



VLSI & System Lab.

# 딥러닝 프레임워크 기반 인공지능의 이해

한양대 IDEC 강좌 – Day1

숙명여자대학교 ICT융합공학부 전자공학과 최웅

# Outline

---

## □ 인공지능 관련 동향

## □ 딥러닝의 기초

- 용어 및 분류(Classification)/회기(Regression) 모델
- 퍼셉트론 (Perceptron) 및 MNIST Example

## □ 학습 및 추론 알고리즘의 이해

- Backpropagation
- Challenge of Learning
- Optimizer of Learning

## □ 딥러닝 가속기

- CNN 및 인공 신경망 가속기 연구 동향

# Outline

---

## □ 인공지능 관련 동향

## □ 딥러닝의 기초

- 용어 및 분류(Classification)/회기(Regression) 모델
- 퍼셉트론 (Perceptron) 및 MNIST Example

## □ 학습 및 추론 알고리즘의 이해

- Backpropagation
- Challenge of Learning
- Optimizer of Learning

## □ 딥러닝 가속기

- CNN 및 인공 신경망 가속기 연구 동향

# 산업계 동향

## □ 선도 업체들의 동향 : 자체 칩 개발을 진행

### 세트 업체의 독자 칩 설계

세트 업체들은 제품 차별화 및 세상에 없던 제품 출시를 목적으로 자체 칩 확보를 활발히 추진 중

- 시스템 반도체는 제품 성능을 좌우하는 핵심 역할을 수행
- 예 : TV 화질/8K, 모바일 그래픽 성능/인공지능

TV 세트 업체 중 분야 글로벌 선도 업체는 독자 화질 엔진을 탑재한 자체 칩 솔루션 확보 (LG/삼성)

스마트 폰 분야 글로벌 선도 업체는 독자 AP 솔루션 확보 (애플/삼성/화웨이/샤오미)

- 화웨이는 자회사인 하이실리콘을 통해 인공지능 엔진 및 카메라 성능 개선한 독자 AP 개발

### 세트 업체의 시장 지배력 강화

- 칩 개발 역량 + 마케팅 소구점 → 시장 지배력 강화

### 서비스 업체의 독자 칩 설계

인공지능과 5G 기술이 결합, 스마트 홈/City/Factory/Farm 등으로 발전하며, 인공지능 칩 시장은 급격한 성장이 예상됨

- Connected Things의 개수가 '18년 179억개에서 '20년 500억개로 증가
- 인공지능이 클라우드에서 Edge 기기로 확대되며, 칩은 연평균 53% 성장하여 '21년 5조 규모 시장 예상됨

구글/아마존 등 서비스 업체들은 새로운 사용자 접점에서의 데이터 확보를 위해 독자 인공지능 칩 개발을 진행중

- 구글은 클라우드용 칩 TPU 개발에 이어 Edge 기기용 TPU까지 개발을 진행하여 IoT 및 스마트 홈 시장 확장을 추진함
- 아마존/마이크로소프트/페이스북 등도 이런 경쟁에 가세 (구글 : '10년 Agnilux 인수, 페이스북 : '18년 자체 칩 설계팀 구성, 아마존 '15년 Annapurna Labs 인수)

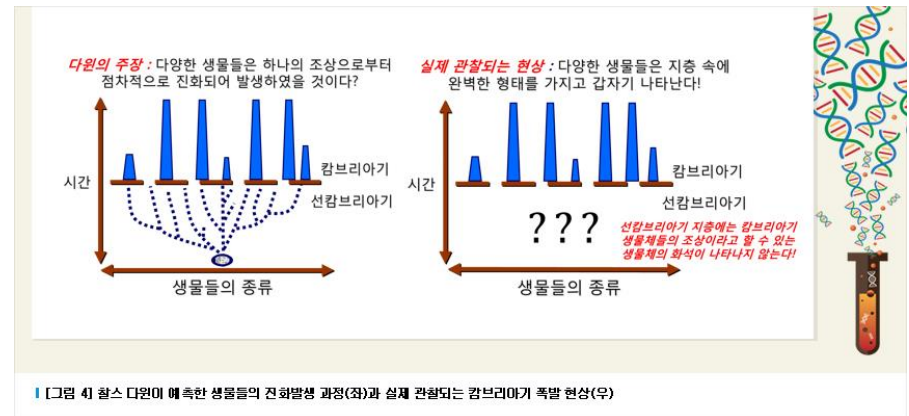
# 생명체와 인공지능

## □ 생명체

- 오너: 유전자, 오퍼레이터: 뇌, 디텍션: 신경계
- 생명체 (유전자)는 살아남기 위해 (종족번식) 신경계와 뇌를 발전시킴
- 뇌는 태어나서 죽을 때까지 '학습'
  - 감각 수용기의 70%가 눈에 밀집, 시각 인지에 뇌의 50% 관여, 장면 인식에 10ms 소요
- 캄브리아기 대폭발 ('눈'의 탄생) : 현존하는 거의 모든 종의 원형이 등장

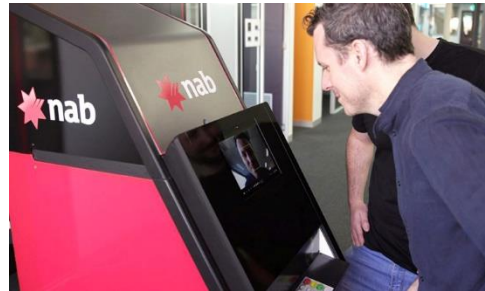
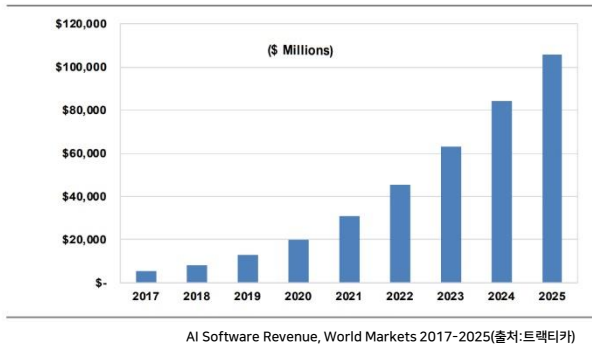


■ [그림 1] 지질주상도에 사용되는 진화론적인 지층 연대 순서

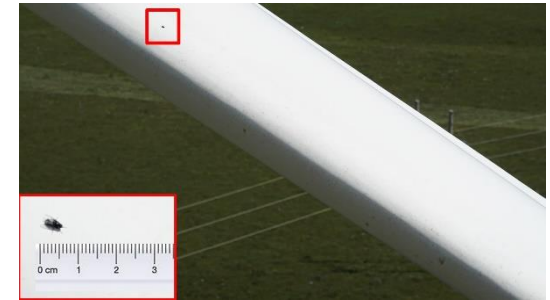


# 인공지능의 영향력

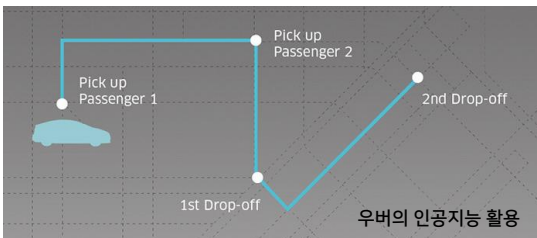
## □ 다양한 산업에서 인공지능(AI)의 현재와 미래



얼굴이 곧 카드가 되어 간편하게 입출금이 가능한 서비스  
(국립 호주은행)



풍력발전 점검 : 5명의 인력이 6시간 동안 위험을 무릅쓰고 작업을 했지만, 하나의 드론으로 25분 만에 모든 프로세스를 완료

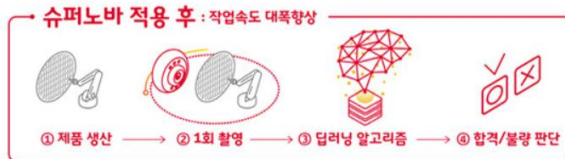
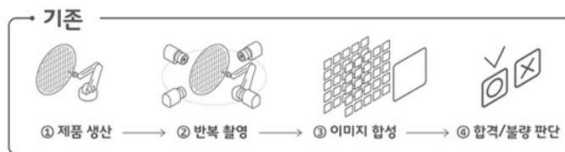


109년 전통의 로레알, 신제품 (18개월 → 6개월), AI 기반 스타 일리스트도 개발중

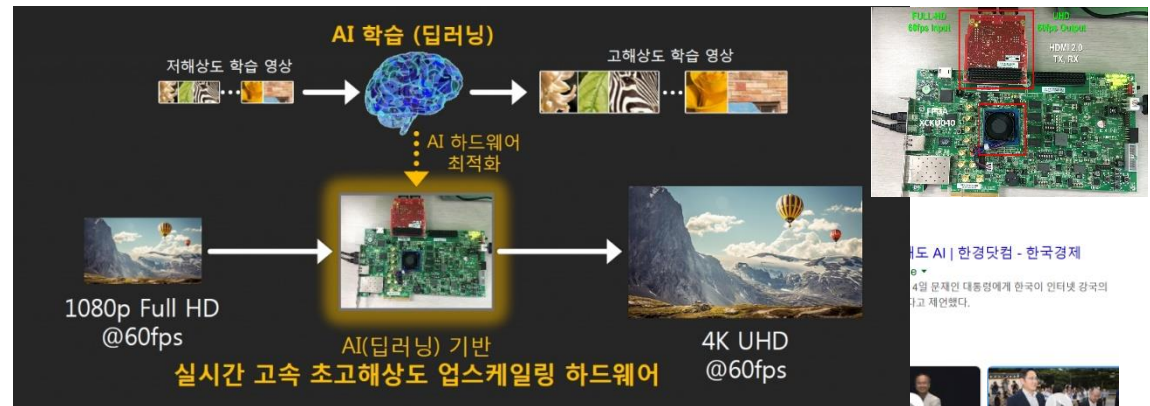


# 인공지능의 영향력

## □ 다양한 산업에서 인공지능(AI)의 현재와 미래



SK텔레콤은 지난해 10월 SK하이닉스 반도체 품질평가(QA) 공정에 슈퍼노바를 시범 적용



3월 19일(현지시각) 마르세유에 모인 노란 조끼 시위자들. AFP연합뉴스 정치인만큼은 인공지능(AI)이 대체했으면 좋겠다는 여론이 생겨나고 있다

무디스 “힐러리 당선”...모그니는 “트럼프가 승자” 엇갈린 예측 ...  
[news.kmib.co.kr > article > view](https://news.kmib.co.kr/article/view)

미국 대선 또다른 승자는 AI...'04년부터 매년 예상 적중 - 중앙일보  
[https://news.joins.com > article](https://news.joins.com/article)



손정의가 "AI에 집중하라" 강조하는 이유 - 주간조선  
[weeklychosun.com > client > news > view](https://www.weeklychosun.com/client/news/view)  
 2019. 7. 15. - 손정의 회장의 소프트뱅크는 2016년 7월 영국의 반도체 설계기업인 ARM을 인수하면서 본격적으로 AI 시대의 준비에 나섰다. photo 조선일보 ...

손정의 "AI 지배하는 자가 미래 지배" - 헬로디디  
<https://www.hellodd.com> > ...  
 2018. 7. 23. - 손정의 회장의 강연 메시지를 요약하면 미래는 AI를 거머쥔 자가 지배한다는 것이다. 특이점에 의해 AI는 인류 사상 최대의 역량이 될 것이고, 그것은 ...

손정의 "한국, 첫째도 둘째도 셋째도 AI에 집중해야" - 매일경제  
[https://www.mk.co.kr > news > economy > view > 2019/07](https://www.mk.co.kr/news/economy/view/2019/07)  
 2019. 7. 4. - 세계적인 혁신 기업가인 손정의 소프트뱅크 회장은 4일 "일본과 한국 모두 중국, 미국에 비해서 AI(인공지능) 대응이 늦었다"며 "한국이 인터넷 광국 ...

# Outline

---

## □ 인공지능 관련 동향

## □ 딥러닝의 기초

- 용어 및 분류(Classification)/회기(Regression) 모델
- 퍼셉트론 (Perceptron) 및 MNIST Example

## □ 학습 및 추론 알고리즘의 이해

- Backpropagation
- Challenge of Learning
- Optimizer of Learning

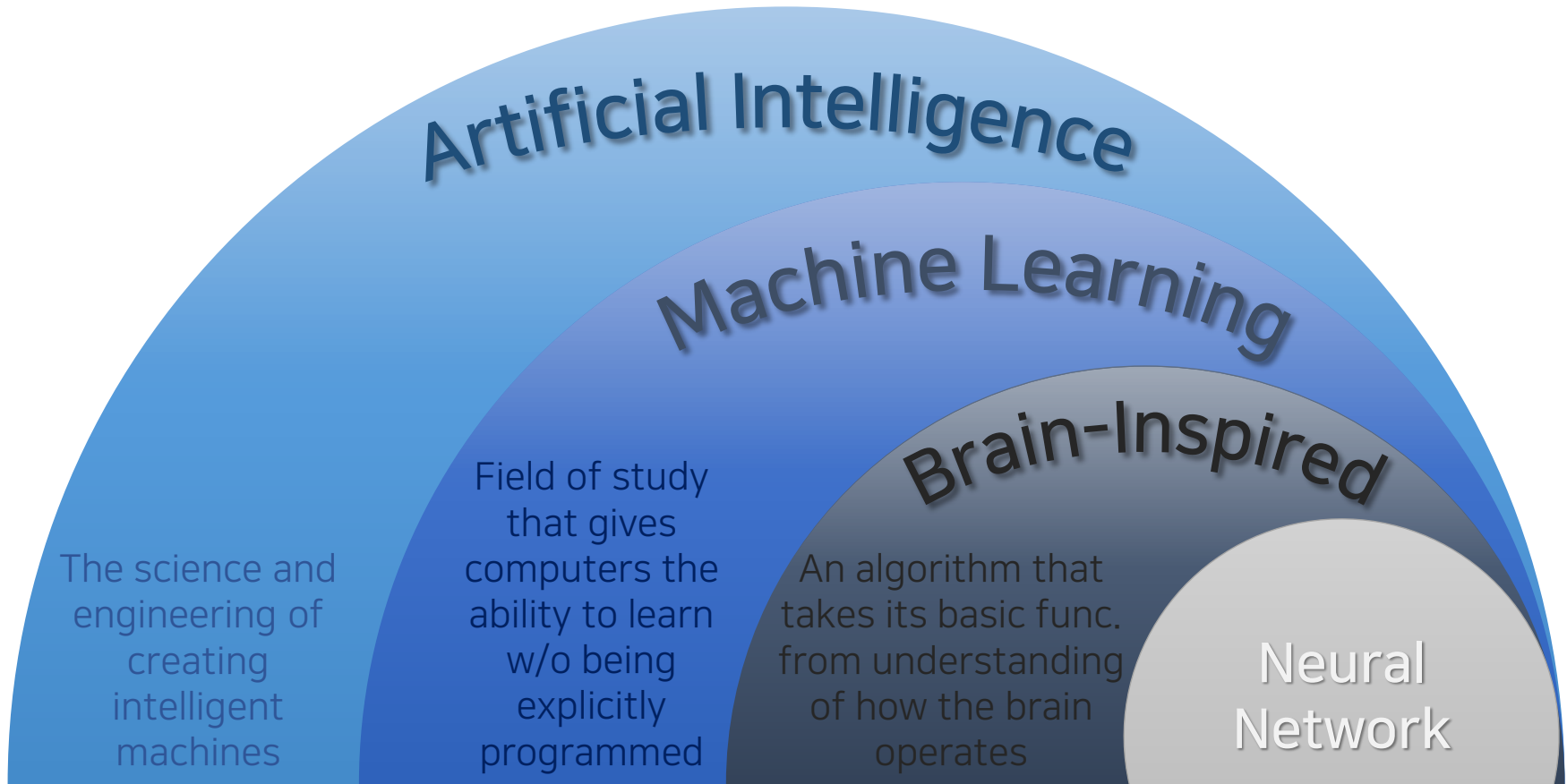
## □ 딥러닝 가속기

- CNN 및 인공 신경망 가속기 연구 동향



# AI & Neural Network

---



# Deep (Artificial) Neural Network

## ❑ Deep Neural Network (심층 인공 신경망)

### Available Big Data



350M Images / day

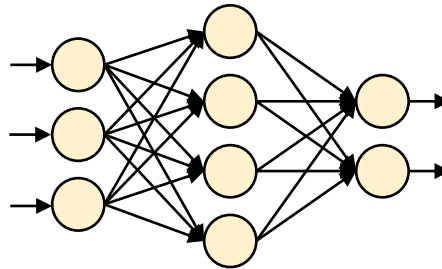


300 hours videos /1 min

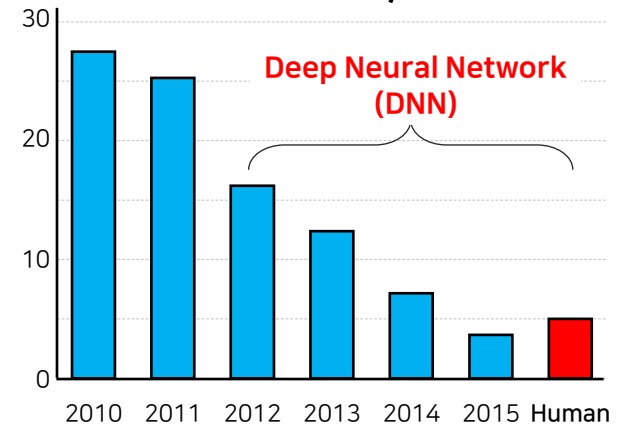
### GPU Acceleration



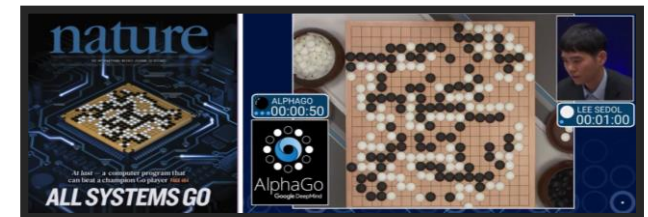
### New ML Techniques



### Top-5 Image Classification Accuracy

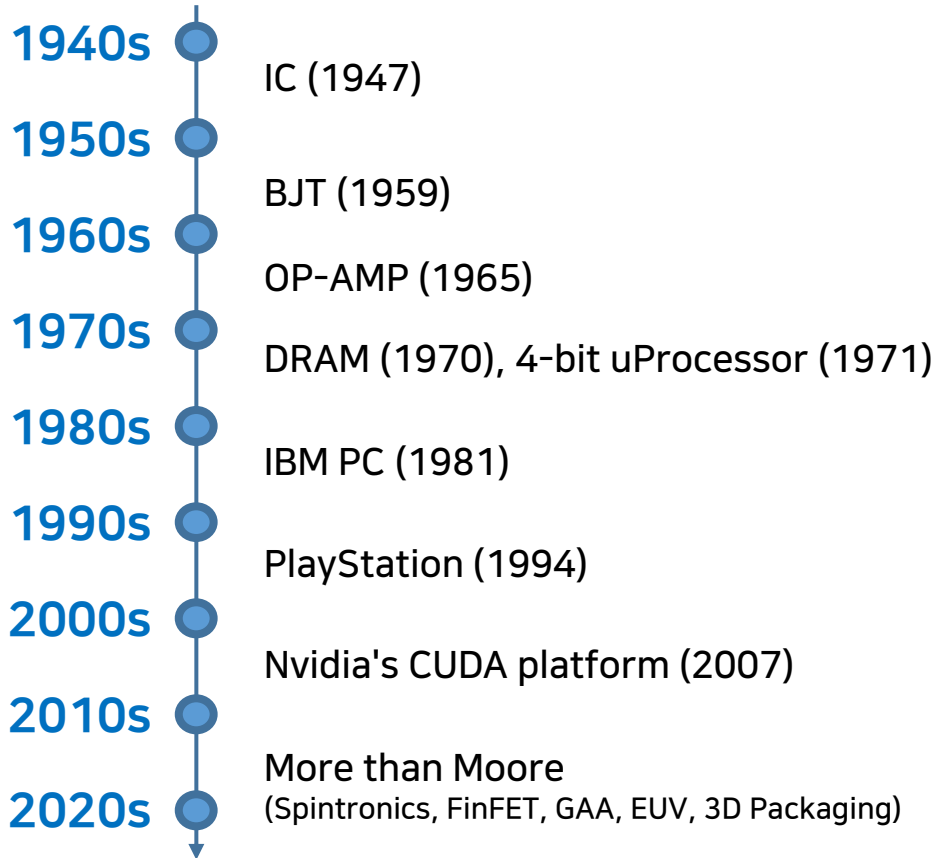


[ Russakovsky et al., IJCV 2015 ]



# AI & Hardware Resource

## IC (Integrated Circuit)



## AI (Artificial Intelligence)

수학자: 계산 기계와 지능 (1950s) → 폰 노이만

신경외과의사+논리학자: 인공신경망 (1950s)  
→ 퍼셉트론 (Perceptron, 1958)

1<sup>st</sup> AI Winter (1974-1980)  
XOR Problem w/ 퍼셉트론 (1969) → DARPA 펀딩 중단

기업의 R&D - 통계기술 및 데이터 (1970s)  
→ 베이즈 (Bayes) 기반 확률적 방법, Fuzzy 이론

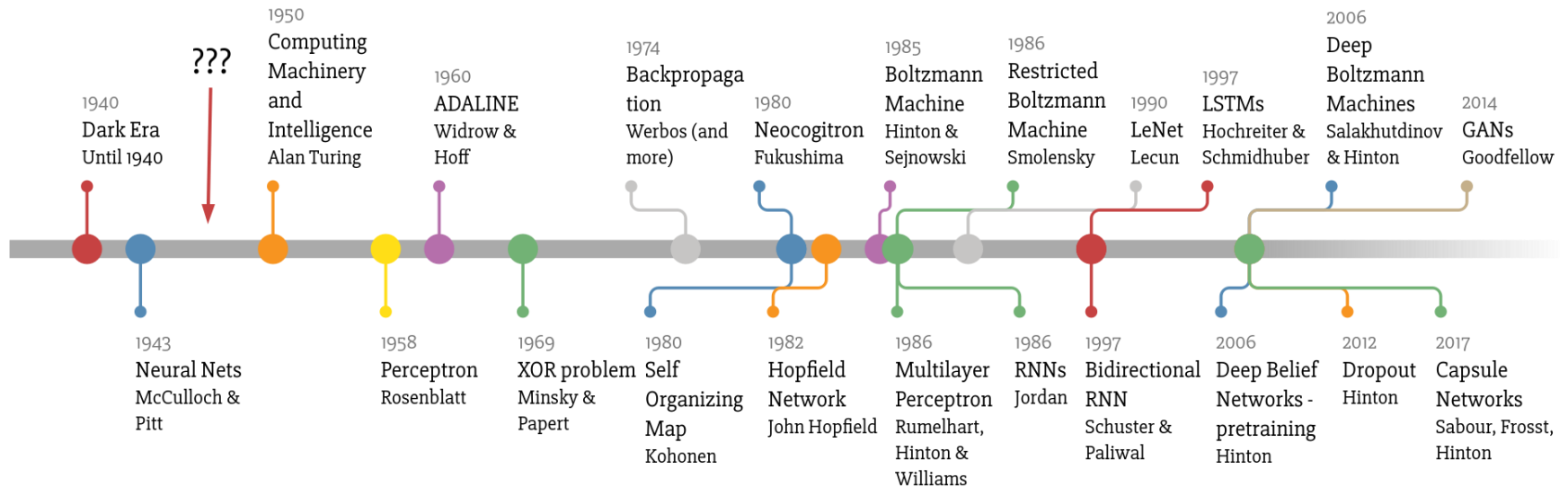
2<sup>nd</sup> AI Winter (1987-1993)  
→ 대부분의 연구가 슈퍼컴퓨터 & 시뮬레이션으로 전환  
→ But, 강화학습, 역전파 (Backpropagation) 등 개발

심층신경망의 가능성 확인 (2000s)  
→ 힌트교수, CNN, MNIST, LSTM, ImageNet

Big Data, GPU, 알고리즘 (2010s)  
→ AlexNet, Dropout, AlphaGo/Zero, BERT

# More Detail AI History

## Deep Learning Timeline



Made by Favio Vázquez

# Terminology

---

## □ Key Terminology

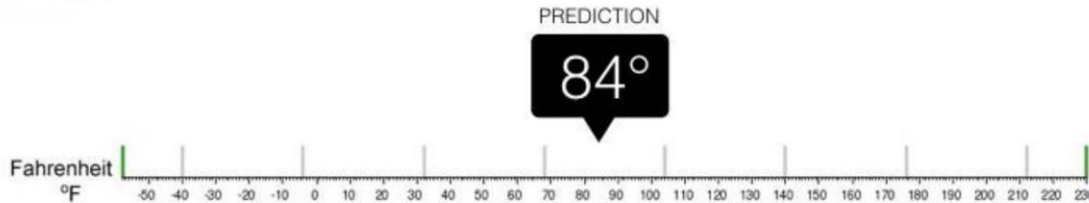
- 라벨 (Labels)
  - 예측하는 항목 (변수  $y$ ) : 부동산 향후 가격, 사진 속 동물의 종류 등등
- 특성 (Features)
  - 입력 변수 (변수  $x$ ) : 스팸 감지 예 - 이메일의 단어, 보낸 사람 주소 및 시간
- 모델 (Models)
  - 모델은 특성과 라벨의 관계를 정의
  - 학습 (Training) : 모델을 만들거나 배우는 것을 의미
  - 추론 (Inference) : 학습된 모델을 라벨이 없는 예에 적용하는 것을 의미
  - 회귀 모델 : 연속적인 값 (부동산 가격) 예측
  - 분류 모델 : 불연속적인 값 (강아지/고양이) 예측

# Regression vs Classification



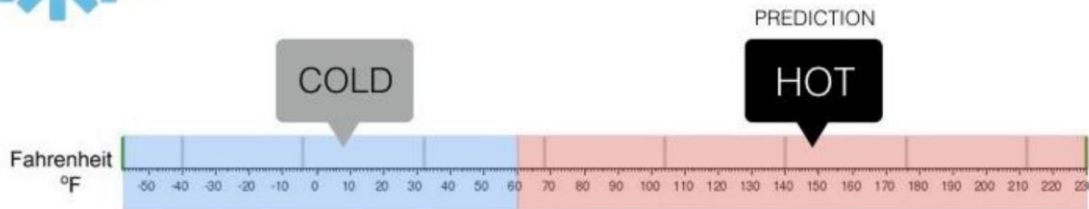
## Regression

What is the temperature going to be tomorrow?

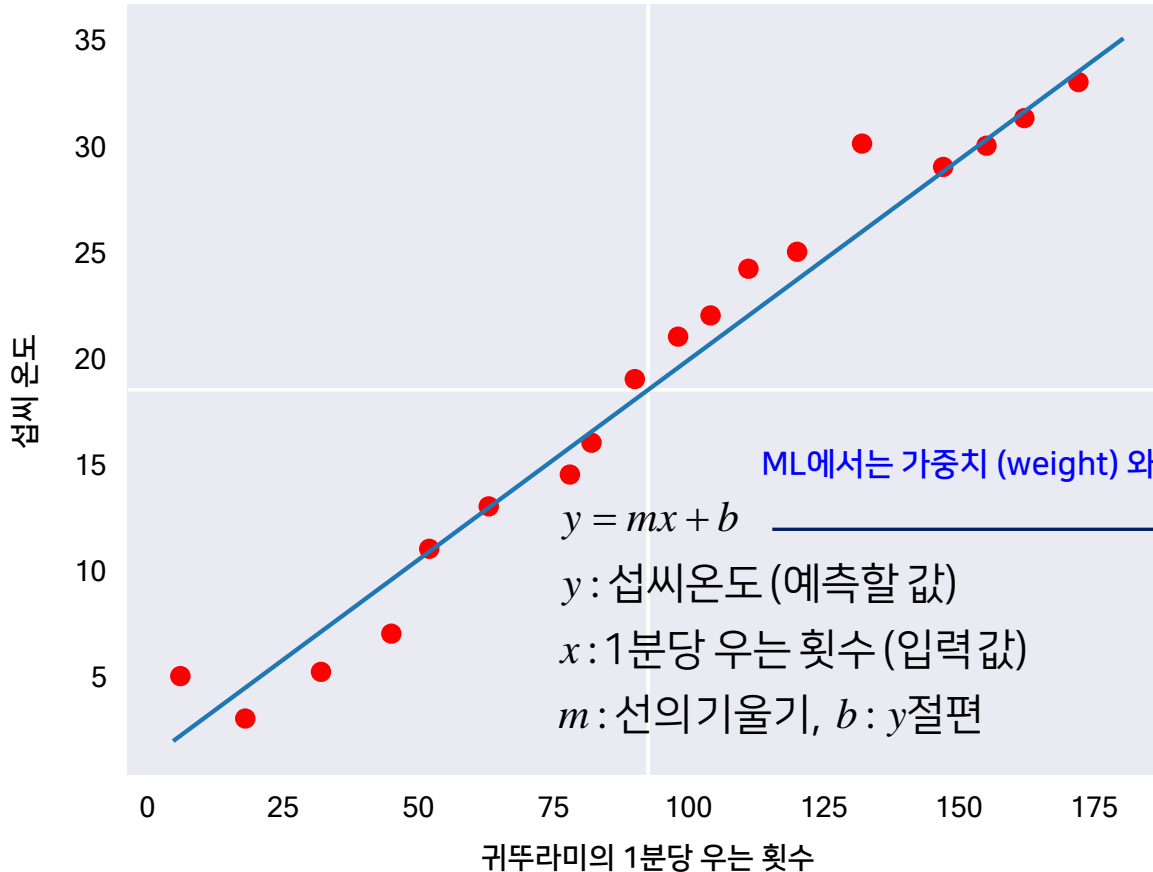


## Classification

Will it be Cold or Hot tomorrow?



# Linear Regression



오랫동안 귀뚜라미는 시원날 날보다 더운 날 더 자주 우는 것으로 알려져 왔습니다. 전문가 및 아마추어 곤충 학자들이 수십 년에 걸쳐서 **1분당 귀뚜라미가 우는 횟수와 온도에 관한 데이터**를 목록으로 작성했습니다. 고모가 생일 선물로 두 특성의 관계를 예측하는 모델을 학습시켜 보라고 아끼던 귀뚜라미 데이터베이스를 줬다고 해 봅시다.

$$H(x, b) = w_1 x_1 + b$$

입력 특성이 세 가지 일때,

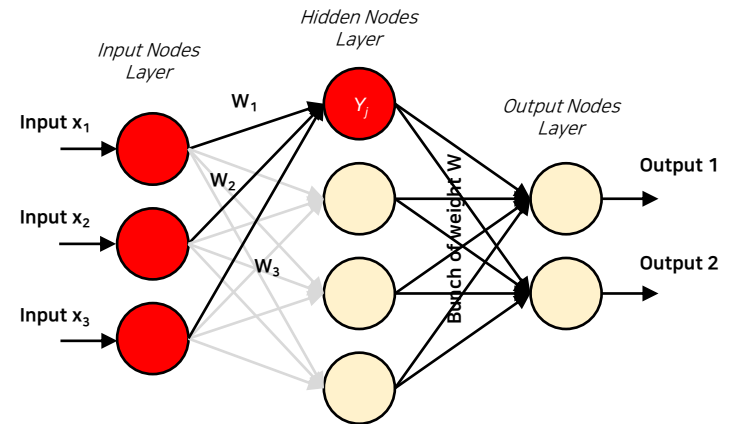
$$H(x, b) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

# Biological Inspiration

## □ Key Difference



Neuron & Synapse

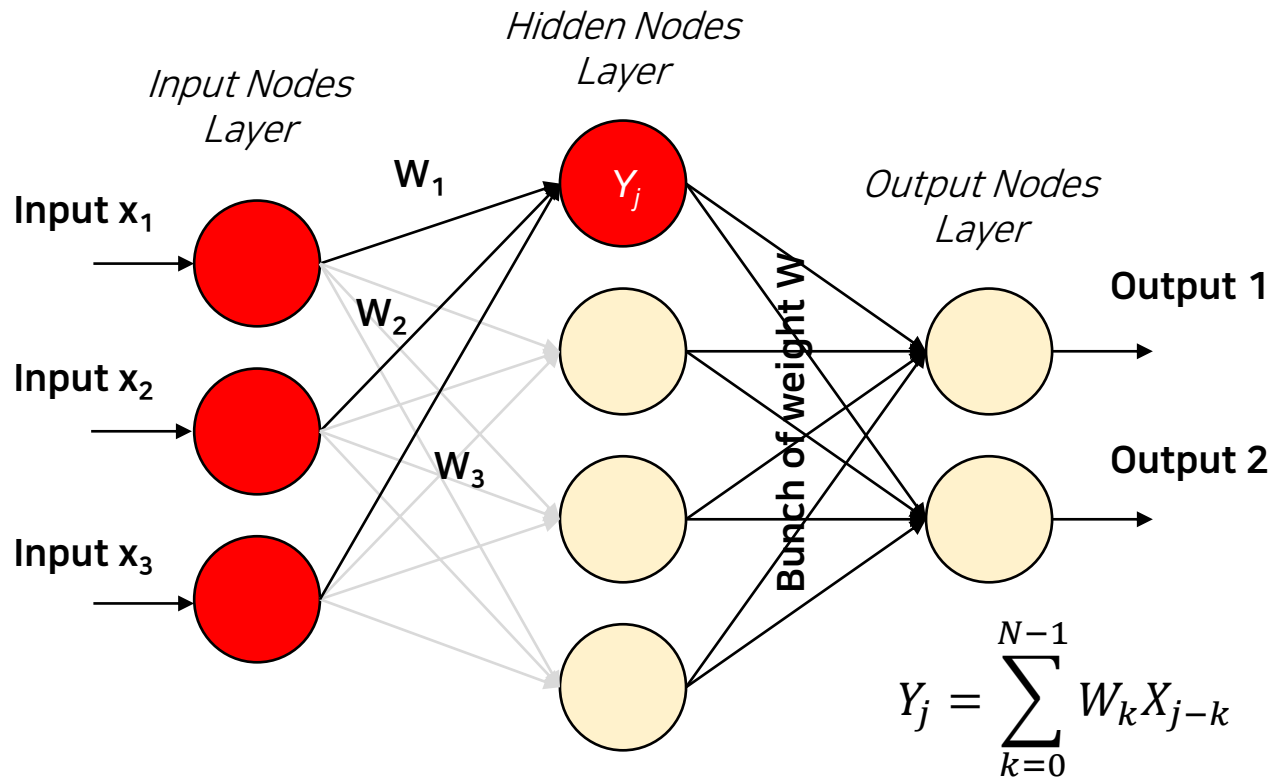


	Human (Biological) Neural Network	Artificial Neural Network
Parameter	Human brains have $\sim 10^6$ times synapses than artificial neural networks.	
Topology	Async	Sync
Learning algorithm	We don't know	Gradient Descent
Power consumption	Biological neural networks use very little power than artificial networks	
Stages	Never stop learning	First train then test



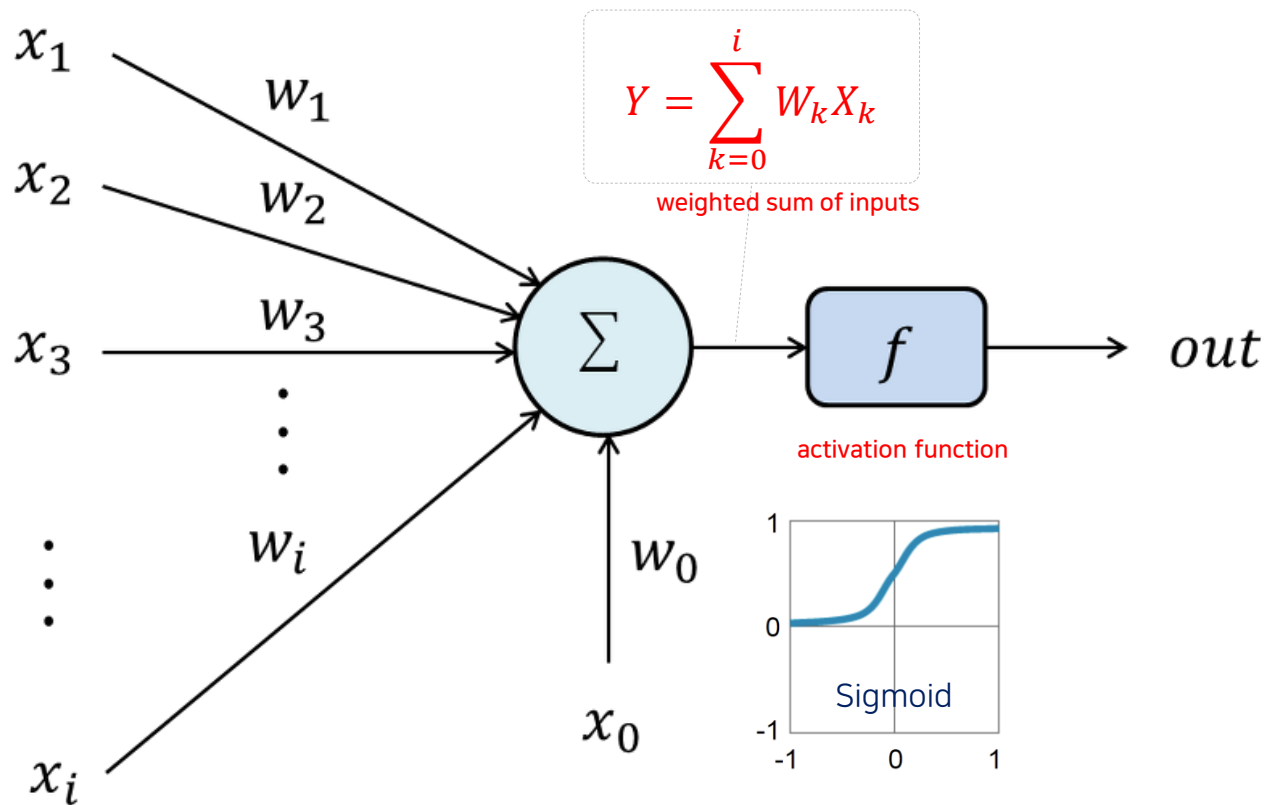
# Multi Layer Perceptron (MLP)

## □ 다중 레이어 퍼셉트론



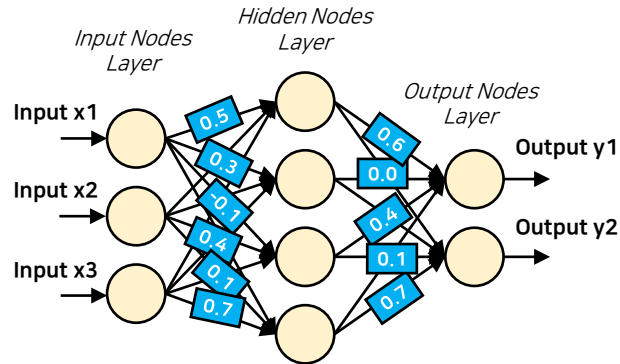
# Perceptron

## □ 퍼셉트론 (Perceptron)

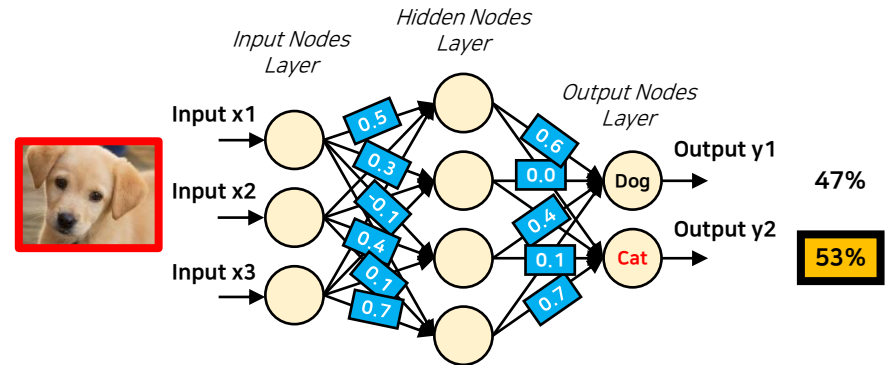


# Simple Neural Network

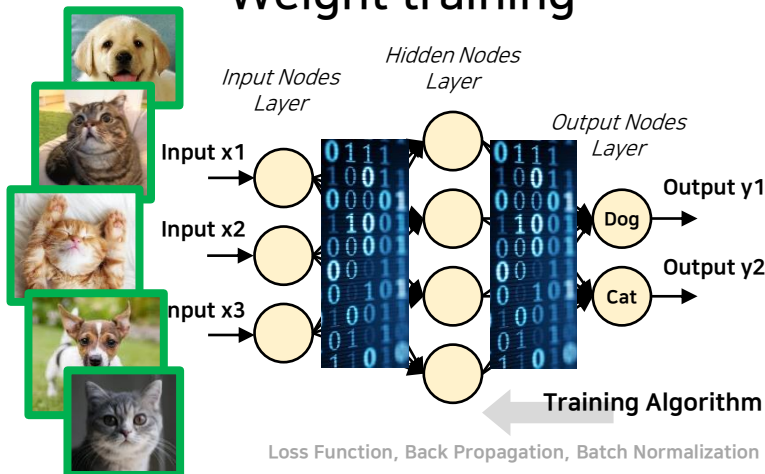
## Random initialization



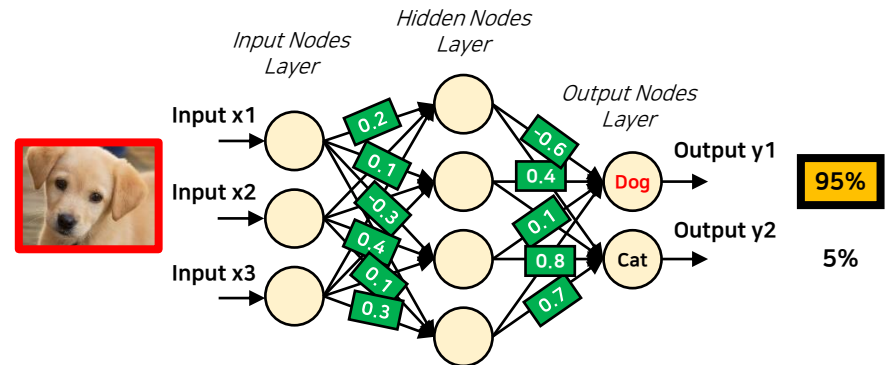
## Let's try a dog w/ un-trained NN



## Weight training



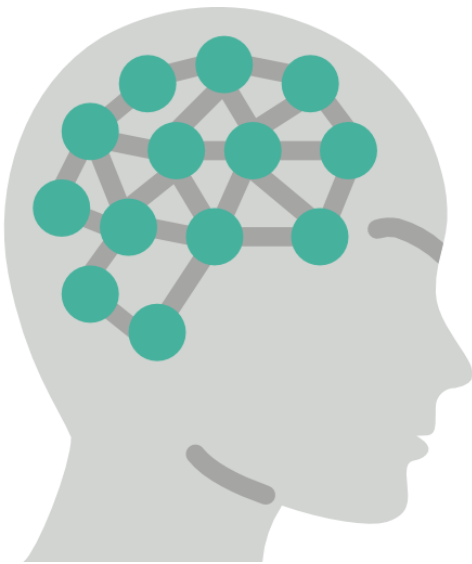
## Let's try again inference



# Neural Network

## ❑ MNIST Example

- $28 \times 28 = 784$  pixels Images



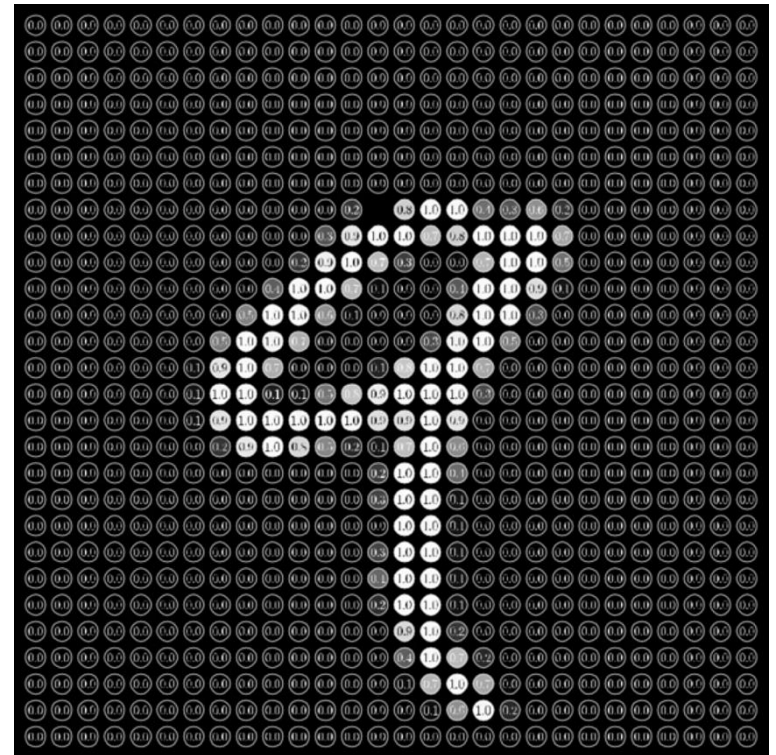
# Neural Network

## ❑ Neuron

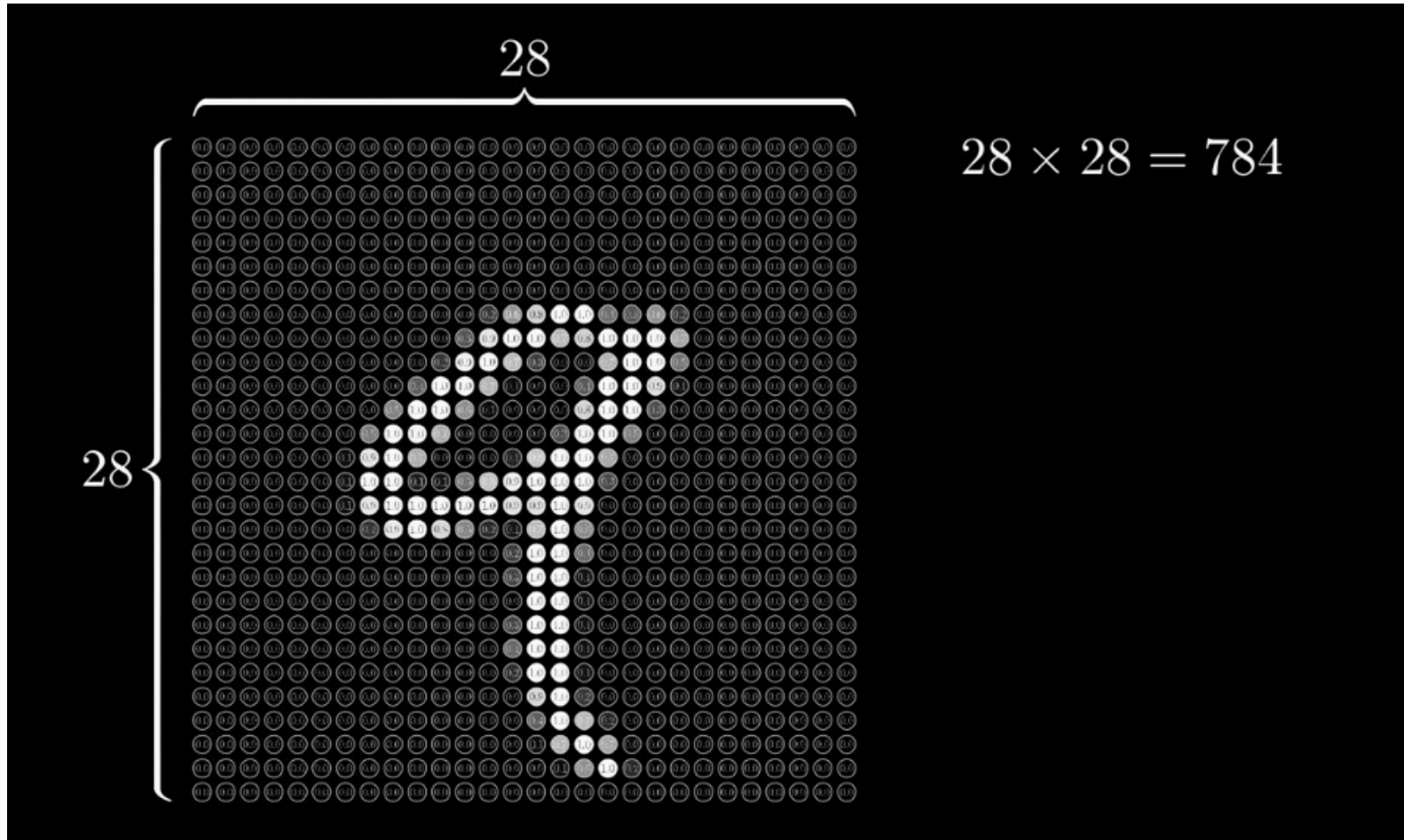
- Thing that holds a number

## ❑ Activation

- The number in Neuron
- Range: 0.0 ~ 1.0
- Gray Scale
  - 0.0 : Black
  - 1.0 : White



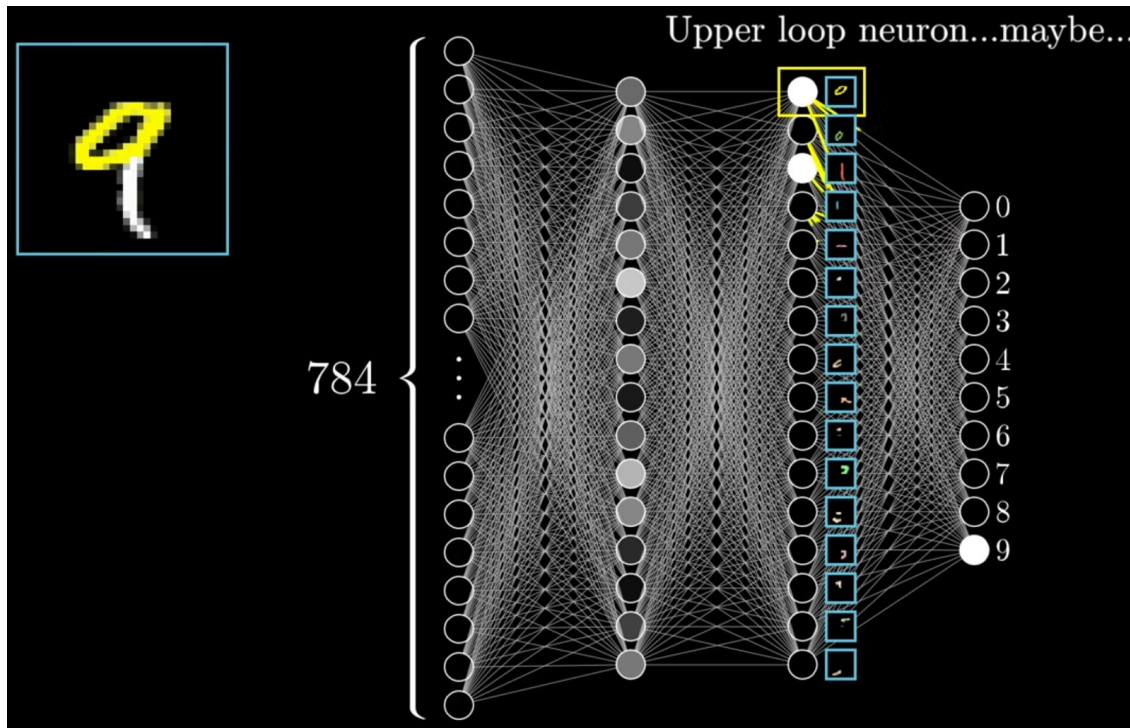
# Neural Network



# Neural Network

## ❑ Neural Network : How to work?

- Feature Extraction & Combine

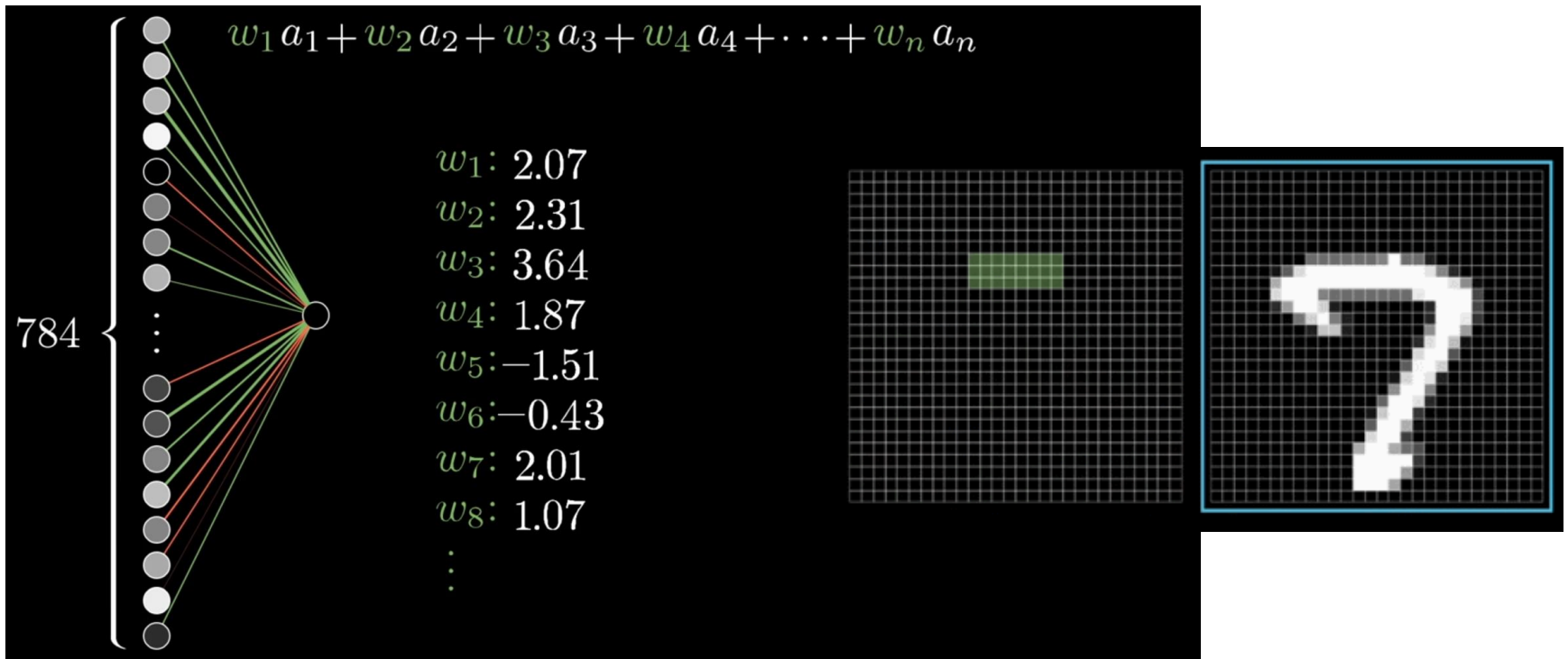




# Neural Network

## ❑ Neural Network : How to work?

- Weighted Sum make correlation w/ some shapes

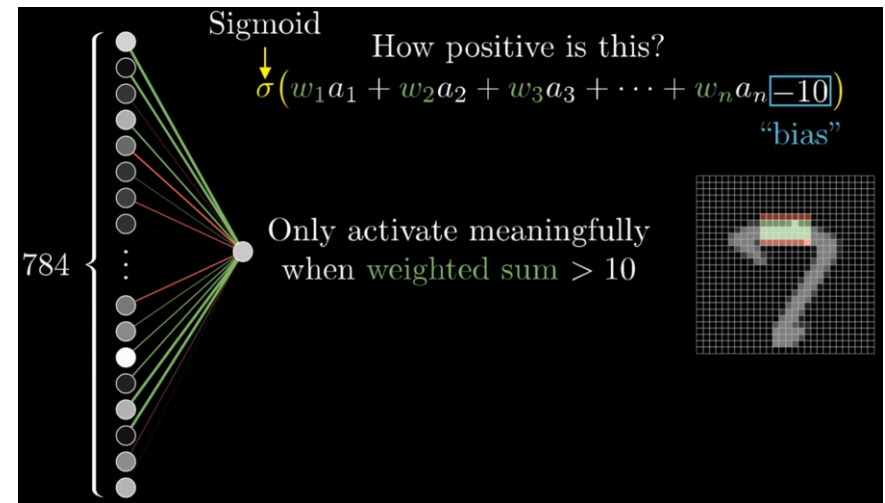
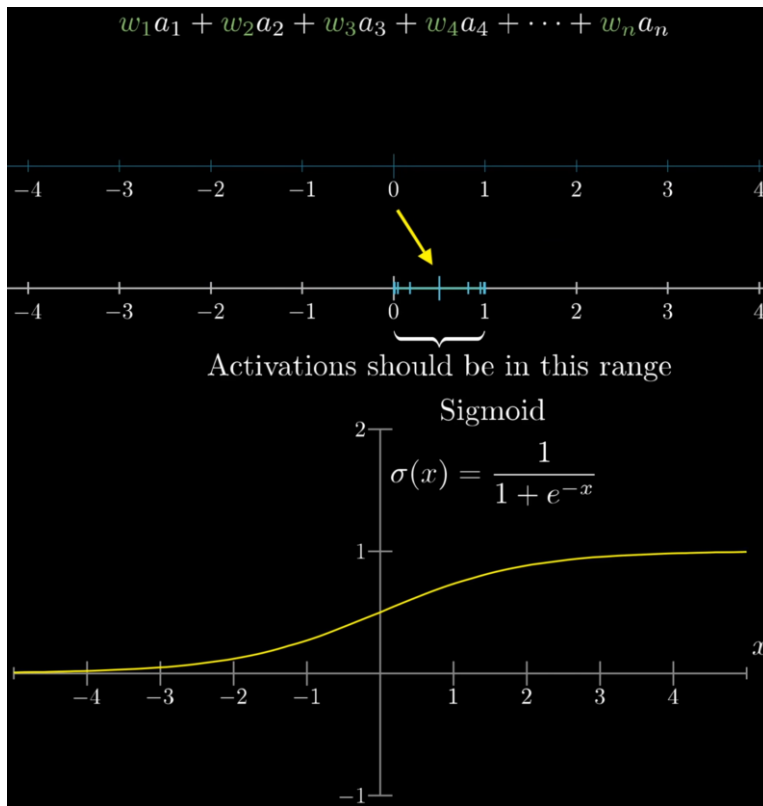




# Neural Network

## ❑ Neural Network : How to work?

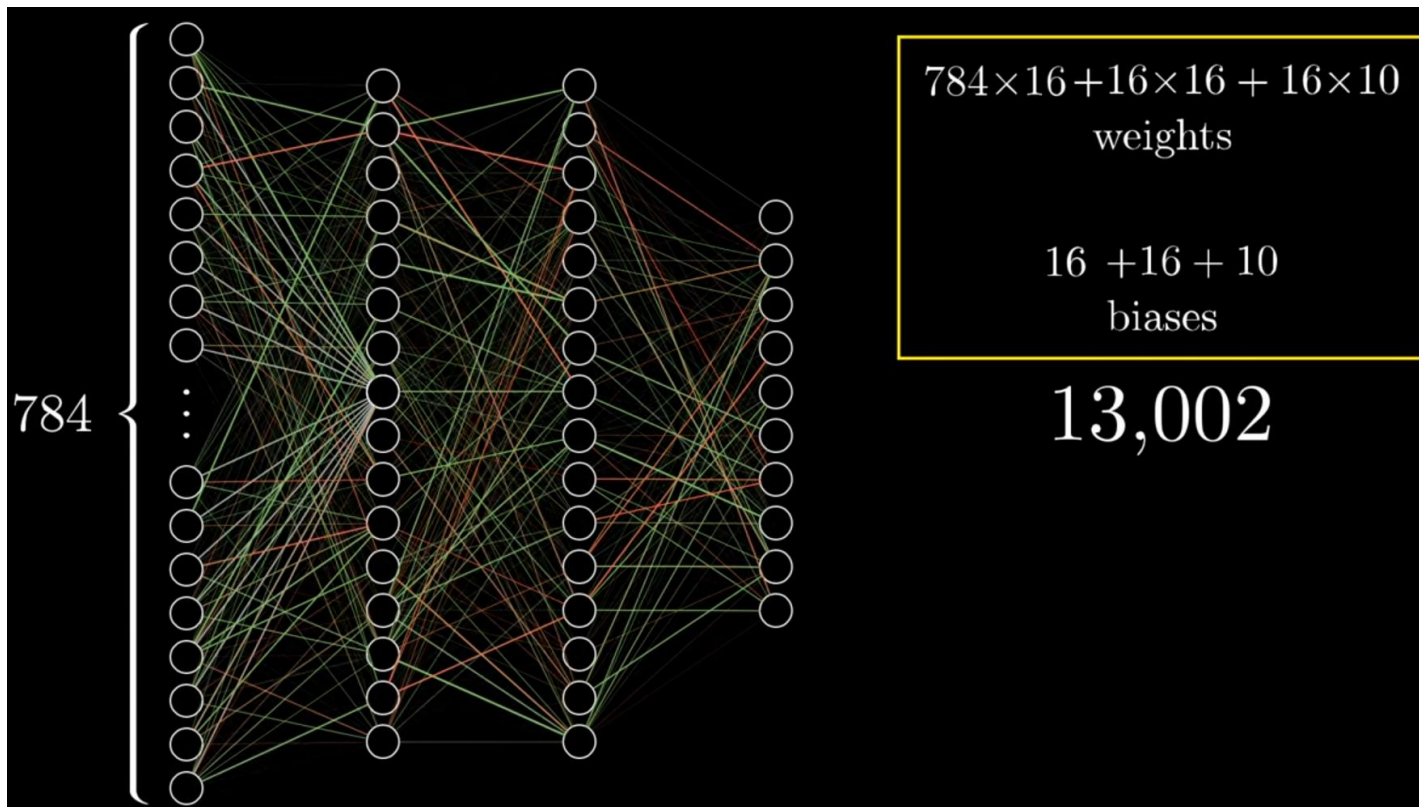
### - Activation Function & Bias



# Neural Network

## ❑ Neural Network : How to work?

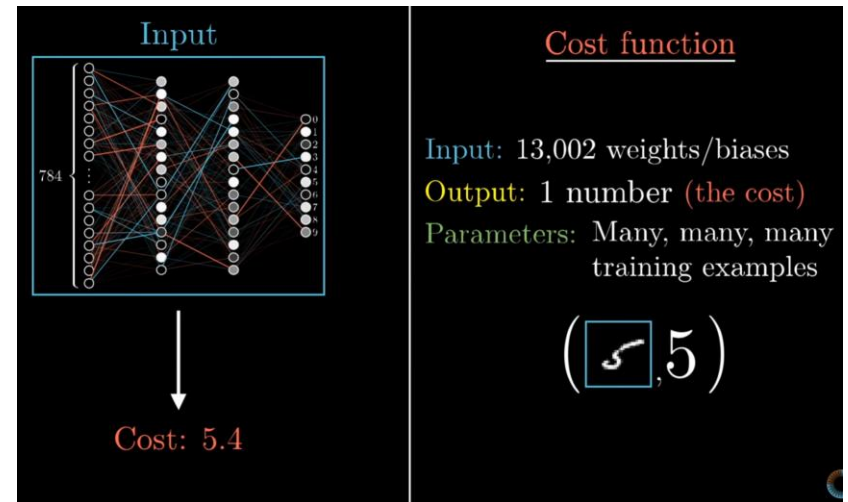
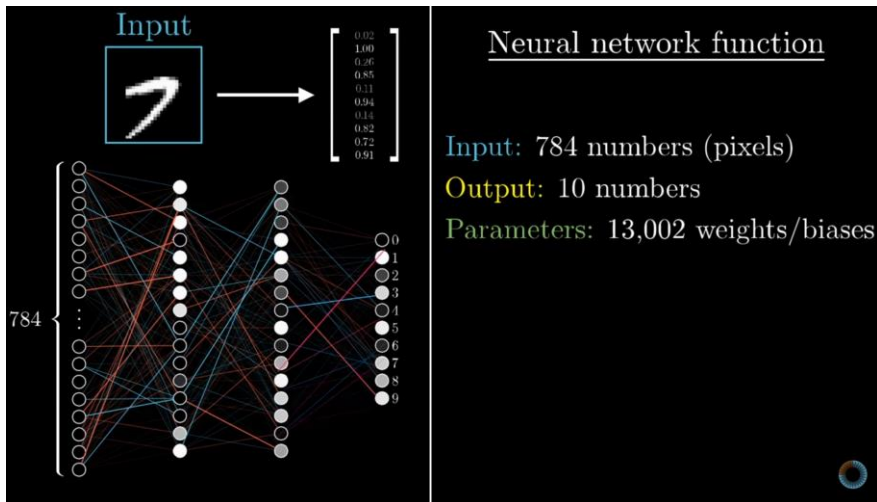
- Learning: Find right parameters



# Neural Network

## ❑ Neural Network : How to work?

- Learning: Find right parameters



# Neural Network

## ❑ Neural Network : How to work?

- Learning: Find right parameters

Cost function

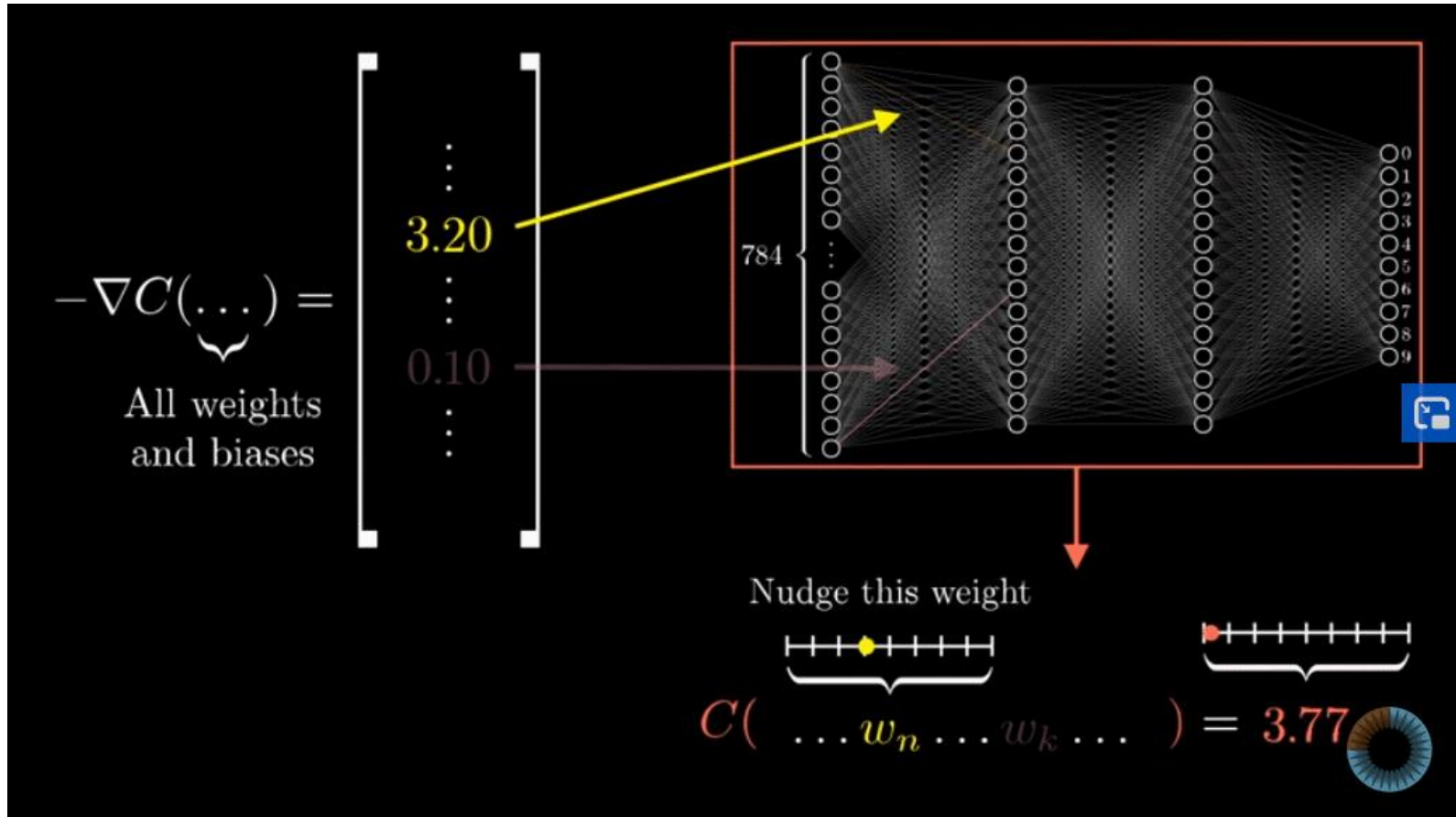
$$C(\underbrace{w_1, w_2, \dots, w_{13,002}}_{\text{Weights and biases}})$$

Weights and biases



# Neural Network

## □ Neural Network : How to work?



# Outline

---

## □ 인공지능 관련 동향

## □ 딥러닝의 기초

- 용어 및 분류(Classification)/회기(Regression) 모델
- 퍼셉트론 (Perceptron) 및 MNIST Example

## □ 학습 및 추론 알고리즘의 이해

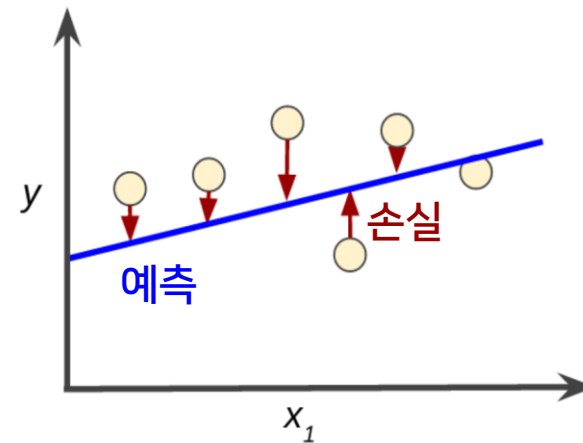
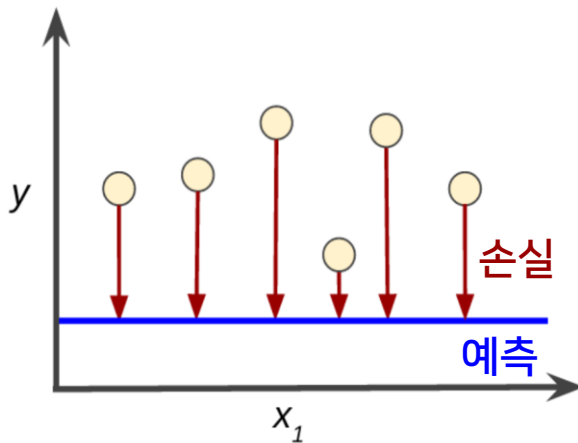
- Backpropagation
- Challenge of Learning
- Optimizer of Learning

## □ 딥러닝 가속기

# Training & Loss

## □ 학습

- 올바른 가중치 (Weight)와 편향값 (Biase)을 학습(결정)하는 것



Forward Pass

Input Data

Neural Network

Prediction

Backward Pass  
(Backpropagation)

Neural Network

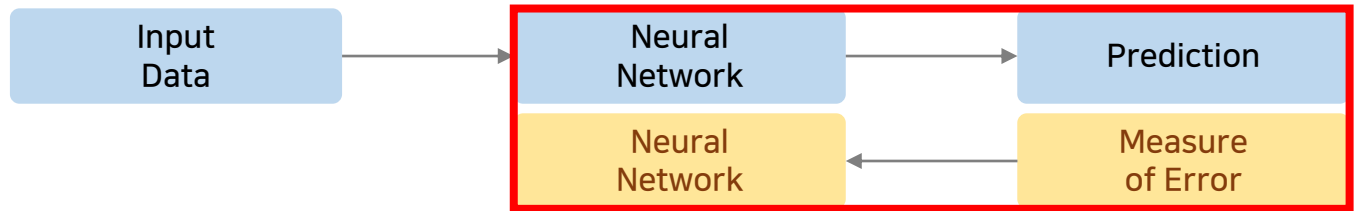
Measure of Error

# Backpropagation

## ❑ 오차역전파법 기반 학습

Forward Pass

Backward Pass  
(Backpropagation)



학습: 올바른 가중치와 편향 값을 학습(결정)하는 것



미분을 통해 각 파라미터가  
오류에 주는 영향도 파악 가능



인공신경망 내의 가중치와 편향 값

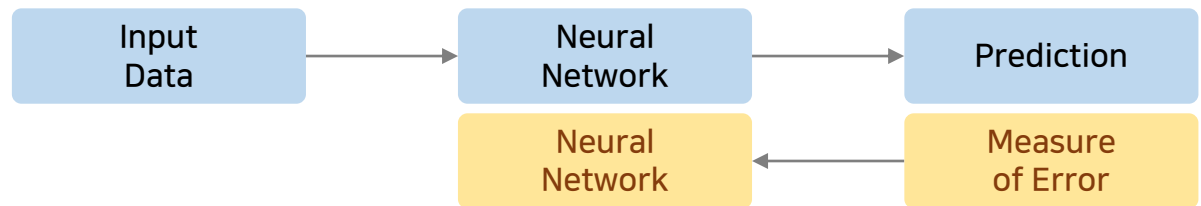


# Backpropagation

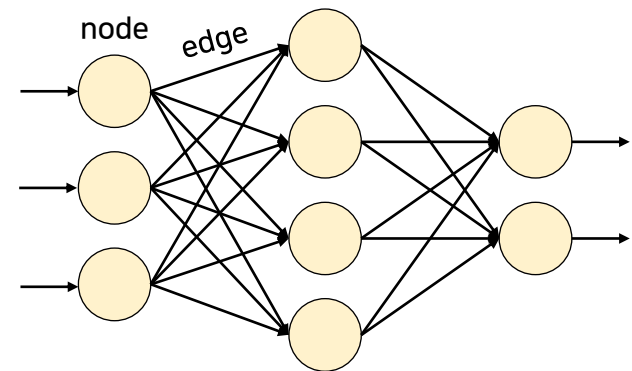
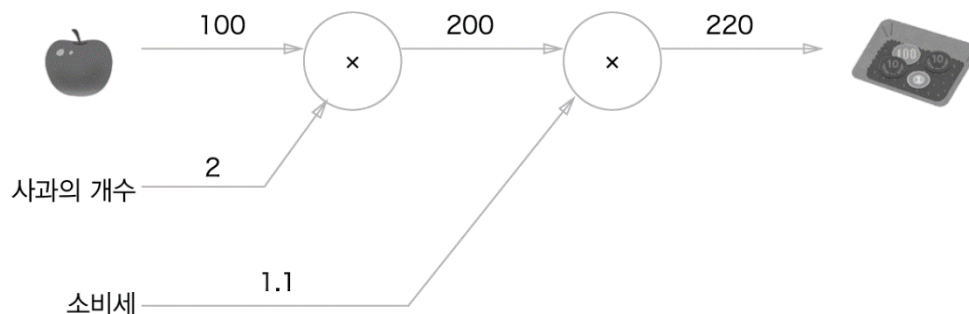
## □ Training (학습) Stage

Forward Pass

Backward Pass  
(Backpropagation)



계산 그래프 (복수의 Node와 Edge)

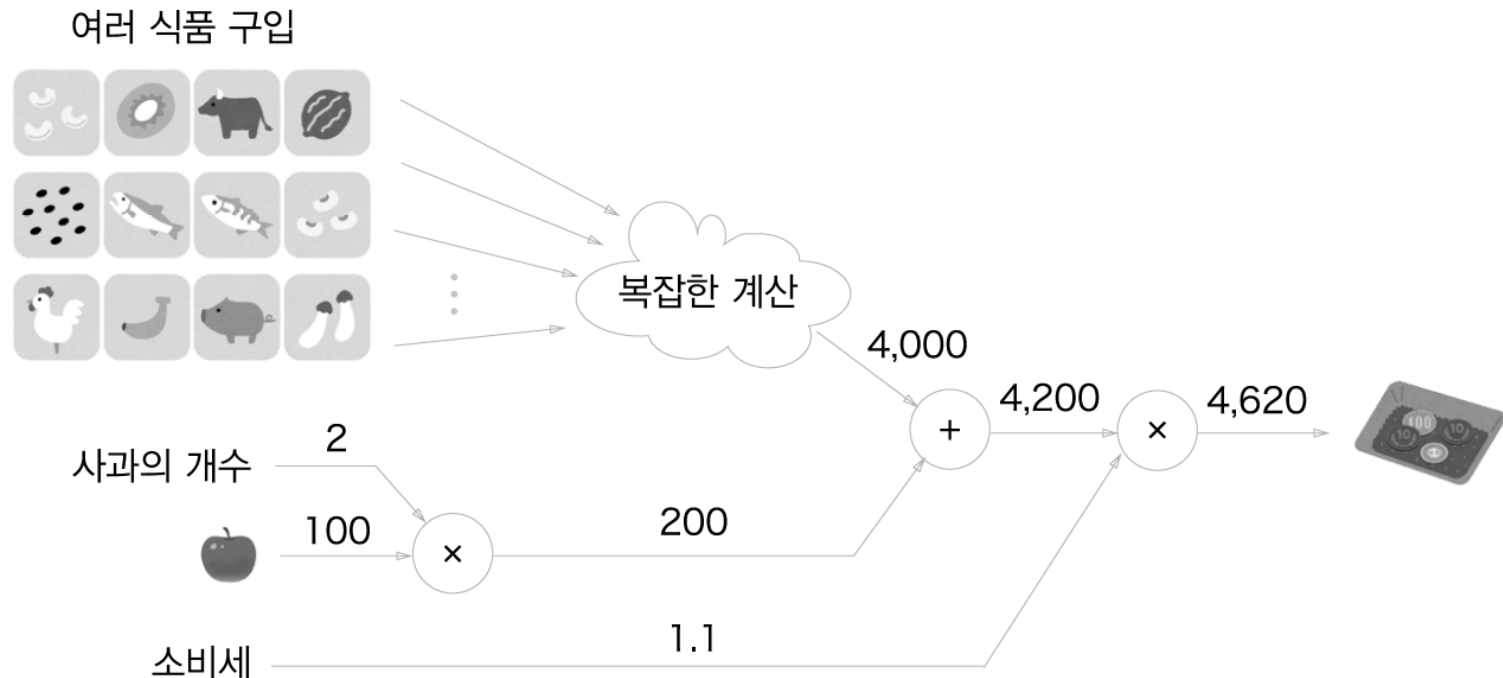


TensorFlow도 계산그래프와 동일한 의미

# Backpropagation

## □ 왜 계산 그래프로 문제를 푸는가?

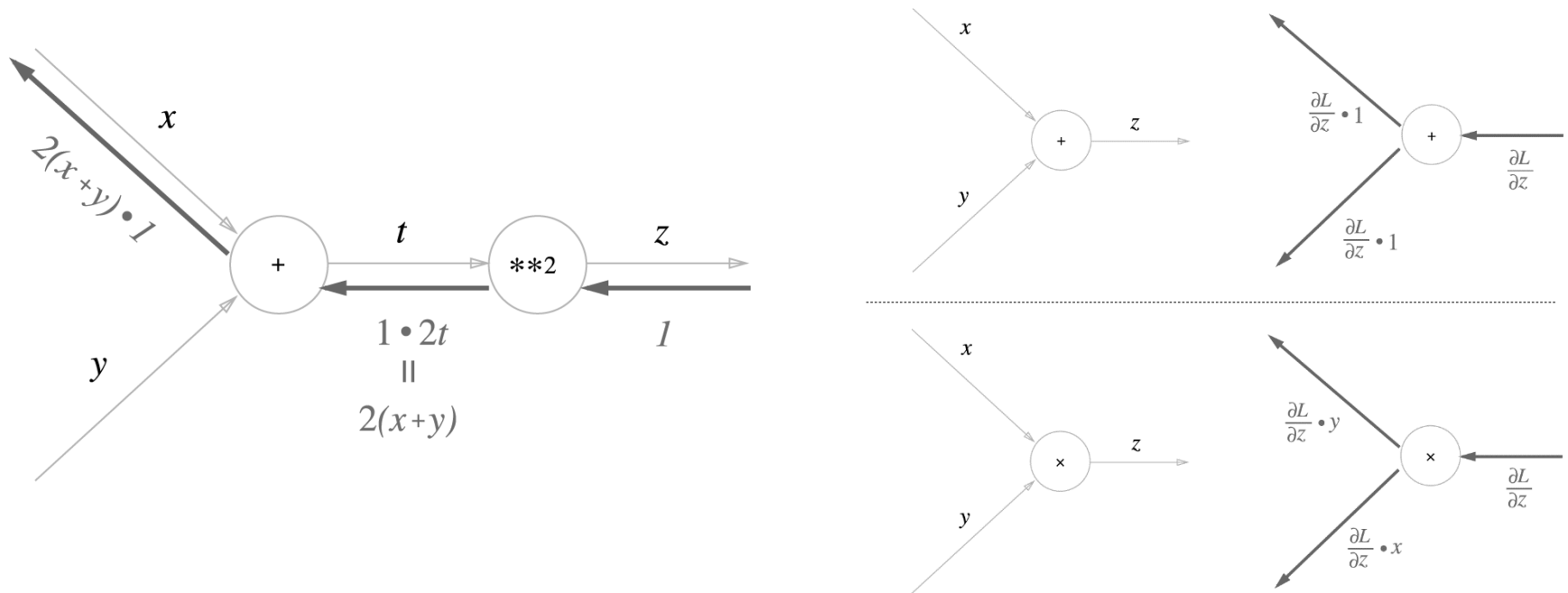
- 국소적 계산으로 문제 단순화 가능



# Backpropagation

## □ 왜 계산 그래프로 문제를 푸는가?

- 역전파를 통해 미분을 효율적으로 계산 가능



# Training & Loss

## □ 손실 함수 : 신경망이 학습할 수 있도록 해주는 지표

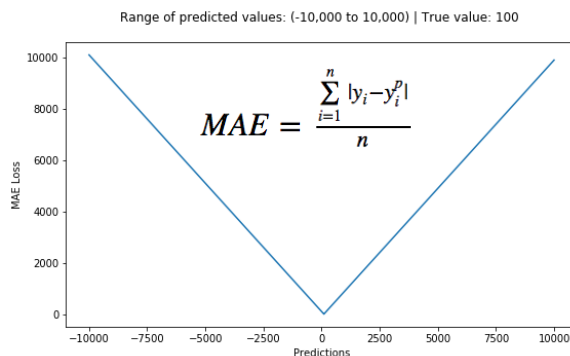
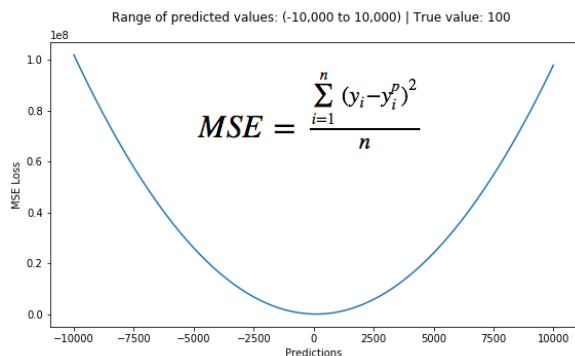
### - 목적

- 머신러닝 모델의 최종적인 목적은 높은 정확도를 끌어내는 매개변수(가중치, 편향)를 찾는 것이다. 신경망 학습에서는 최적의 매개변수를 탐색할 때 손실함수의 값을 가능한 한 작게 하는 매개변수 값을 찾는다. 이 때, 매개변수의 미분(기울기)을 계산하고, 그 미분 값을 토대로 매개변수 값을 갱신하는 과정을 반복한다.
- 주요한 점은, 정확도와는 달리 손실 함수는 매개변수의 변화에 따라 연속적으로 변화한다는 점이다. 손실 함수와는 달리 정확도는 매개변수의 변화에 둔감하고, 또한 변화가 있다하여도 불연속적으로 변화하기 때문에 미분을 할 수 없다. 미분이 되지 않으면 최적화를 할 수 없으므로 정확도가 아닌 손실 함수를 지표로 삼아 학습을 해나가는 것이다

# 5 Regression Loss Functions

## ❑ MSE vs MAE

- Mean Square Error (MSE, L2 Loss)
- Mean Absolute Error (MAE, L1 Loss)



MAE vs. RMSE for cases with slight variance in data

ID	Error	Error	Error <sup>2</sup>
1	0	0	0
2	1	1	1
3	-2	2	4
4	-0.5	0.5	0.25
5	1.5	1.5	2.25

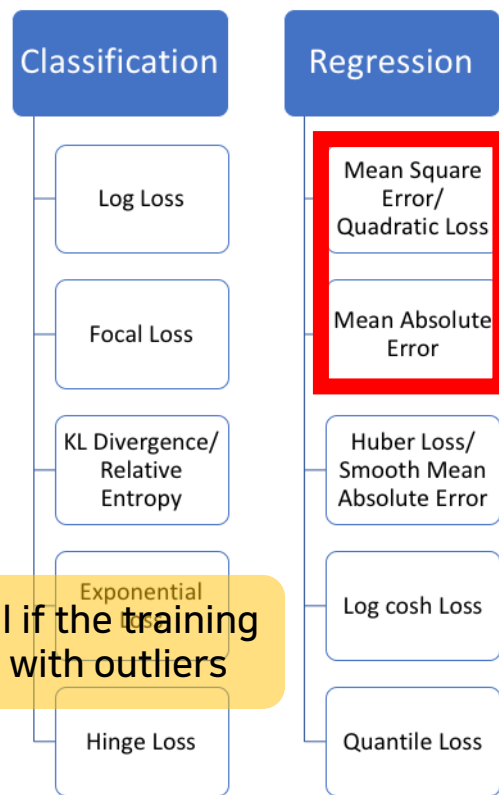
MAE: 1      RMSE: 1.22

MAE vs. RMSE for cases with outliers in data

ID	Error	Error	Error <sup>2</sup>
1	0	0	0
2	1	1	1
3	1	1	1
4	-2	2	4
5	15	15	225

MAE: 3.8      RMSE: 6.79

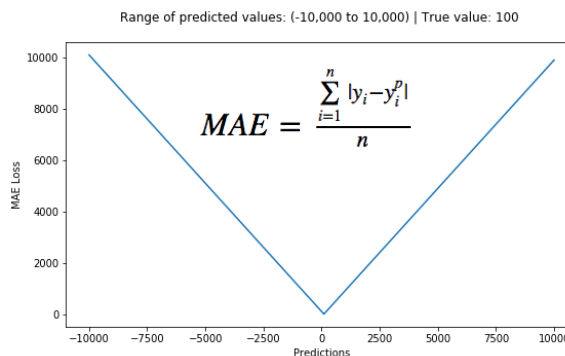
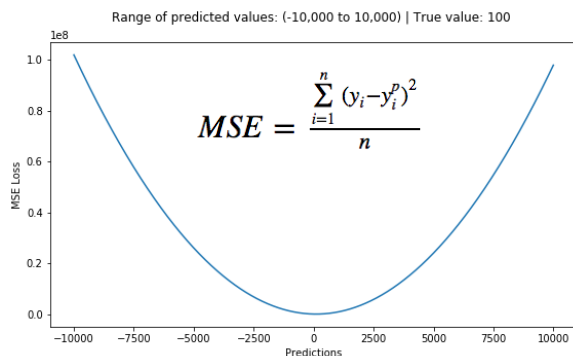
MAE loss is useful if the training data is corrupted with outliers



# 5 Regression Loss Functions

## ❑ MSE vs MAE

- Mean Square Error (MSE, L2 Loss)
- Mean Absolute Error (MAE, L1 Loss)



MAE vs. RMSE for cases with slight variance in data

ID	Error	Error	Error <sup>2</sup>
1	0	0	0
2	1	1	1
3	-2	2	4
4	-0.5	0.5	0.25
5	1.5	1.5	2.25

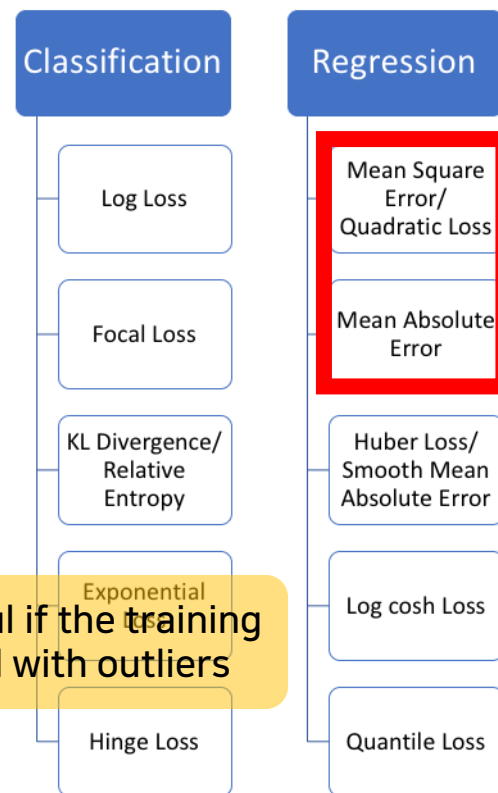
MAE: 1      RMSE: 1.22

MAE vs. RMSE for cases with outliers in data

ID	Error	Error	Error <sup>2</sup>
1	0	0	0
2	1	1	1
3	1	1	1
4	-2	2	4
5	15	15	225

MAE: 3.8      RMSE: 6.79

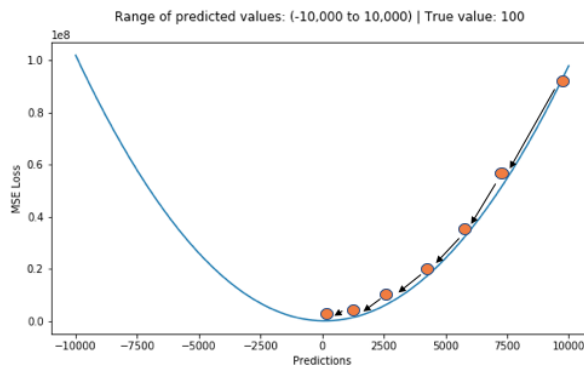
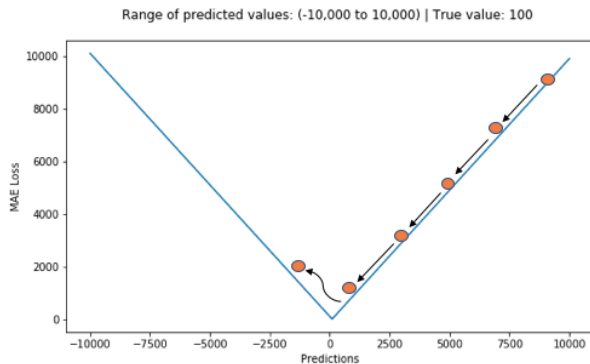
MAE loss is useful if the training data is corrupted with outliers



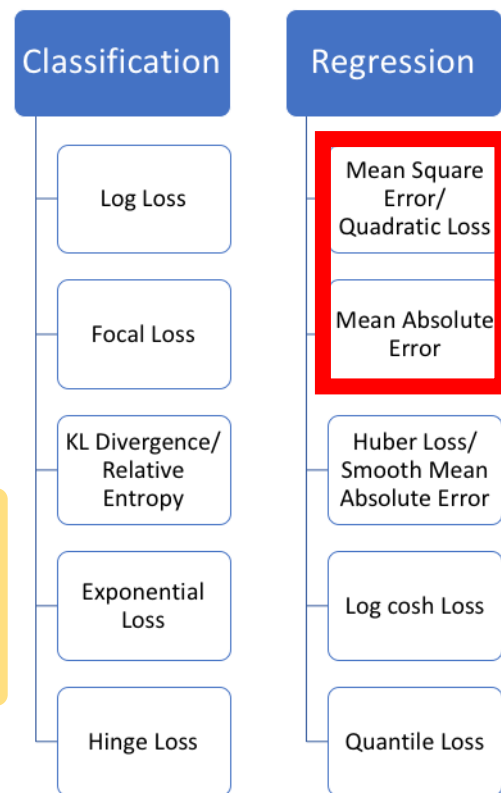
# 5 Regression Loss Functions

## ❑ MSE vs MAE

- One big problem in using MAE loss



If the outliers represent anomalies that are important for business and should be detected, then we should use MSE. On the other hand, if we believe that the outliers just represent corrupted data, then we should choose MAE as loss.

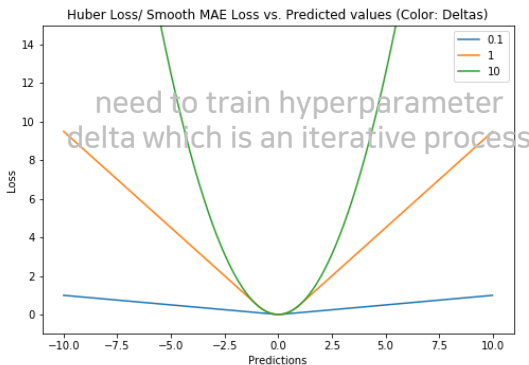


# 5 Regression Loss Functions

## Alternative Solutions

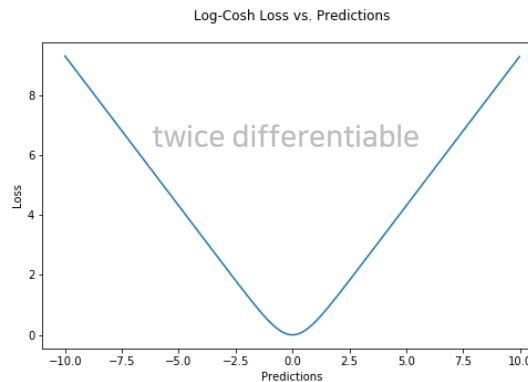
### Huber Loss (Smooth Mean Absolute Error)

$$L_{\delta}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

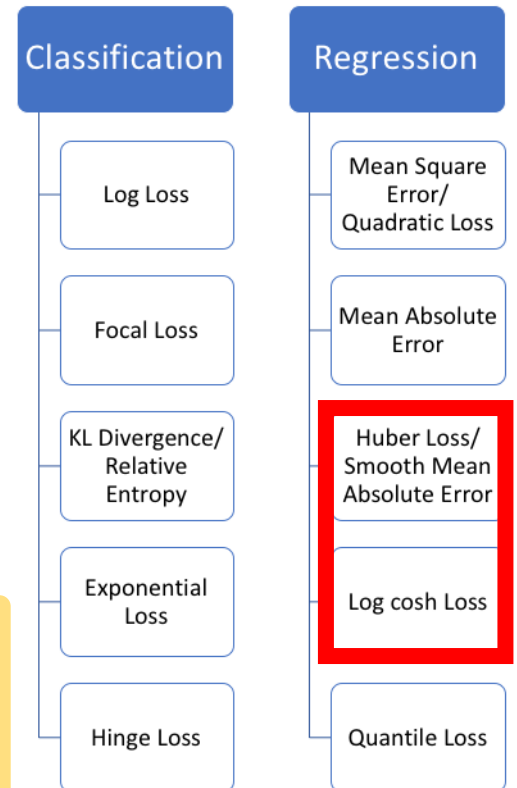


### Log-Cosh Loss

$$L(y, y^p) = \sum_{i=1}^n \log(\cosh(y_i^p - y_i))$$



**Why do we need a 2nd derivative?** Many ML model implementations like [XGBoost](#) use Newton's method to find the optimum, which is why the second derivative (Hessian) is needed. For ML frameworks like XGBoost, twice differentiable functions are more favorable.

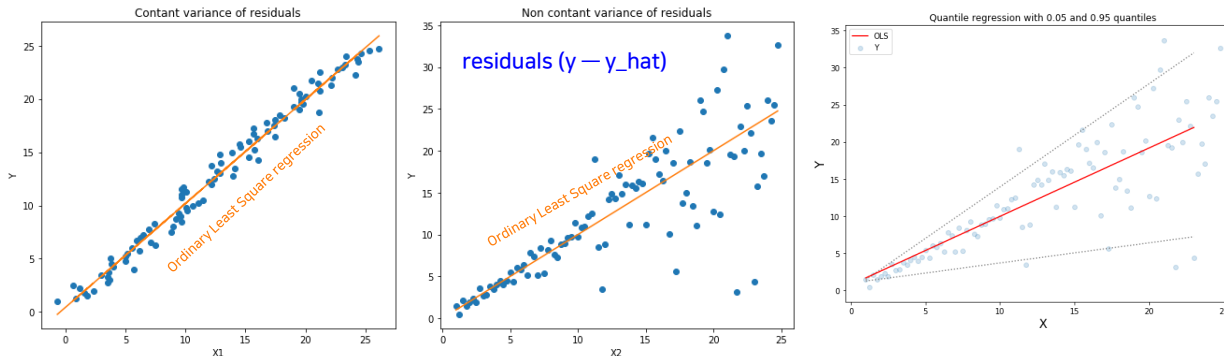




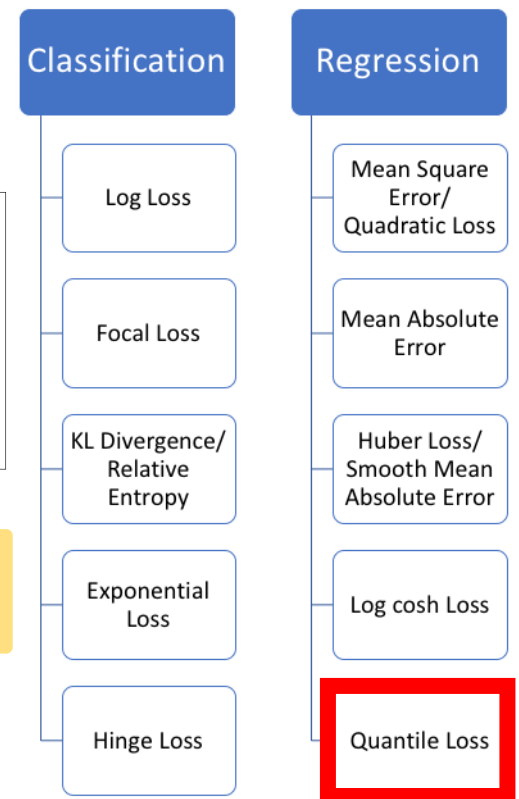
# 5 Regression Loss Functions

## Quantile (변위치) Loss

- actually just an extension of MAE (when quantile is 50th percentile)



useful when we are interested in predicting an interval instead of only point predictions

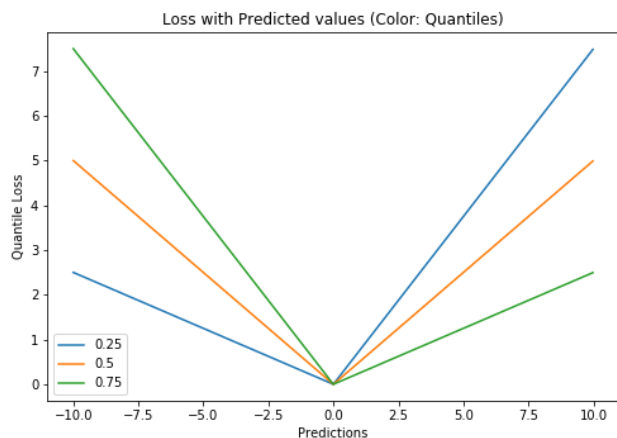


# 5 Regression Loss Functions

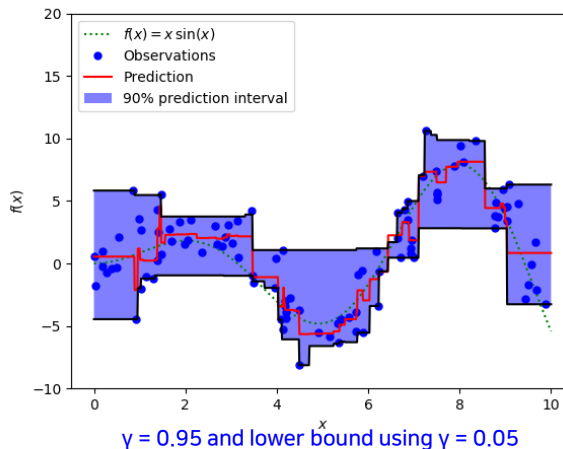
## Quantile Loss

- For example, a quantile loss function of  $\text{quantile}(y) = 0.25$  gives more penalty to overestimation and tries to keep prediction values a little below median

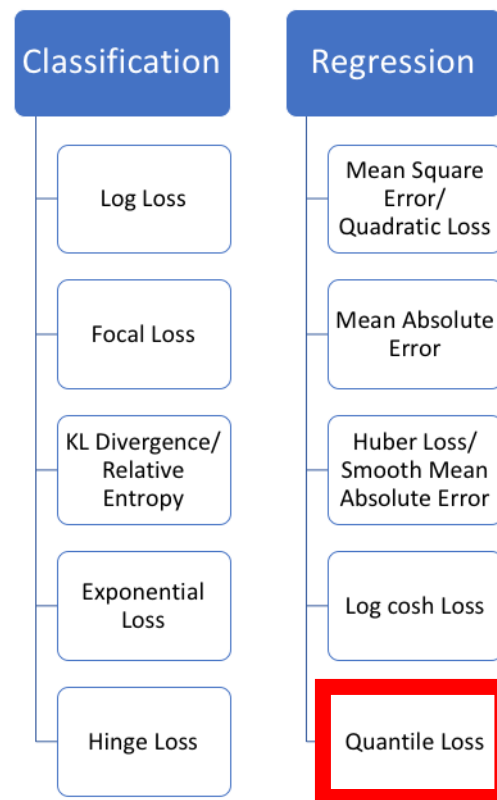
$$L_\gamma(y, y^p) = \sum_{i=y_i < y_i^p} (\gamma - 1) \cdot |y_i - y_i^p| + \sum_{i=y_i \geq y_i^p} (\gamma) \cdot |y_i - y_i^p|$$



Sklearn implementation for gradient boosted tree regressors



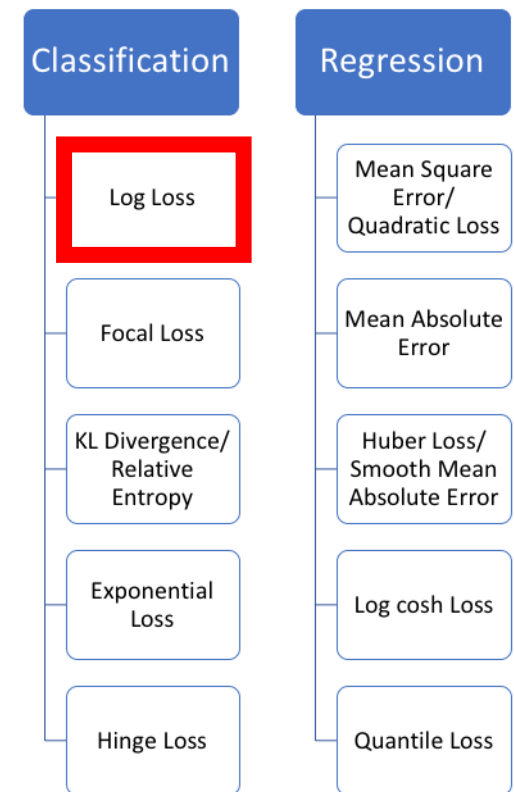
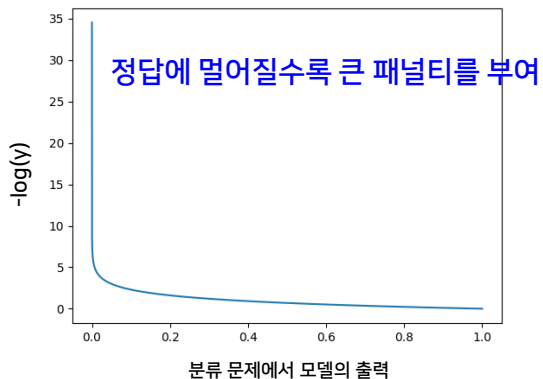
$\gamma = 0.95$  and lower bound using  $\gamma = 0.05$



# Classification Loss Functions

## □ Cross Entropy Loss (Log Loss)

- 원-핫 인코딩(one-hot encoding)했을 경우에만 사용할 수 있는 오차 계산법
- $E = -\sum_k t_k \log y_k$   $t$ 값 : 원-핫 인코딩된 벡터
- 결과적으로 교차 엔트로피 오차는 정답일 때의 모델 값에 자연로그를 계산하는 식



# TensorFlow API : Loss Func.

---

## Module: tf.losses

### Classes

[class Reduction](#): Types of loss reduction.

### Functions

[absolute\\_difference\(...\)](#): Adds an Absolute Difference loss to the training procedure.

[add\\_loss\(...\)](#): Adds a externally defined loss to the collection of losses.

[compute\\_weighted\\_loss\(...\)](#): Computes the weighted loss.

[cosine\\_distance\(...\)](#): Adds a cosine-distance loss to the training procedure. (deprecated arguments)

[get\\_losses\(...\)](#): Gets the list of losses from the loss\_collection.

[get\\_regularization\\_loss\(...\)](#): Gets the total regularization loss.

[get\\_regularization\\_losses\(...\)](#): Gets the list of regularization losses.

[get\\_total\\_loss\(...\)](#): Returns a tensor whose value represents the total loss.

[hinge\\_loss\(...\)](#): Adds a hinge loss to the training procedure.

[huber\\_loss\(...\)](#): Adds a Huber Loss term to the training procedure.

[log\\_loss\(...\)](#): Adds a Log Loss term to the training procedure.

[mean\\_pairwise\\_squared\\_error\(...\)](#): Adds a pairwise-errors-squared loss to the training procedure.

[mean\\_squared\\_error\(...\)](#): Adds a Sum-of-Squares loss to the training procedure.

[sigmoid\\_cross\\_entropy\(...\)](#): Creates a cross-entropy loss using `tf.nn.sigmoid_cross_entropy_with_logits`.

[softmax\\_cross\\_entropy\(...\)](#): Creates a cross-entropy loss using `tf.nn.softmax_cross_entropy_with_logits_v2`.

[sparse\\_softmax\\_cross\\_entropy\(...\)](#): Cross-entropy loss using [tf.nn.sparse\\_softmax\\_cross\\_entropy\\_with\\_logits](#).

# Outline

---

## □ 인공지능 관련 동향

## □ 딥러닝의 기초

- 용어 및 분류(Classification)/회기(Regression) 모델
- 퍼셉트론 (Perceptron) 및 MNIST Example

## □ 학습 및 추론 알고리즘의 이해

- Backpropagation
- Challenge of Learning
- Optimizer of Learning

## □ 딥러닝 가속기

- CNN 및 인공 신경망 가속기 연구 동향