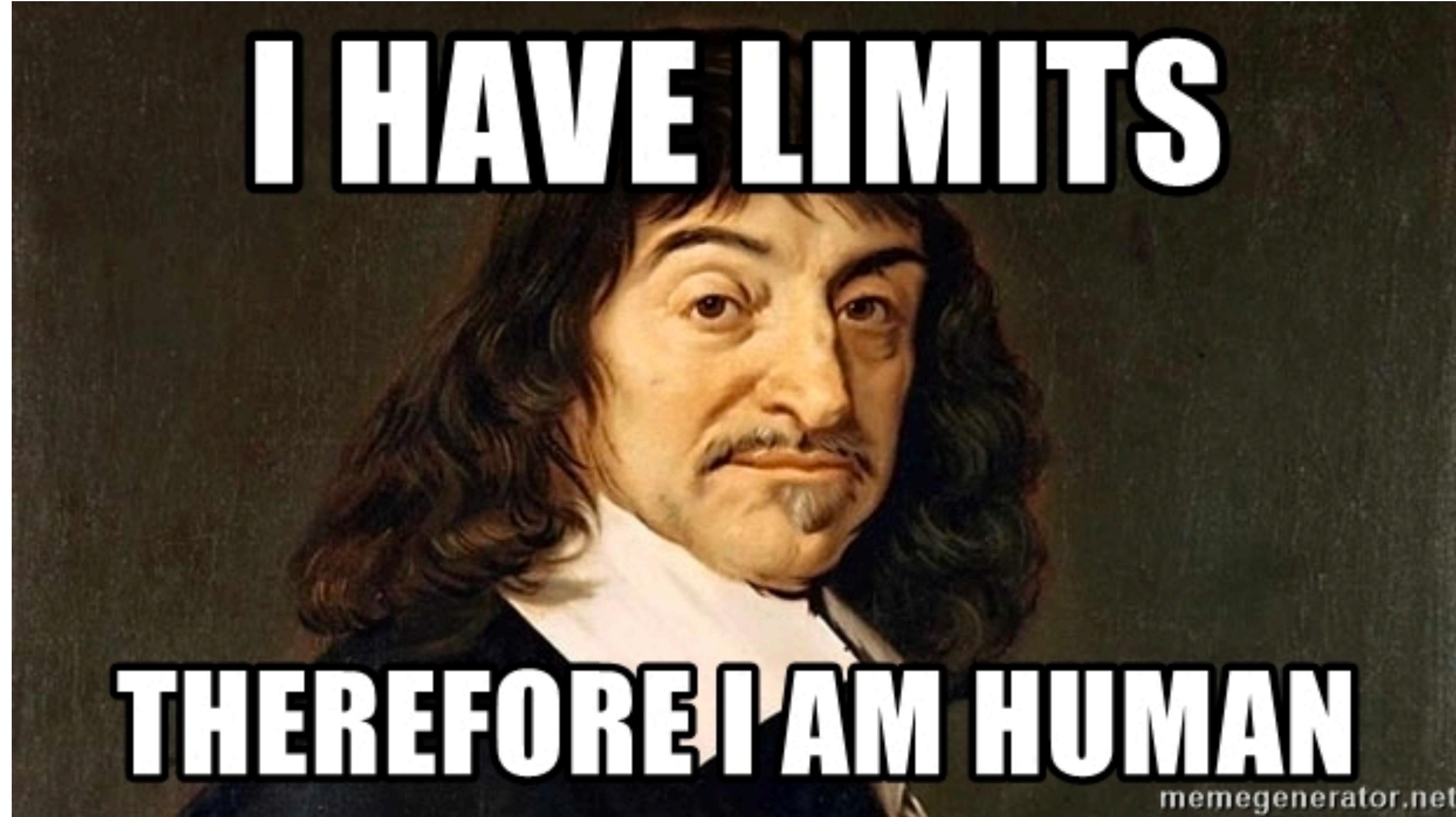
# 1. Deep Learning 개요

## 딥러닝이란?



# 1. Deep Learning 개요

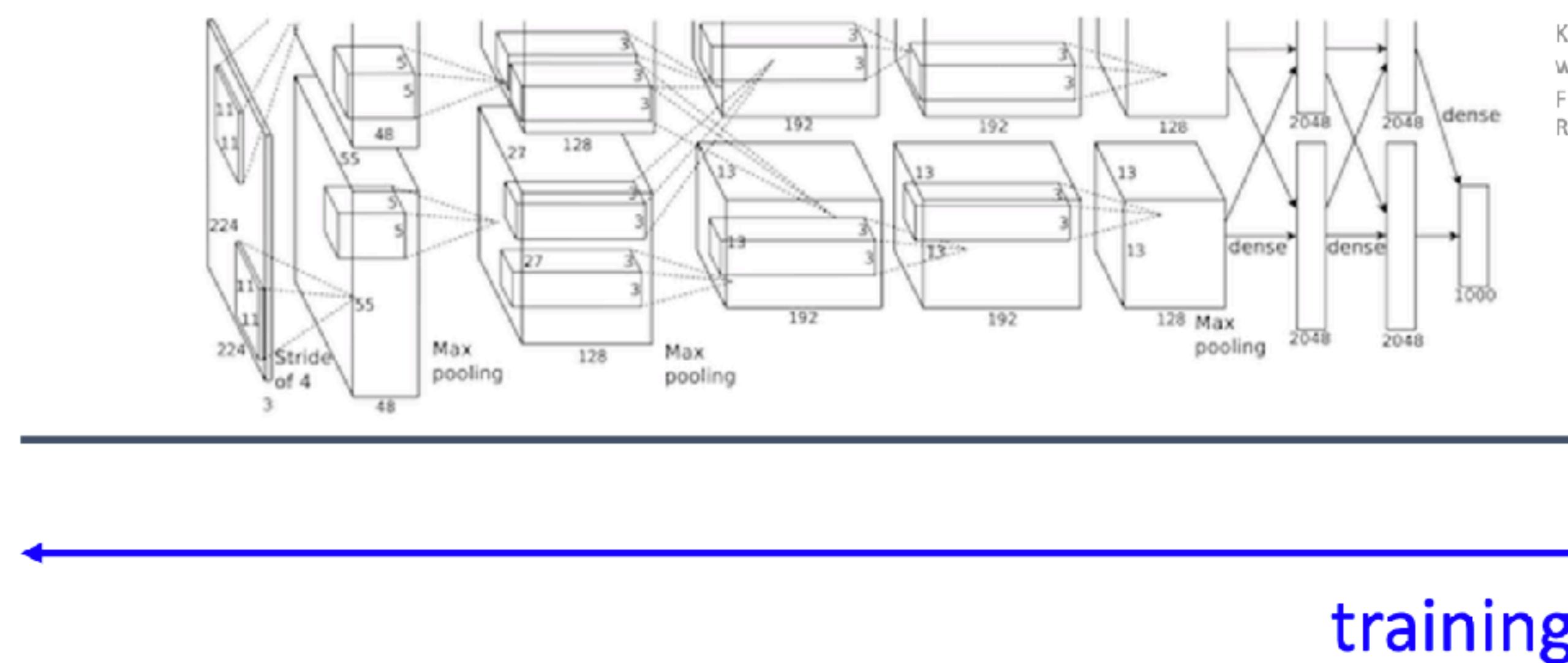
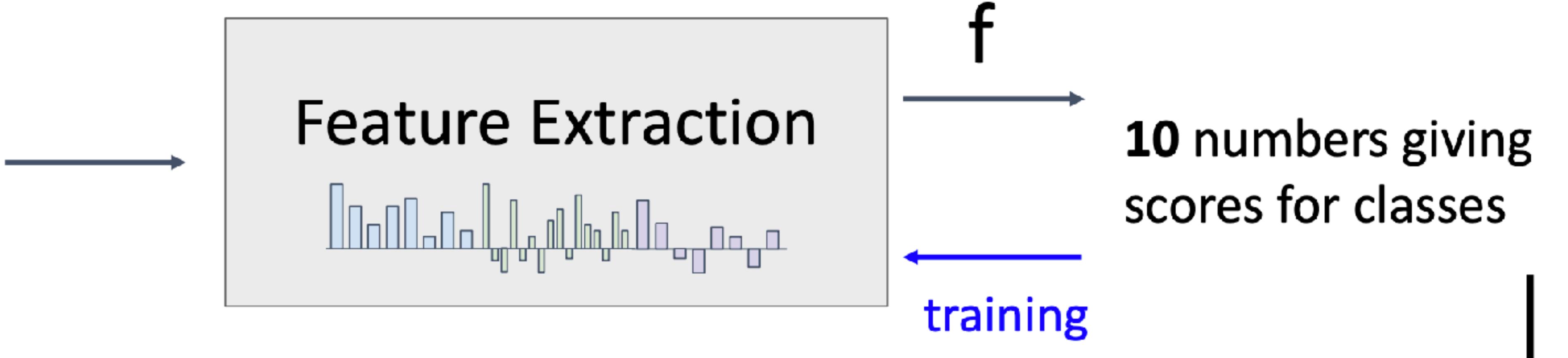
## 딥러닝이란?



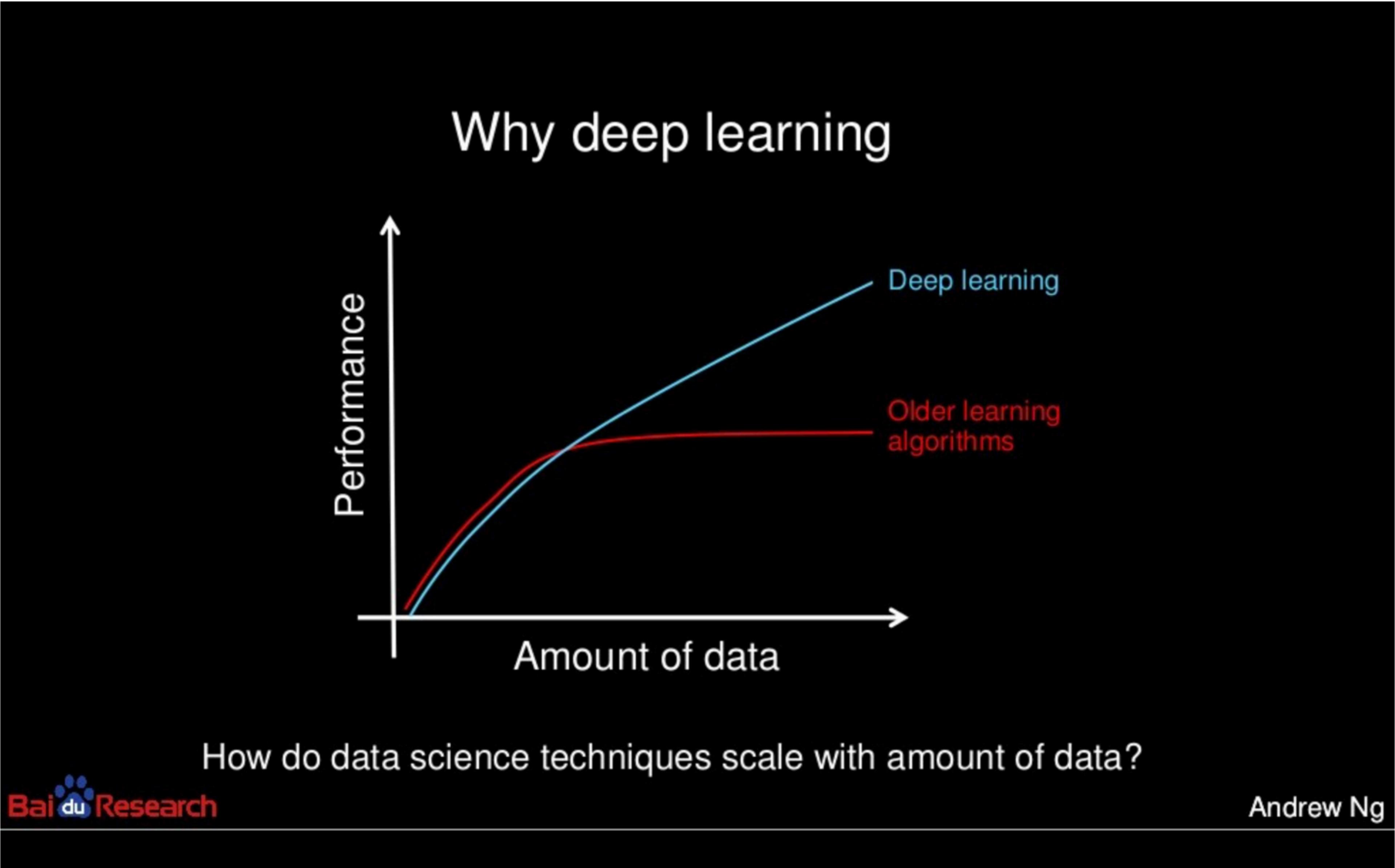
## 1. Deep Learning 개요 딥러닝이란?

# Representation Learning

# 1. Deep Learning 개요 딥러닝이란?



# 1. Deep Learning 개요 딥러닝이란?



# 1. Deep Learning 개요

## 딥러닝이란?



# 1. Deep Learning 개요

## 딥러닝이란?



### Python For Data Science Cheat Sheet

#### Keras

[www.techringe.com](http://www.techringe.com)

#### Keras

Keras is a powerful and easy-to-use deep learning library for Theano and TensorFlow that provides a high-level neural networks API to develop and evaluate deep learning models.

#### A Basic Example

```
>>> import numpy as np
>>> from keras.models import Sequential
>>> from keras.layers import Dense
>>> from sklearn.datasets import make_classification
>>> data = np.random.randint(1000,100)
>>> labels = np.random.randint(2, size=[1000,1])
>>> model = Sequential()
>>> model.add(Dense(32,
>>>                 activation='relu',
>>>                 input_dim=100))
>>> model.add(Dense(1, activation='sigmoid'))
>>> model.compile(optimizer='sgd',
>>>                 loss='binary_crossentropy',
>>>                 metrics=['accuracy'])
>>> model.fit(data,labels,epochs=10,batch_size=32)
>>> predictions = model.predict(data)
```

#### Data

Also see NumPy, Pandas & Scikit-Learn  
Your data needs to be stored as NumPy arrays or as a list of NumPy arrays. Ideally, you split the data in training and test sets, for which you can also resort to the `train_test_split` module of `sklearn.cross_validation`.

#### Keras Data Sets

```
>>> from keras.datasets import boston_housing,
>>>                      mnist,
>>>                      cifar10,
>>>                      imdb
>>> (x_train,y_train),(x_test,y_test) = mnist.load_data()
>>> (x_train,y_train2),(x_test,y_test2) = boston_housing.load_data()
>>> (x_train,y_train3),(x_test,y_test3) = cifar10.load_data()
>>> (x_train,y_train4),(x_test4,y_test4) = imdb.load_data(num_words=20000)
>>> num_classes = 10
```

#### Other

```
>>> from urllib.request import urlopen
>>> data = np.loadtxt(urlopen("http://archive.ics.uci.edu/
>>> ai/machine-learning-databases/pima-indians-diabetes/
>>> pima-indians-diabetes.data"), delimiter=",")
>>> X = data[:,0:8]
>>> y = data[:,8]
```

#### Preprocessing

#### Model Architecture

##### Sequential Model

```
>>> from keras.models import Sequential
>>> model = Sequential()
>>> model2 = Sequential()
>>> model3 = Sequential()
```

##### Multilayer Perceptron (MLP)

**Binary Classification**

```
>>> from keras.layers import Dense
>>> model.add(Dense(12,
>>>                 input_dim=8,
>>>                 kernel_initializer='uniform',
>>>                 activation='relu'))
```

**Multi-Class Classification**

```
>>> from keras.layers import Dense
>>> model.add(Dense(512,activation='relu',input_shape=(784,)))
>>> model.add(Dropout(0.2))
>>> model.add(Dense(512,activation='relu'))
>>> model.add(Dropout(0.2))
>>> model.add(Dense(10,activation='softmax'))
```

**Regression**

```
>>> model.add(Dense(64,activation='relu',input_dim=train_data.shape[1]))
>>> model.compile(optimizer='sgd',
>>>                 loss='mse',
>>>                 metrics=['mae'])
```

##### Convolutional Neural Network (CNN)

```
>>> from keras.layers import Activation,Conv2D,MaxPooling2D,Flatten
>>> model2.add(Conv2D(32,(3,3),padding='same',input_shape=x_train.shape[1:]))
>>> model2.add(Activation('relu'))
>>> model2.add(Conv2D(32,(3,3)))
>>> model2.add(Activation('relu'))
>>> model2.add(MaxPooling2D(pool_size=(2,2)))
>>> model2.add(Dropout(0.25))
>>> model2.add(Conv2D(64,(3,3),padding='same'))
>>> model2.add(Activation('relu'))
>>> model2.add(Conv2D(64,(3,3)))
>>> model2.add(Activation('relu'))
>>> model2.add(MaxPooling2D(pool_size=(2,2)))
>>> model2.add(Dropout(0.25))
>>> model2.add(Flatten())
>>> model2.add(Dense(512))
>>> model2.add(Activation('relu'))
>>> model2.add(Dropout(0.5))
>>> model2.add(Dense(num_classes))
>>> model2.add(Activation('softmax'))
```

##### Recurrent Neural Network (RNN)

```
>>> from keras.layers import Embedding,LSTM
>>> model3.add(Embedding(20000,128))
>>> model3.add(LSTM(128,dropout=0.2,recurrent_dropout=0.2))
>>> model3.add(Dense(1,activation='sigmoid'))
```

##### Inspect Model

```
>>> model.output_shape
>>> model.summary()
>>> model.get_config()
>>> model.get_weights()
```

Model output shape  
Model summary representation  
Model configuration  
List all weight tensors in the model

##### Compile Model

**MLP: Binary Classification**

```
>>> model.compile(optimizer='adam',
>>>                 loss='binary_crossentropy',
>>>                 metrics=['accuracy'])
```

**MLP: Multi-Class Classification**

```
>>> model.compile(optimizer='sgd',
>>>                 loss='categorical_crossentropy',
>>>                 metrics=['accuracy'])
```

**MLP: Regression**

```
>>> model.compile(optimizer='sgd',
>>>                 loss='mse',
>>>                 metrics=['mae'])
```

##### Recurrent Neural Network

```
>>> model3.compile(loss='binary_crossentropy',
>>>                  optimizer='adam',
>>>                  metrics=['accuracy'])
```

##### Model Training

```
>>> model3.fit(x_train,
>>>             y_train,
>>>             batch_size=32,
>>>             epochs=15,
>>>             verbose=1,
>>>             validation_data=(x_test,y_test))
```

##### Evaluate Your Model's Performance

```
>>> score = model3.evaluate(x_test,
>>>                           y_test,
>>>                           batch_size=32)
```

##### Prediction

```
>>> model3.predict(x_test, batch_size=32)
>>> model3.predict_classes(x_test,batch_size=32)
```

##### Save / Reload Models

```
>>> from keras.models import load_model
>>> model3.save('model.h5')
>>> my_model = load_model('my_model.h5')
```

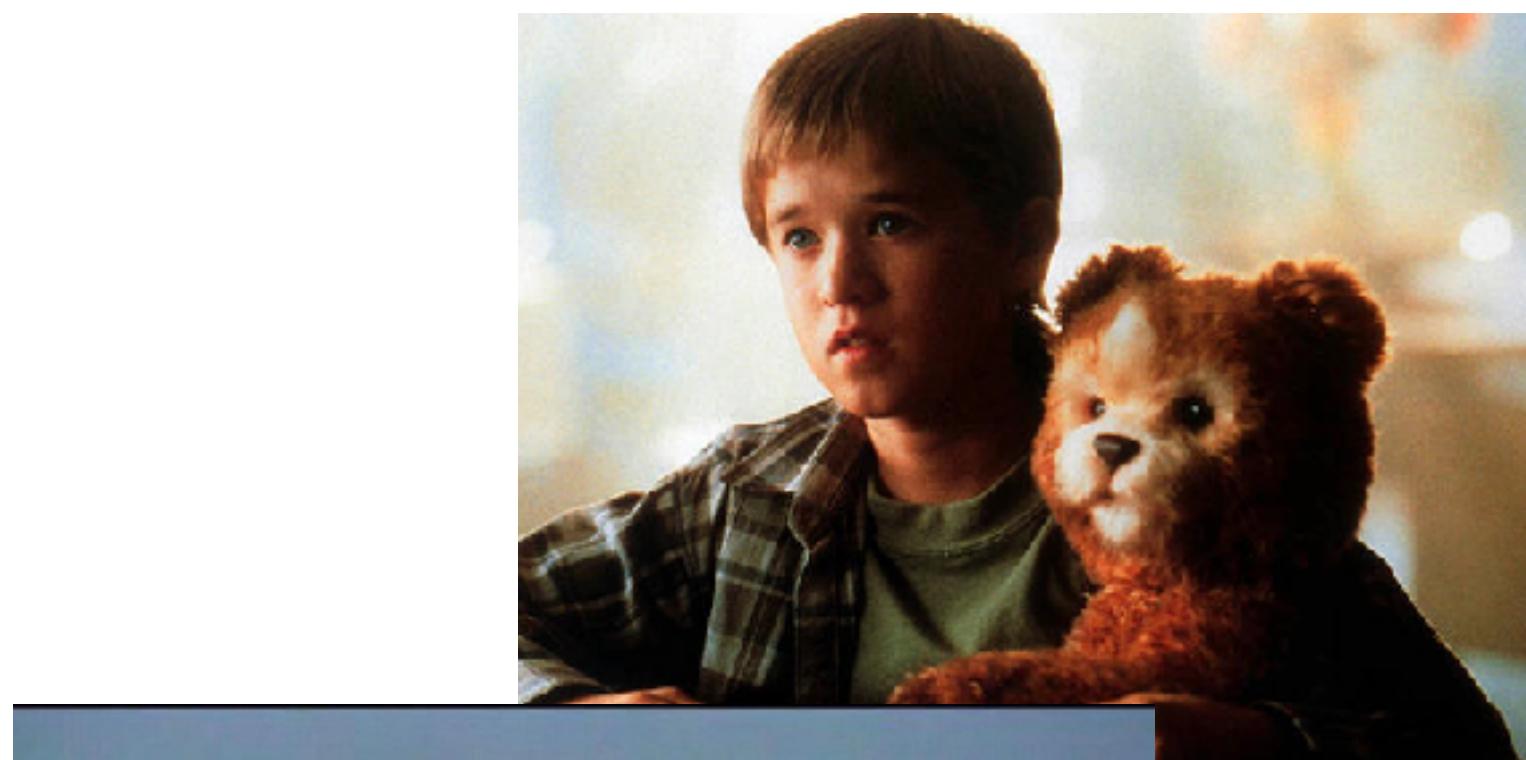
##### Model Fine-tuning

**Optimization Parameters**

```
>>> from keras.optimizers import RMSprop
>>> opt = RMSprop(lr=0.0001, decay=1e-6)
>>> model3.compile(loss='categorical_crossentropy',
>>>                  optimizer=opt,
>>>                  metrics=['accuracy'])
```

# 1. Deep Learning 개요

## 딥러닝이란?



Python

What the....

```
Theano and TensorFlow networks API to develop and train neural networks. It provides a high-level interface for defining, optimizing and evaluating mathematical expressions involving tensors. It is built on top of NumPy and can run on CPU or GPU. It is designed to be used with other Python libraries such as Scikit-learn, Pandas, and Matplotlib. It is also compatible with TensorFlow and Keras.
```

A Basic Example

```
>>> import numpy as np
>>> from keras.models import Sequential
>>> from keras.layers import Dense
>>> data = np.random.randint(1000,1000)
>>> data = np.random.randint(2, size=[1000,1])
>>> model = Sequential()
>>> model.add(Dense(32, activation='relu'))
>>> model.add(Dense(1, kernel_initializer='uniform', activation='sigmoid'))
>>> model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Multi-Class Classification

```
>>> from keras.layers import Dropout
>>> model.add(Dense(512, activation='relu', input_shape=(784,)))
>>> model.add(Dropout(0.2))
>>> model.add(Dense(512, activation='relu'))
>>> model.add(Dropout(0.2))
>>> model.add(Dense(10, activation='softmax'))
```

Regression

```
>>> model.add(Dense(64, activation='relu', input_dim=train_data.shape[1]))
>>> model.add(Dense(1))
```

Convolutional Neural Network (CNN)

```
>>> from keras.layers import Activation, Conv2D, MaxPooling2D, Flatten
>>> model2.add(Conv2D(32, (3, 3), padding='same', input_shape=x_train.shape[1:]))
>>> model2.add(Activation('relu'))
>>> model2.add(Conv2D(32, (3, 3)))
>>> model2.add(Activation('relu'))
>>> model2.add(MaxPooling2D(pool_size=(2, 2)))
>>> model2.add(Flatten())
>>> model2.add(Dense(128))
>>> model2.add(Activation('relu'))
>>> model2.add(Dense(1))
```

Data Sets

```
>>> from keras.datasets import boston_housing, mnist, cifar10, imdb
>>> (x_train, y_train), (x_test, y_test) = mnist.load_data()
>>> (x_train, y_train), (x_test, y_test) = boston_housing.load_data()
>>> (x_train, y_train), (x_test, y_test) = cifar10.load_data()
>>> (x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=20000)
>>> num_classes = 10
```

Other

```
>>> from urllib.request import urlopen
>>> data = np.loadtxt(urlopen("http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data"), delimiter=",")
>>> X = data[:, :-1]
>>> y = data[:, -1]
```

Preprocessing

```
>>> from keras.layers import Embedding, LSTM
>>> model3.add(Embedding(20000, 128))
>>> model3.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
>>> model3.add(Dense(1, activation='sigmoid'))
```

Inspect Model

```
>>> model.output_shape
>>> model.summary()
>>> model.get_config()
>>> model.get_weights()
```

Compile Model

```
>>> MLP: Binary Classification
>>> model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
>>> MLP: Multi-Class Classification
>>> model.compile(optimizer='nagprop', loss='categorical_crossentropy', metrics=['accuracy'])
>>> MLP: Regression
>>> model.compile(optimizer='nagprop', loss='mse', metrics=['mse'])
```

Recurrent Neural Network

```
>>> model3.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Model Training

```
>>> model3.fit(x_train, y_train, batch_size=32, epochs=15, verbose=1, validation_data=(x_test, y_test))
```

Evaluate Your Model's Performance

```
>>> score = model3.evaluate(x_test, y_test, batch_size=32)
```

Prediction

```
>>> model3.predict(x_test, batch_size=32)
>>> model3.predict_classes(x_test, batch_size=32)
```

Save / Reload Models

```
>>> from keras.models import load_model
>>> model3.save('model.h5')
>>> my_model = load_model('my_model.h5')
```

Model Fine-tuning

Optimization Parameters

```
>>> from keras.optimizers import RMSprop
>>> opt = RMSprop(lr=0.0001, decay=1e-6)
>>> model3.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
```

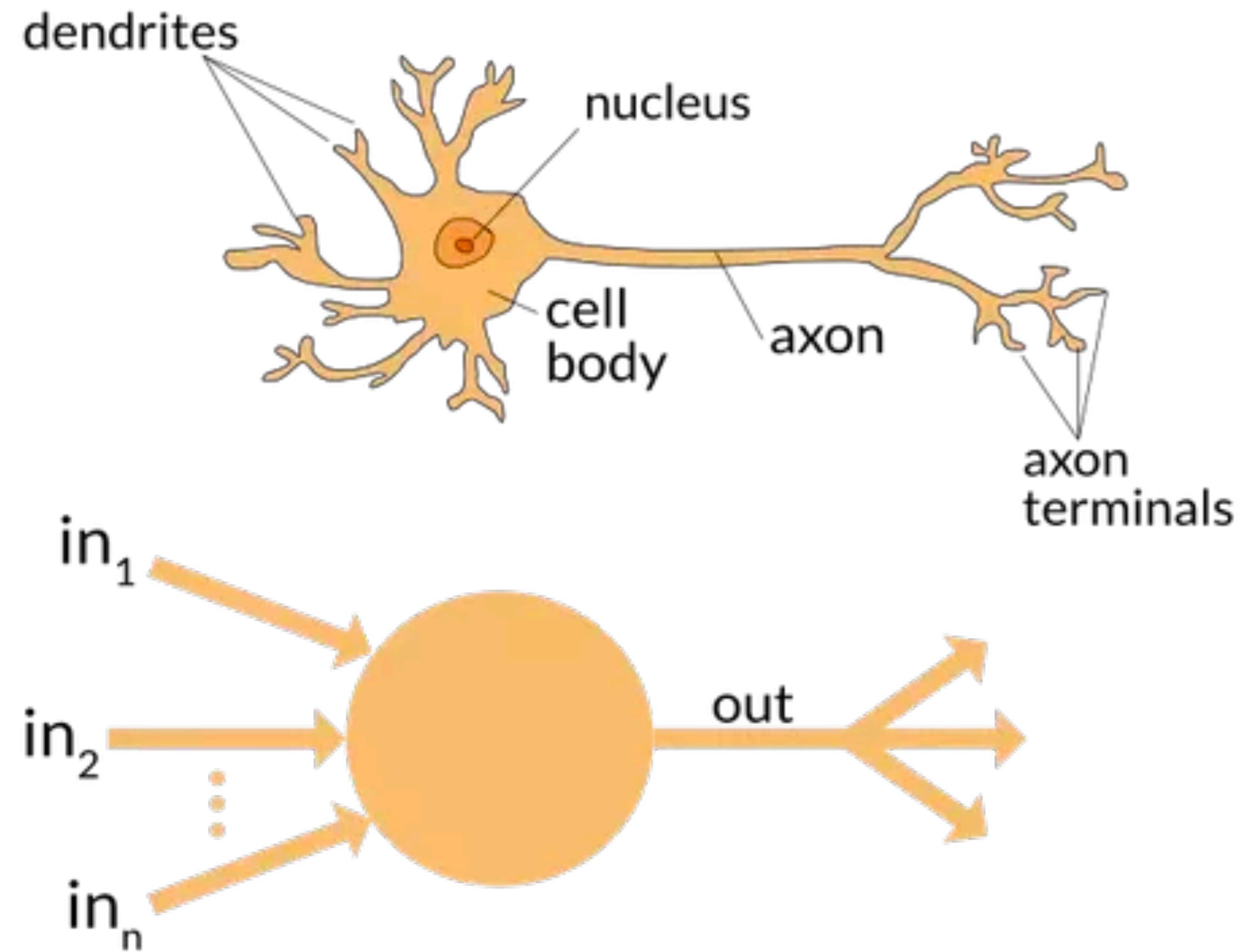
# Deep learning

Yann LeCun<sup>1,2</sup>, Yoshua Bengio<sup>3</sup> & Geoffrey Hinton<sup>4,5</sup>

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

# 1. Deep Learning 개요

## 딥러닝이란?



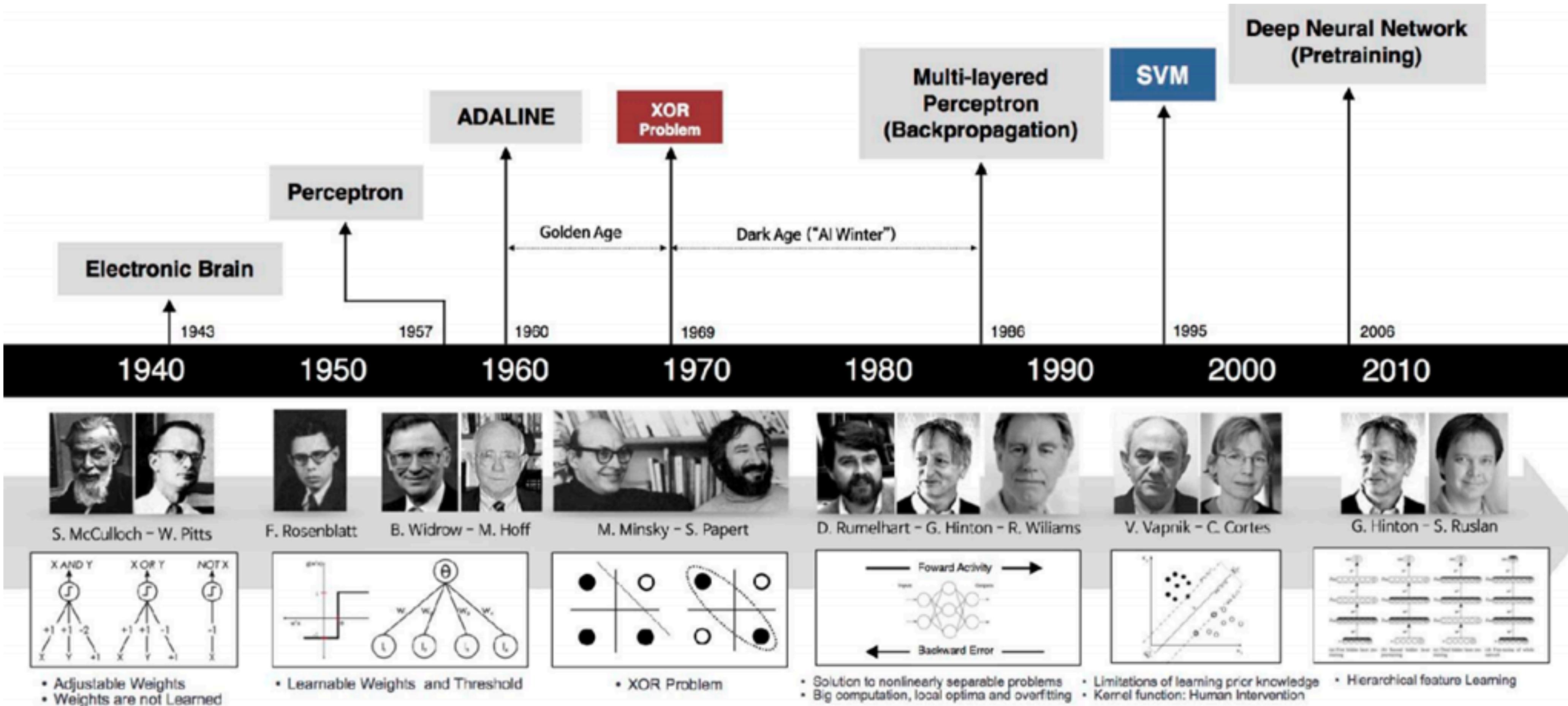
1. Deep Learning 개요  
딥러닝이란?

**Human brain : 100 trillion synapses**

**GPT-3 : 175 billion parameters**

# 1. Deep Learning 개요

## 딥러닝의 역사



아날로그 사이언스[시즌 7 인공지능] 6화. 마빈 민스키의 폭탄 발언

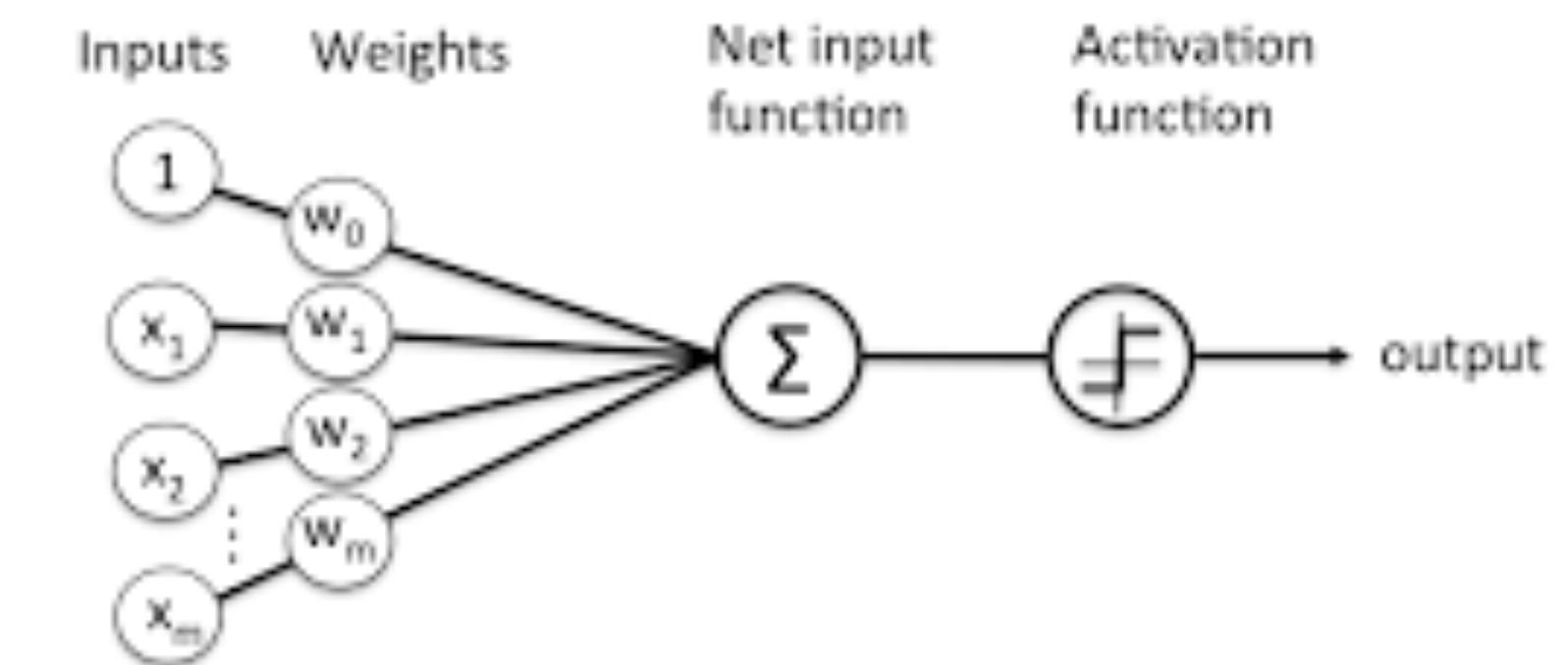
# 1. Deep Learning 개요

## 딥러닝의 역사

$$y(x) = f(W^T \phi(x))$$

The perceptron algorithm

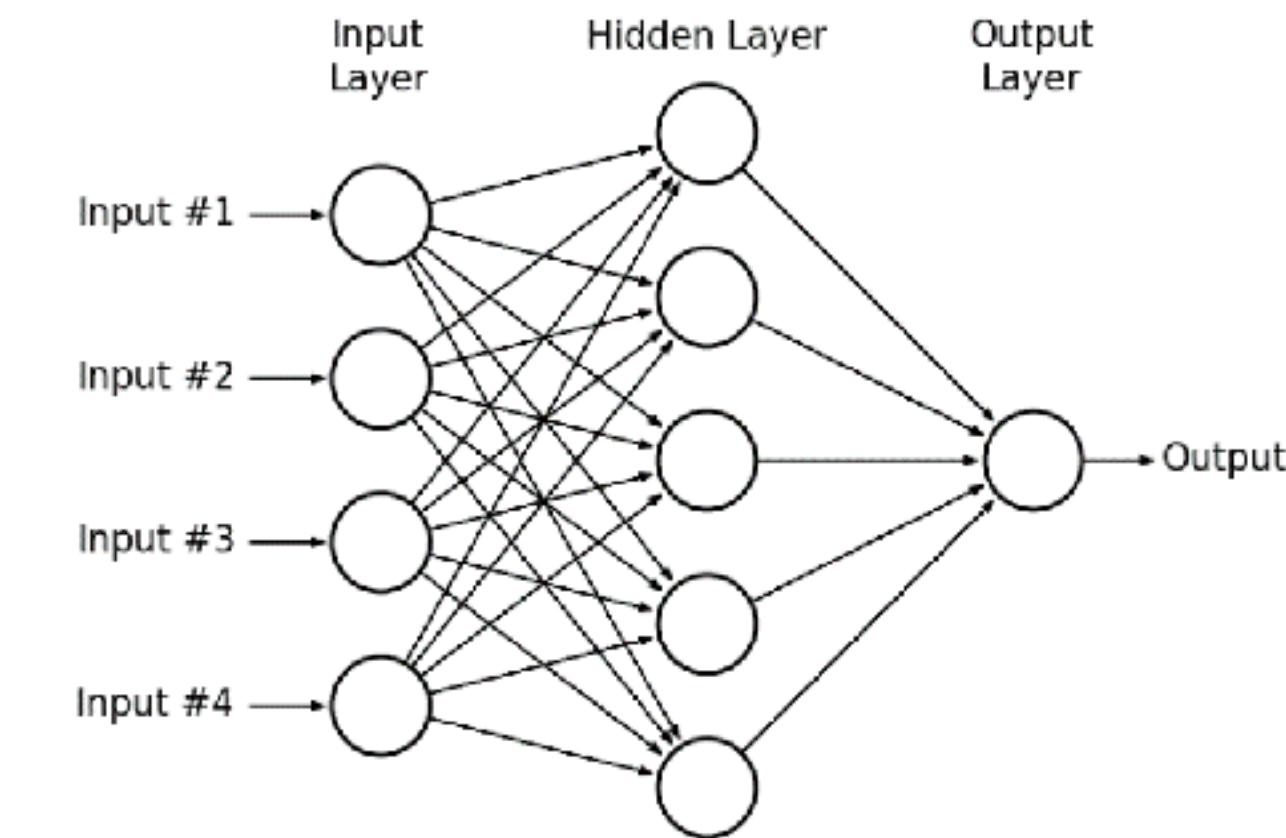
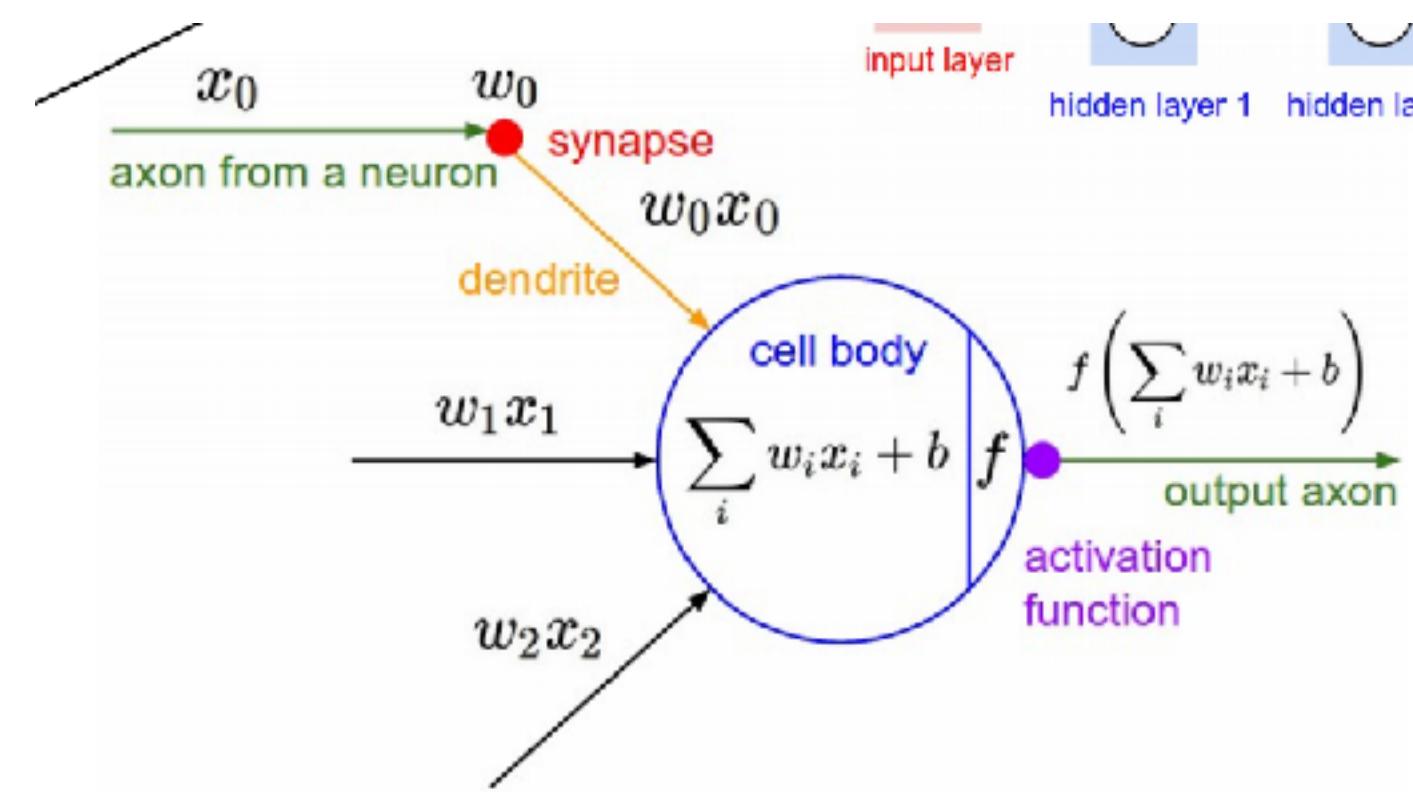
$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1 & a < 0. \end{cases}$$



Schematic of Rosenblatt's perceptron.

# Backpropagation

Multi-Layer Perceptron

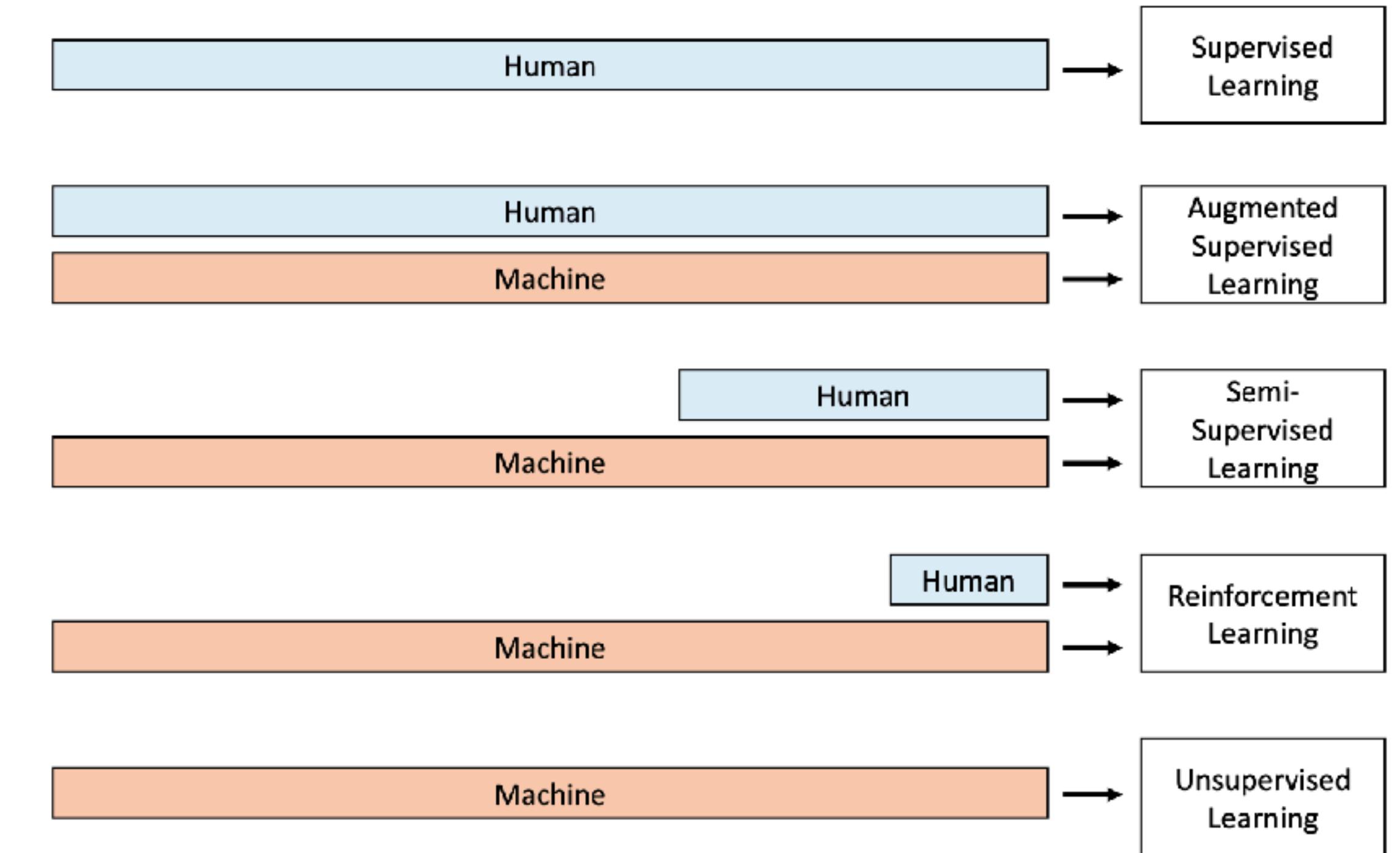
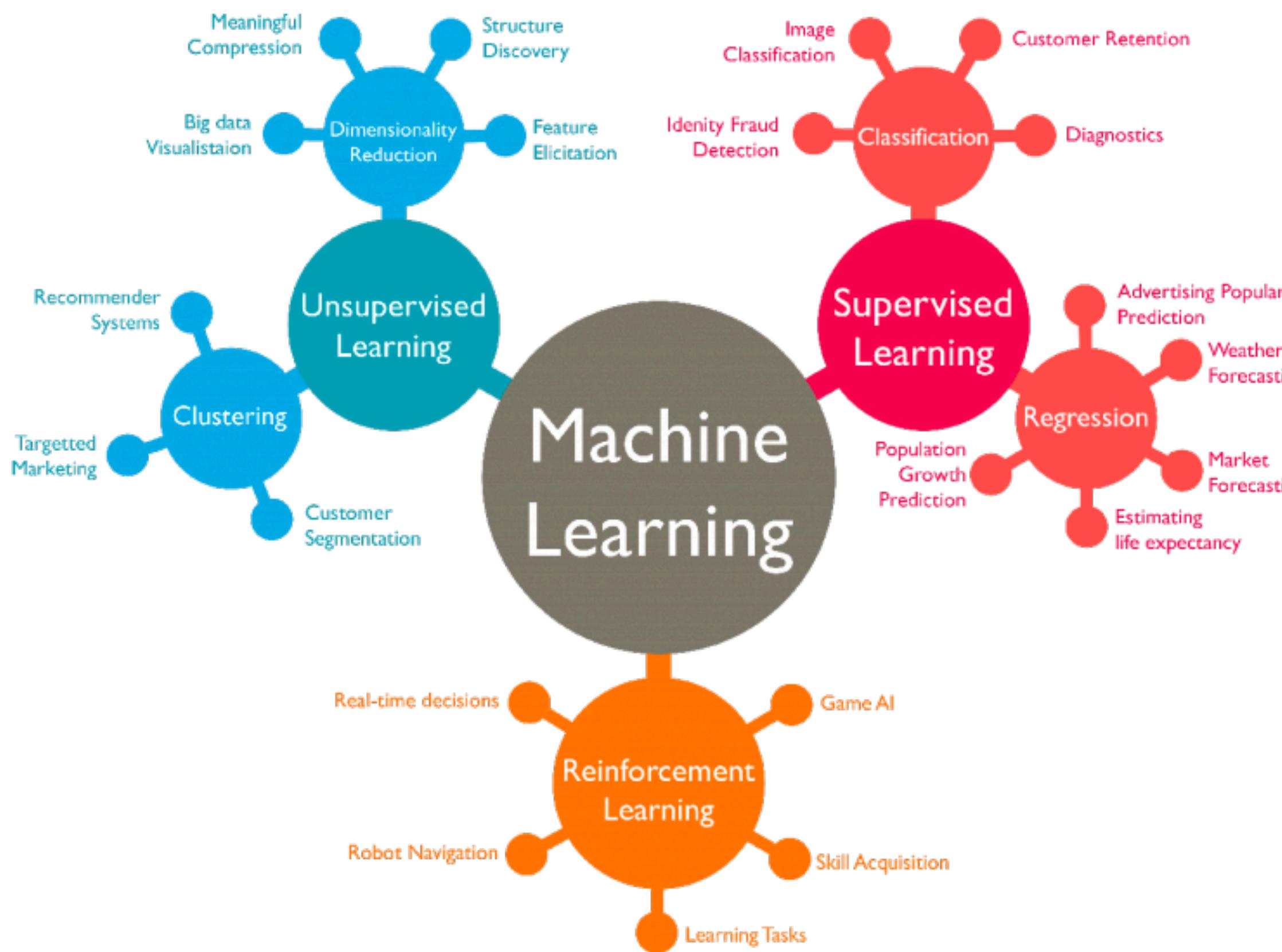


598\_FA2020\_lecture05

Schematic of Rosenblatt's perceptron

# 1. Deep Learning 개요

## 딥러닝의 활용



[Deep learning basic - Lex Fridman](#)

참고 : [Machine Learning Roadmap 2020](#)

[Great Mind Maps for Learning Machine Learning](#)

# 1. Deep Learning 개요

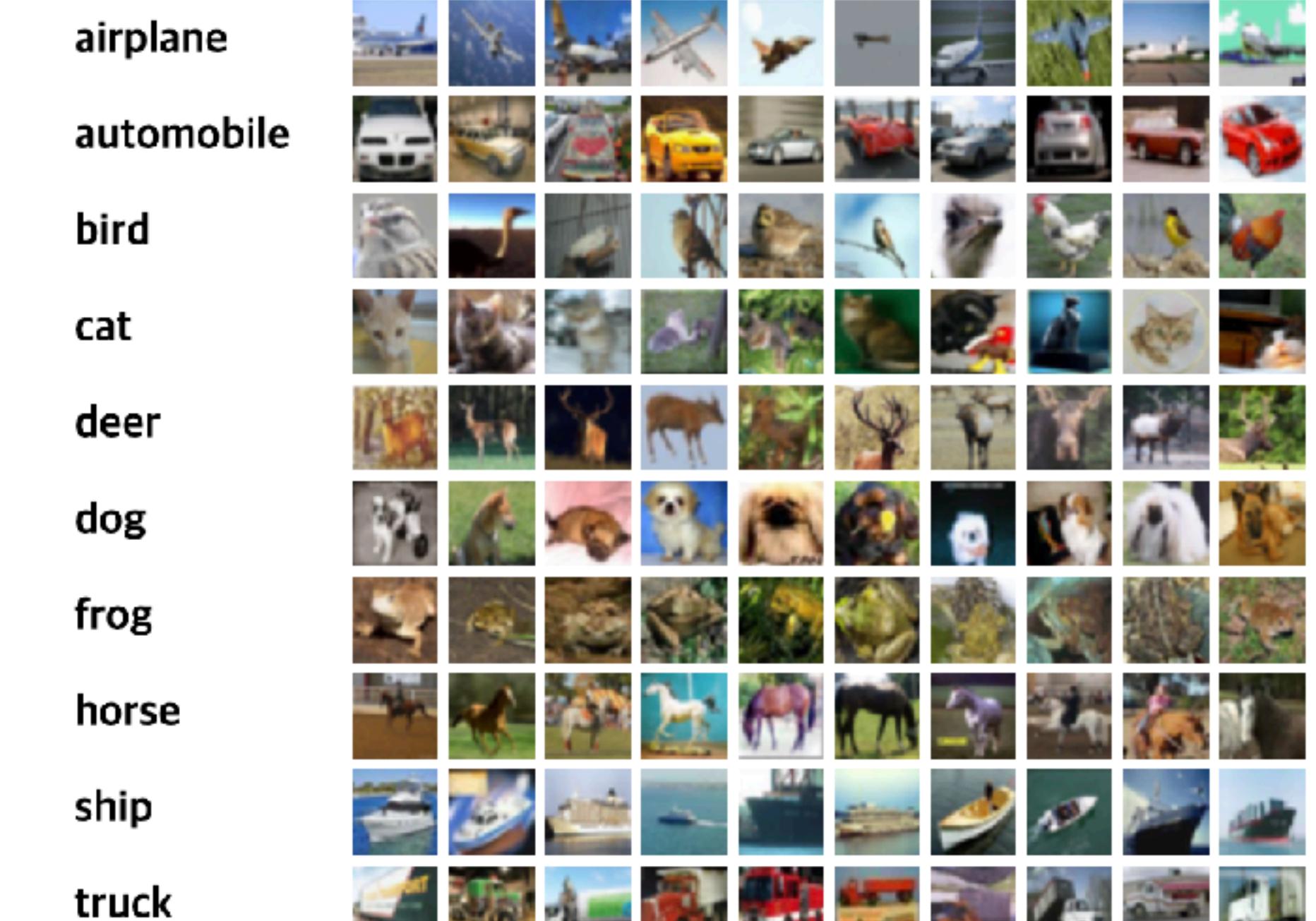
## 딥러닝의 활용

### Deep learning for Computer Vision

#### 1. Classification



IMAGENET

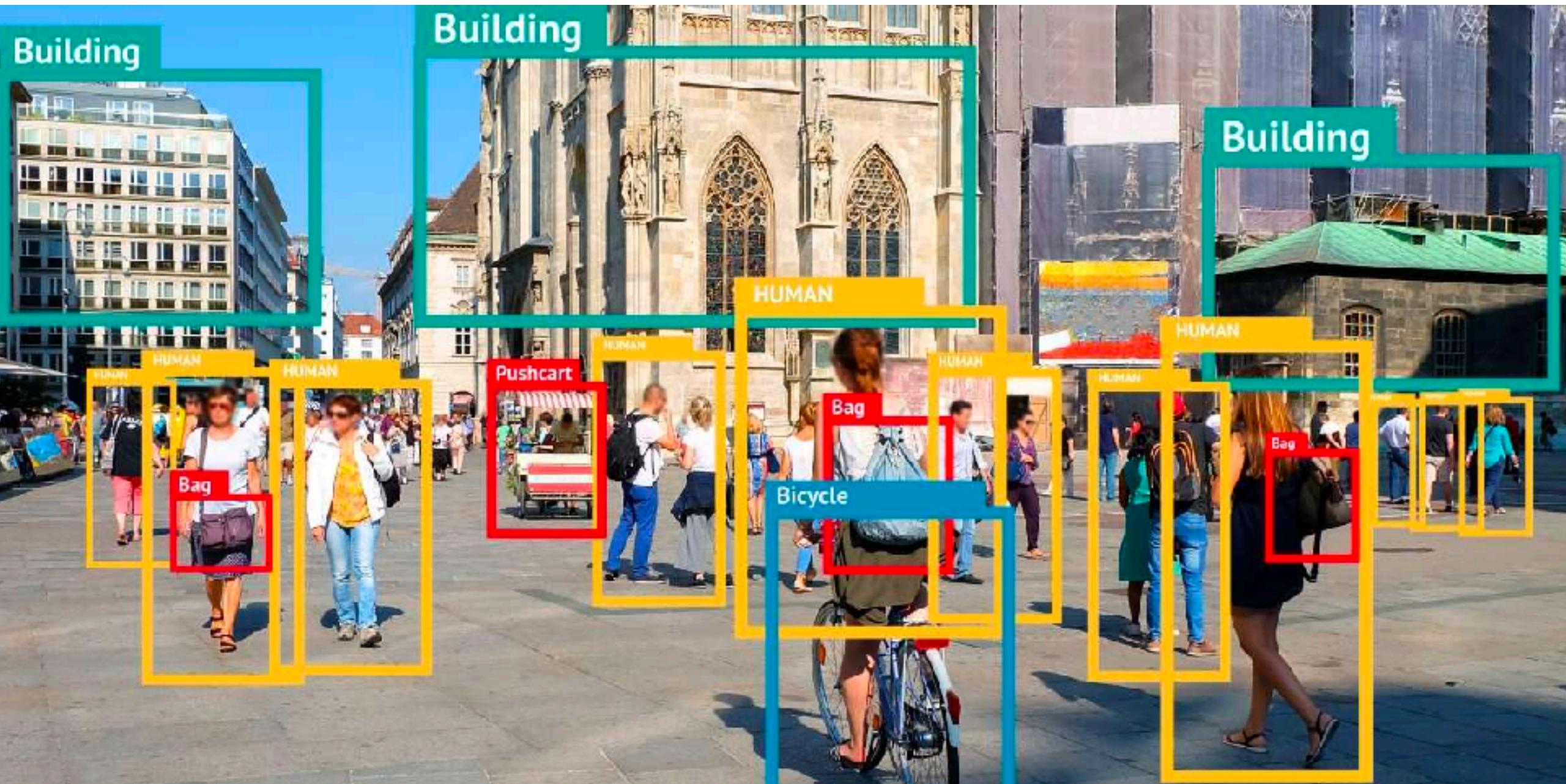


CIFAR-10

# 1. Deep Learning 개요

## 딥러닝의 활용

Deep learning for Computer Vision  
2. Object detection

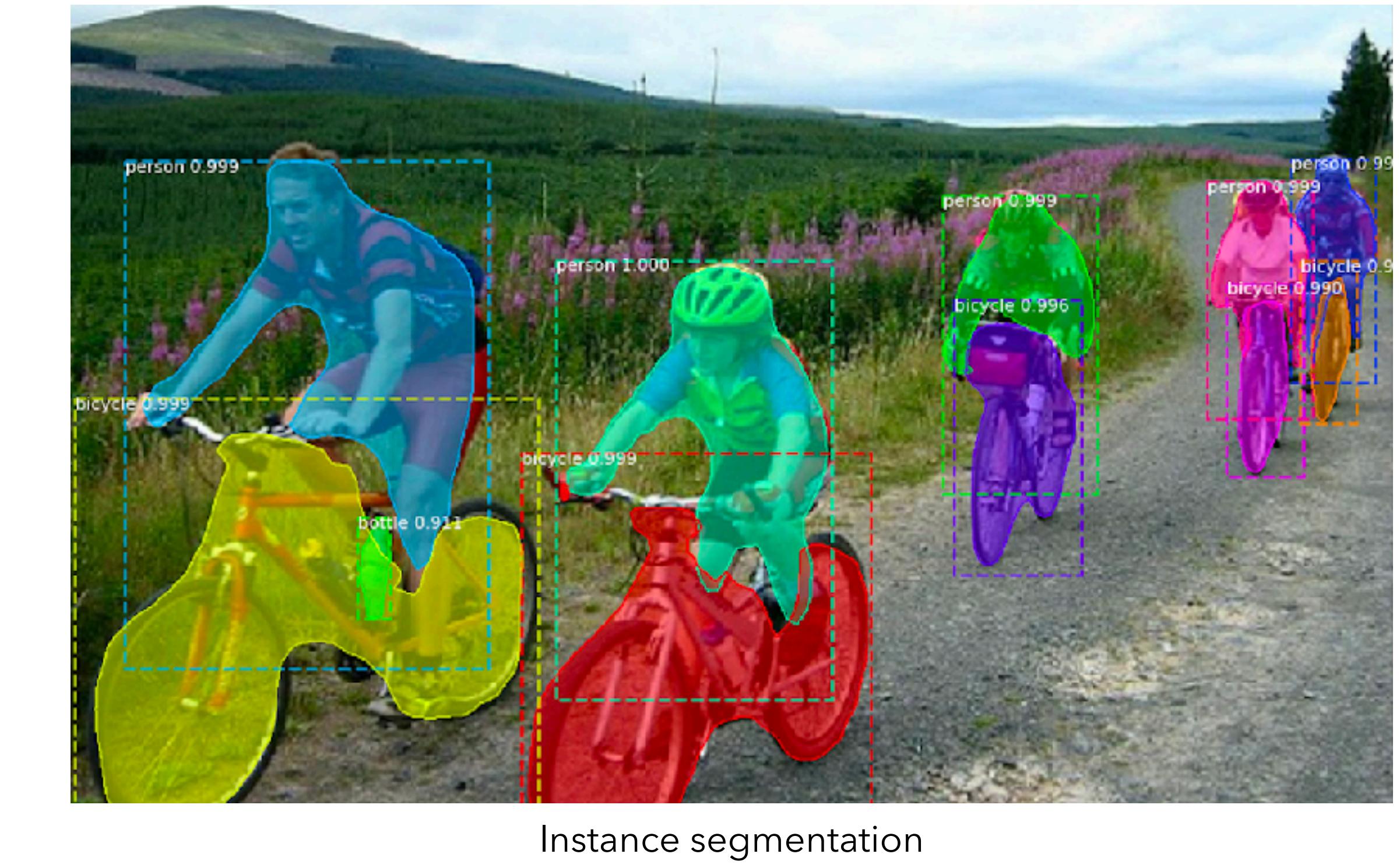
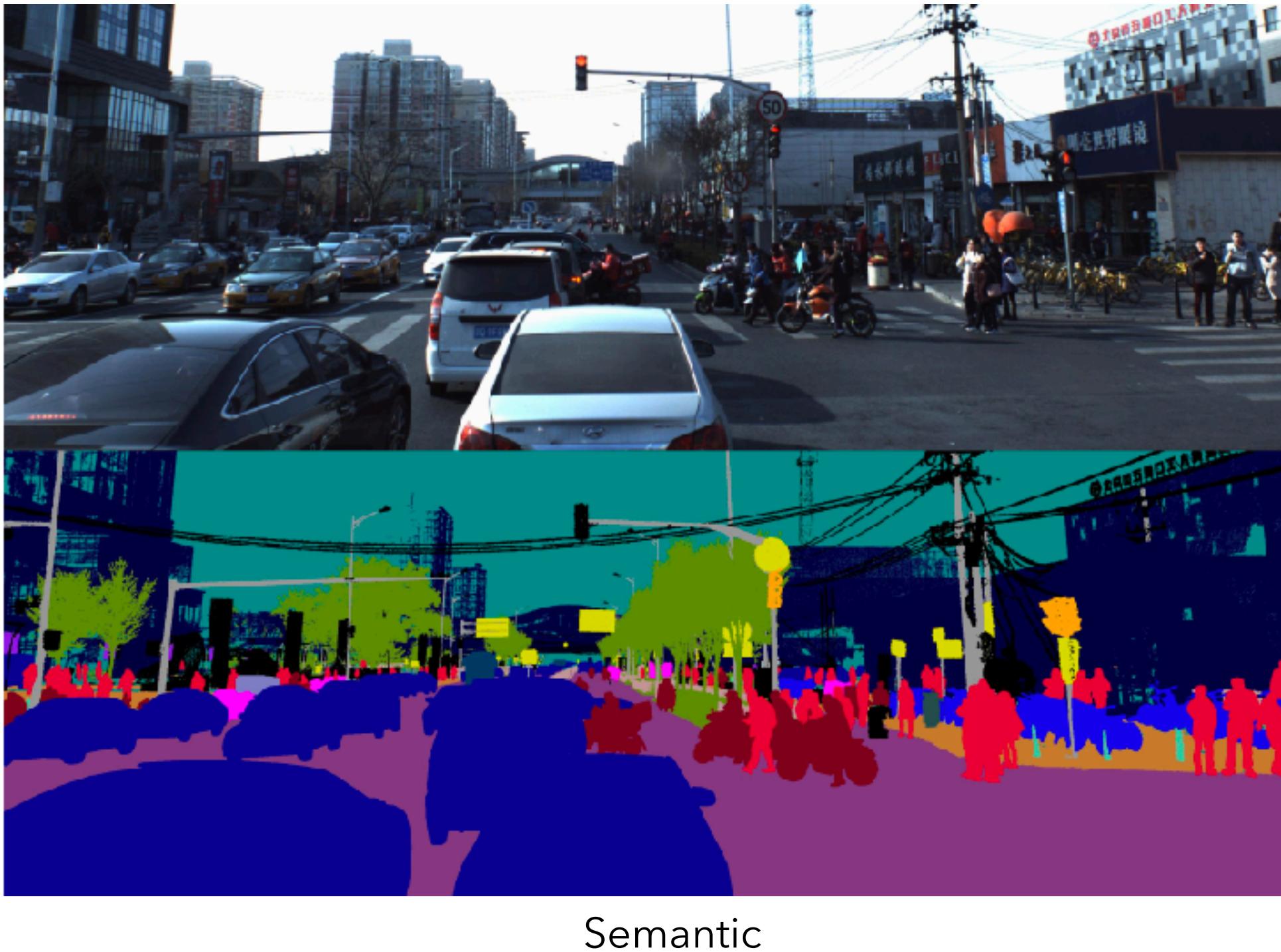


YOLO

# 1. Deep Learning 개요

## 딥러닝의 활용

### Deep learning for Computer Vision 3. Segmentation



<https://towardsdatascience.com/overcome-overfitting-during-instance-segmentation-with-mask-rcnn-32db91f400bc>

<https://autonomous-driving.org/2018/07/15/semantic-segmentation-datasets-for-urban-driving-scenes/>

# 1. Deep Learning 개요

## 딥러닝의 활용

Deep learning for Computer Vision  
4. Pose estimation



<https://medium.com/beyondminds/an-overview-of-human-pose-estimation-with-deep-learning-d49eb656739b>

# 1. Deep Learning 개요

## 딥러닝의 활용

# Deep learning for Computer Vision

4. Video



# 1. Deep Learning 개요

## 딥러닝의 활용

### Deep learning for Natural Language 1. Sentiment Analysis

The screenshot shows a list of movie reviews from a Korean website. Each review includes the user's profile picture, their rating (from 1 to 5 stars), their name, the date and time of the review, and the number of likes and dislikes.

- 작년** (작년) | ★★★★★ 10 | 도움돼요 17 | 2017.06.28 20:04 | 신고  
혜가 수치스러웠습니다.  
우리집 개 먹으라고 줬더니  
갑자기 두 발로 일어서서  
  
"주인님 아무리 제가 짬개지만 이건  
너무하지 않습니까?  
이런식이면 짬개 노릇도 못해먹습니다."  
이러더군요. 감사합니다 덕분에 우리집 개가  
말을합니다.
- 베스트** | ★★★★★ 10 | 2017.06.28 03:01 | 신고  
이 영화를 본 제가 별 10개를 받아야겠습니다
- | 2017.06.28 05:51 | 신고  
저는 10년째 불면증을 겪고있습니다 수면제없인 잠을못잡니다..처음으로 수면제없이 상쾌한잠을  
잤습니다 값비싼 프로포폴대신 싸고저렴한 "리얼"을 강력추천합니다
- | 2017.06.28 01:22 | 신고  
씨리얼 보는게 더 재미있어요
- | 2017.06.28 09:24 | 신고  
5년동안 끊었던 담배를 다시 피게 되었습니다

The screenshot shows two additional movie reviews from the same website.

- 관람객** | ★★★★★ 1 | 2017.06.28 20:04 | 신고  
옆에서 폰을 해도 화나지 않는 영화...??  
+ + + (dmsq\*\*\*\*) | 2017.06.28 20:04 | 신고  
246 | 59
- 거노(cnzk\*\*\*\*)** | ★★★★★ 10 | 2017.06.28 17:46 | 신고  
시험공부에 흥미가 생겼습니다  
거노(cnzk\*\*\*\*) | 2017.06.28 17:46 | 신고  
192 | 14

# 1. Deep Learning 개요

## 딥러닝의 활용

### Deep learning for Natural Language 2. Summarization

The screenshot displays two main sections: 'ARTICLE\_ORIGINAL' and 'ARTICLE\_ABSTRACTIVE'.

**ARTICLE\_ORIGINAL:**

**당진시대**

[당진시 문화관광과를 대상으로 하는 행정사무감사에서 당진시립합창단 관계자가 보낸 것으로 추정되는 문자와 관련해 당진시의회가 행정사무조사특별위원회를 구성해 조사하겠다고 밝혔다.] '당진시의회 행정사무감사 3일차였던 지난 6일 문화관광과를 대상으로 하는 행감에서 최창용 의원은 "(당진시립합창단 정기연주회를 앞두고) 문자메세지를 제보받았다"며 "(음향팀에 보낸 것으로 추정되는) 해당 문자에는 '합창단이 소리를 작게 낼 것이니 알고 있으라'는 내용이었다"고 말했다.] '이어 "공연에서 소리를 작게 낸다는 것은 합창단으로서 그 임무를 하지 않겠다는 것"이며 "공연 자체를 무력화해 당진시를 망신 주려는 행위"라며 해당 문자를 보낸 단원 등 연루된 사람들을 찾아 사실관계를 확인하고 징계 등 책임을 물어야 한다고 지적했다.] '문제의 문자메세지를 전달받은 문화관광과는 감사법무담당관에게 조사를 의뢰했다.] '권경선 문화관광과장은 "누가, 어떻게 해서 해당 문자가 나온 것인지 정확히 조사해봐야 할 수 있다"며 "전달받은 문자 내용도 최 의원이 언급한 부분만 있어, 중간 내용만 가지고는 전체를 유추할 수 없다"고 전했다.] '하지만 감사법무담당관실에서 아직 조사가 이뤄지지 않고 있어 당진시가 사태의 심각성을 인지하지 못하고 있다는 지적이다.] '그동안 행정사무감사가 진행되고 있어 사태를 지켜봤던 감사법무담당관실에서는 "관계된 사람들을 조사해 사태를 파악해야 하는데, 아직 조사에 대한 뚜렷한 계획이 없다"고 답했다.] '한편 행감이 끝난 지난 12일 당진시의회에서는 당진시립합창단 문제를 비롯해 구체적인 조사가 필요한 부분에 대해 행정사무조사특별위원회를 구성해 운영하겠다고 밝혔다.] '김기재 의장은 "본회의 의결과 제적의원 1/3의 발의가 있으면 행정사무조사특별위원회를 구성할 수 있다"며 "다음 달 초 위원들과 상의해 위원회를 구성한다면 당진시립합창단 관련 사안을 비롯해 사회복지기관 위수탁 등에 대해 다룰 계획"이라고 말했다.]'

**ARTICLE\_ABSTRACTIVE:**

**당진시대**

지난 6일 당진시의회 행정사무감사에서 '합창단이 소리를 작게 낼 것이니 알고 있으라'라는 문자 등으로 불거진 합창단의 의무 불이행 논란에 대해 행정사무조사특별위원회를 구성해 조사를 수행하겠다는 의견을 표명했다.

**EXTRACTIVE**

[0, 1, 2]

**ABSTRACTIVE**

LA 에인절스의 최지만이 맹활약을 하여 시즌 타율 0.250에서 0.313으로 올리며 주전 입지를 굳혀가는 반면 텍사스 레인저스의 추신수는 개막전 선발에서 밀려나는 등 반등을 하지 못하는 모습을 보이며, 미국 메이저리그에서 활동하는 한국 선수들의 시즌 초반 희비가 엇갈리고 있다.

**ARTICLE\_ORIGINAL:**

**국제신문**

[미국 메이저리그(MLB)에서 활동하는 한국 선수들의 시즌 초반 희비가 엇갈리고 있다.] 'LA 에인절스의 최지만은 맹활약으로 주전 입지를 굳혀가는 반면 텍사스 레인저스의 추신수는 개막전 선발에서 밀려나는 수모를 당한 뒤에서도 좀처럼 반등을 하지 못하고 있다.] '최지만은 2일(한국시간) 미국 플로리다주 세인트피터즈버그 트로피카나 필드에서 열린 2019 미국프로야구 메이저리그 콜로라도 로키스와 홈경기에 3번 타자 1루수로 선발 출전해 4타수 2안타를 치고 1득점을 올렸다.] '시즌 타율은 0.250에서 0.313(16타수 5안타)로 올랐다.] '최지만은 개막전에서 4타수 무안타로 침묵했으나 이후 3경기 연속 안타이자 2경기 연속 멀티 힙트를 뽑아내며 입지를 굳혀갔다.] '2016년 LA 에인절스 소속으로 메이저리그에 데뷔한 최지만이 2경기 연속 멀티 힙트를 친 것은 이번이 처음이다.] '스코어가 벌어진 탓에 콜로라도 불펜의 필승조인 오승환과 최지만의 맞대결은 성사되지 않았다.] '추신수는 들쭉날쭉한 출전 일정 탓인지 타격감을 찾지 못하고 있다.] '추신수는 이날 텍사스주 알링턴 글로브 라이프 파크에서 열린 휴斯顿 애스트로스와 홈경기에 1번 지명타자로 선발 출전했지만 4타수 무안타로 침묵했다.] '시즌 타율은 0.167에서 0.100(10타수 1안타)로 떨어졌다.] '추신수는 11년 만에 개막전 선발 라인업에서 제외되는 등 이날까지 텍사스가 치른 4경기 가운데 2경기에만 선발 출전했다.] '크리스 우드워드 텍사스 감독은 상대 선발이 좌완이면 좌타자인 추신수 대신 우타자인 헌터 펜스를 선발 지명타자로 기용하고 있다.] '텍사스(2승 2패)는 휴斯顿 마운드에 2안타로 꽁꽁 묶이며 1-2로 패해 2연승 행진을 마감했다.] '한편 피츠버그 파이리츠의 강정호는 914일 만에 홈경기에 나섰지만 안타를 만들어내지 못했다.] '이날 펜실베이니아주 피츠버그의 PNC 파크에서 열린 세인트루이스 카디널스와 홈 개막전에 선발 라인업에서 제외된 강정호는 벤치에서 경기를 시작했다.] '강정호는 4-4 동점이 된 8회 초 2사 2루에서 3루 대수비로 교체 출전했다.] '강정호가 PNC 파크 그라운드를 다시 밟은 것은 2016년 9월 30일 시카고 컵스와의 홈 최종전 이후 정확히 914일 만이다.] '강정호는 8회 말 선두타자 모란이 솔로포를 터트려 스코어가 5-4가 된 이후 후속 타석에 들어섰지만 중견수 뜬 공으로 물려났다.] '연장 10회 말 1사에서 다시 타석에 들어선 강정호는 세인트루이스의 우완 '파이어볼러' 조던 힉스에게 3구 삼진으로 물려났다.] '2타수 무안타 1삼진으로 침묵한 강정호의 시즌 타율은 0.222(9타수 2안타)가 됐다.] '피츠버그는 연장 11회 혈투 끝에 5-6으로 패했다.]'

**ARTICLE\_ABSTRACTIVE:**

LA 에인절스의 최지만이 맹활약을 하여 시즌 타율 0.250에서 0.313으로 올리며 주전 입지를 굳혀가는 반면 텍사스 레인저스의 추신수는 개막전 선발에서 밀려나는 등 반등을 하지 못하는 모습을 보이며, 미국 메이저리그에서 활동하는 한국 선수들의 시즌 초반 희비가 엇갈리고 있다.

**EXTRACTIVE**

[0, 1, 3]

# 1. Deep Learning 개요

## 딥러닝의 활용

Deep learning for Natural Language  
3. Machine Translation

The screenshot shows a machine translation interface with two panels. The left panel is for Korean input, and the right panel is for English output. Both panels have dropdown menus for language selection ('한국어' and '영어') and a double-headed arrow icon between them.

**Korean Panel:**

- Input text: 끝도 없이 먼 하늘 날아가는 새처럼 뒤돌아 보지 않을래. ×
- Count: 31 / 5000
- Buttons:Speaker icon, Copy icon, Share icon, and a green '번역하기' (Translate) button.

**English Panel:**

- Output text: I won't look back like a bird flying endlessly into the distant sky.
- Text below: 아이 워운트 룩 백 라익 어 버드 플라이잉 엔들러슬리 인투 더 디스턴트 스카이.
- Buttons: Speaker icon, Copy icon, Star icon, Share icon, and links for '번역 수정' (Edit Translation) and '번역 평가' (Evaluate Translation).

# 1. Deep Learning 개요

## 딥러닝의 활용

### Deep learning for Natural Language 4. Question Answering

#### Passage Sentence

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity.

#### Question

What causes precipitation to fall?

#### Answer Candidate

gravity

- Between question and answer

cause---gravity  
precipitation---gravity  
fall---gravity  
what---gravity

#### Passage Sentence

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity.

#### Question

What causes precipitation to fall?

#### Answer Candidate

gravity

- Between question and passage sentence (around candidate)

cause---under  
cause---fall  
precipitation---fall  
fall---under

# 1. Deep Learning 개요

## 딥러닝의 활용

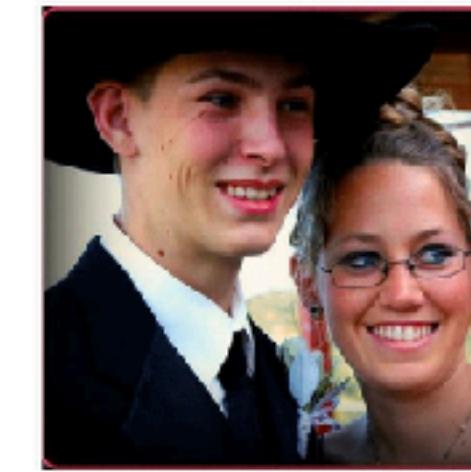
Deep learning for Multi modal  
1. VQA

Who is wearing glasses?

man



woman



Where is the child sitting?

fridge



arms



Is the umbrella upside down?

yes

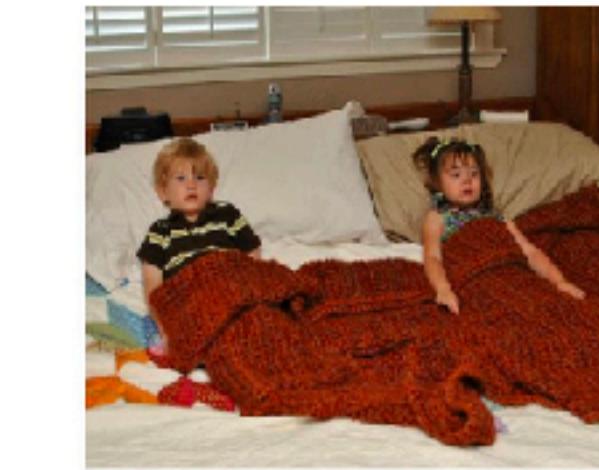


no



How many children are in the bed?

2



1



# 1. Deep Learning 개요

## 딥러닝의 활용

Deep learning for Multimodal  
2. Image Captioning

A young boy is playing basketball. 	Two dogs play in the grass. 	A dog swims in the water. 
A group of people walking down a street. 	A group of women dressed in formal attire. 	Two children play in the water. 
A skier is skiing down a snowy hill. 	A little girl in a pink shirt is swinging. 	A dog jumps over a hurdle. 



오레오로 사람 못죽일것 같지?

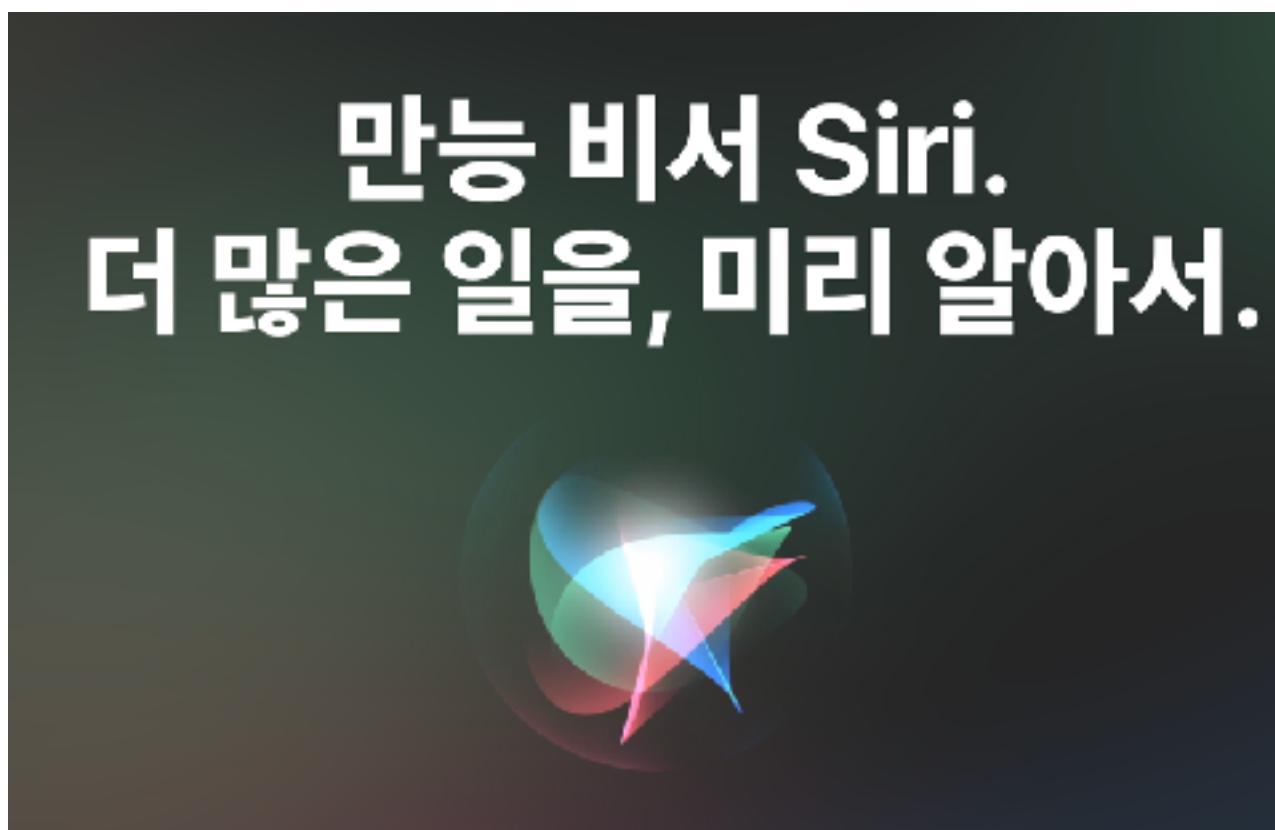
제목학원 수강생 'ジョジョ'



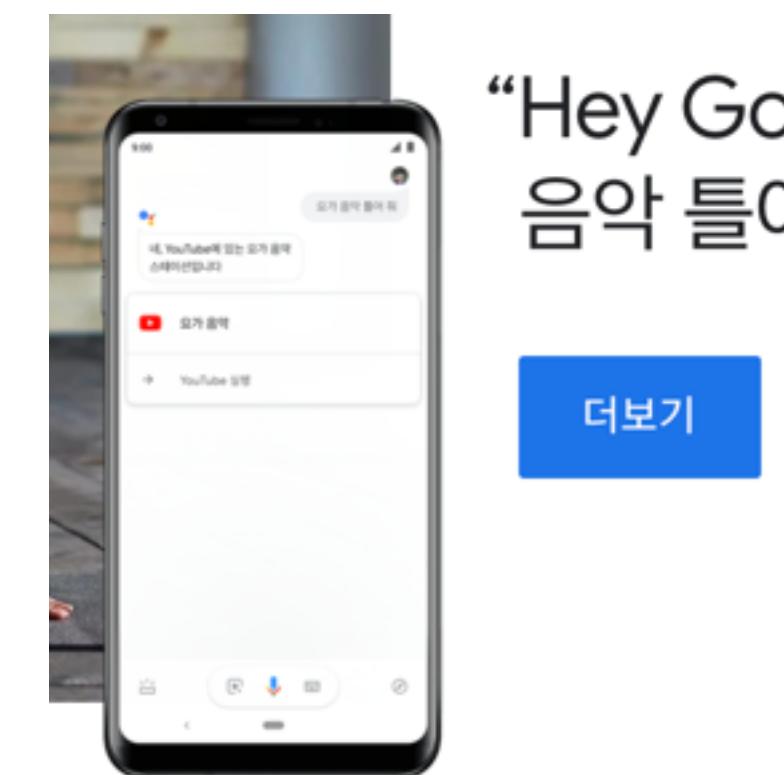
# 1. Deep Learning 개요

## 딥러닝의 활용

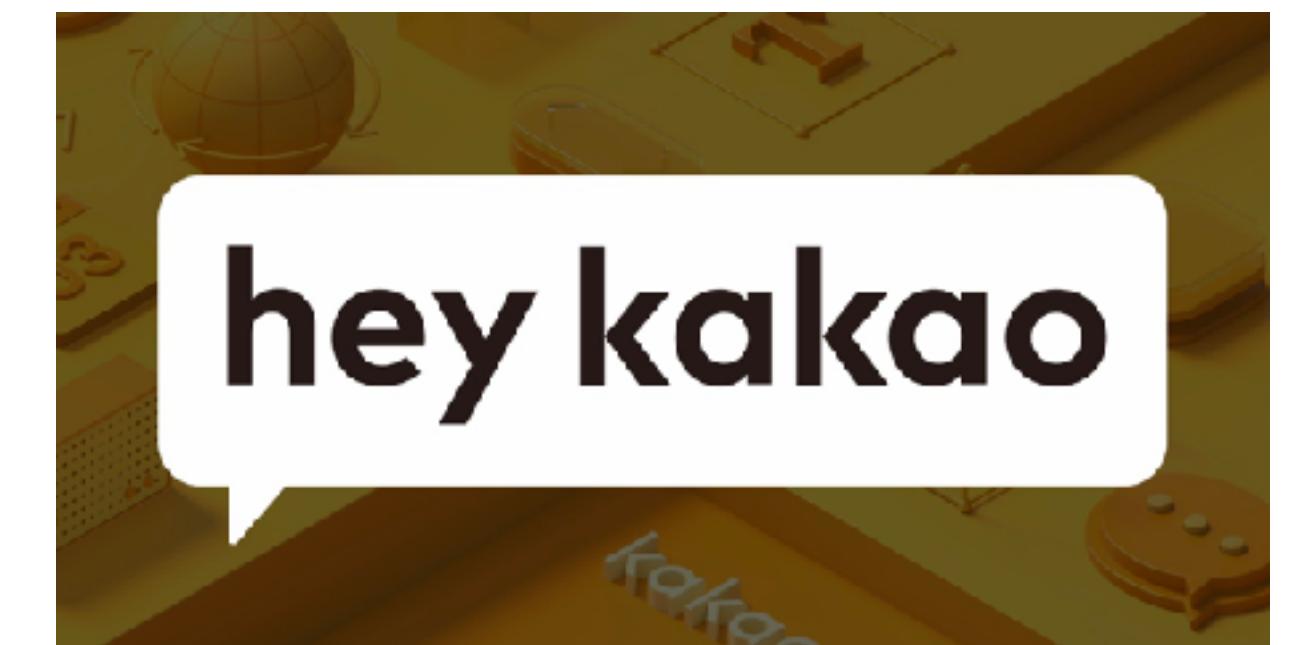
Deep learning for Speech Recognition



Say less, do more.



“Hey Google, 요가  
음악 틀어 줘”

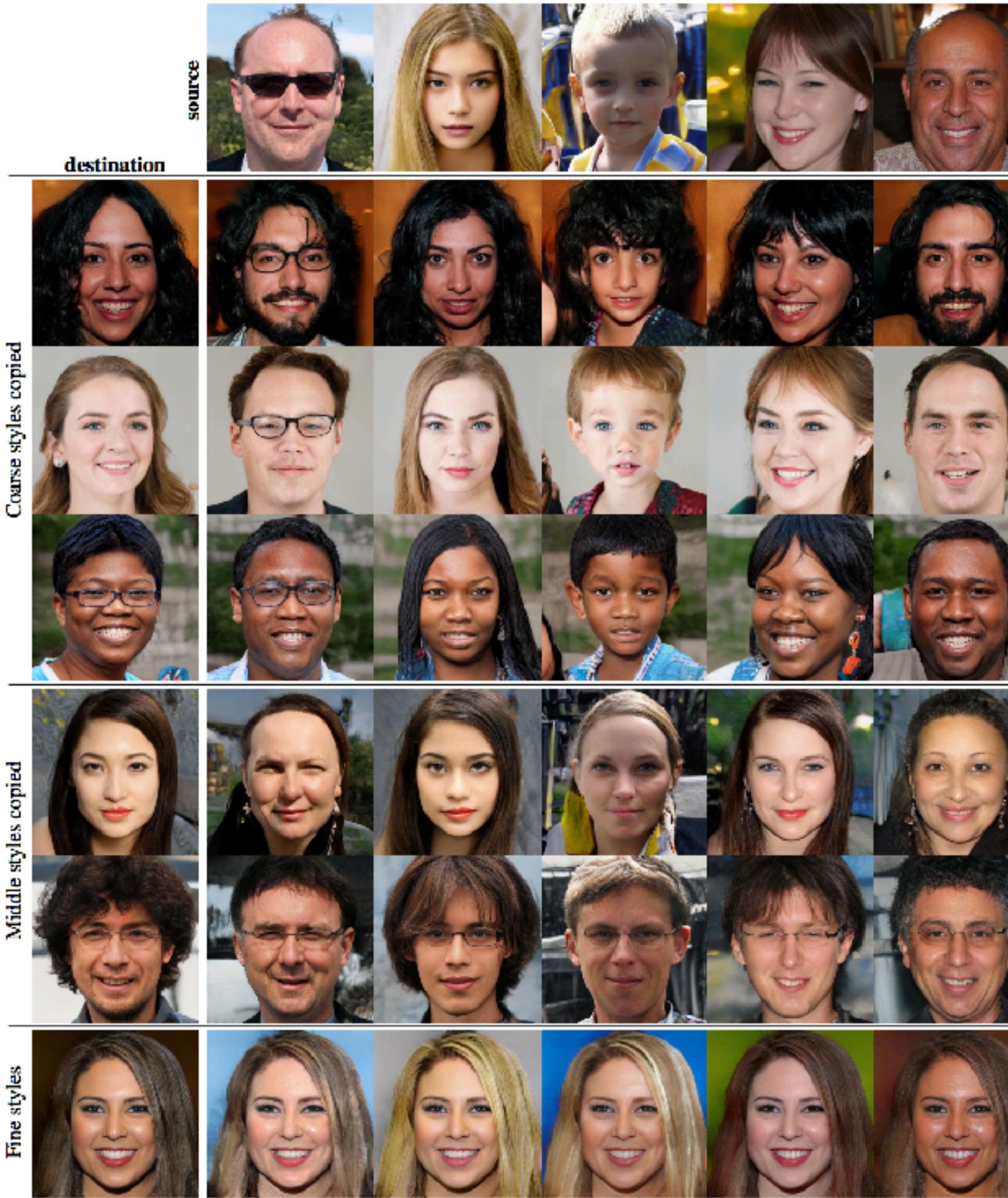


# 1. Deep Learning 개요

## 딥러닝의 활용

### Deep learning for Generation

#### 1. Image generation



DALL·E: Creating Images from Text

# 1. Deep Learning 개요

## 딥러닝의 활용

### Deep learning for Generation

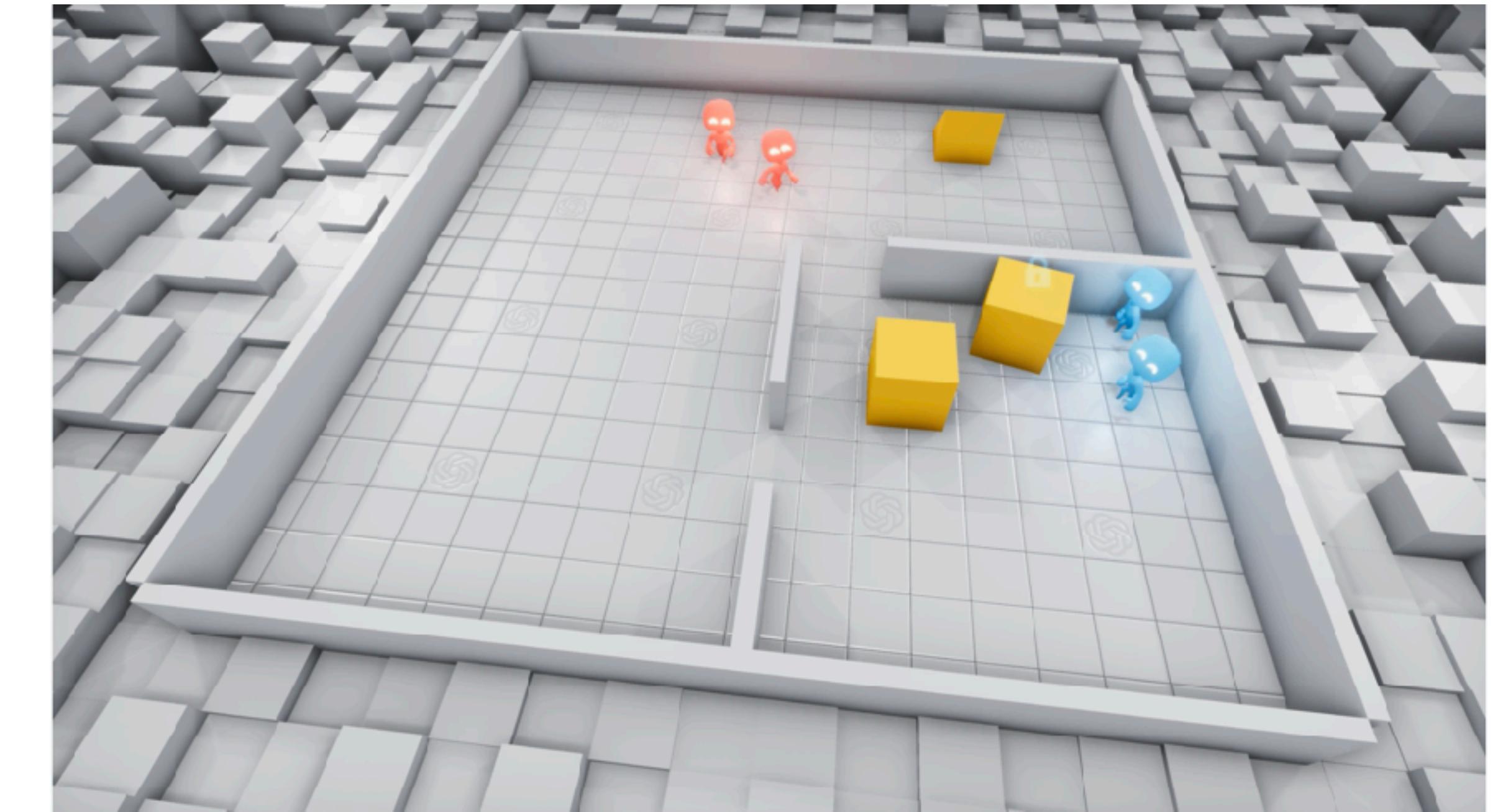
#### 2. Text generation

SYSTEM PROMPT (HUMAN-WRITTEN)	<p><i>In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.</i></p>
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	<p>The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.</p> <p>Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.</p> <p>Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.</p> <p>Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.</p> <p>Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.</p> <p>While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."</p>

# 1. Deep Learning 개요

## 딥러닝의 활용

Deep learning for Reinforcement Learning



<https://openai.com/blog/emergent-tool-use/>

<https://openai.com/blog/solving-rubiks-cube/>

# 1. Deep Learning 개요

## 딥러닝의 활용

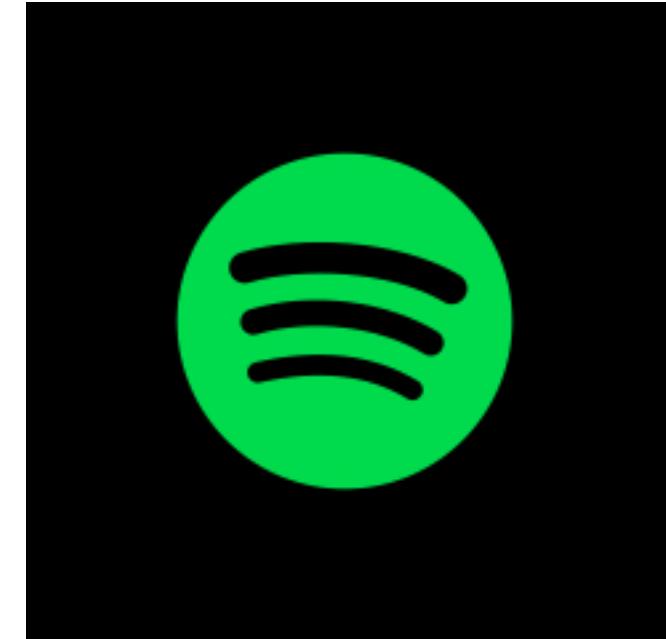
Recommender Systems



Google

facebook

amazon



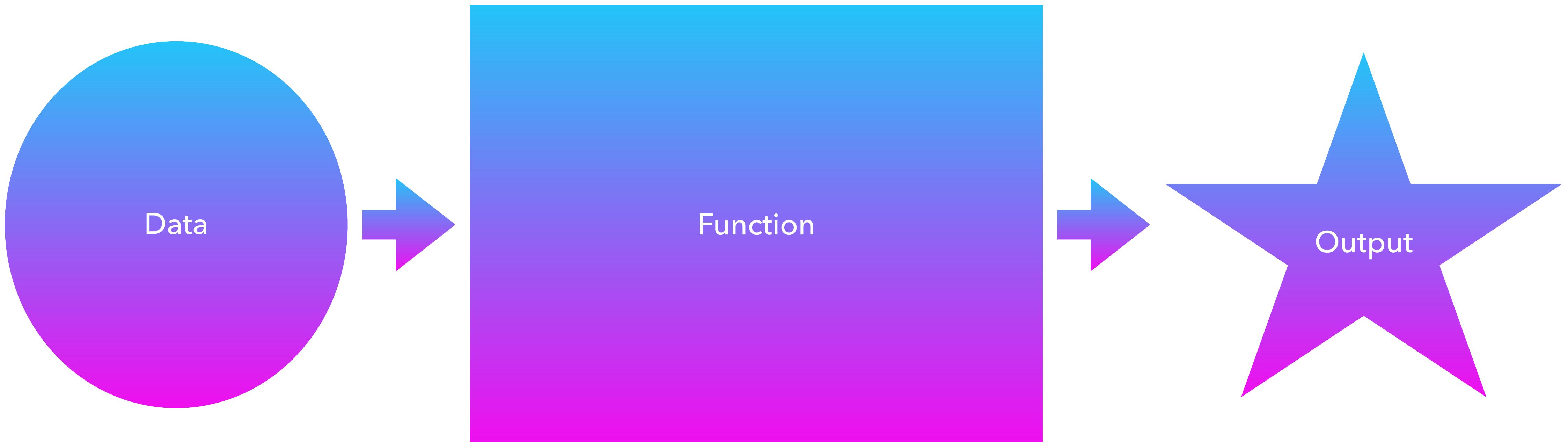
Spotify

Amazon Wikipedia

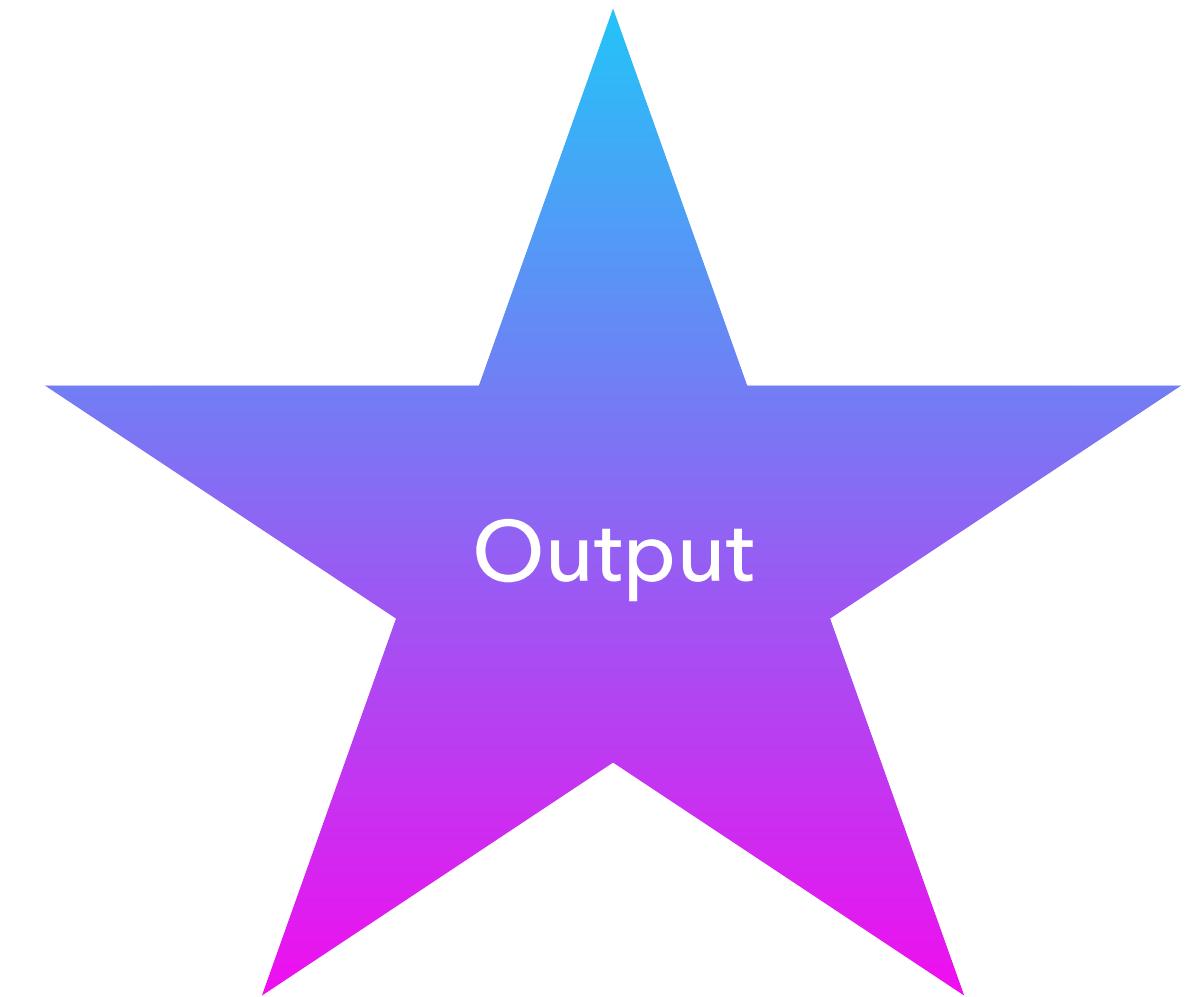
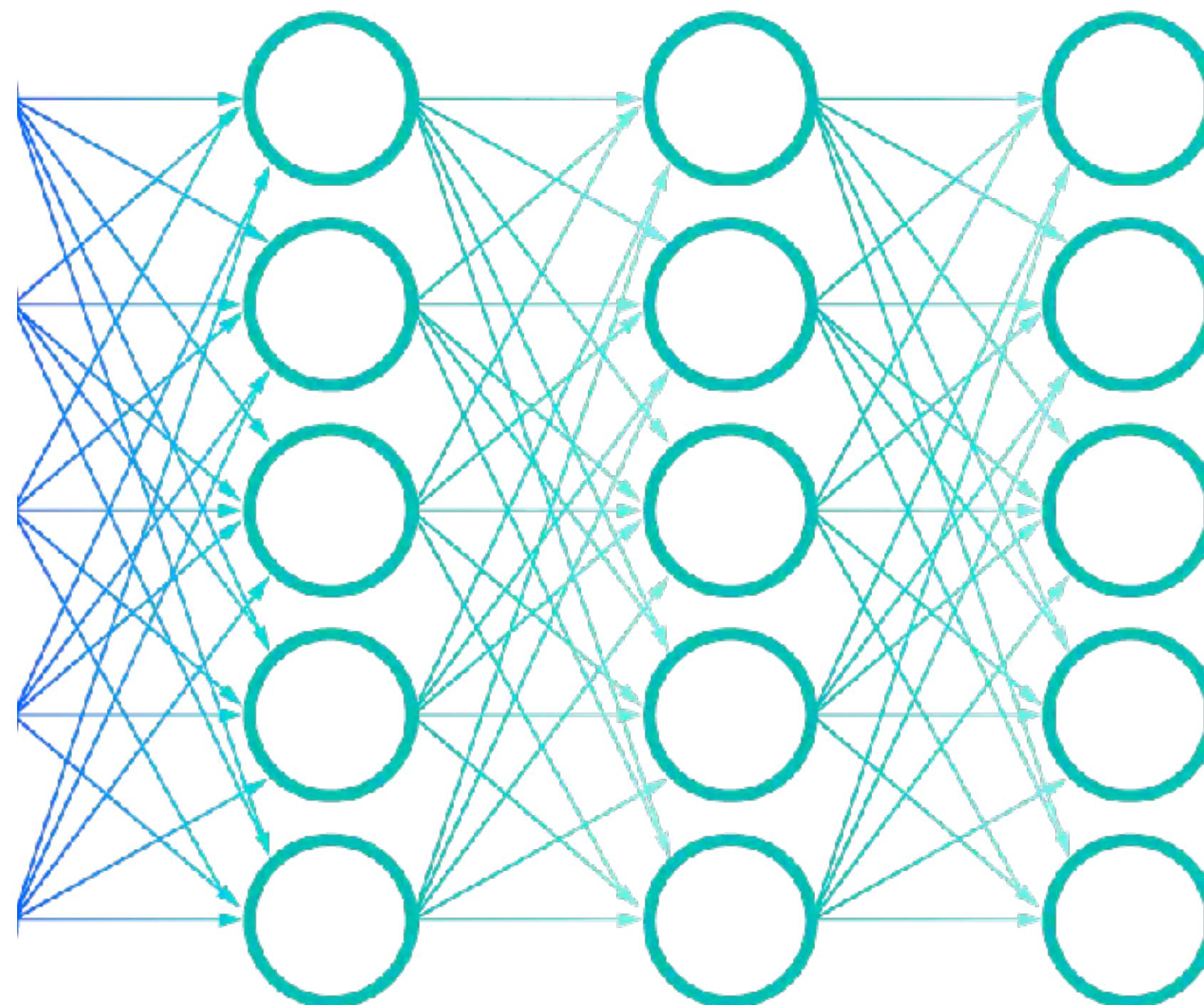
Netflix Facebook

# Feed-Forward Function

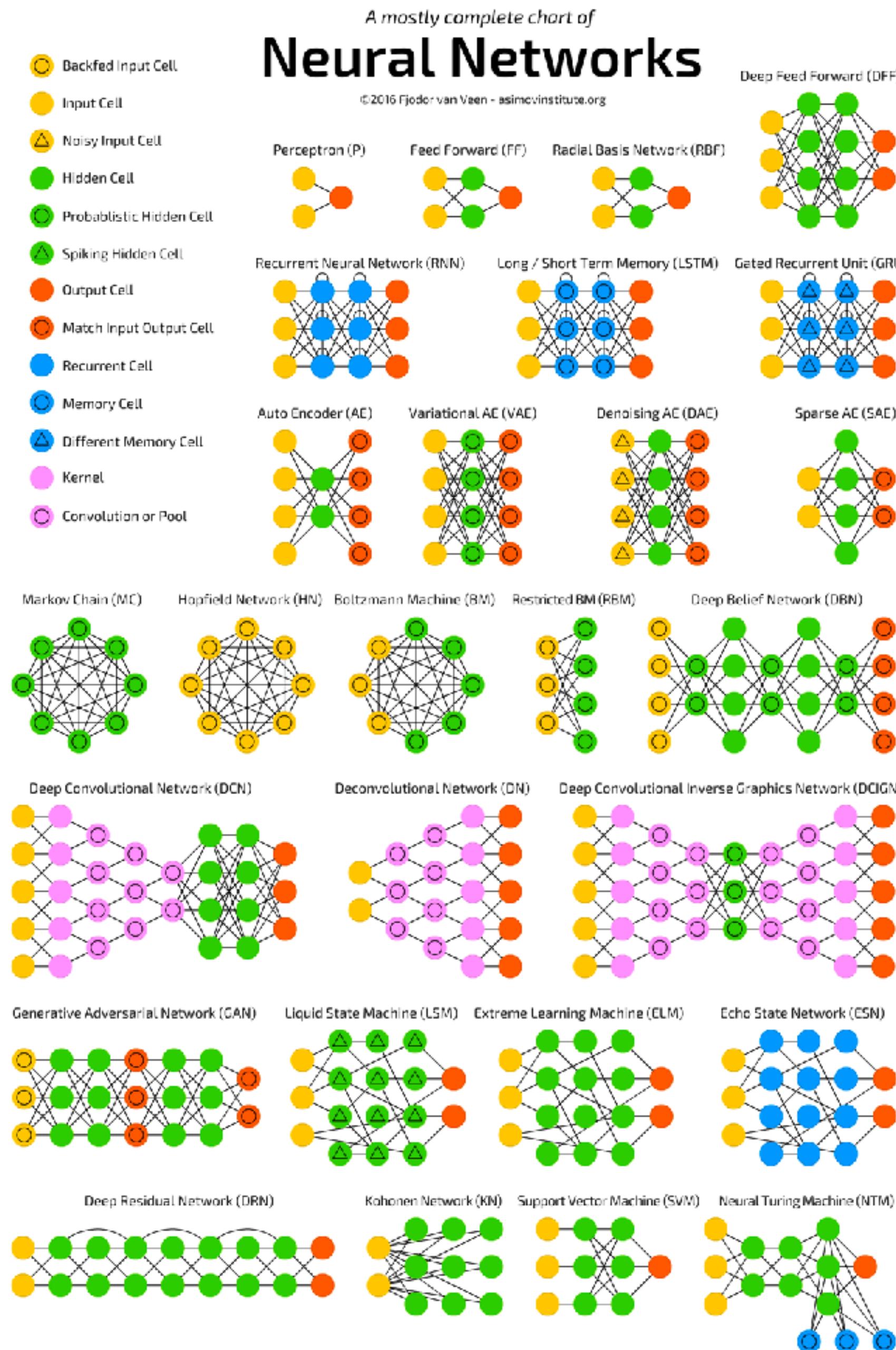
## 2. Deep Learning 이론 FeedForward function.



## 2. Deep Learning 이론 FeedForward function.

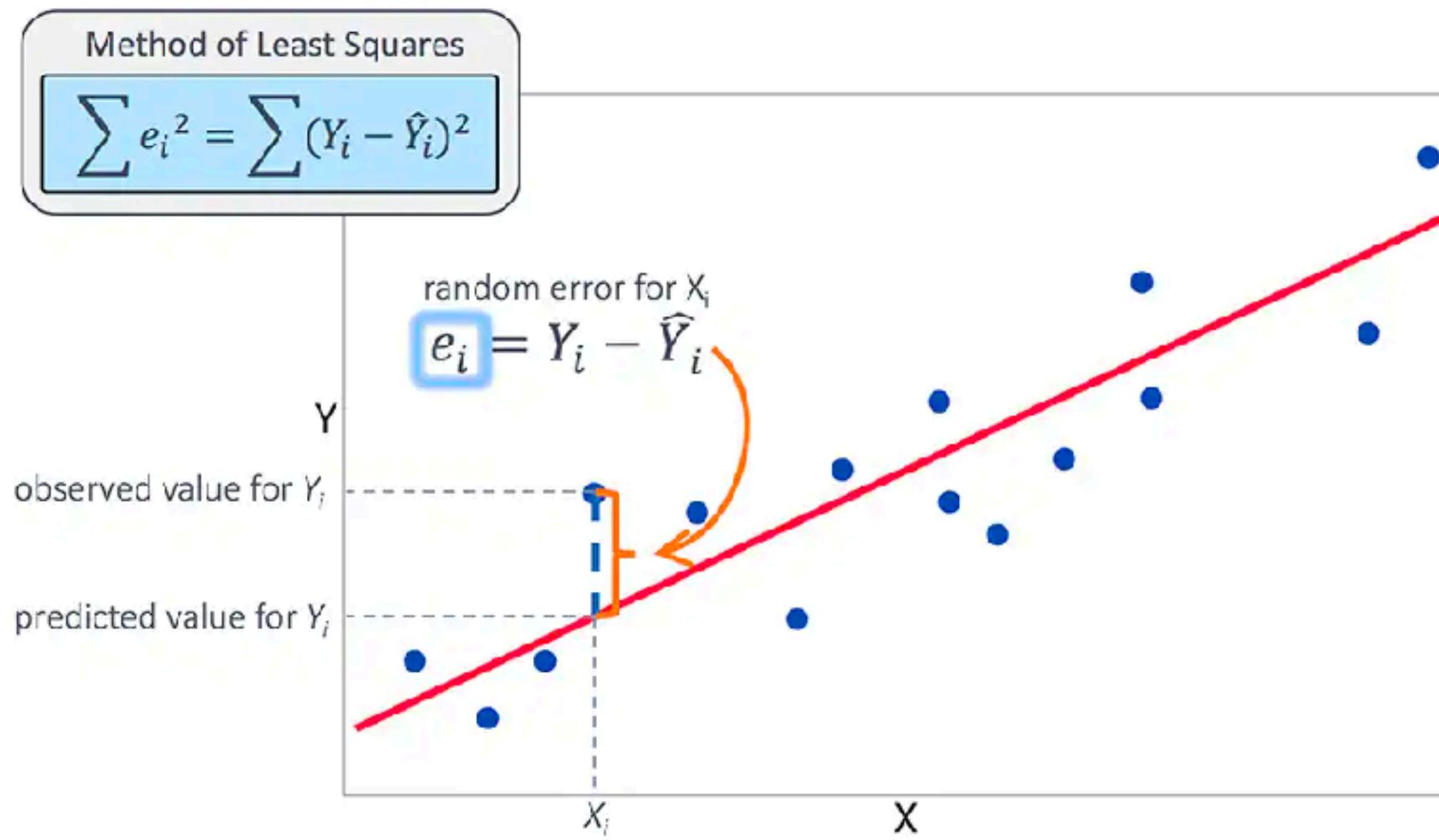


## 2. Deep Learning 이론 FeedForward function.

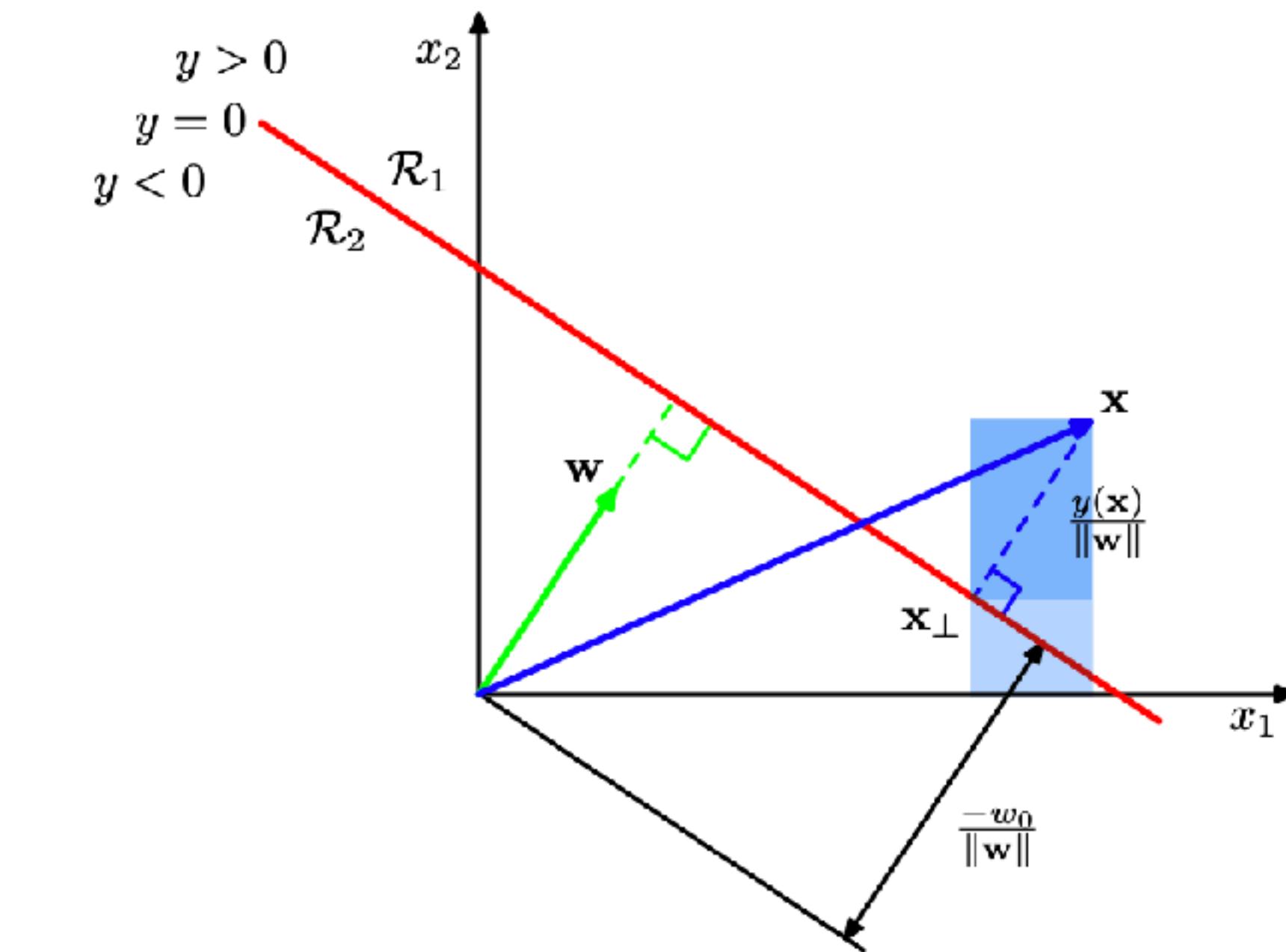


## 2. Deep Learning 이론 FeedForward function.

Linear Regression



Linear Classification



## 2. Deep Learning 이론 FeedForward function.

approach in which we model the class-conditional densities  $p(\mathbf{x}|\mathcal{C}_k)$ , as well as the class priors  $p(\mathcal{C}_k)$ , and then use these to compute posterior probabilities  $p(\mathcal{C}_k|\mathbf{x})$  through Bayes' theorem.

Consider first of all the case of two classes. The posterior probability for class  $\mathcal{C}_1$  can be written as

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} = \sigma(a) \end{aligned} \quad (4.57)$$

where we have defined

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \quad (4.58)$$

and  $\sigma(a)$  is the *logistic sigmoid* function defined by

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \quad (4.59)$$

which is plotted in Figure 4.9. The term ‘sigmoid’ means S-shaped. This type of function is sometimes also called a ‘squashing function’ because it maps the whole real axis into a finite interval. The logistic sigmoid has been encountered already in earlier chapters and plays an important role in many classification algorithms. It satisfies the following symmetry property

$$\sigma(-a) = 1 - \sigma(a) \quad (4.60)$$

as is easily verified. The inverse of the logistic sigmoid is given by

$$a = \ln \left( \frac{\sigma}{1 - \sigma} \right) \quad (4.61)$$

and is known as the *logit* function. It represents the log of the ratio of probabilities  $\ln [p(\mathcal{C}_1|\mathbf{x})/p(\mathcal{C}_2|\mathbf{x})]$  for the two classes, also known as the *log odds*.

Note that in (4.57) we have simply rewritten the posterior probabilities in an equivalent form, and so the appearance of the logistic sigmoid may seem rather vacuous. However, it will have significance provided  $a(\mathbf{x})$  takes a simple functional form. We shall shortly consider situations in which  $a(\mathbf{x})$  is a linear function of  $\mathbf{x}$ , in which case the posterior probability is governed by a generalized linear model.

For the case of  $K > 2$  classes, we have

$$\begin{aligned} p(\mathcal{C}_k|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} \\ &= \frac{\exp(a_k)}{\sum_j \exp(a_j)} \end{aligned} \quad (4.62)$$

which is known as the *normalized exponential* and can be regarded as a multiclass generalization of the logistic sigmoid. Here the quantities  $a_k$  are defined by

$$a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k). \quad (4.63)$$

The normalized exponential is also known as the *softmax function*, as it represents a smoothed version of the ‘max’ function because, if  $a_k \gg a_j$  for all  $j \neq k$ , then  $p(\mathcal{C}_k|\mathbf{x}) \simeq 1$ , and  $p(\mathcal{C}_j|\mathbf{x}) \simeq 0$ .

We now investigate the consequences of choosing specific forms for the class-conditional densities, looking first at continuous input variables  $\mathbf{x}$  and then discussing briefly the case of discrete inputs.

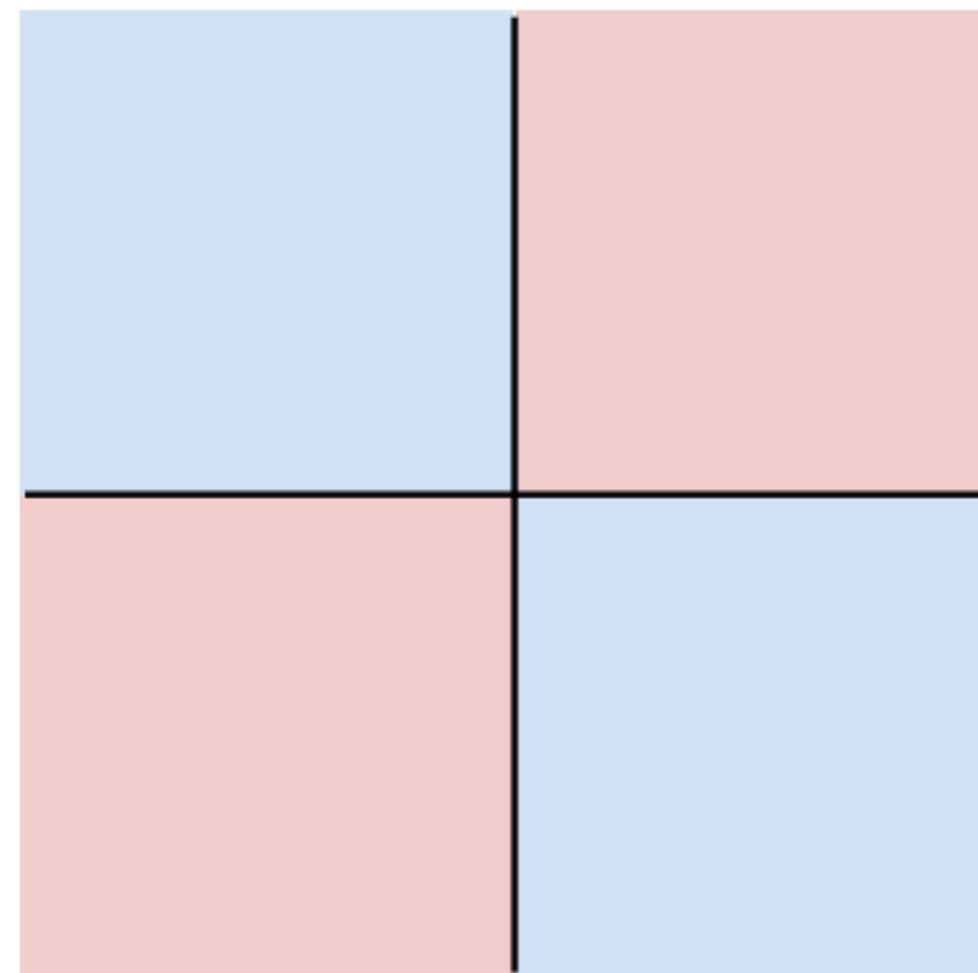
## Hard Cases for a Linear Classifier

**Class 1:**

First and third quadrants

**Class 2:**

Second and fourth quadrants

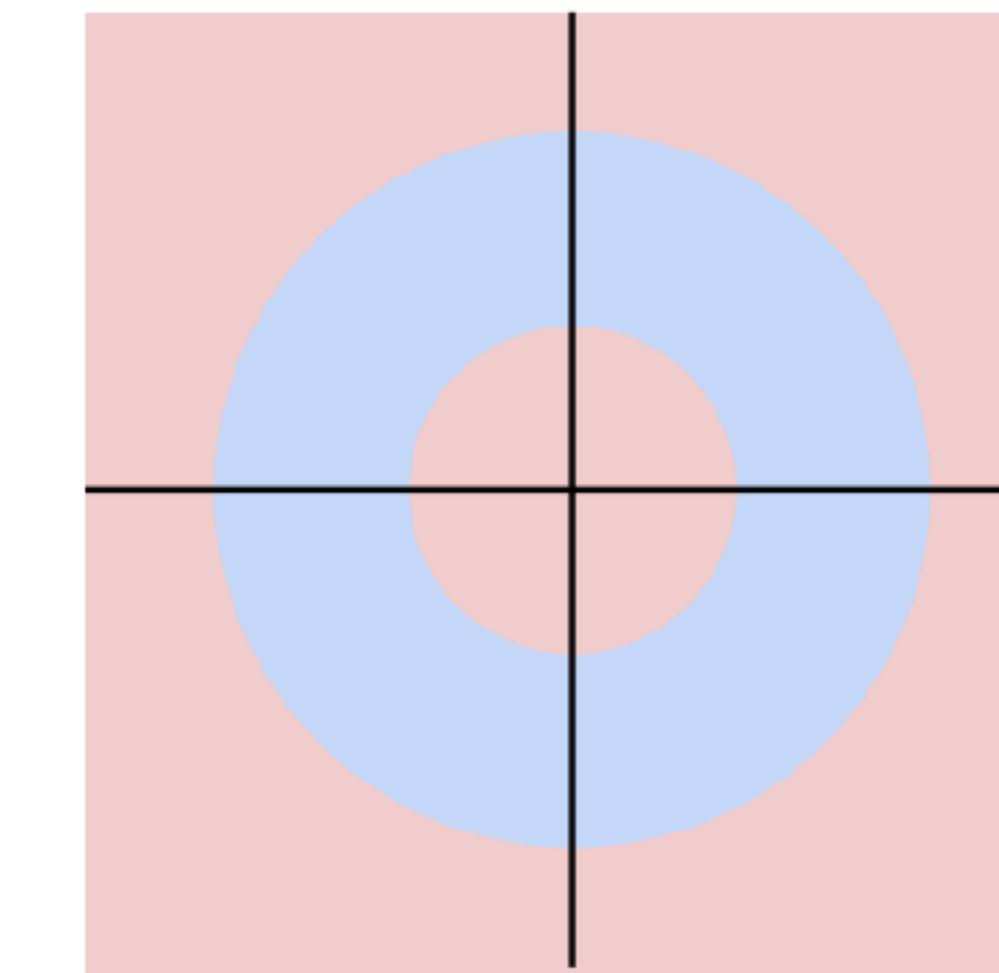


**Class 1:**

$1 \leq L_2 \text{ norm} \leq 2$

**Class 2:**

Everything else

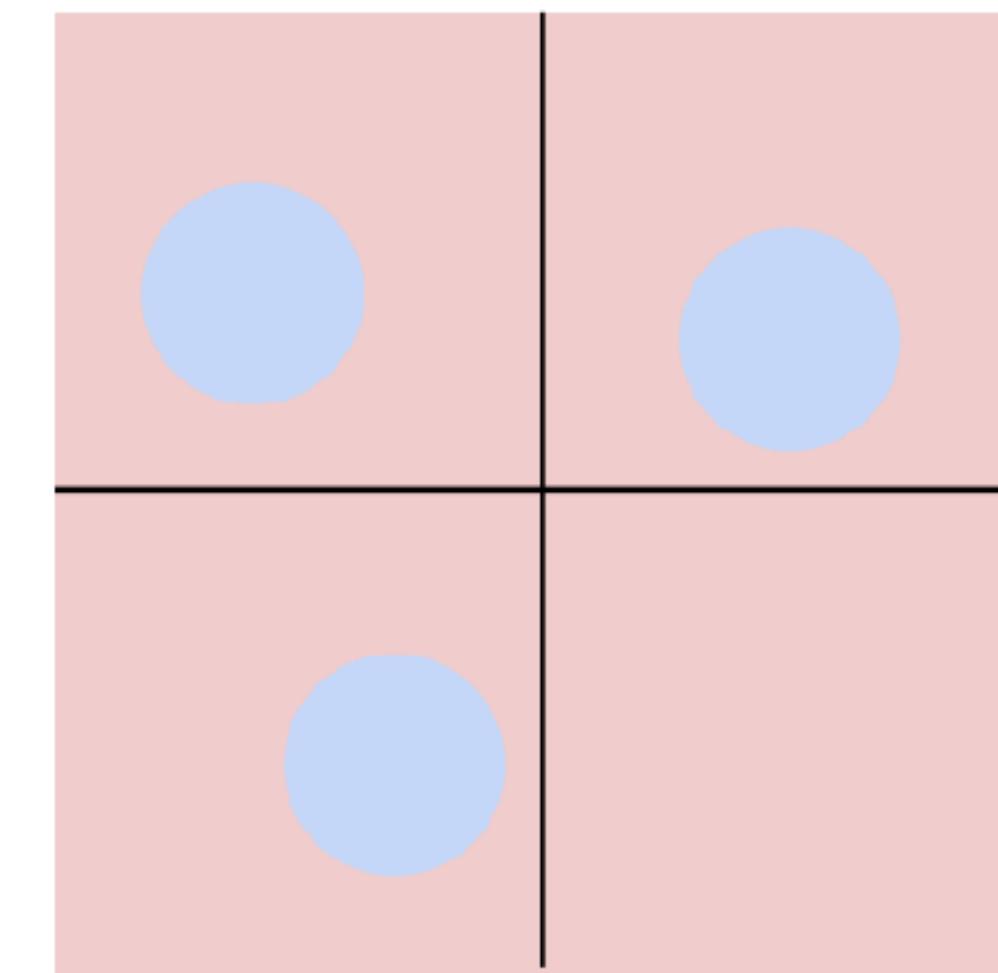


**Class 1:**

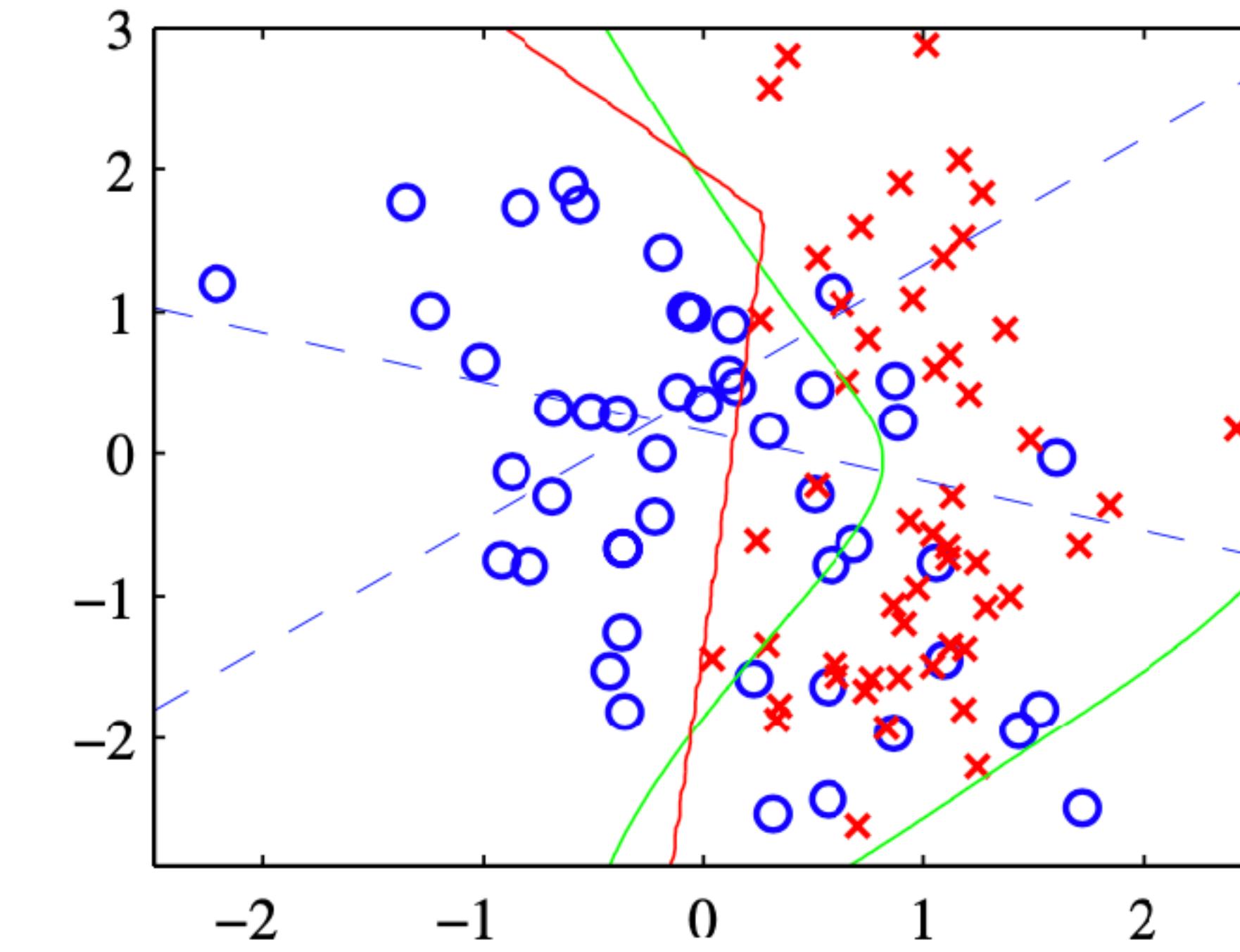
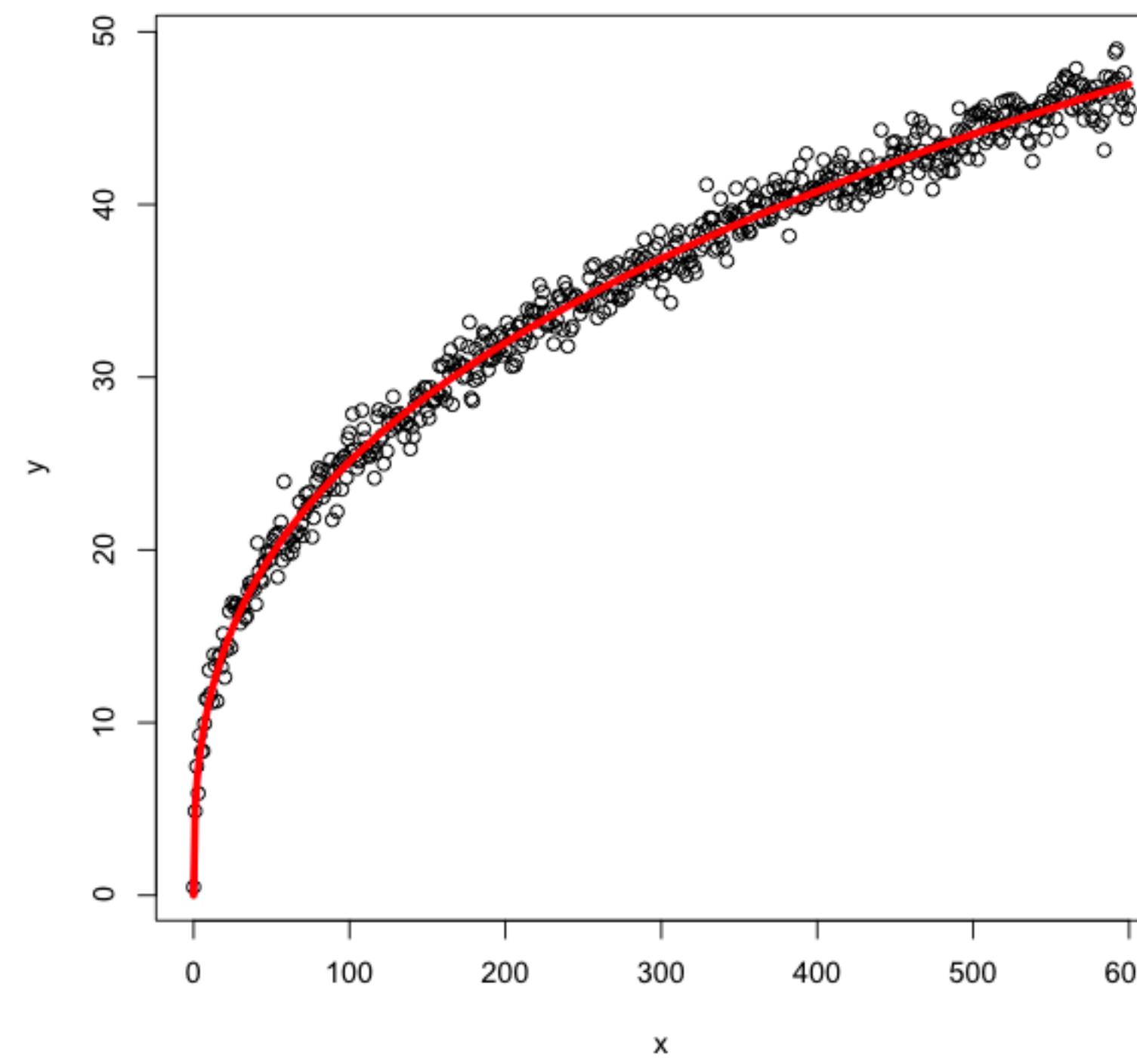
Three modes

**Class 2:**

Everything else



## 2. Deep Learning 이론 FeedForward function.



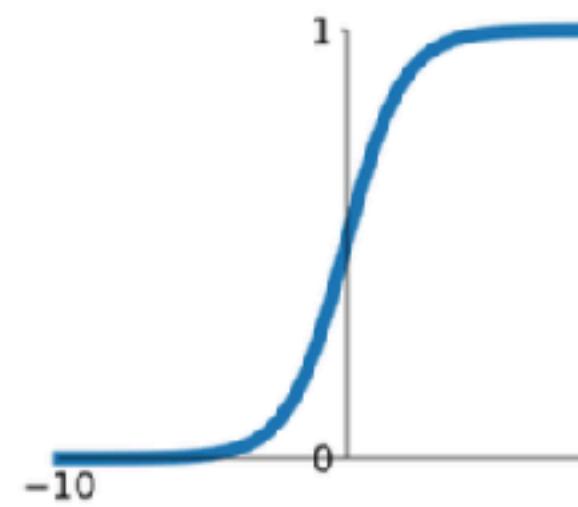
Nonlinear regression in R

2. Deep Learning 이론  
FeedForward function.

## Activation Functions

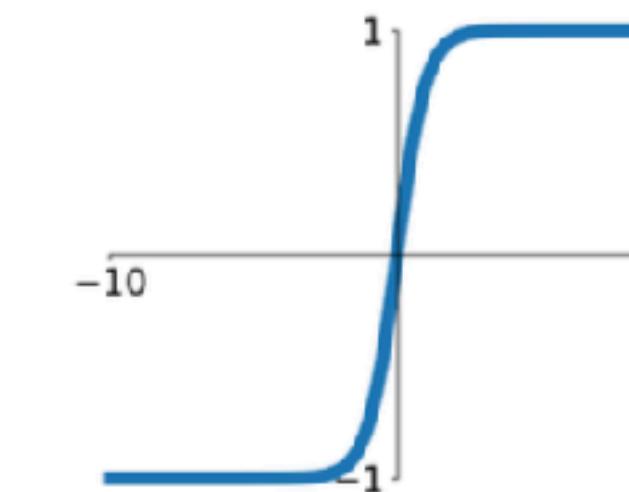
### Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



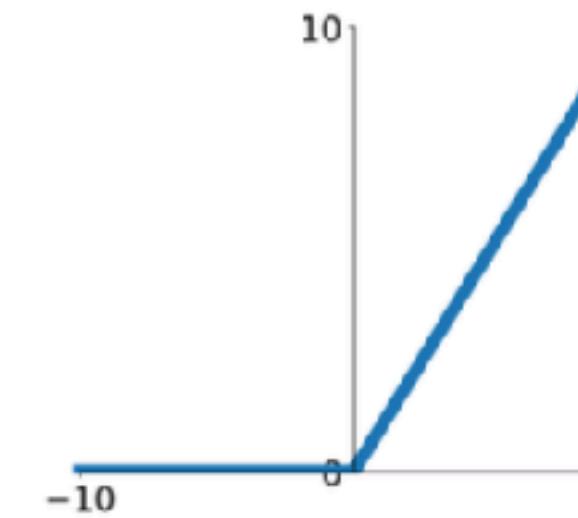
### tanh

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$



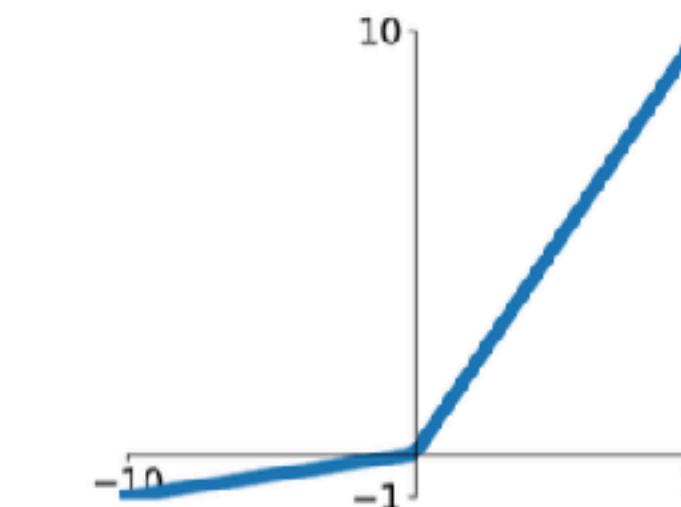
### ReLU

$$\max(0, x)$$



### Leaky ReLU

$$\max(0.2x, x)$$

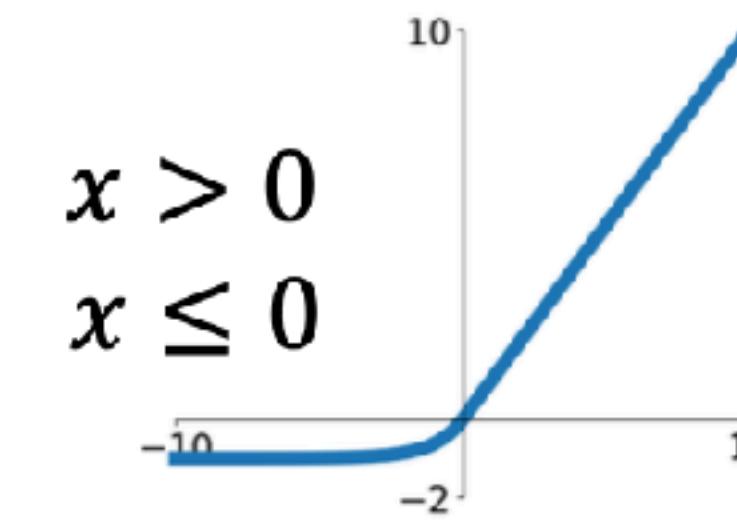


### Softplus

$$\log(1 + \exp(x))$$

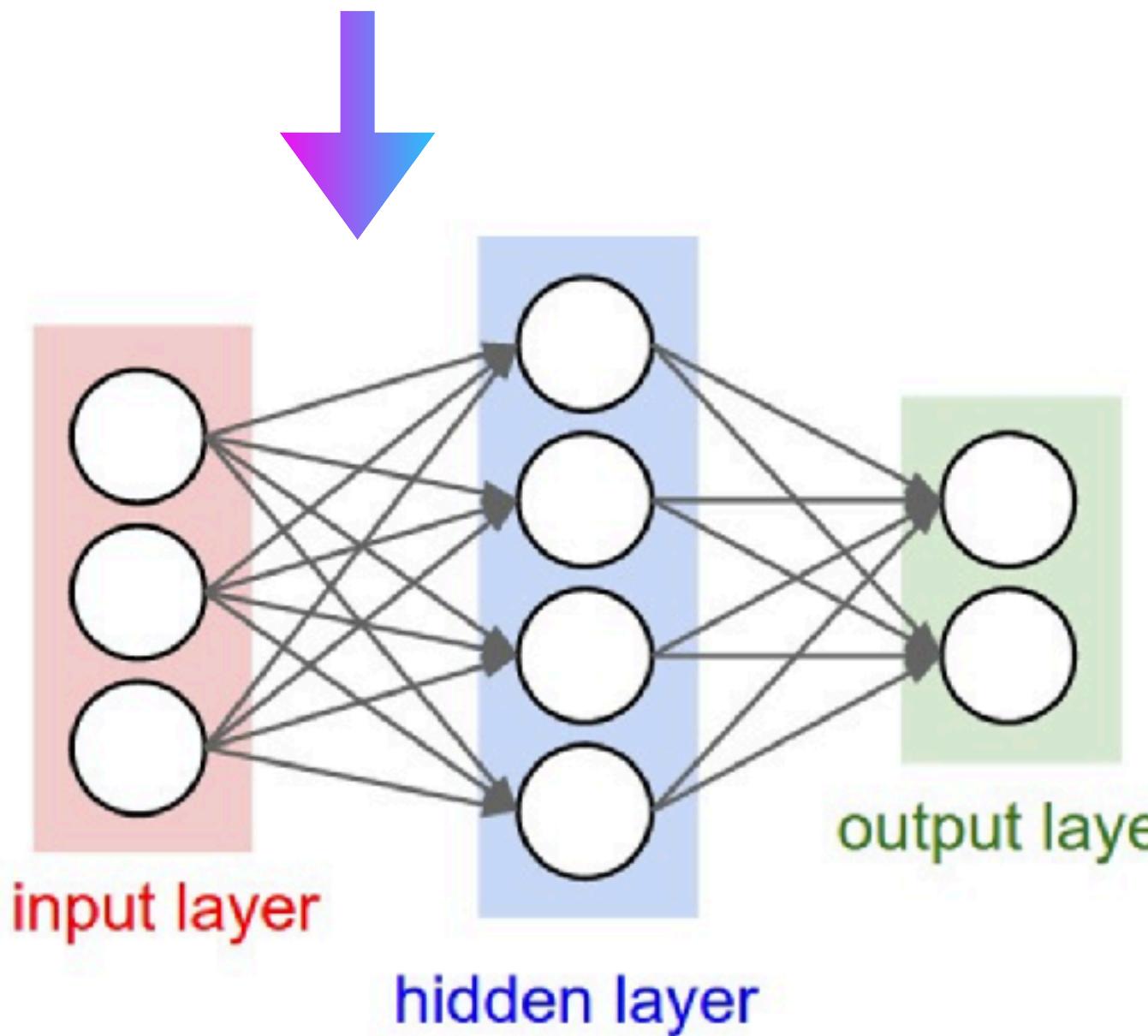
### ELU

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(\exp(x) - 1), & x \leq 0 \end{cases}$$



## 2. Deep Learning 이론 FeedForward function.

What is Feed Forward Function

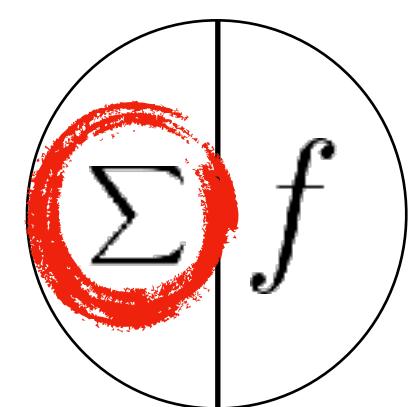


$$z_m^{[l]} = w_{m1}x_1 + w_{m2}x_2 + w_{m3}x_3$$

$$= \sum_{d=1}^D w_{md}x_d$$

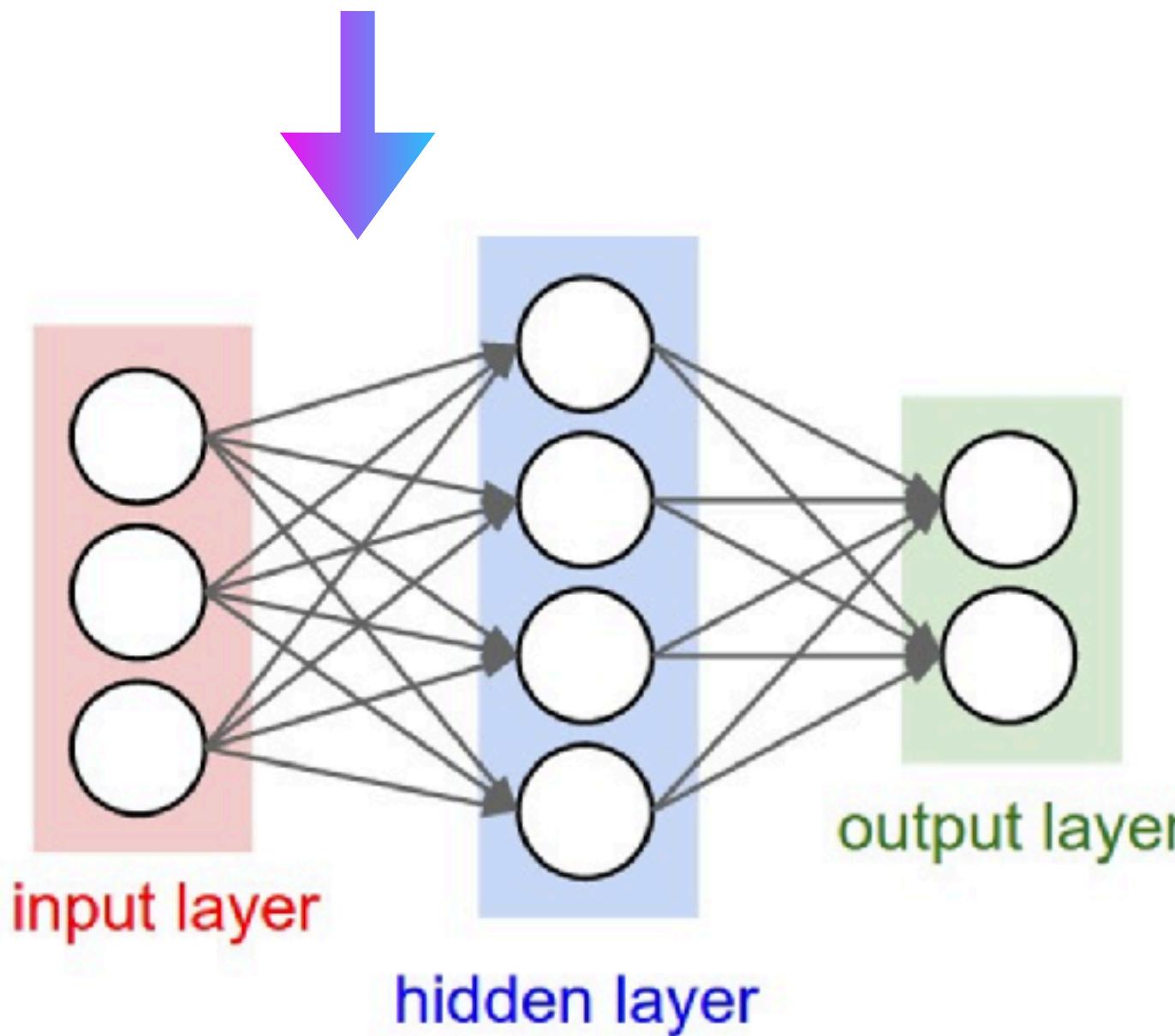
$$= \mathbf{w}_m^{[l]} \mathbf{x}$$

$$\mathbf{w}^{[l]} \in \mathbb{R}^{M \times D}, \mathbf{x} \in \mathbb{R}^D$$



## 2. Deep Learning 이론 FeedForward function.

What is Feed Forward Function

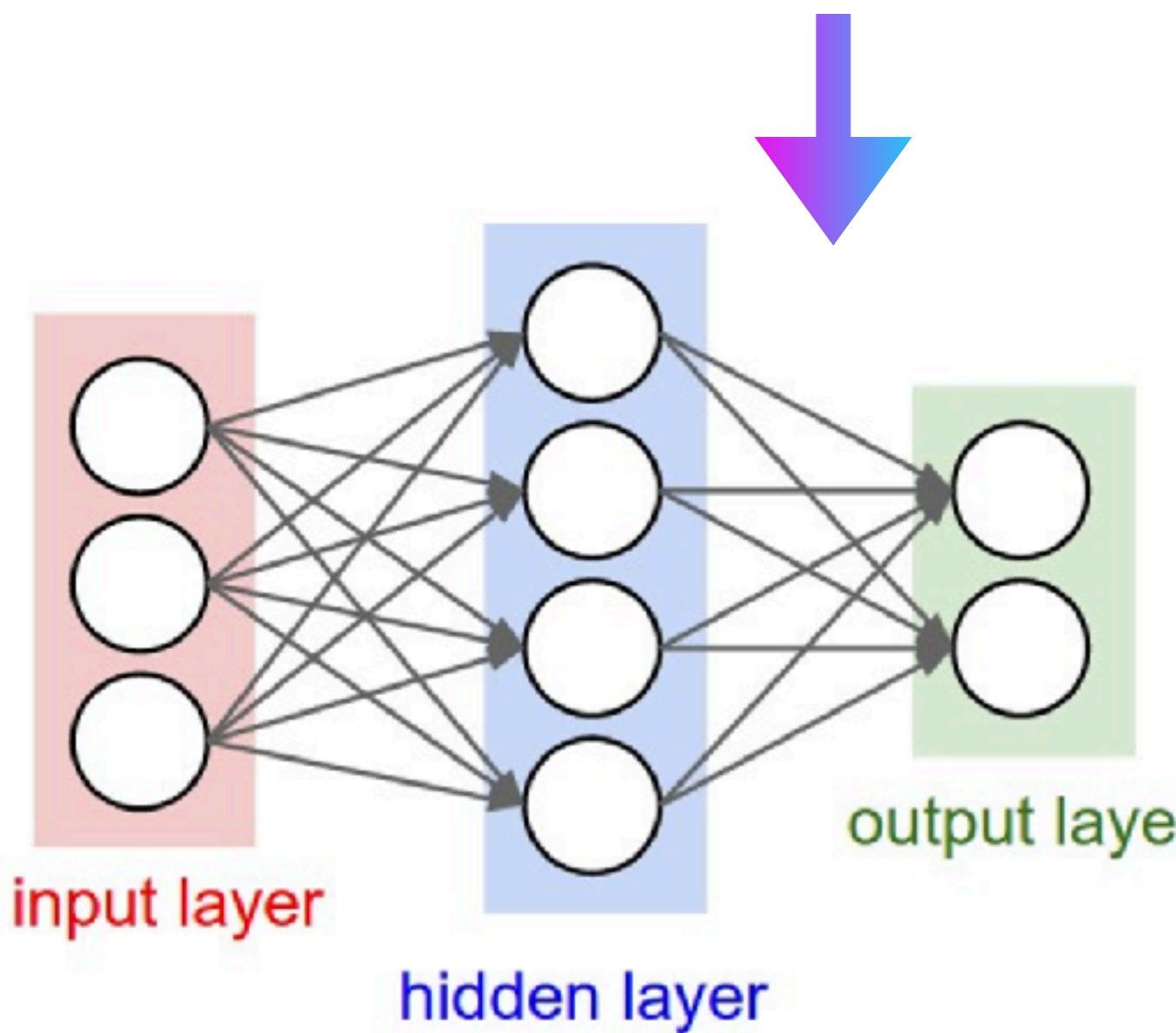


$$a_m^{[l]} = f(z_m^{[l]})$$

A circular diagram representing a single neuron's computation. Inside the circle, there is a summation symbol ( $\Sigma$ ) on the left and a function symbol ( $f$ ) on the right. A red circle highlights the function symbol ( $f$ ).

## 2. Deep Learning 이론 FeedForward function.

What is Feed Forward Function



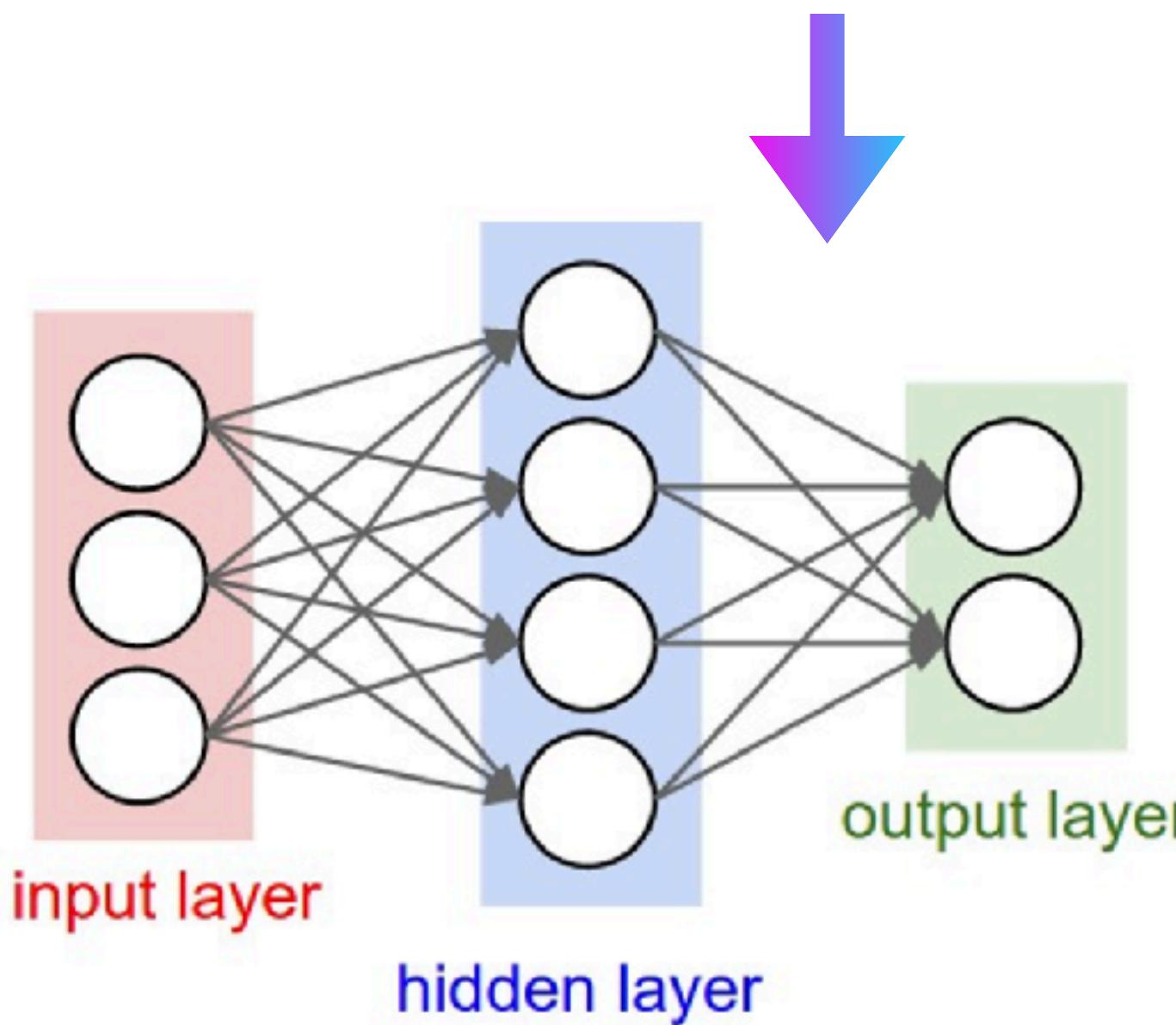
$$z_k^{[l+1]} = w_{k1}a_1^{[l]} + w_{k2}a_2^{[l]} + w_{k3}a_3^{[l]} + w_{k4}a_4^{[l]}$$

$$\mathbf{W}^{[l+1]} \in \mathbb{R}^{K \times M}, \mathbf{A}^{[l]} \in \mathbb{R}^M$$

A circular diagram representing a single neuron's computation. It is divided vertically. The left half contains the summation symbol ( $\sum$ ) with a red circle around it, and the right half contains the activation function symbol ( $f$ ).

## 2. Deep Learning 이론 FeedForward function.

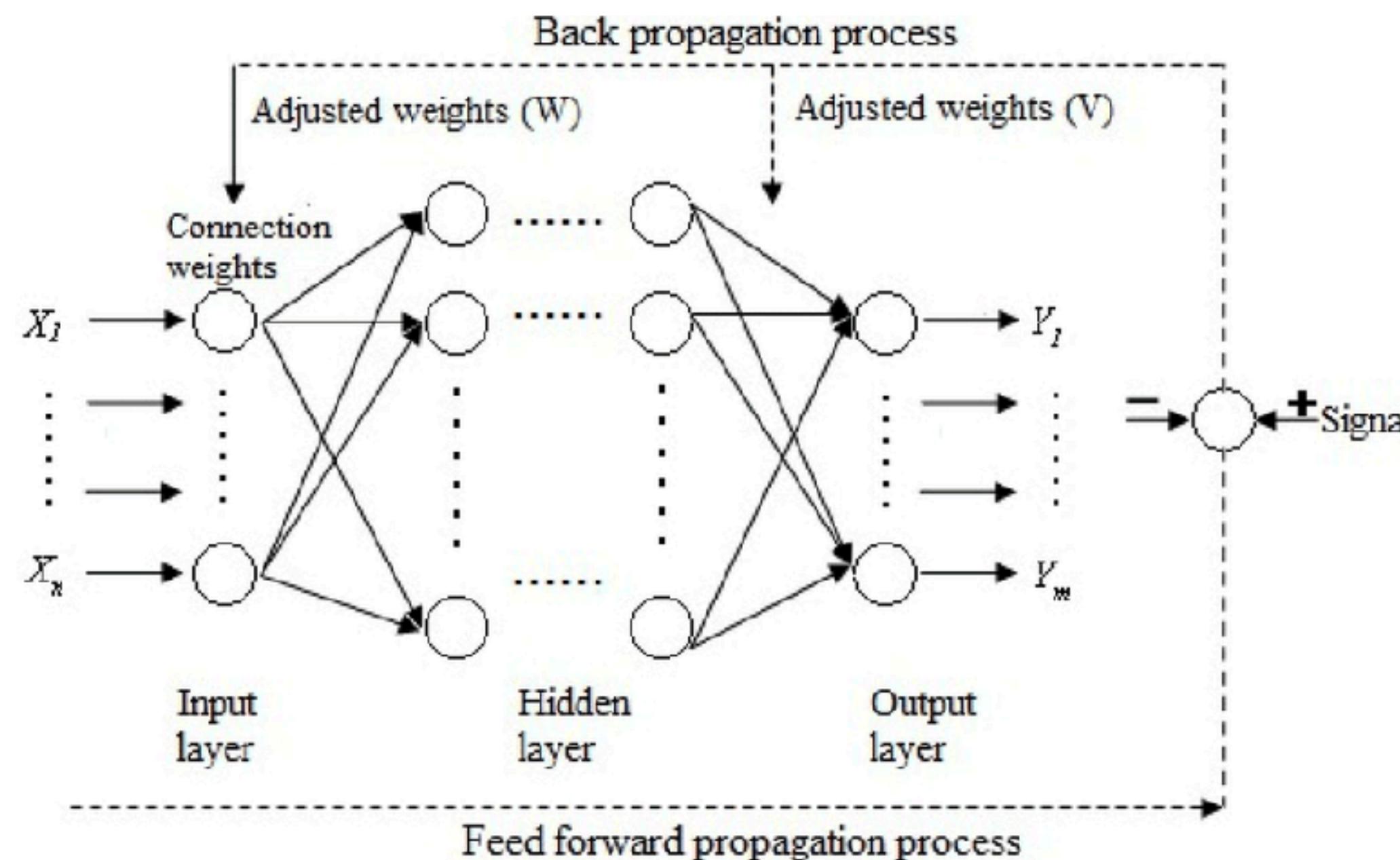
What is Feed Forward Function



$$a_k^{[l+1]} = f(z_k^{[l+1]})$$
$$y_k = a_k^{[l+1]}$$

A diagram of a single neuron's activation function. It is represented by a circle divided into four quadrants. The bottom-right quadrant contains the letter 'f' in red, with a red circle around it, indicating the activation function. The other three quadrants are empty.

## 2. Deep Learning 이론 Network Training.



Iteration

Model output

Loss function

Weight update

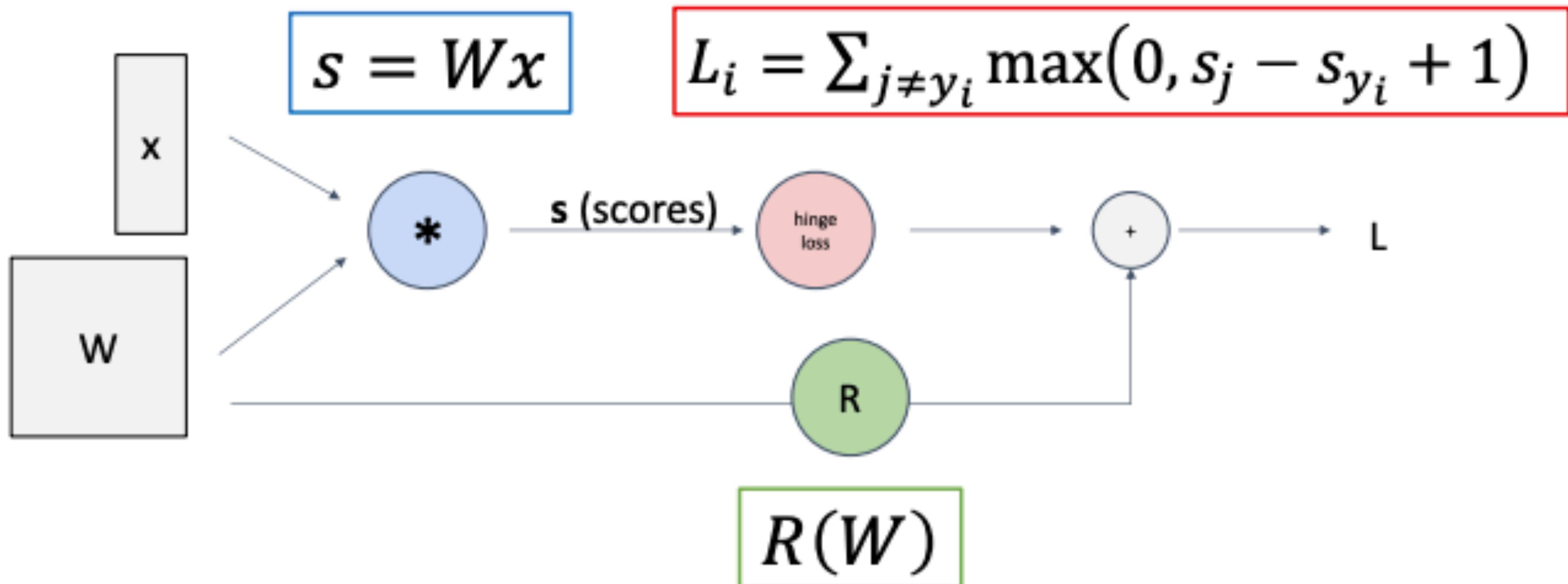
$$\hat{y}(x) = f(\mathbf{W}^T \mathbf{X} + w_0)$$

$f(\cdot)$  : nonlinear function (activation function)

$$E(W) = \frac{1}{2} \sum_{n=1}^N \{t_n - \hat{y}(x)\}^2$$

$$w^{(t+1)} = w^{(t)} - lr \nabla E(W)$$

## Better Idea: Computational Graphs



## Backpropagation: Simple Example

$$f(x, y, z) = (x + y) \cdot z$$

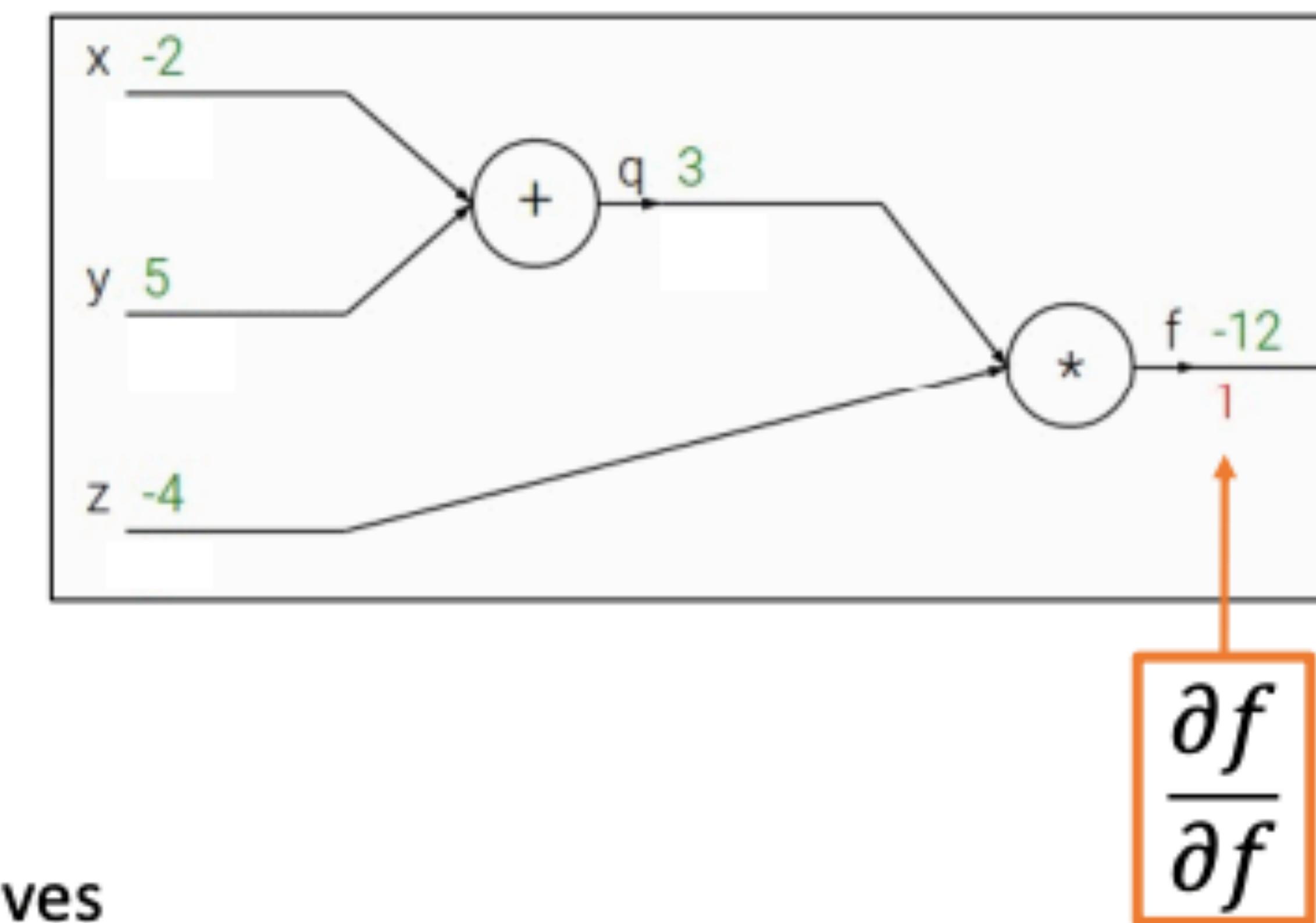
e.g.  $x = -2, y = 5, z = -4$

**1. Forward pass:** Compute outputs

$$q = x + y \quad f = q \cdot z$$

**2. Backward pass:** Compute derivatives

Want:  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



## Backpropagation: Simple Example

$$f(x, y, z) = (x + y) \cdot z$$

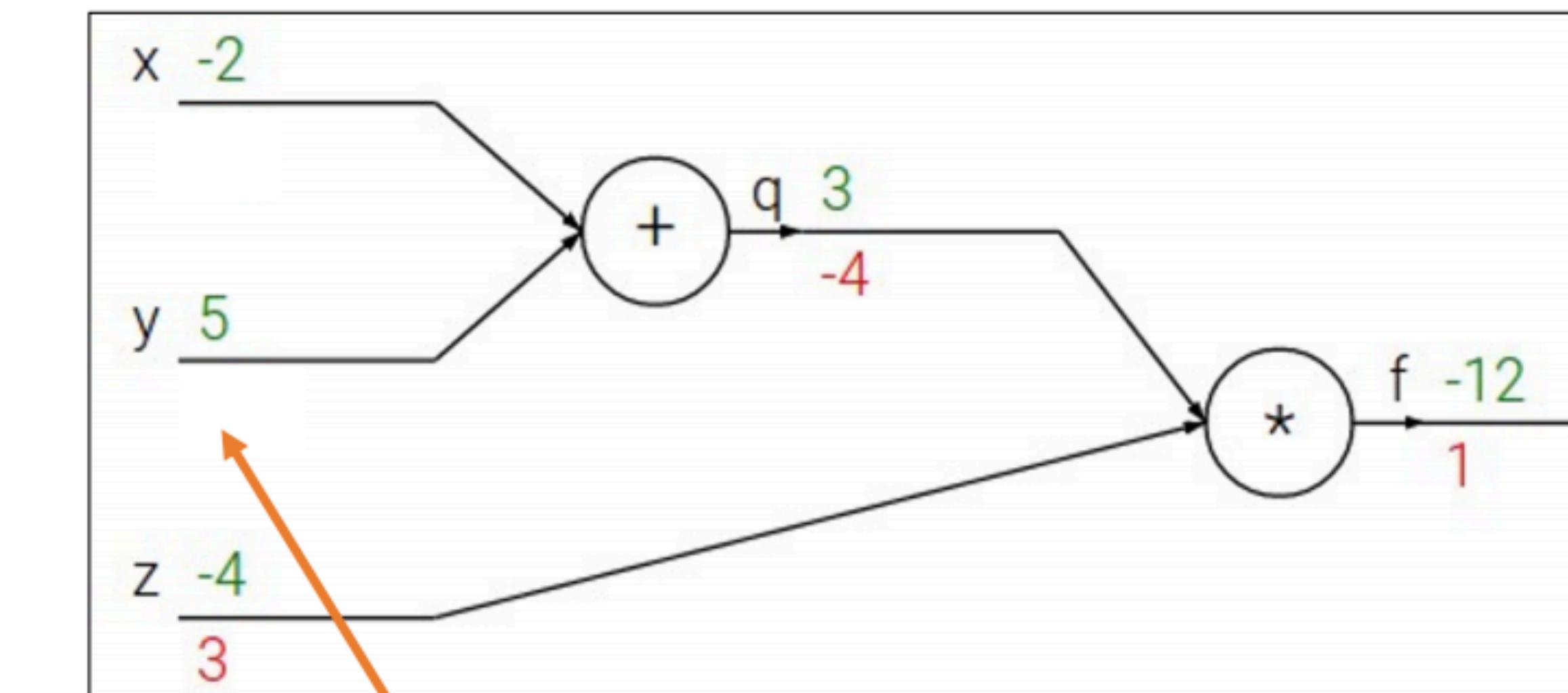
e.g.  $x = -2$ ,  $y = 5$ ,  $z = -4$

**1. Forward pass:** Compute outputs

$$q = x + y \quad f = q \cdot z$$

**2. Backward pass:** Compute derivatives

Want:  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



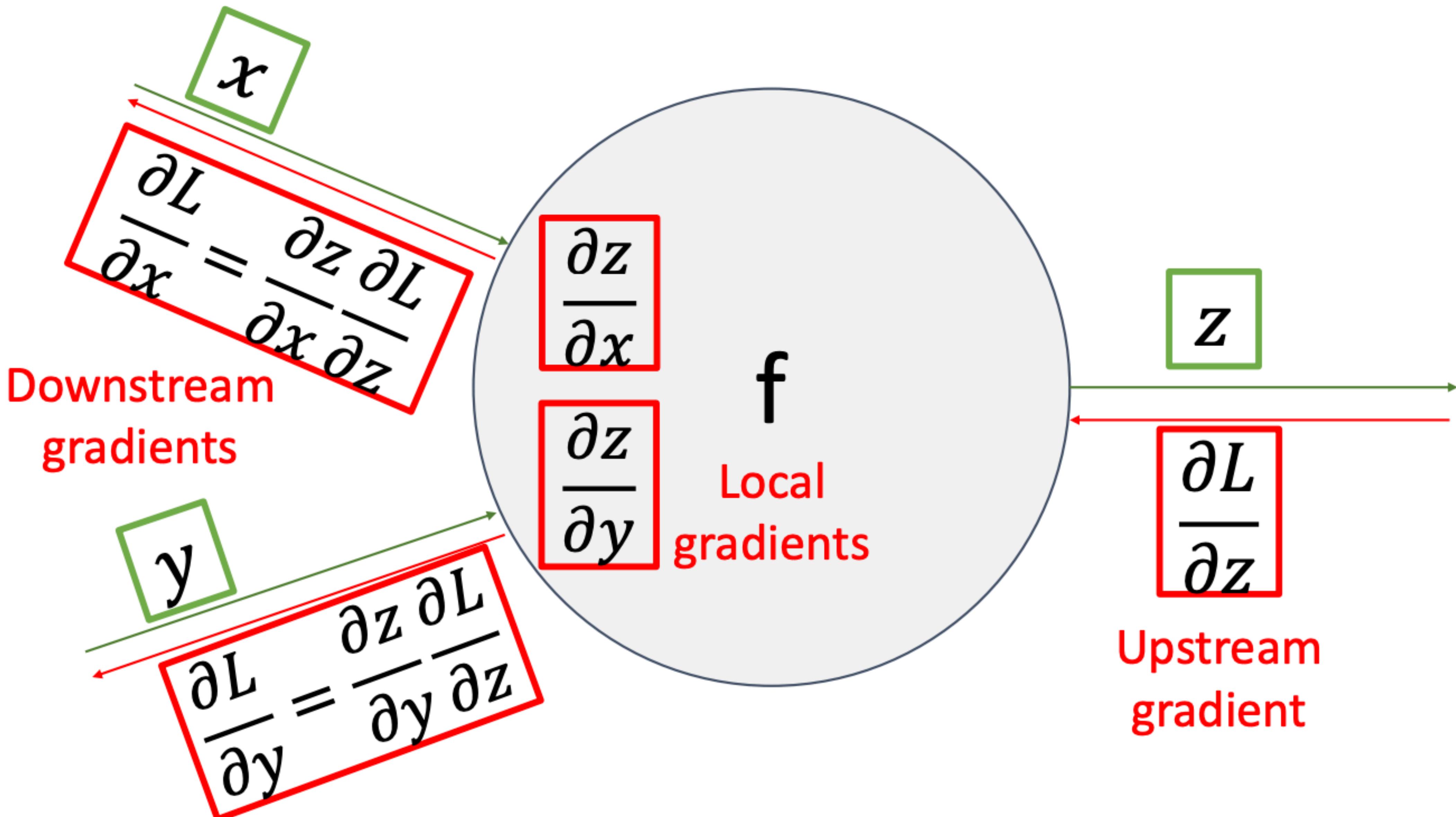
**Chain Rule**

$$\frac{\partial f}{\partial y} = \frac{\partial q}{\partial y} \frac{\partial f}{\partial q}$$

Downstream Gradient      Local Gradient      Upstream Gradient

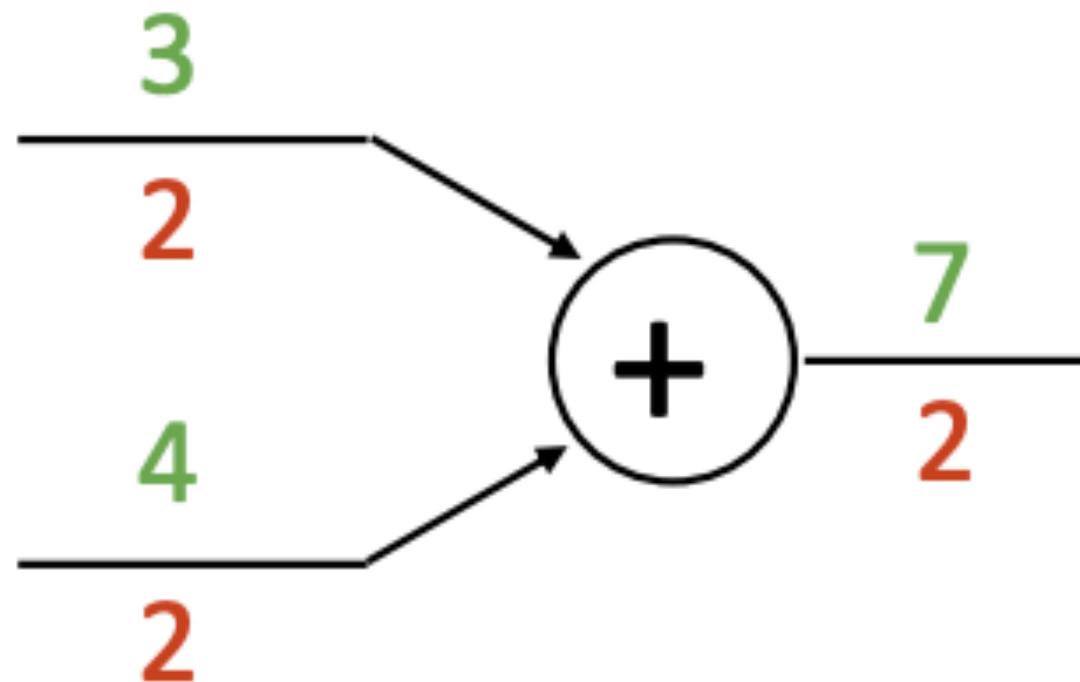
$$\frac{\partial q}{\partial y} = 1$$

2. Deep Learning 이론  
Error Backpropagation.

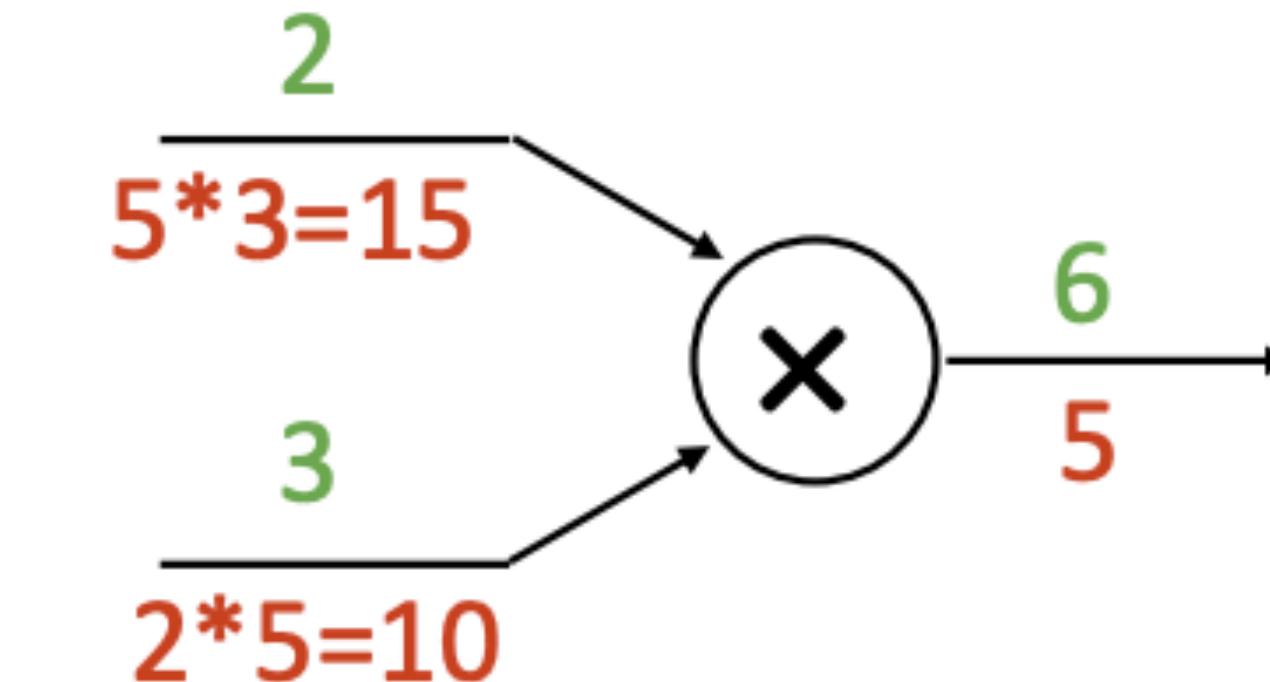


# Patterns in Gradient Flow

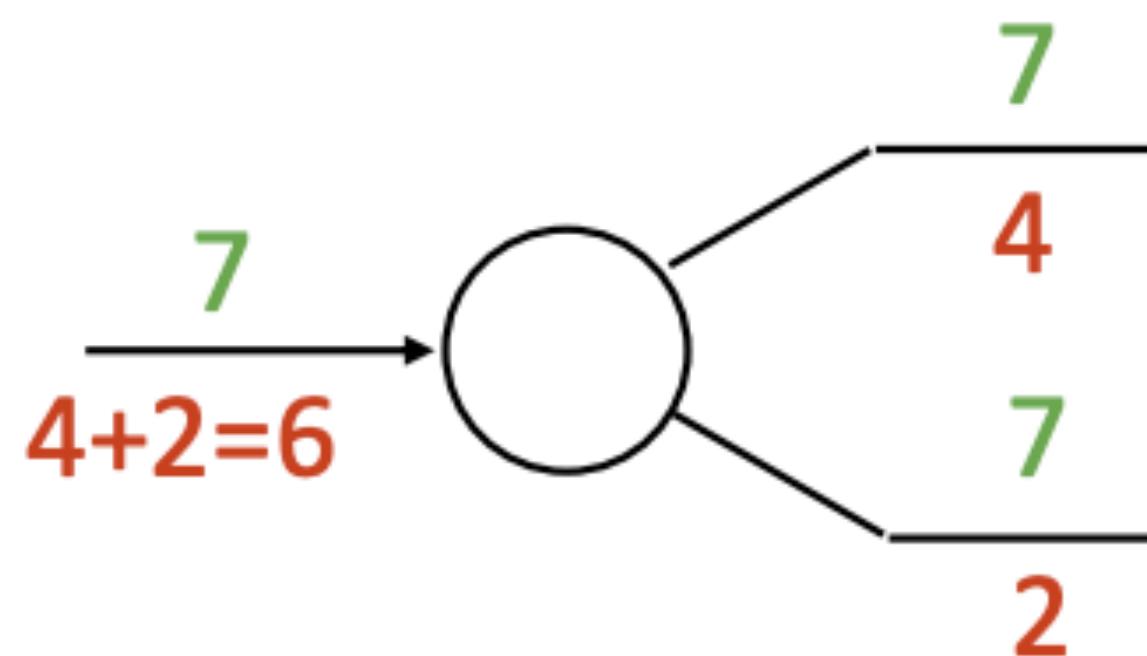
**add gate: gradient distributor**



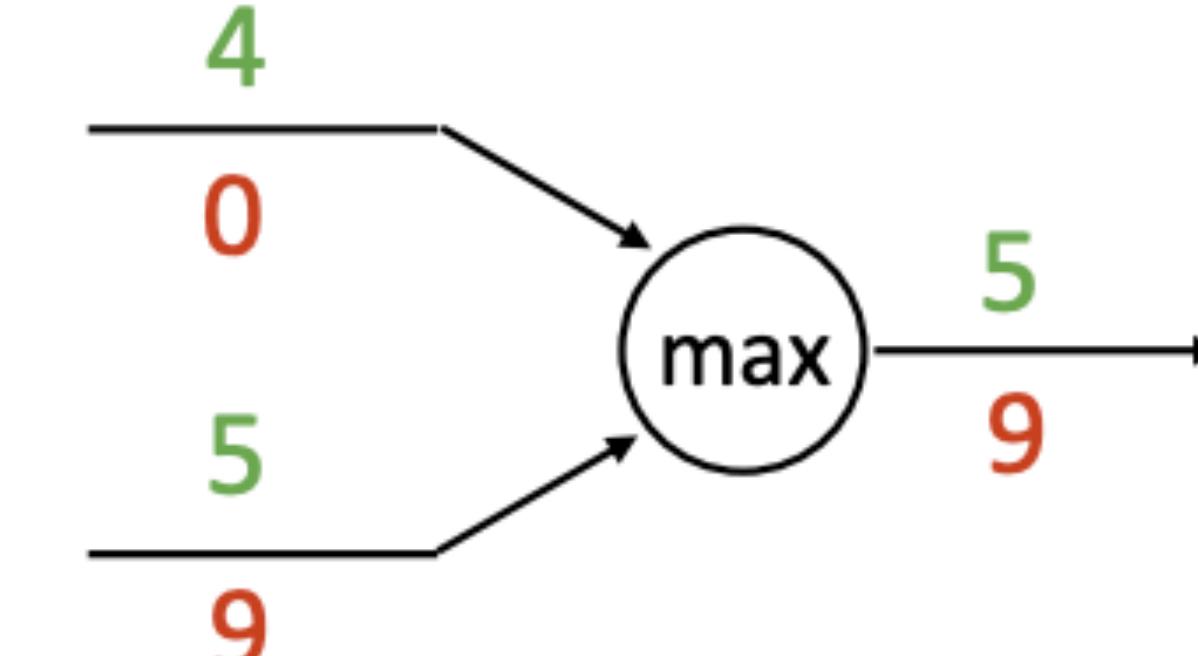
**mul gate: “swap multiplier”**



**copy gate: gradient adder**



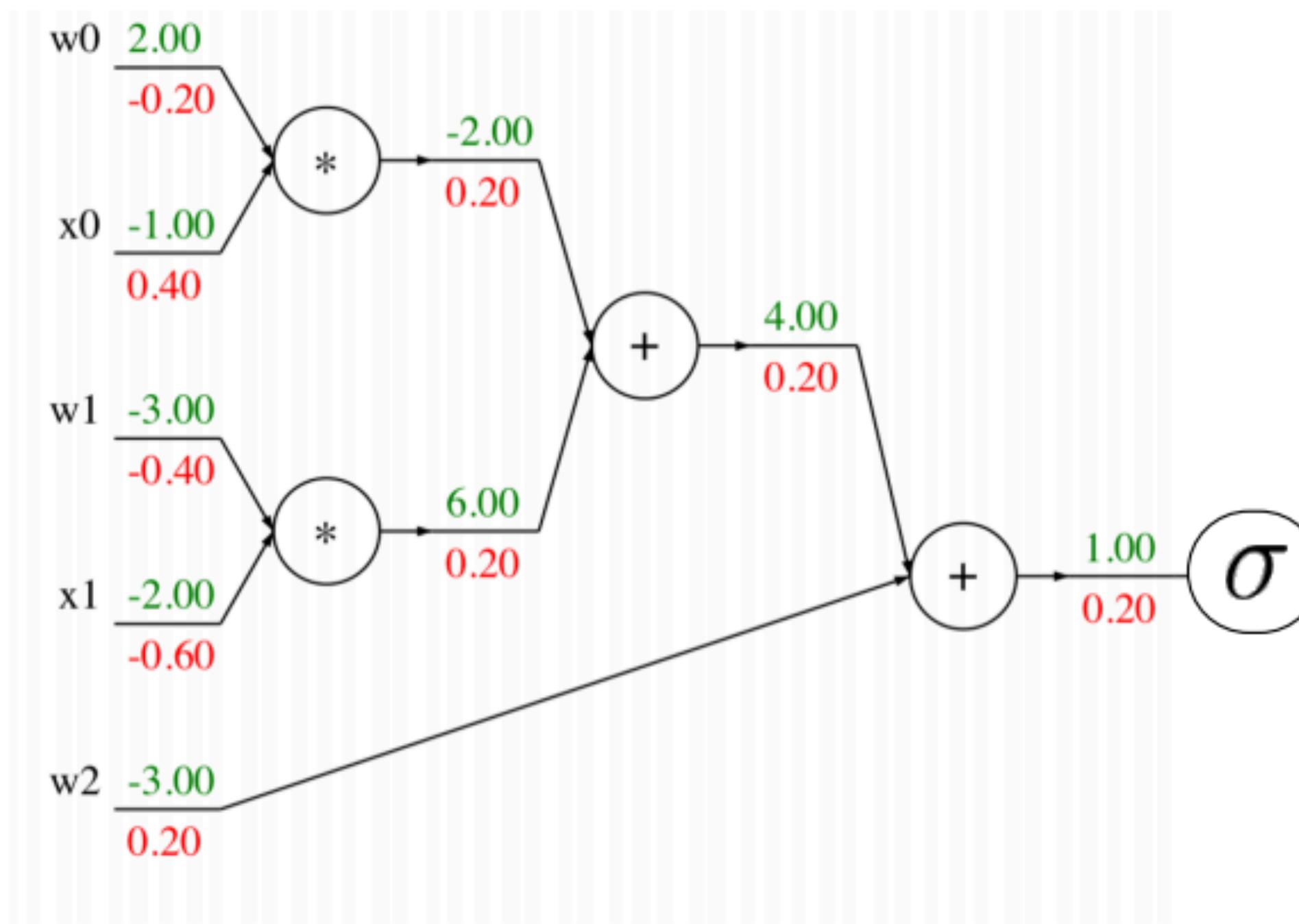
**max gate: gradient router**



## 2. Deep Learning 이론 Error Backpropagation.

### Backprop Implementation: "Flat" gradient code:

Forward pass:  
Compute output



Backward pass:  
Compute grads

```
def f(w0, x0, w1, x1, w2):  
    s0 = w0 * x0  
    s1 = w1 * x1  
    s2 = s0 + s1  
    s3 = s2 + w2  
    L = sigmoid(s3)
```

```
grad_L = 1.0  
grad_s3 = grad_L * (1 - L) * L  
grad_w2 = grad_s3  
grad_s2 = grad_s3  
grad_s0 = grad_s2  
grad_s1 = grad_s2  
grad_w1 = grad_s1 * x1  
grad_x1 = grad_s1 * w1  
grad_w0 = grad_s0 * x0  
grad_x0 = grad_s0 * w0
```

### 3. Numpy와 Matplotlib 실습

# 실습

<https://colab.research.google.com/drive/1jcv9qZ-uBZS6v6vbWejhMxxp0aOMpv9g?usp=sharing>

# 실습

[https://colab.research.google.com/drive/1tOYIueHGU\\_FCDeQLSbAkHZOROfD3-Npi?usp=sharing](https://colab.research.google.com/drive/1tOYIueHGU_FCDeQLSbAkHZOROfD3-Npi?usp=sharing)

# 실습