VLSI & System Lab.

딥러닝 가속기 기초 및 최적화 설계

한양대 IDEC 강의

숙명여자대학교

최 웅 (woongchoi@sookmyung.ac.kr)

Outline

- **WSL (Window Subsystem for Linux) 기반 개발 환경**
  - WSL 설치 및 Setup 파일 설명
- **딥러닝 기초 및 실습**
  - TensorFlow 기반 실습
- **딥러닝 하드웨어 가속기 연구 동향**

Outline

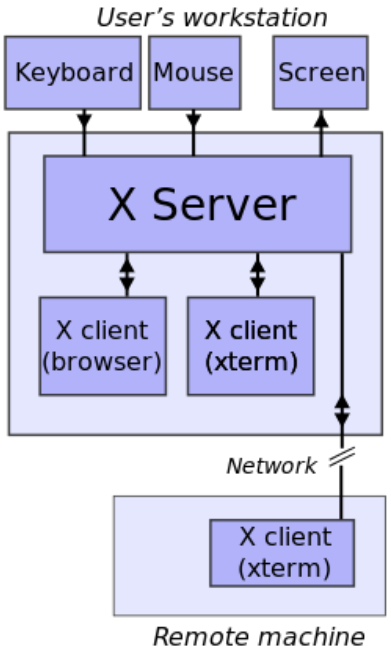
- **WSL (Window Subsystem for Linux) 기반 개발 환경**
  - WSL 설치 및 Setup 파일 설명
- **딥러닝 기초 및 실습**
  - TensorFlow 기반 실습
- **딥러닝 하드웨어 가속기 연구 동향**

개발 환경 구축 : Xming

❑ Xming? X Window?

- Xming : 윈도우에서 동작하는 X 서버
- X Window : 플랫폼과 독립적으로 작동하는 윈도 시스템
- 설치
  - https://sourceforge.net/projects/xming/
  - 'xming' 구글링 시 맨 위 링크

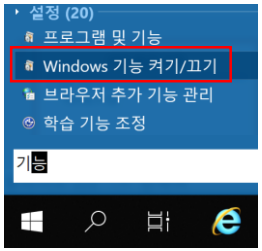
윈도우에는 Xming 과 WSL만 설치

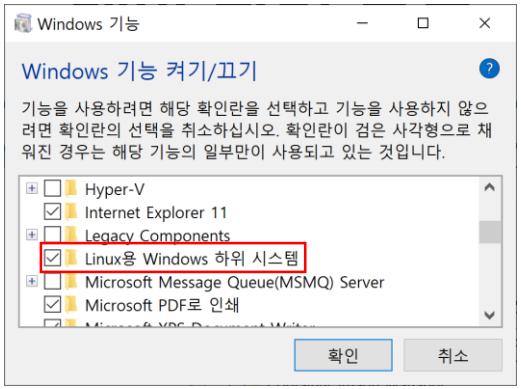


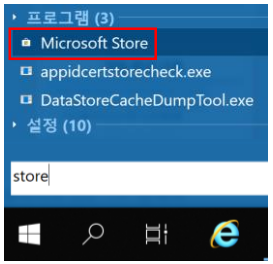
개발 환경 구축 : WSL

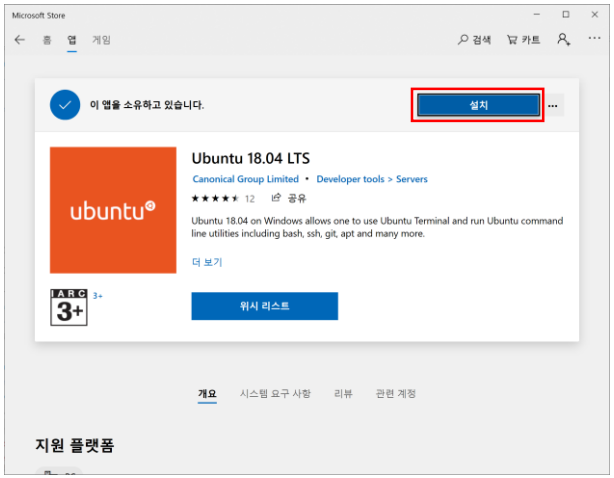
❑ WSL (Windows Subsystem for Linux) 설치

- Win Key → ‘Windows 기능 켜기/끄기’ 검색 및 실행
- ‘Linux용 Windows 하위 시스템’ 활성화 → 재부팅
- Win Key → ‘Microsoft Store’ 검색 및 실행
- ‘Ubuntu 18.04 LTS’ 설치 (설치 클릭 시 새로 뜨는 창은 닫아 버려도 됨)







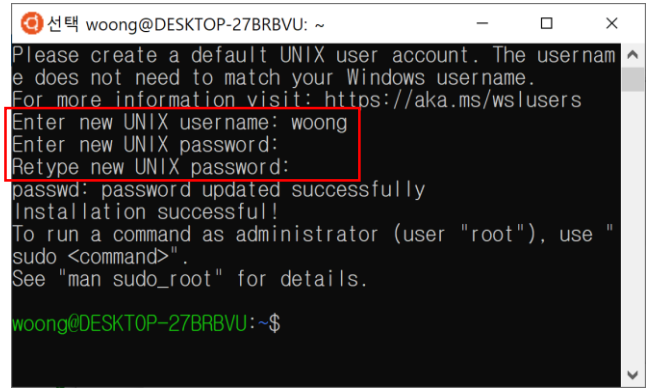


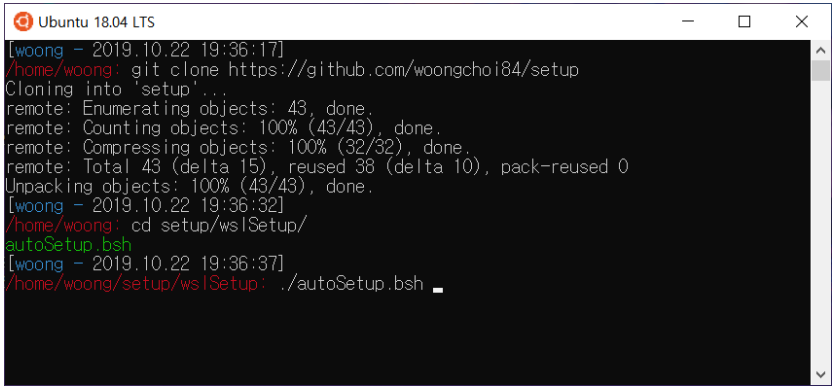
개발 환경 구축 : WSL

❑ WSL (Windows Subsystem for Linux) 설치

- username/password 설정
- Linux/ TensorFlow 기본 환경 설정 및 필수 패키지 설치

```
git clone https://github.com/woongchoi84/setup
cd setup/wslSetup
./autoSetup.bsh
```





개발 환경 구축 : WSL

❑ Customized Setup File : autoSetup.bsh

```
#! /bin/bash
# ================================================
#  Coded by Woong
# ================================================
# [Make Backup File]
sudo cp /etc/apt/sources.list /etc/apt/sources.list_bk

# [Change Repository]
sudo sed -i 's/archive.ubuntu.com/ftp.daumkakao.com/g' /etc/apt/sources.list
sudo sed -i 's/security.ubuntu.com/ftp.daumkakao.com/g' /etc/apt/sources.list

# [Update & Upgrade]
sudo apt-get -y update
sudo DEBIAN_FRONTEND=noninteractive apt-get -y full-upgrade
```

셔뱅 : 인터프리터 할당

더 빠른 한국 서버로 리포지토리 (저장소) 변경

업그레이드 시 유저에게
물어보는 (interactive) 것들 제거

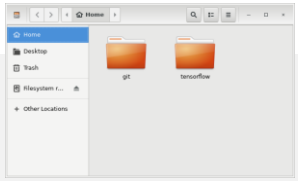
개발 환경 구축 : WSL

❑ Customized Setup File : autoSetup.bsh

```
# [Install Essential Tools]
sudo apt-get -y install build-essential
sudo apt-get -y install ftpd ssh git vim-gtk3 nautilus
sudo apt-get -y install x11-apps xfonts-base xfonts-100dpi xfonts-75dpi xfonts-cyrillic
dbus-x11
sudo apt-get -y install gnome-terminal gnome-paint
sudo apt-get -y install fonts-unfonts-core fonts-unfonts-extra fonts-baekmuk fonts-nanum
fonts-nanum-coding fonts-nanum-extra
sudo apt-get -y install tcl-dev tk-dev python3-dev python3-pip python3-사
sudo apt-get -y install firefox
#sudo apt-get -y install cmake clang

# [Remove Minor Warning]
sudo chown -R ${USER}:${USER} ~/.cache
```

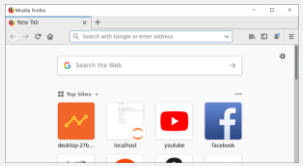
필수 패키지 설치

nautilus :
파일 관리 프로그램



그놈 터미널

'Jupyter Notebook' 과 연동 위해 반드시 설치



https://github.com/woongchoi84/setup/

개발 환경 구축 : WSL

❑ Customized Setup File : autoSetup.bsh

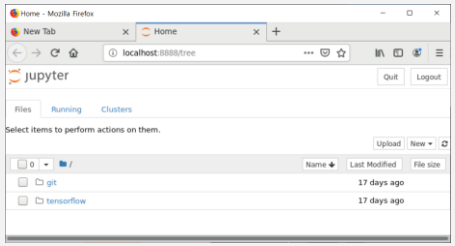
```
# [TensorFlow]
sudo pip3 install --upgrade pip
sudo pip3 install numpy pandas scipy matplotlib pillow pypi progress idx2numpy
sudo pip3 install jupyter jupyterlab
sudo pip3 install tensorflow

# [Bash Environment Setup]
printf "\nsource\t~/.bashrc_add" >> ~/.bashrc
cp .bashrc_add ~
cp .vimrc ~

# [Vim Plugin Install]
#git clone https://github.com/VundleVim/Vundle.vim.git ~/.vim/bundle/Vundle.vim
#vim +PluginInstall +qall
#python3 ~/.vim/bundle/YouCompleteMe/install.py
```

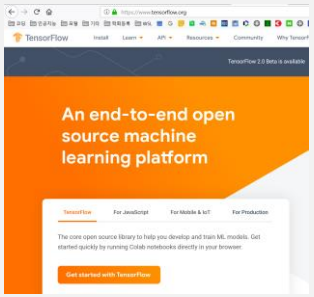
Python 패키지들

Jupyter Notebook/Lab

TensorFlow

Bash & VIM 환경 설정





vim 자동완성 플러그인 (위의 cmake, clang 설치 필요)

개발 환경 구축 : WSL

❑ Customized Setup File : .bashrc_add

```
export DISPLAY=0:0
export PS1="[\[\e[0;36m\]\u - \[\e[0;37m\]\D{%Y.%m.%d} \t\[\e[0;39m\]]\n\[\e[0;31m\]\${PWD}:
\[\e[0;39m\]"
alias  src='source ~/.bashrc'
alias  g='gvim -p'
alias  t='gnome-terminal'

function     cd { if (( $#==0 )); then builtin cd ~ && ls; else builtin cd "$@" && ls; fi }

cd
```

Bash Prompt 설정

Alias (단축키) 설정

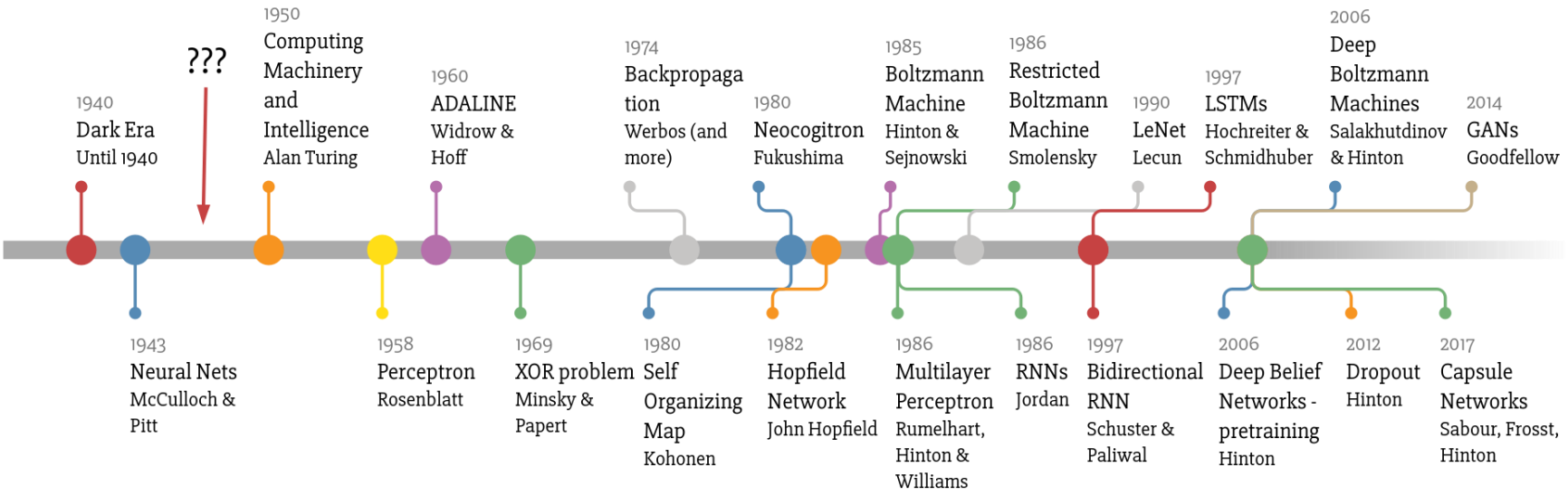
폴더 이동 시 폴더 내 파일/디렉토리 디스플레이

Outline

- WSL (Window Subsystem for Linux) 기반 개발 환경
  - WSL 설치 및 Setup 파일 설명
- **딥러닝 기초 및 실습**
  - TensorFlow 기반 실습
- 딥러닝 하드웨어 가속기 연구 동향

History

Deep Learning Timeline



Made by Favio Vázquez

인공 신경망의 도입

Human

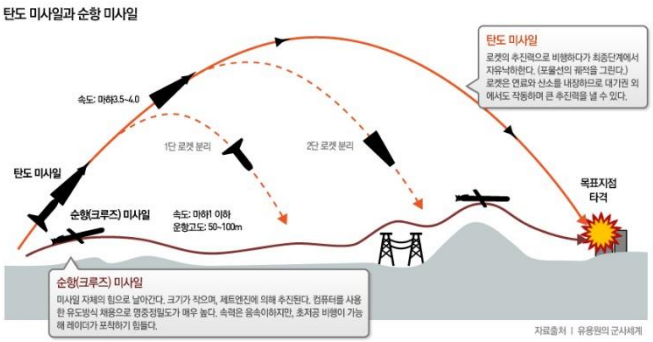
Easy

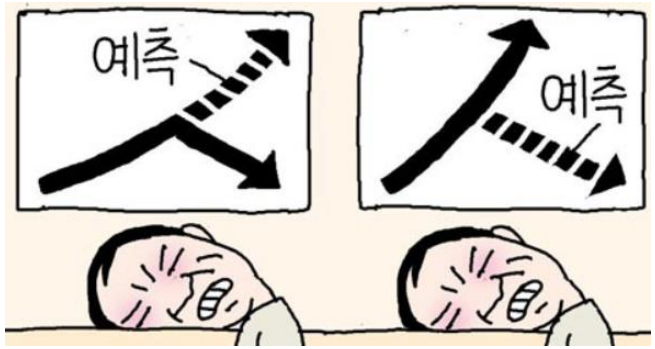
Difficult

Machine

Easy

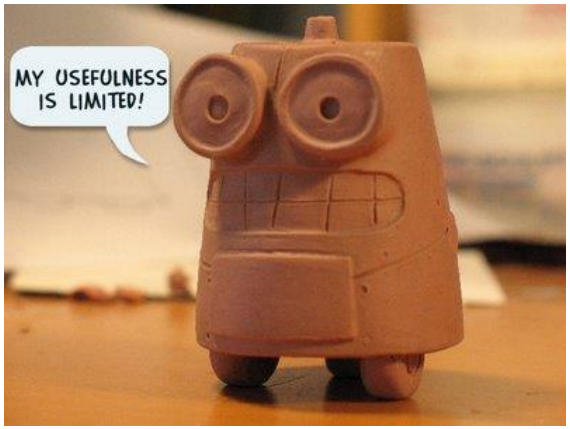
Difficult





인공 신경망의 도입

- **사람에겐 쉽지만 기계에겐 어려운 업무**
  - 손글씨 인식 (MNIST)
  - 음성 인식
  - 질의 응답
  - Etc…

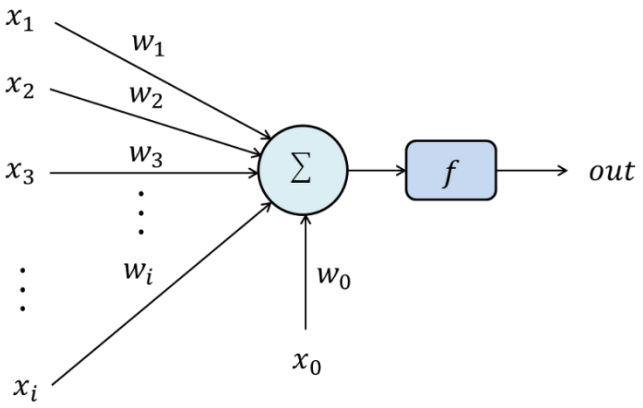


- **인간의 뇌를 모방**
  - 인공 신경망

Biological Inspiration

❑ Key Difference

Neuron & Synapse

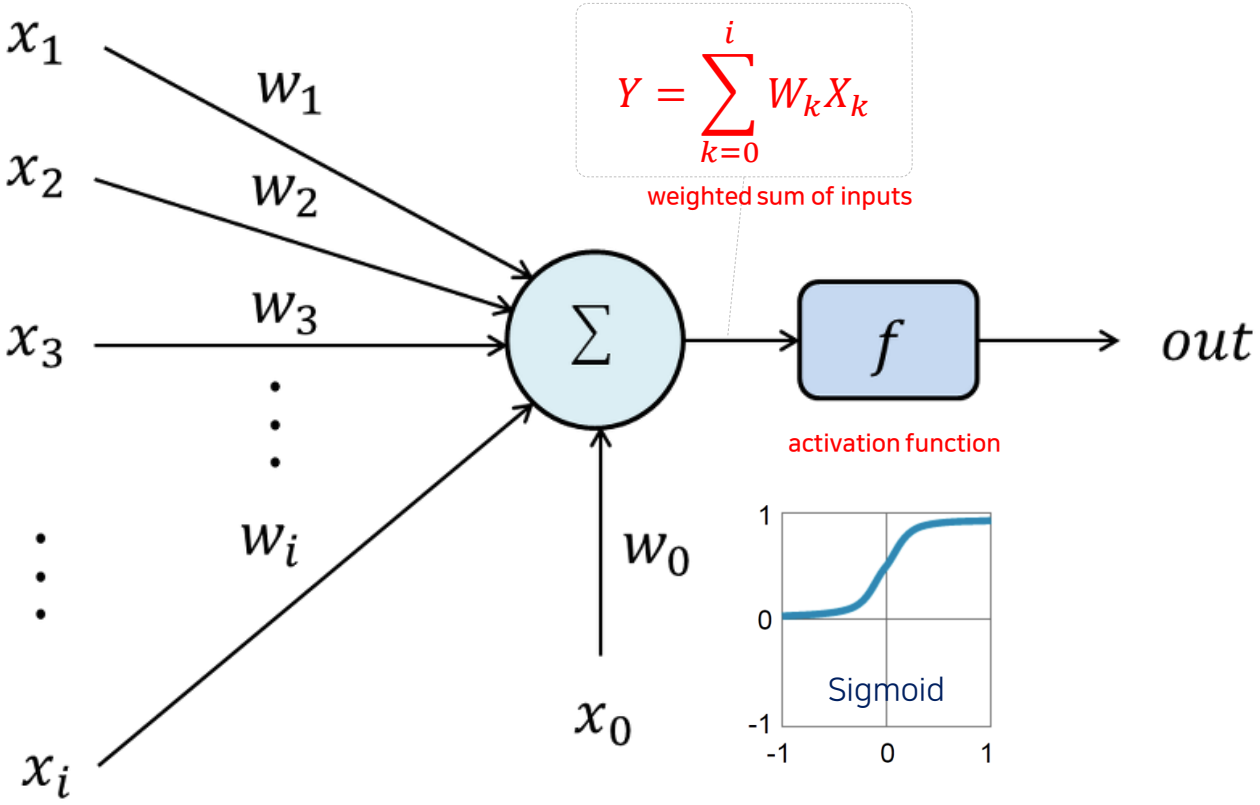


	Human (Biological) Neural Network	Artificial Neural Network
Parameter	Human brains have ~10^6 times synapses than artificial neural networks.	
Topology	Async	Synch
Learning algorithm	We don't know	Gradient Descent
Power consumption	Biological neural networks use very little power than artificial networks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

https://deeplearning.mit.edu/

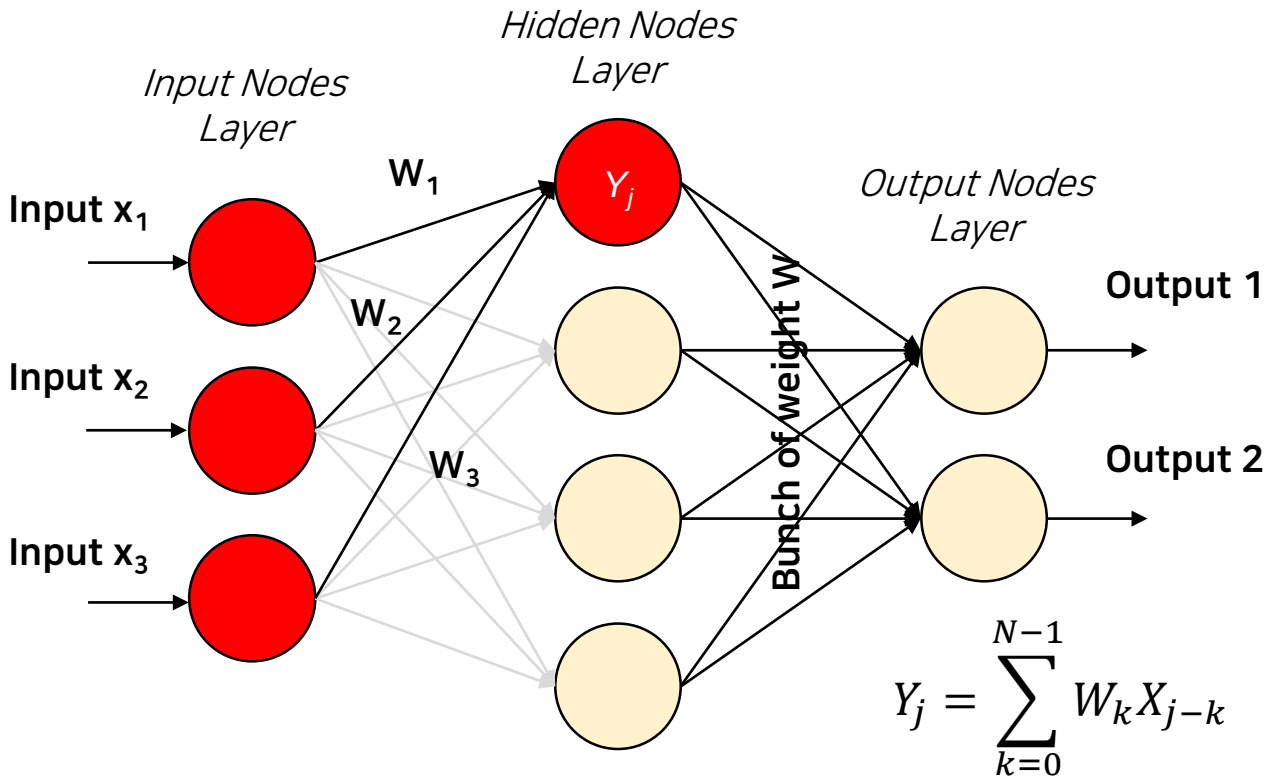
Perceptron

❑ 퍼셉트론 (Perceptron) 에서의 뉴런 (Neuron)



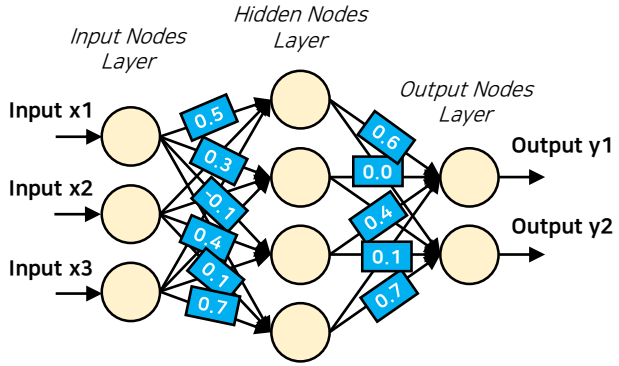
Multi Layer Perceptron (MLP)

❑ 다중 레이어 퍼셉트론

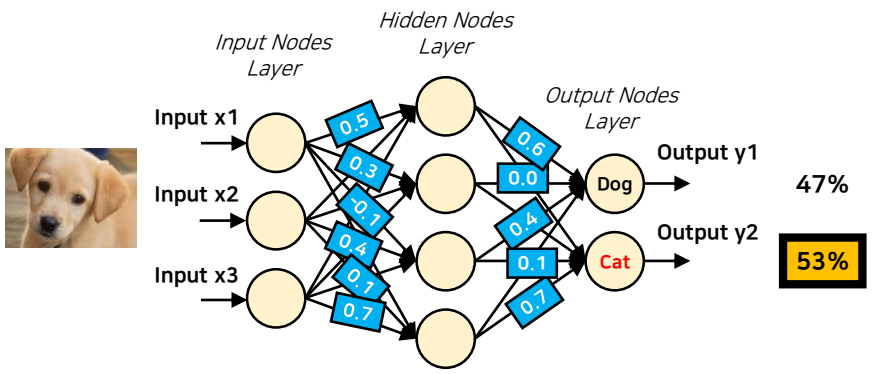


Simple Neural Network

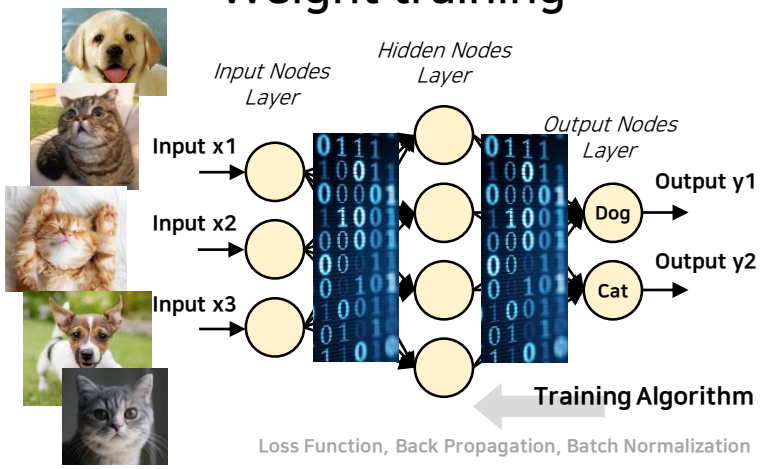
Random initialization



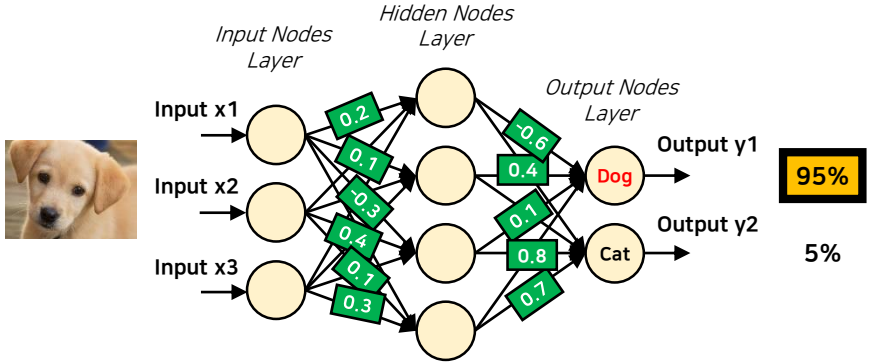
Let's try a dog w/ un-trained NN



Weight training



Let's try again inference



Deep (Artificial) Neural Network

❑ 심층 인공 신경망

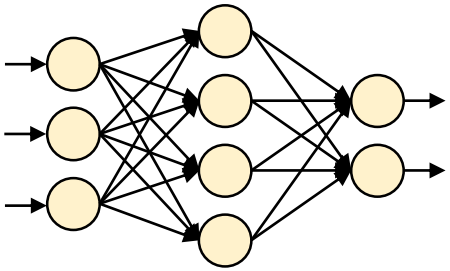
Available Big Data

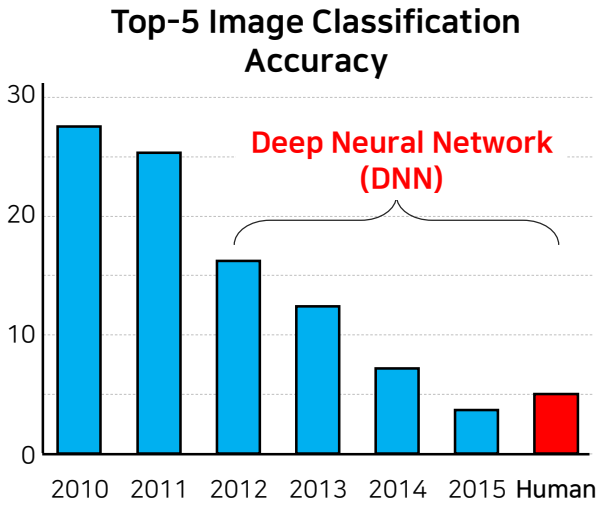
350M Images / day

300 hours videos /1 min

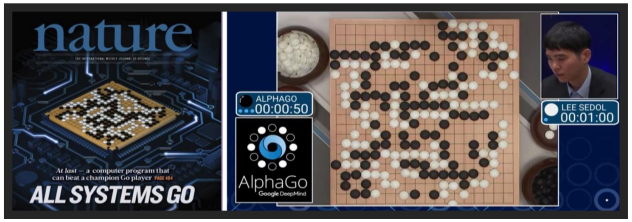
GPU Acceleration

New ML Techniques





[Russakovsky et al., IJCV 2015]



Terminology

❑ Key Machine Learning (ML) Terminology

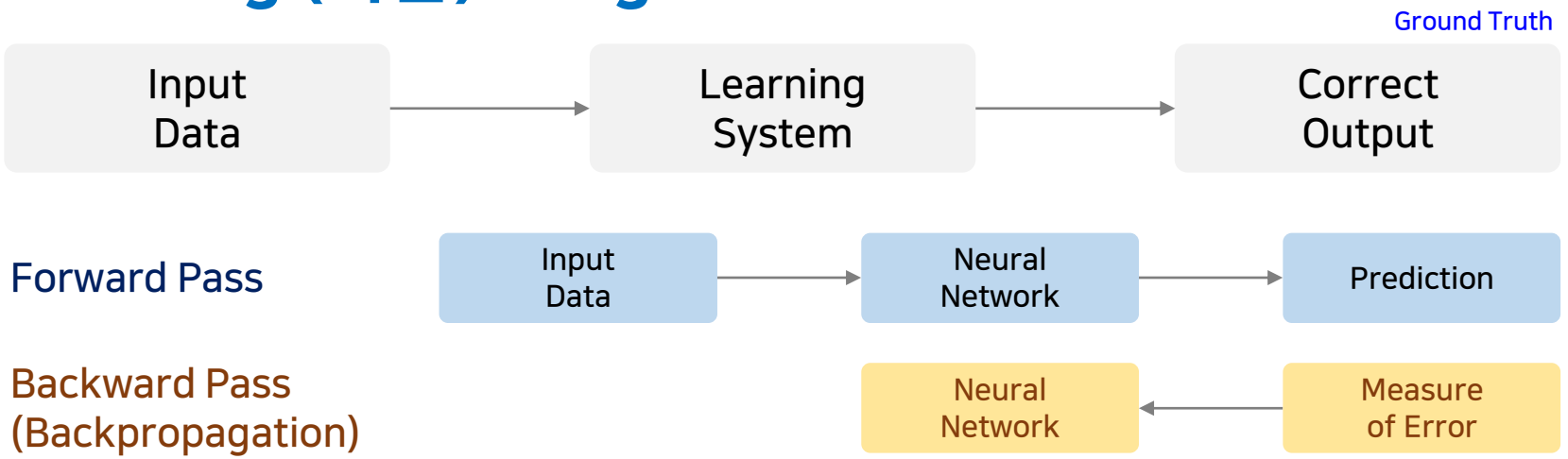
- 머신러닝 (Machine Learning) ?
  - ML 시스템은 입력을 결합하여 이전에 본 적이 없는 데이터를 적절히 예측하는 방법을 학습
- 라벨 (Labels)
  - 예측하는 항목 (변수 y) : 부동산 향후 가격, 사진 속 동물의 종류 등등
- 특성 (Features)
  - 입력 변수 (변수 x) : 스팸 감지 예 - 이메일의 단어, 보낸 사람 주소 및 시간
- 모델 (Models)
  - 모델은 특성과 라벨의 관계를 정의
  - 학습 (Training) : 모델을 만들거나 배우는 것을 의미
  - 추론 (Inference)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

Regression Classification

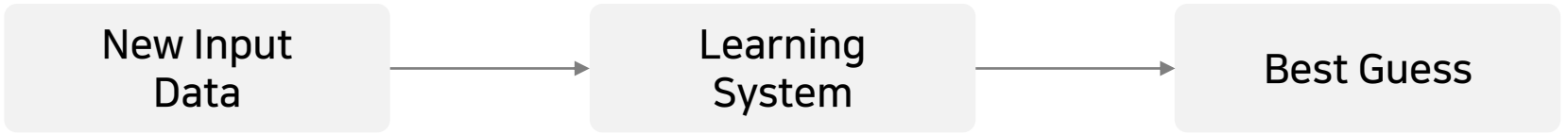
https://developers.google.com/machine-learning/crash-course/

Training & Inference (Test)

❑ Training (학습) Stage



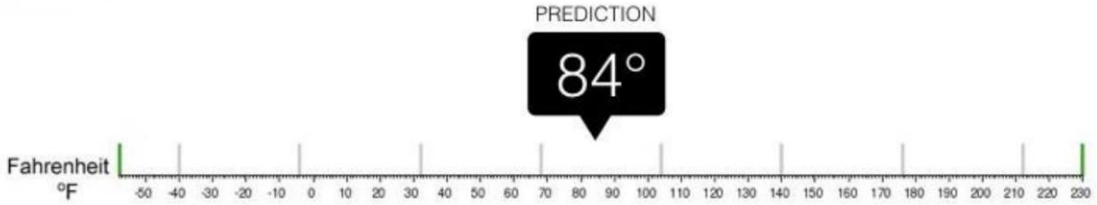
❑ Inference (추론) Stage



Regression vs Classification

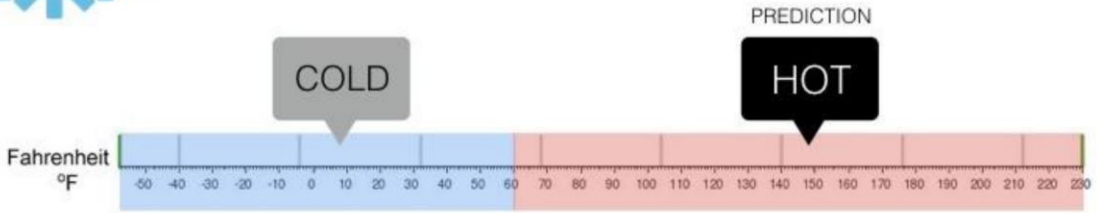
Regression

What is the temperature going to be tomorrow?



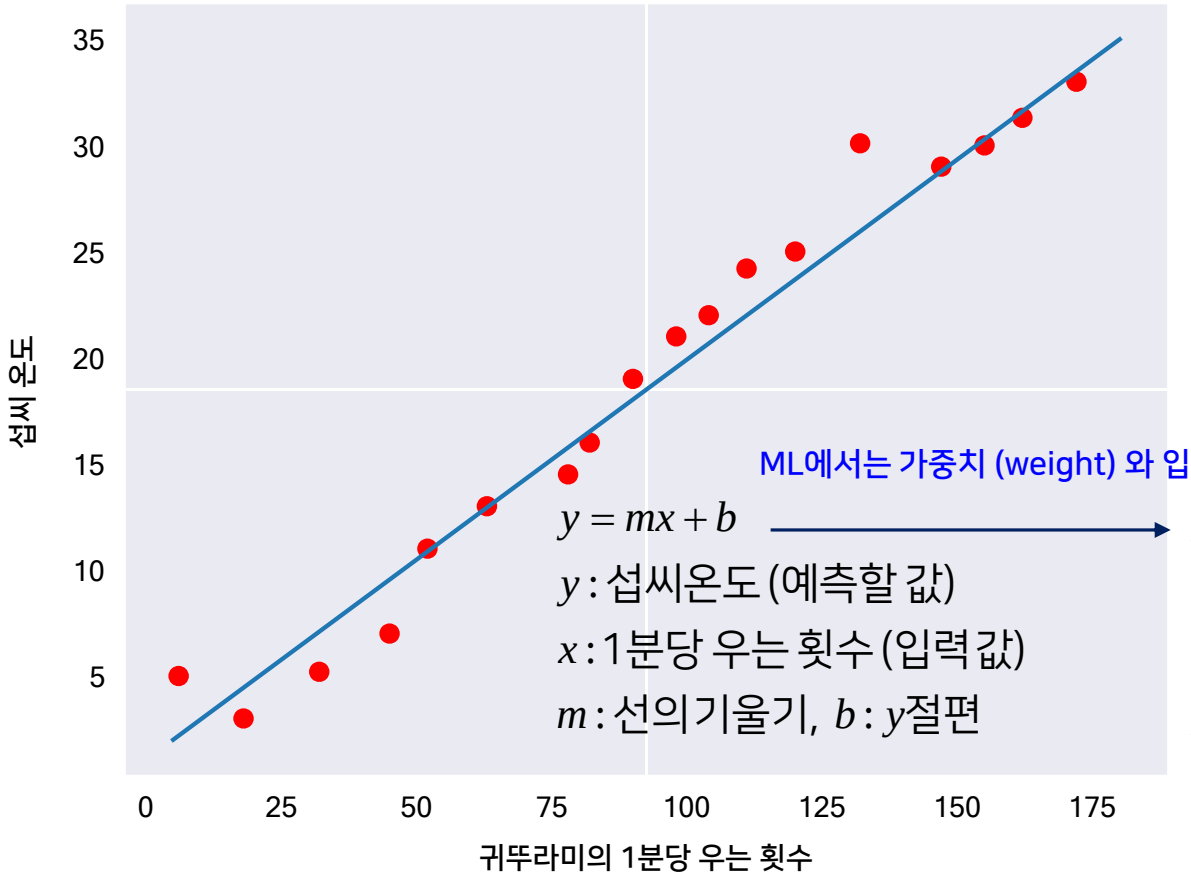
Classification

Will it be Cold or Hot tomorrow?



https://deeplearning.mit.edu/

Linear Regression



$y = mx + b$

y : 섭씨온도 (예측할 값)

x : 1분당 우는 횟수 (입력값)

m : 선의기울기, b : y절편

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.

ML에서는 가중치 (weight) 와 입력 특성(x) 기반 가설 (Hypothesis) 설정

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

입력 특성이 세 가지 일때,

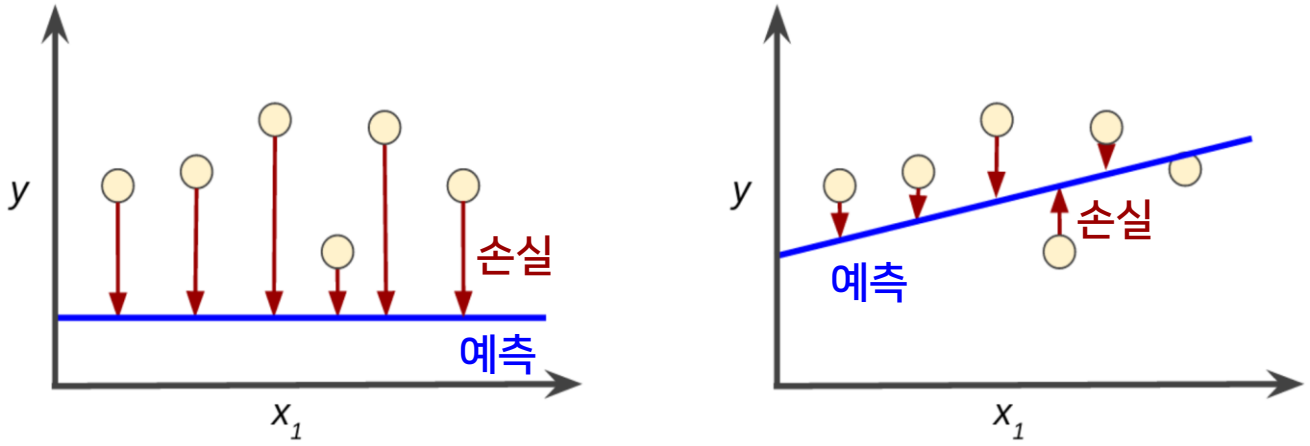
$$H(x,b) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

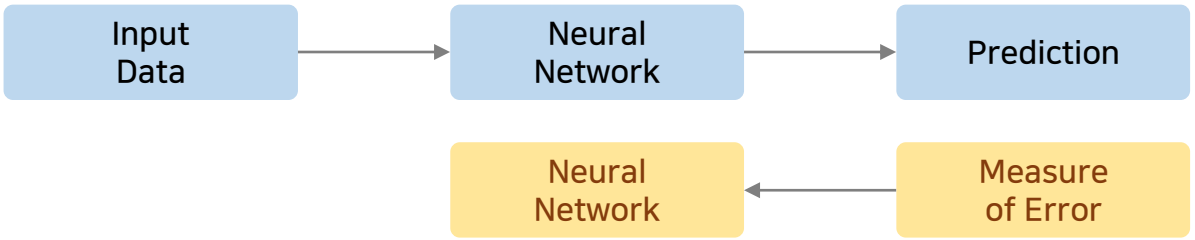
https://developers.google.com/machine-learning/crash-course/

Training & Loss

❑ 학습

- 올바른 가중치와 편향값을 학습(결정)하는 것





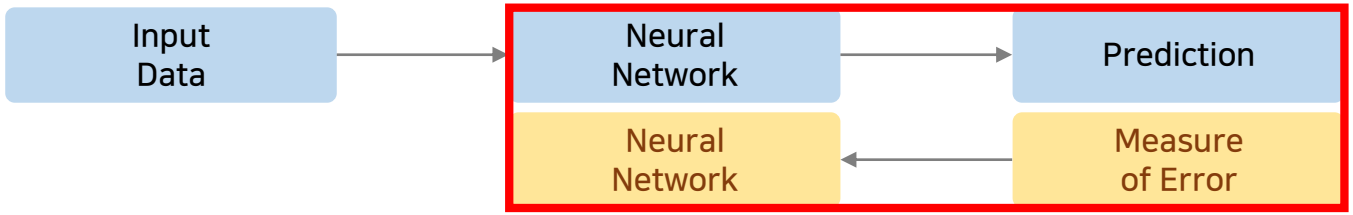
https://developers.google.com/machine-learning/crash-course/

Backpropagation

❑ 오차역전파법 기반 학습

Forward Pass

Backward Pass
(Backpropagation)



학습: 올바른 가중치와 편향 값을 학습(결정)하는 것

인공신경망 내의 가중치와 편향 값

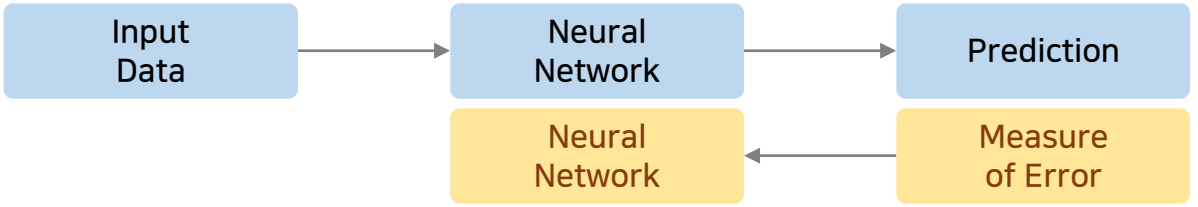
미분을 통해 각 파라미터가
오류에 주는 영향도 파악 가능

Backpropagation

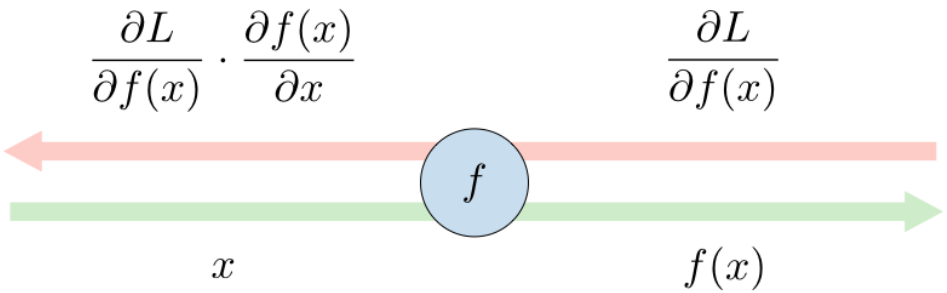
❑ Training (학습) Stage

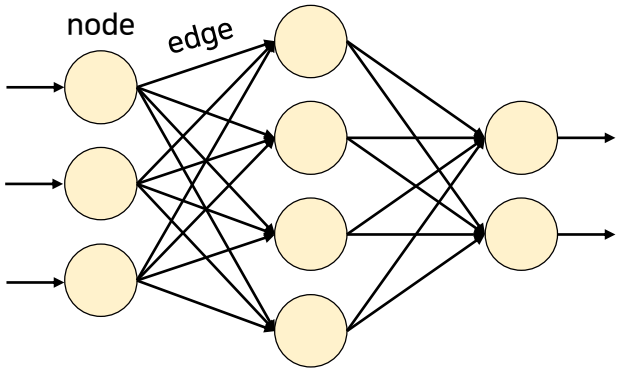
Forward Pass

Backward Pass
(Backpropagation)



계산 그래프 (복수의 Node와 Edge)



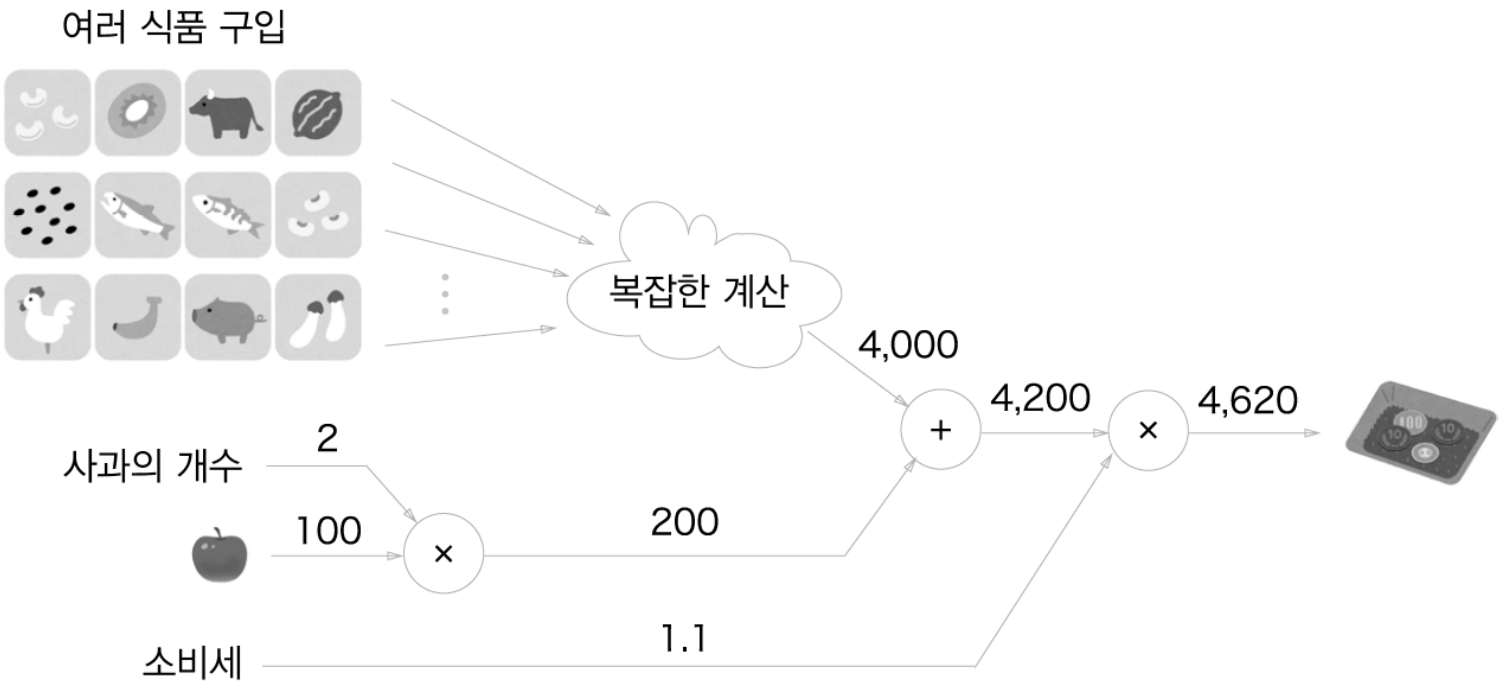


TensorFlow도 계산그래프와 동일한 의미

Backpropagation

❑ 왜 계산 그래프로 문제를 푸는가?

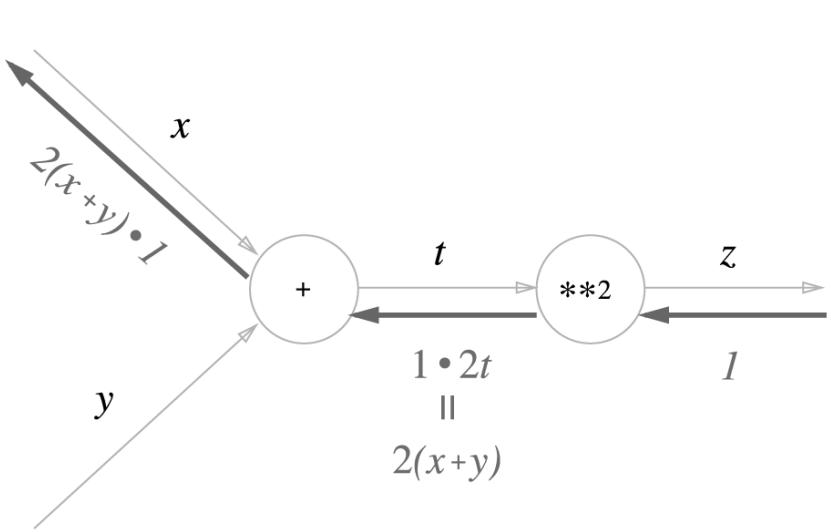
- 국소적 계산으로 문제 단순화 가능

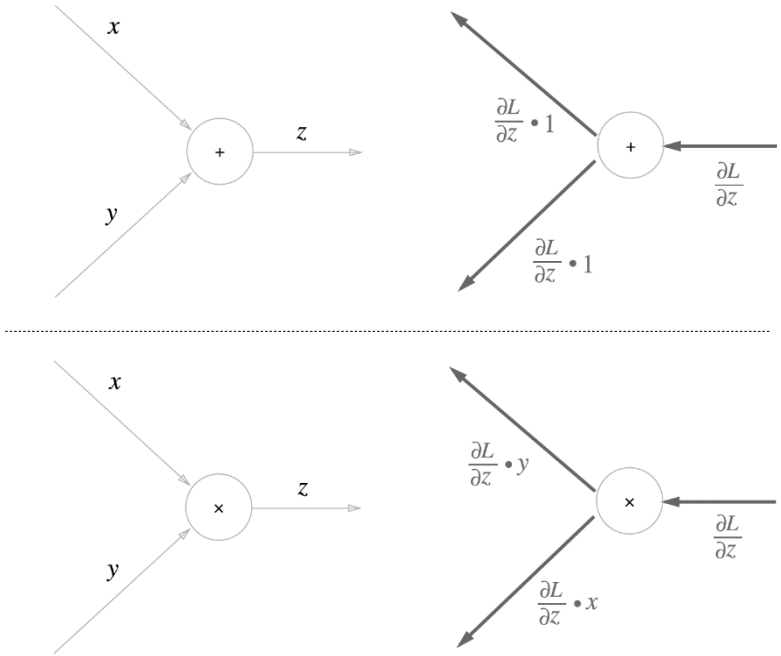


Backpropagation

❑ 왜 계산 그래프로 문제를 푸는가?

- 역전파를 통해 미분을 효율적으로 계산 가능

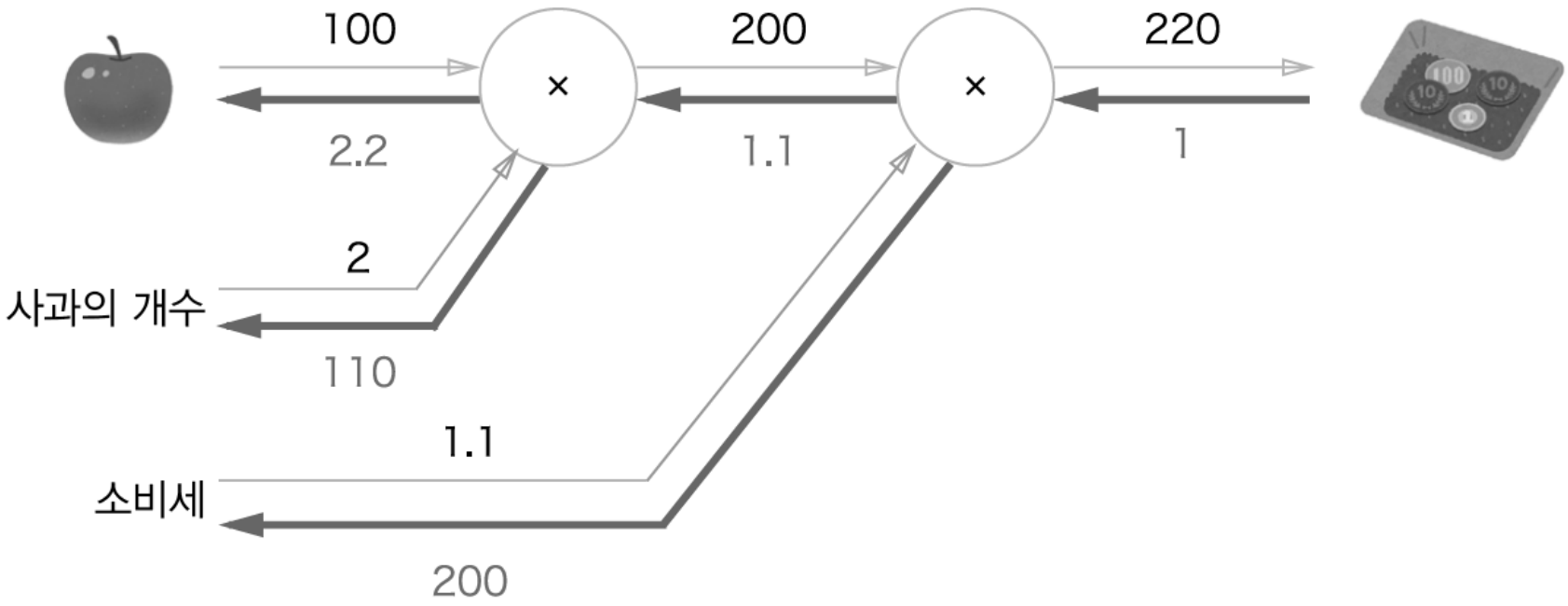




Backpropagation

❑ 왜 계산 그래프로 문제를 푸는가?

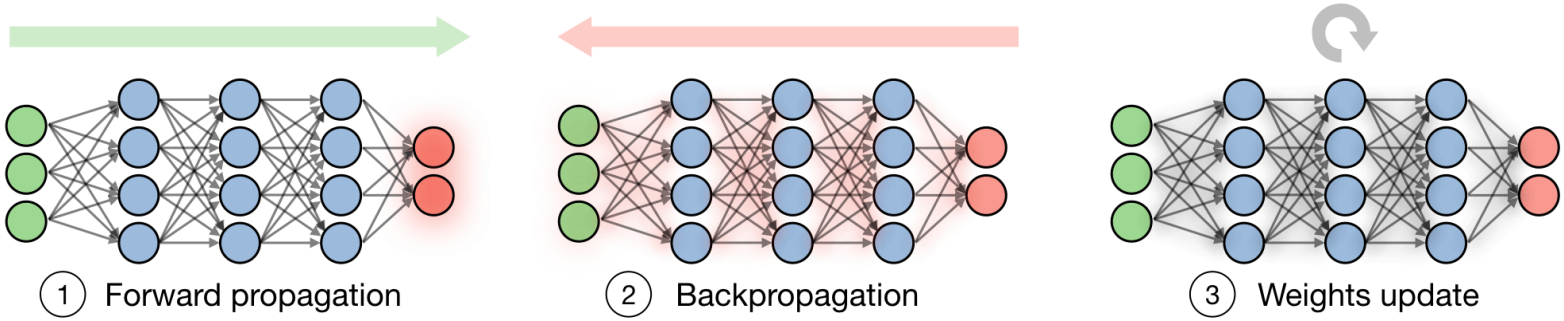
- 역전파를 통해 미분을 효율적으로 계산 가능



Backpropagation

❑ 가중치 업데이트

- 1단계 : 트레이닝 데이터의 배치를 가져와 순전파를 진행하여 손실을 계산
- 2단계 : 각 가중치와 관련하여 그레디언트 손실값을 얻기 위해 역전파
- 3단계 : 네트워크의 가중치를 업데이트 하기 위해서 그레디언트를 사용



$$\theta = \theta - \eta \nabla_\theta J(\theta)$$

θ : Parameter η : Learning Rate $J(\theta)$: Loss Function

$\nabla_\theta J(\theta)$: Gradient of Loss

https://stanford.edu/~shervine/l/ko/teaching/cs-230/cheatsheet-deep-learning-tips-and-tricks

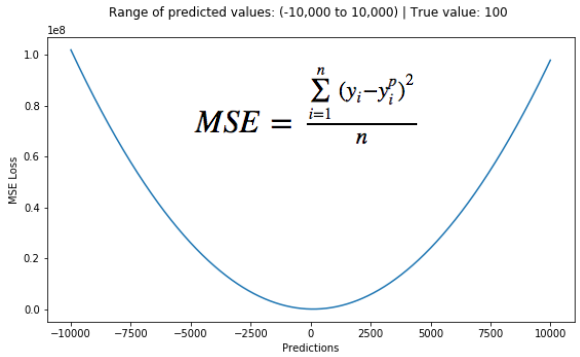
Training & Loss

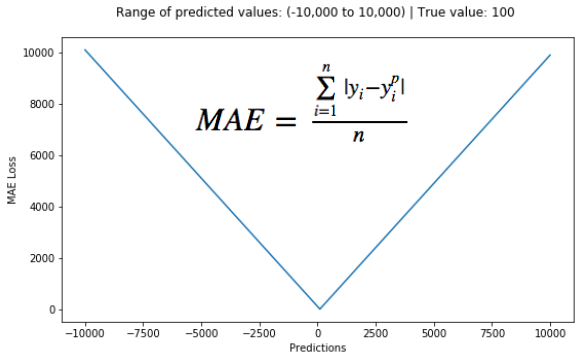
- **손실 함수 : 신경망이 학습할 수 있도록 해주는 지표**
  - 머신러닝 모델의 최종적인 목적은 높은 정확도를 끌어내는 매개변수(가중치, 편향)를 찾는 것
  - 정확도와는 달리 손실 함수는 매개변수의 변화에 따라 연속적으로 변화
  - 손실 함수와는 달리 정확도는 매개변수의 변화에 둔감하고, 또한 변화가 있다하여도 불연속적으로 변화 → 미분 불가
  - 미분이 되지 않으면 최적화를 할 수 없으므로 정확도가 아닌 손실 함수를 지표로 삼아 학습

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|	Error²
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

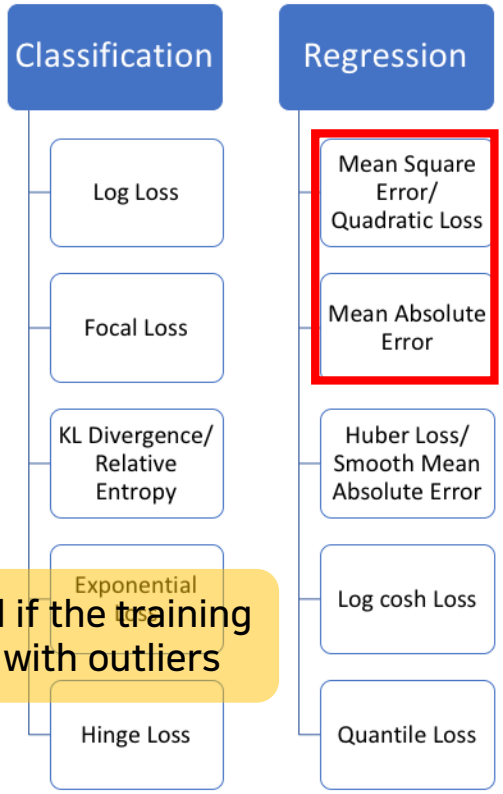
MAE vs. RMSE for cases with outliers in data

ID	Error	|Error|	Error²
1	0	0	0
2	1	1	1
3	1	1	1
4	-2	2	4
5	15	15	225

outlier

MAE: 3.8 RMSE: 6.79

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

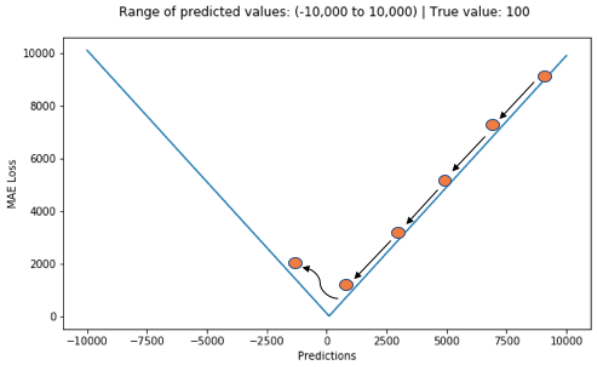


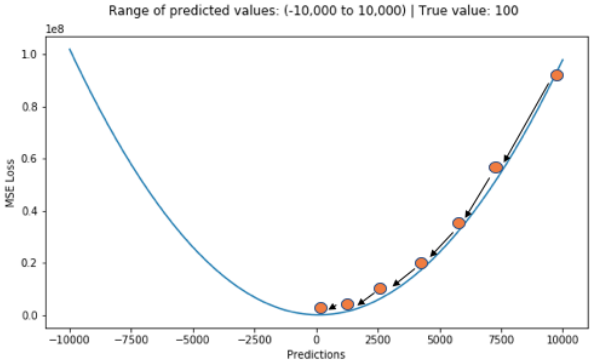
https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0
https://nbviewer.jupyter.org/github/groverpr/Machine-Learning/blob/master/notebooks/05_Loss_Functions.ipynb

5 Regression Loss Functions

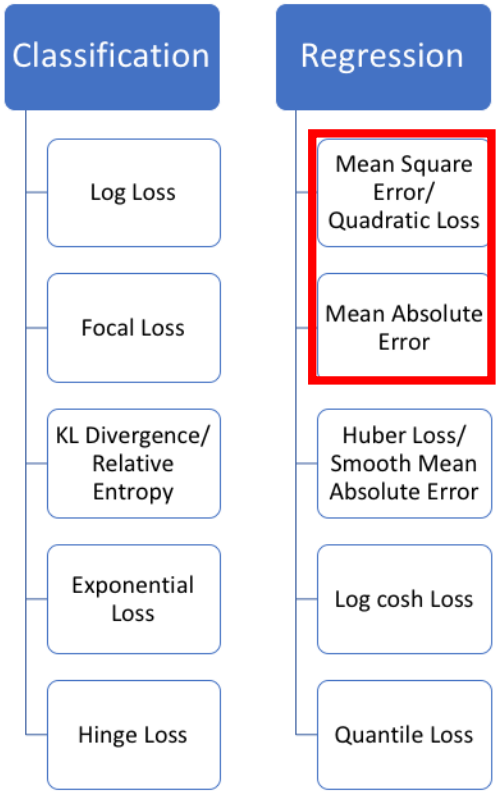
❑ MSE vs MAE

- 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.



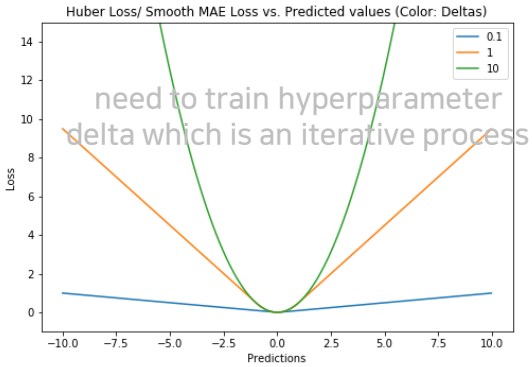
https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0
https://nbviewer.jupyter.org/github/groverpr/Machine-Learning/blob/master/notebooks/05_Loss_Functions.ipynb

5 Regression Loss Functions

❑ Alternative Solutions

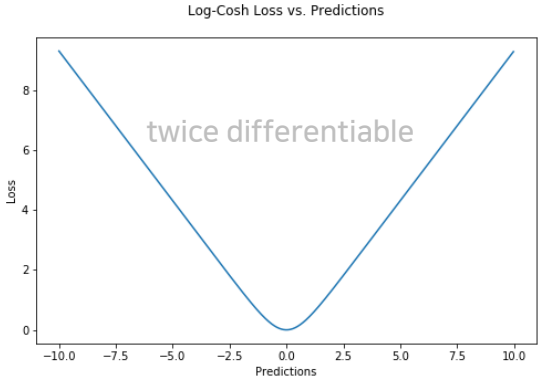
Huber Loss (Smooth Mean Absolute Error)

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

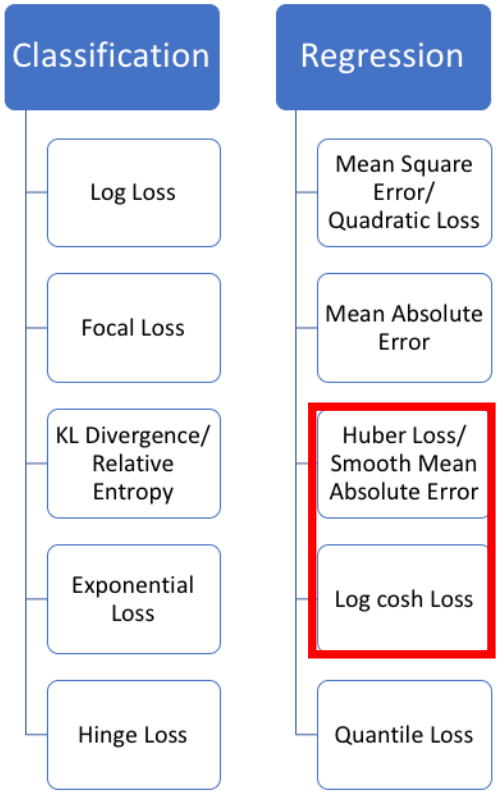


Log-Cosh Loss

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



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.

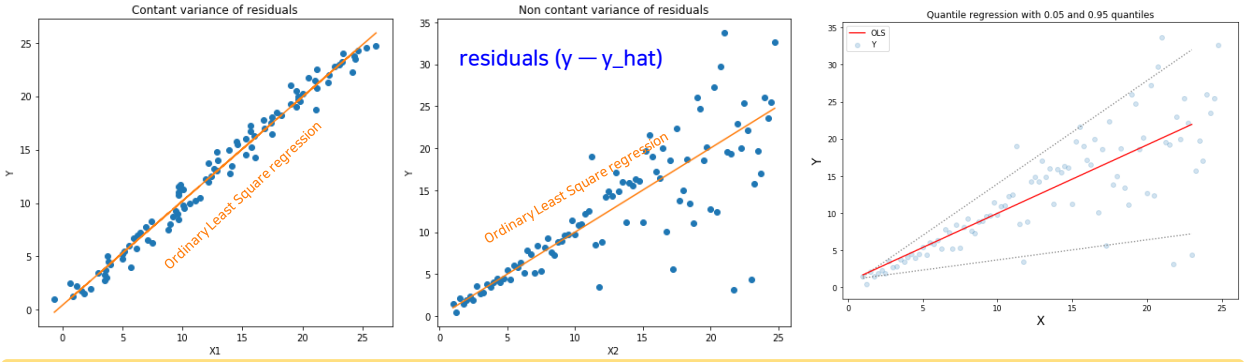


https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0
https://nbviewer.jupyter.org/github/groverpr/Machine-Learning/blob/master/notebooks/05_Loss_Functions.ipynb

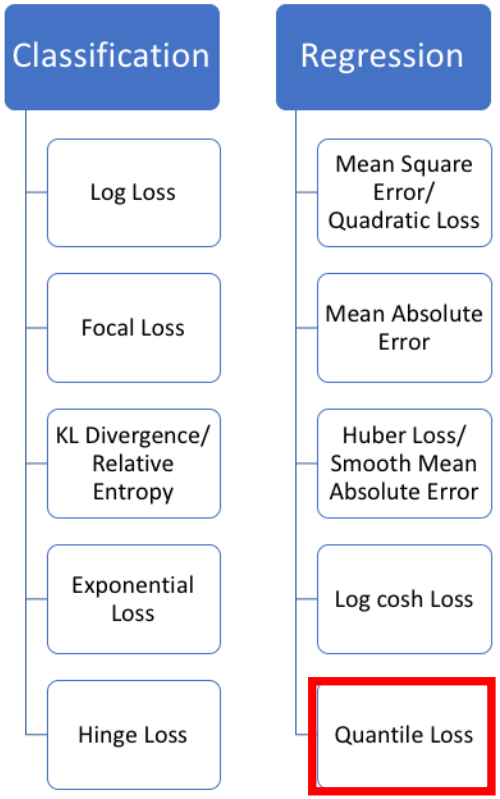
5 Regression Loss Functions

❑ 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



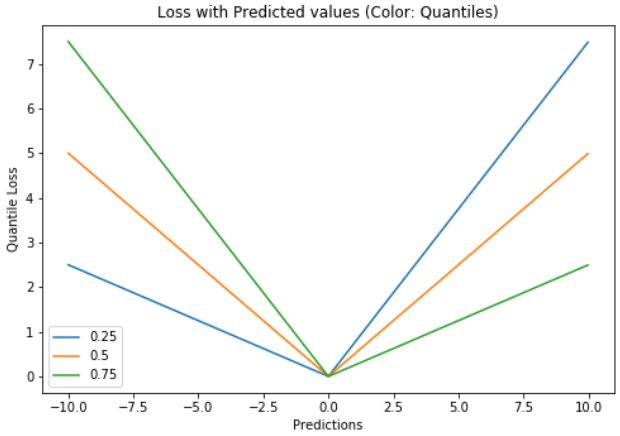
https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0
https://nbviewer.jupyter.org/github/groverpr/Machine-Learning/blob/master/notebooks/05_Loss_Functions.ipynb

5 Regression Loss Functions

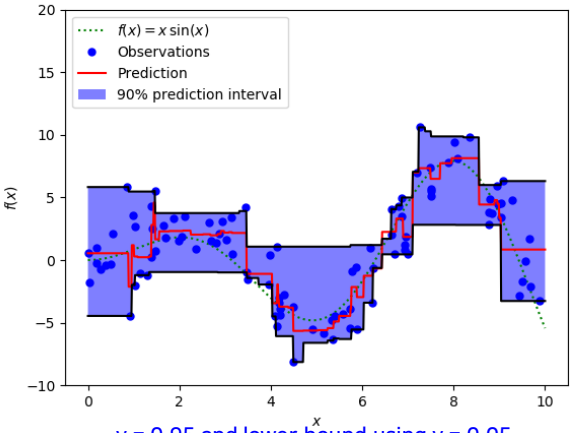
❑ Quantile Loss

- For example, a quantile loss function of quantile(γ) = 0.25 gives more penalty to overestimation and tries to keep prediction values a little below median

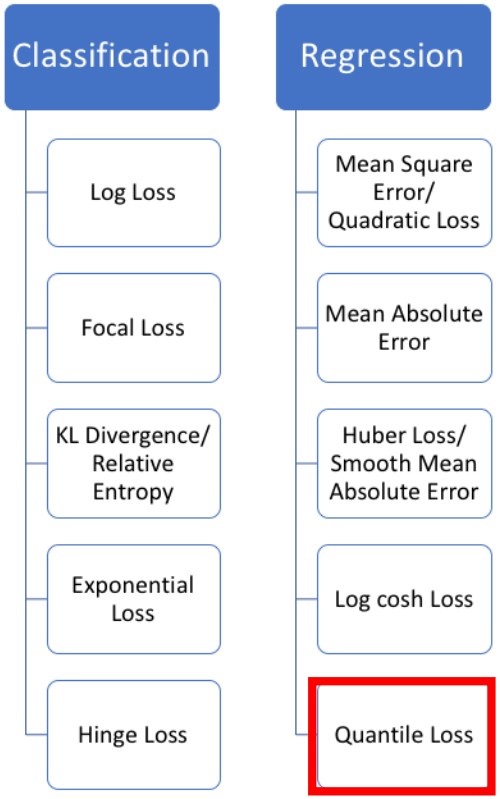
$$L_\gamma(y, y^p) = \sum_{i=y_i<y_i^p} (\gamma - 1).\,|y_i - y_i^p| + \sum_{i=y_i \geq y_i^p} (\gamma).\,|y_i - y_i^p|$$



Sklearn implementation for gradient boosted tree regressors



γ = 0.95 and lower bound using γ = 0.05

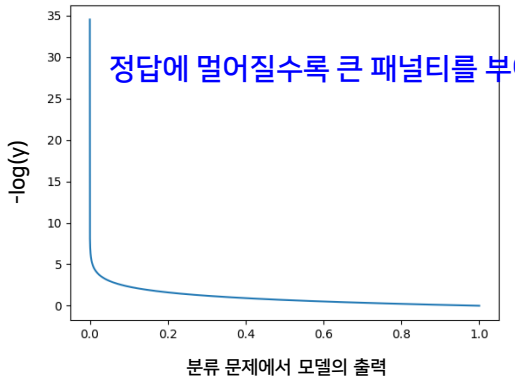


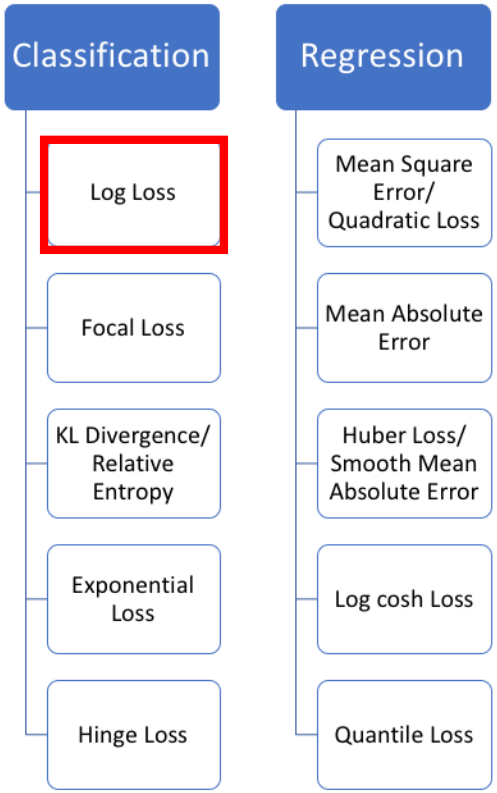
https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0
https://nbviewer.jupyter.org/github/groverpr/Machine-Learning/blob/master/notebooks/05_Loss_Functions.ipynb

Classification Loss Functions

❑ Cross Entropy Loss (Log Loss)

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





https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0
https://kolikim.tistory.com/36

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.

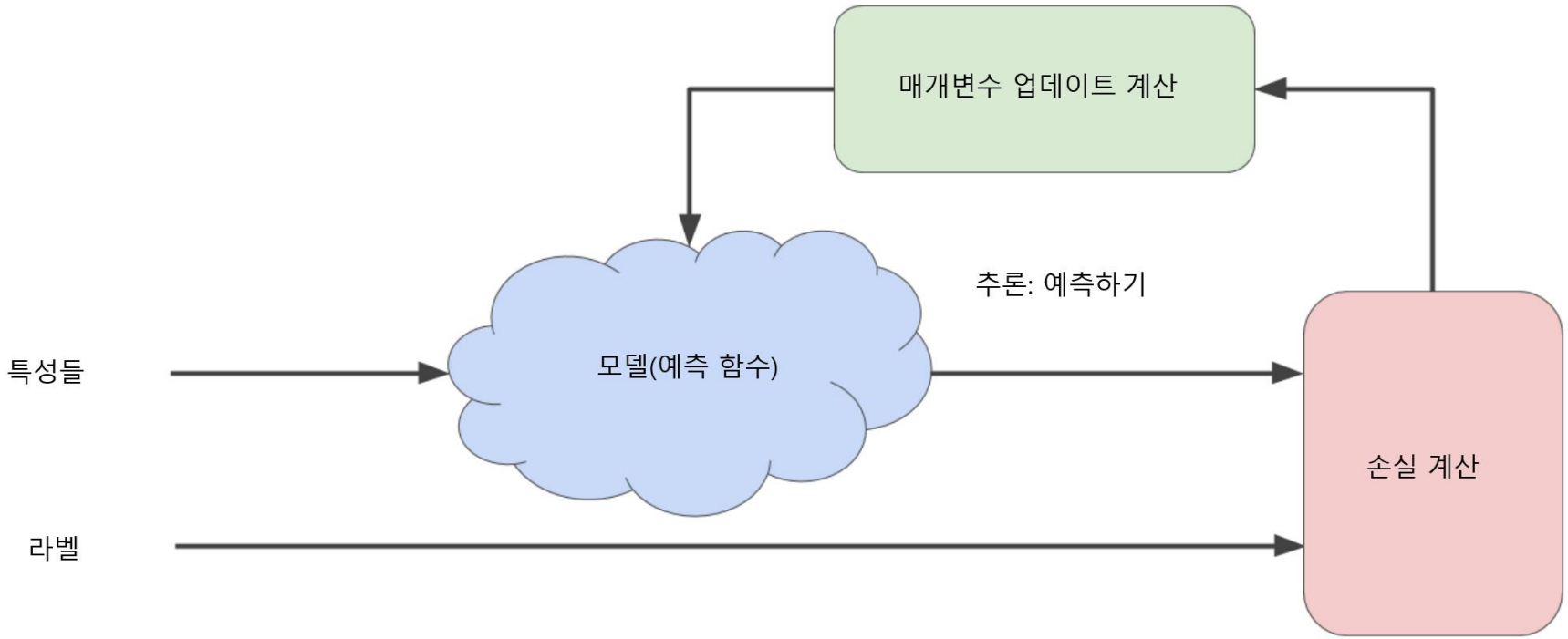
softmax_cross_entropy(...): Creates a cross-entropy loss using tf.nn.softmax_cross_entropy_with_logits_v2.

sparse_softmax_cross_entropy(...): Cross-entropy loss using tf.nn.sparse_softmax_cross_entropy_with_logits.

https://www.tensorflow.org/api_docs/python/tf/losses

Reducing Loss

❑ 경사하강법 (Gradient Descent Algorithm)

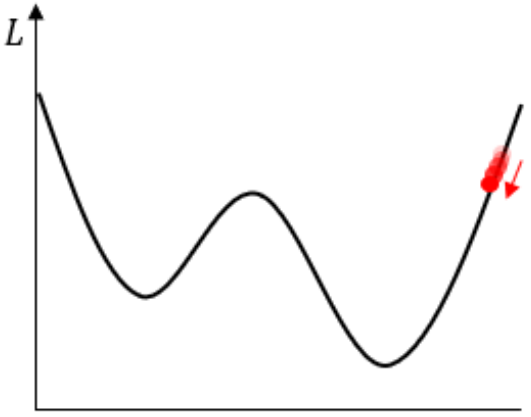


https://developers.google.com/machine-learning/crash-course/

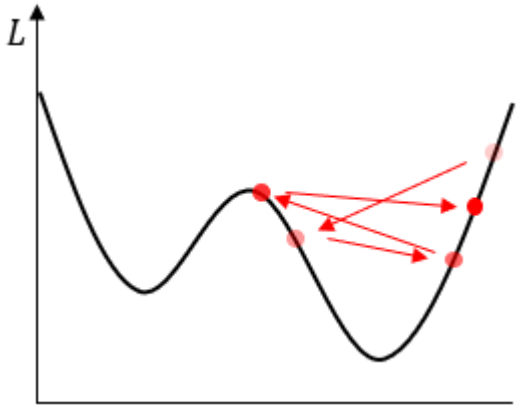
Reducing Loss

❑ 경사하강법 (Gradient Descent Algorithm)

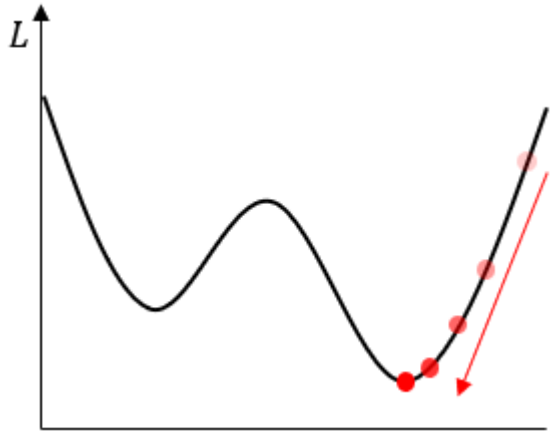
- 경사하강법 알고리즘은 기울기에 학습률 (Learning Rate) 또는 보폭이라 불리는 스칼라를 곱하여 다음 지점을 결정



학습률이 너무 작을 때



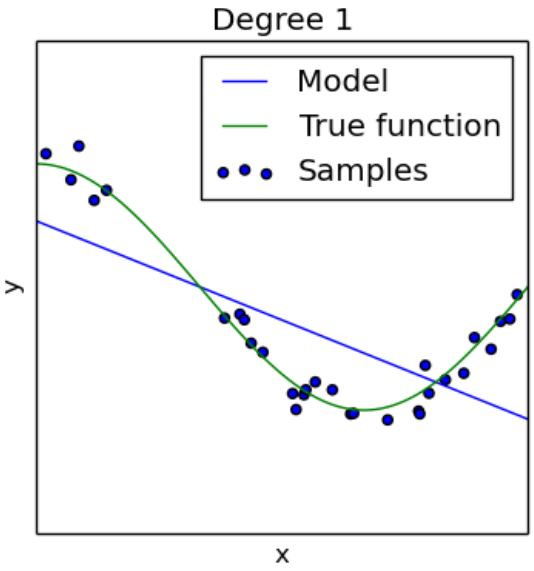
학습률이 너무 클 때



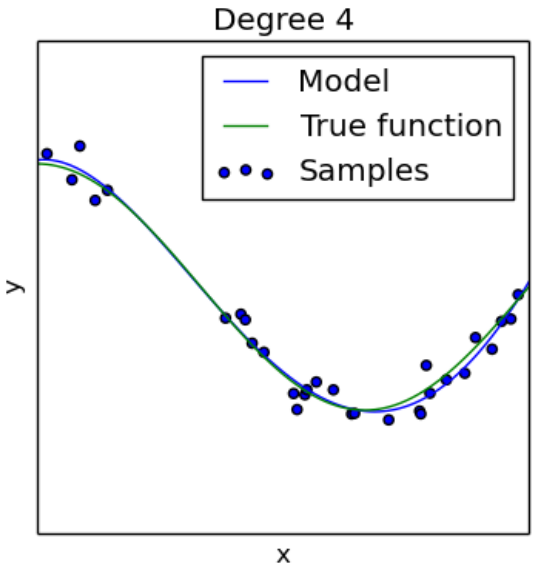
학습률이 너무 적당할 때

Challenge of Learning

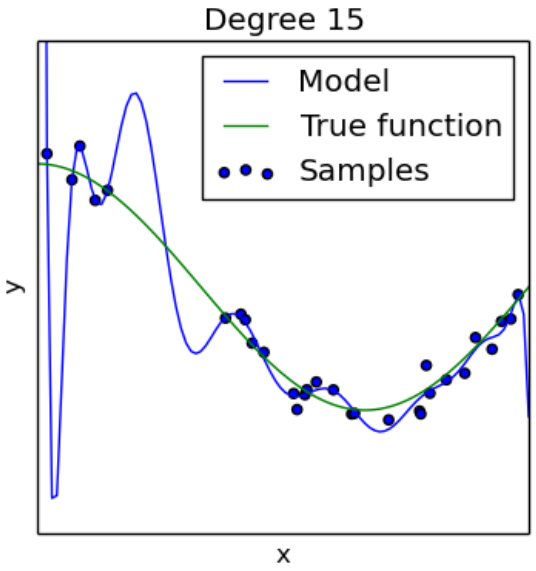
❑ 적당히…



Under-fitted Model



Balanced-Model



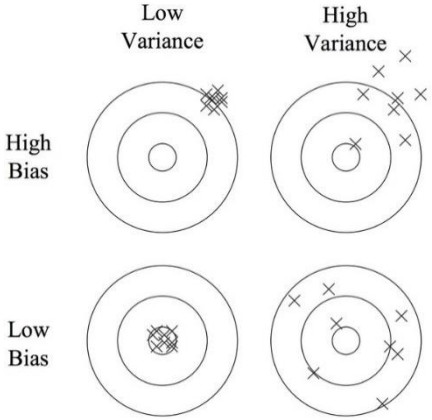
Over-fitted Model

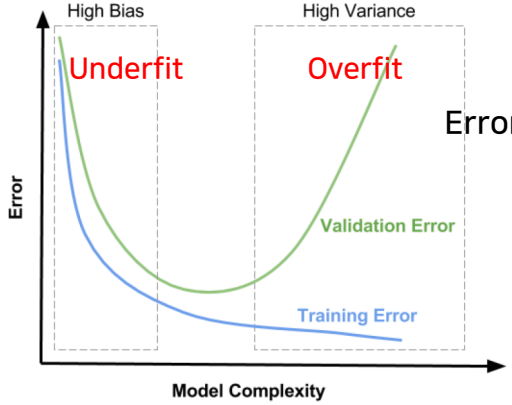
https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

Challenge of Learning

❑ 편향-분산 (Bias-Variance) Trade-off

- 편향 :학습 알고리즘에서 잘못된 가정을 했을 때 발생하는 오차
- 분산 : 트레이닝 셋에 내재된 작은 변동 때문에 발생하는 오차
- 모델을 선택할 때,
  - 트레이닝 데이터의 규칙을 정확하게 포착하는 것 뿐만이 아니라,
  - 보이지 않는 범위에 대해서 일반화(generalization)까지 하는 것이 이상적 (But, 불가능)





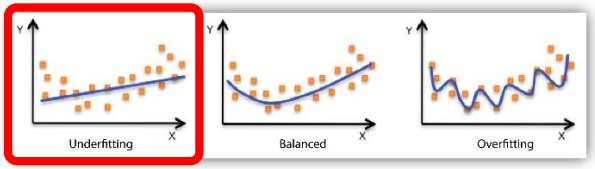
Error(X) = noise(X) + bias(X) + variance(X)

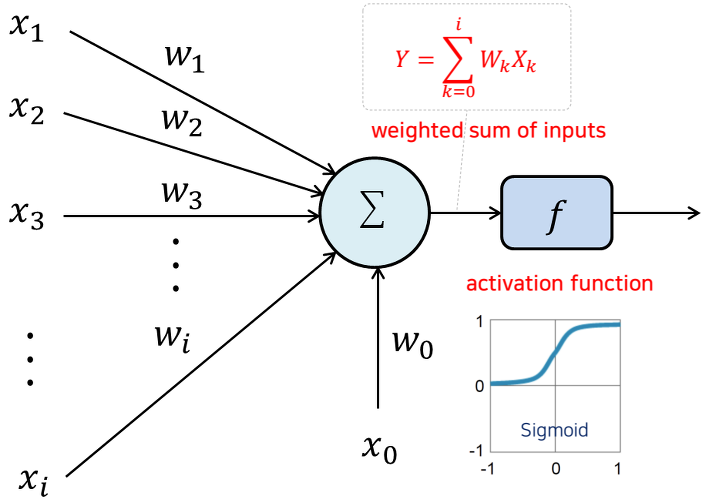
위키피디아
https://www.quora.com/What-is-the-best-way-to-explain-the-bias-variance-trade-off-in-layman%E2%80%99s-terms

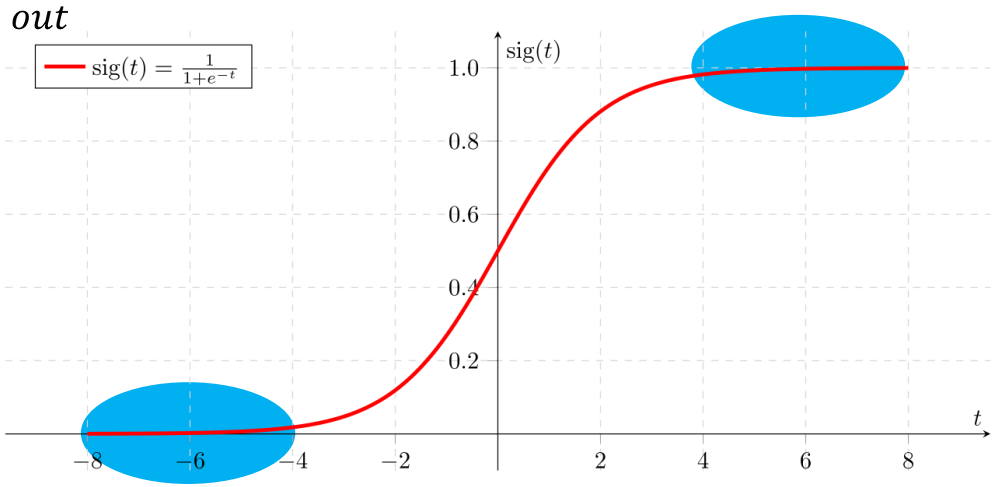
Challenge of Learning

❑ Underfitting

- 원인 : 기울기가 0인 지점 존재



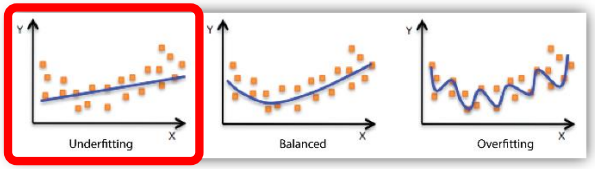


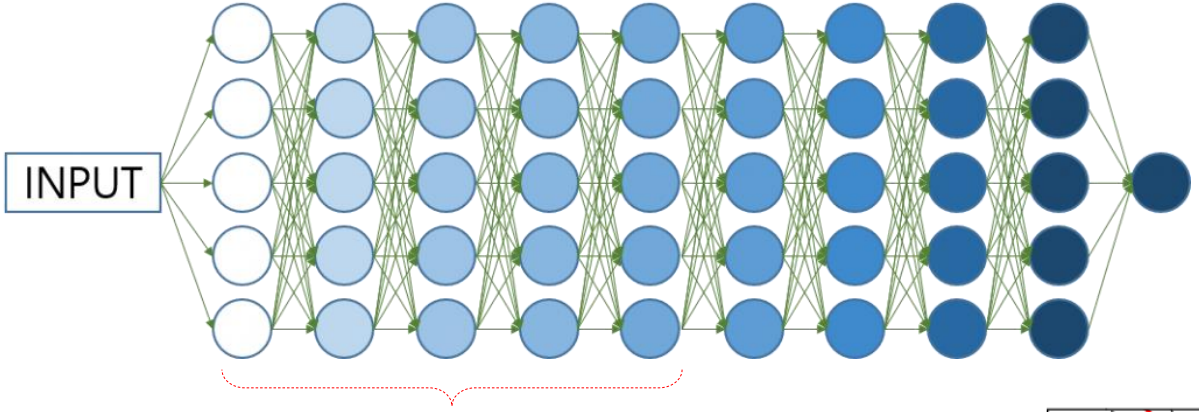


Challenge of Learning

❑ Underfitting

- 기울기 소실 (Vanishing Gradient)

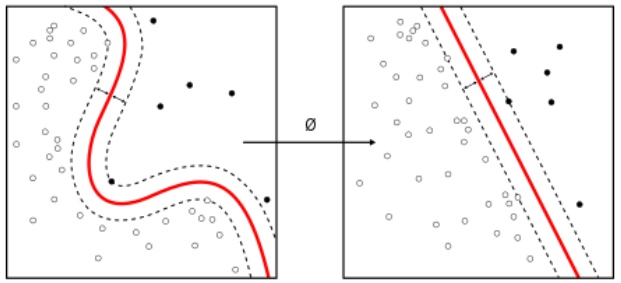




뒤로 갈수록 학습이 잘 안됨

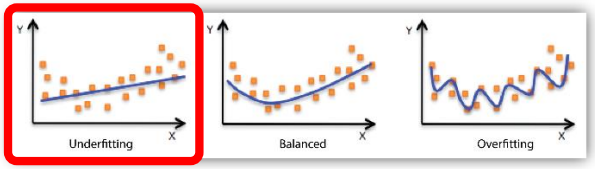
Weight의 업데이트 = 에러 낮추는 방향 X 학습률 X ~~기울기~~

$$-\gamma \nabla F(a^n) \qquad - \qquad \gamma \quad \nabla F(a^n)$$

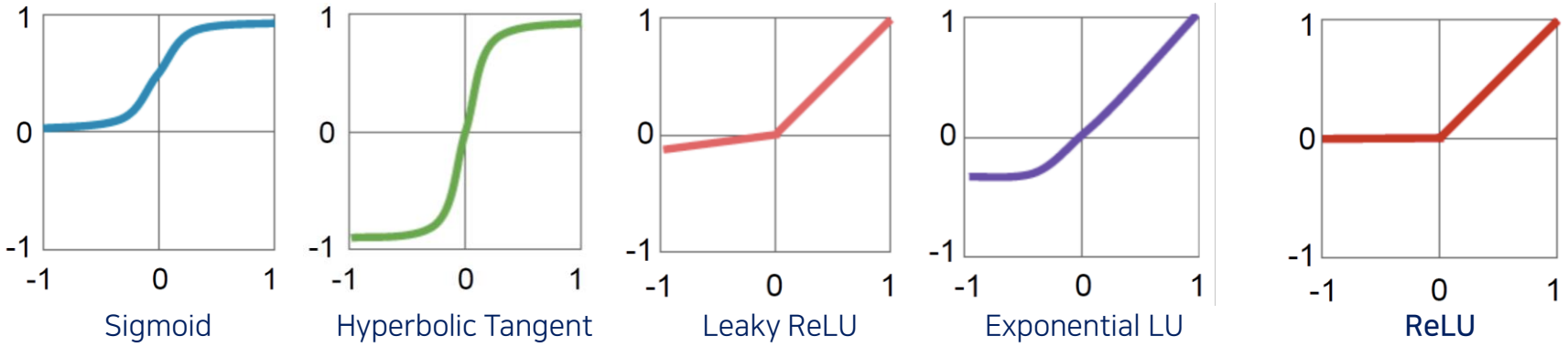


Challenge of Learning

❑ Underfitting



- 해결
  - 활성화 함수의 변경으로 해결
- 다양한 활성화 함수

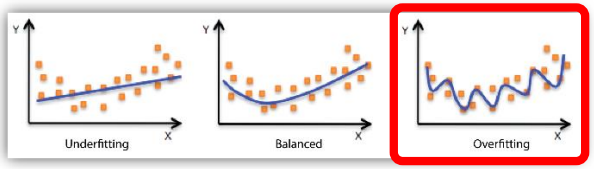


기울기 소실 존재

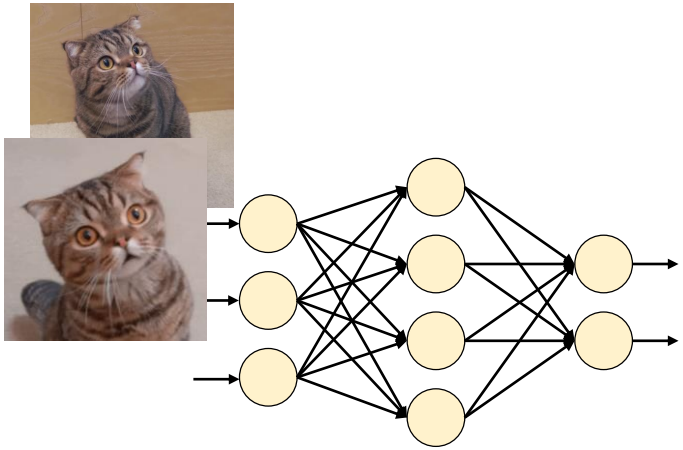
H/W로 만들기 용이

Challenge of Learning

❑ Overfitting



- 원인
  - 매개변수가 많고 표현력이 높은 모델
  - 훈련데이터가 적을 때



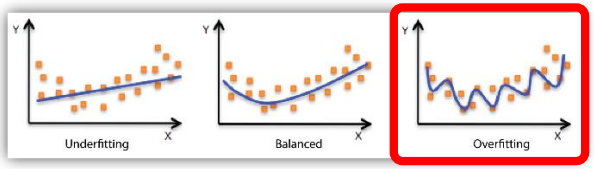
뚱뚱해서 고양이 아님

갈색이라 고양이 아님

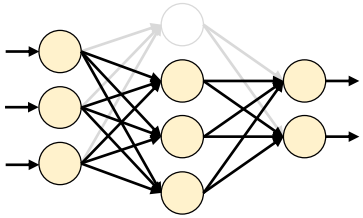
귀쳐져서 고양이 아님

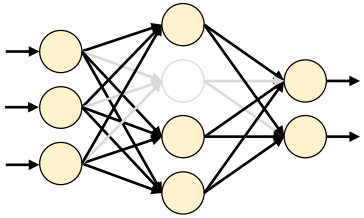
Challenge of Learning

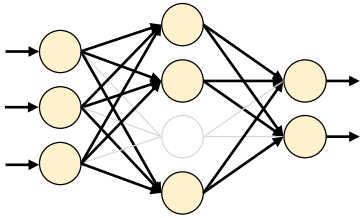
❑ Overfitting



- 해결
  - Dropout







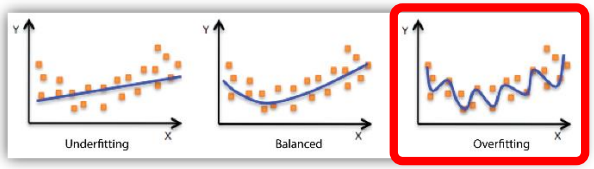
뚱뚱해서 고양이 아님

갈색이라 고양이 아님

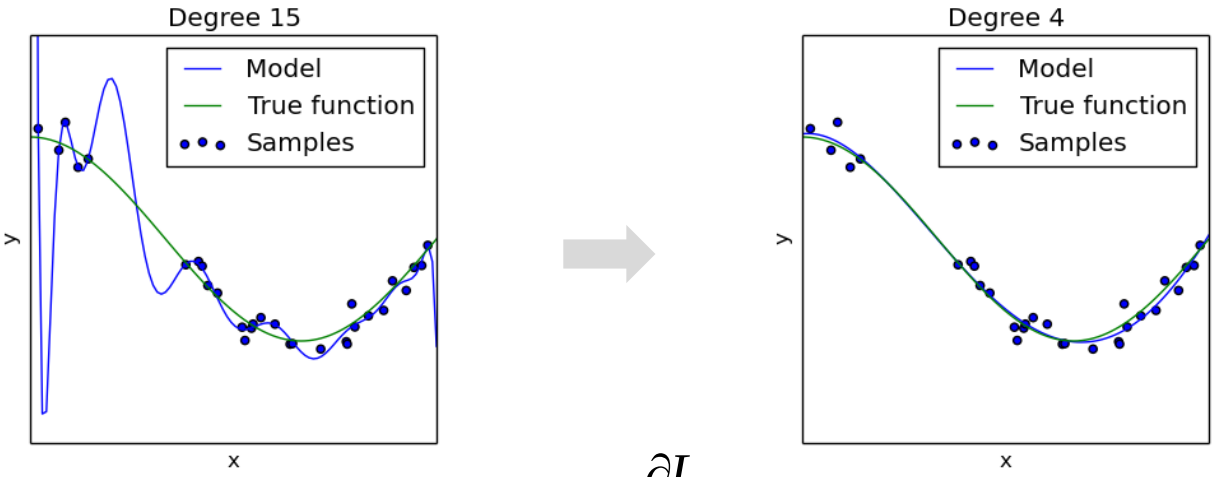
귀쳐져서 고양이 아님

Challenge of Learning

❑ Overfitting



- 해결
  - Weight Decay (Regularization)

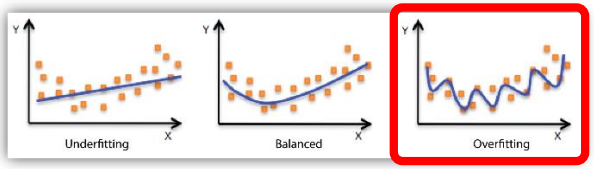


$$W \leftarrow W - \eta\left(\frac{\partial L}{\partial W} + \lambda W\right)$$

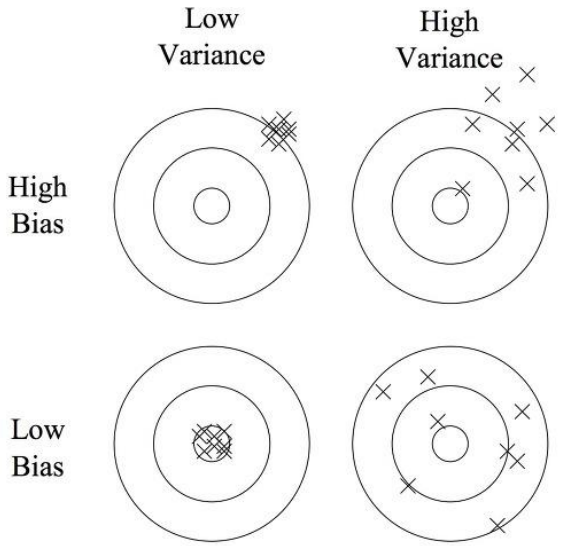
기울기의 급격한 변화를 방지

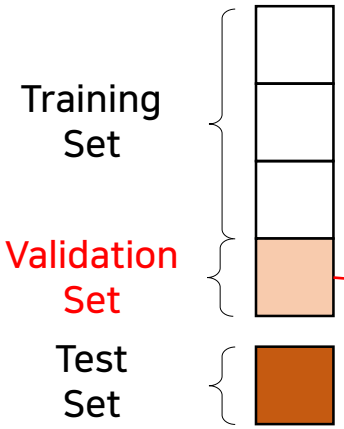
Challenge of Learning

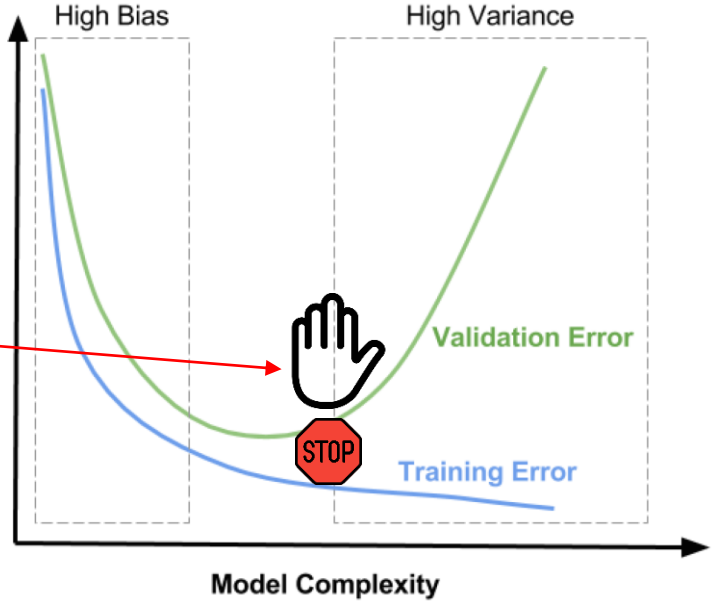
❑ Overfitting



- 해결
  - Validation Set







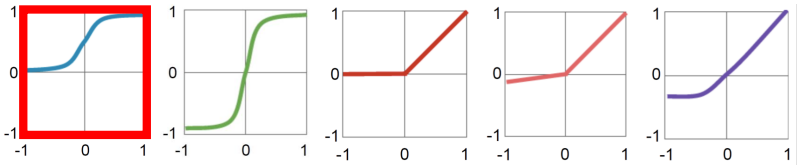
Error(X) = noise(X) + bias(X) + variance(X)

https://www.quora.com/What-is-the-best-way-to-explain-the-bias-variance-trade-off-in-layman%E2%80%99s-terms

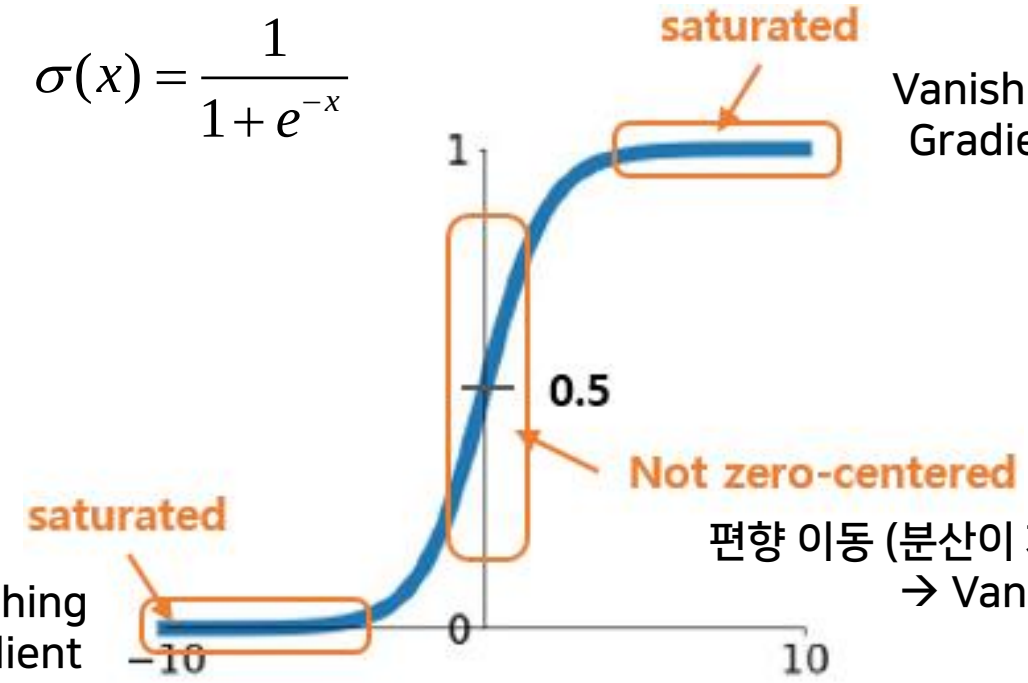
Challenge of Learning

❑ Activation Functions

- Sigmoid



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



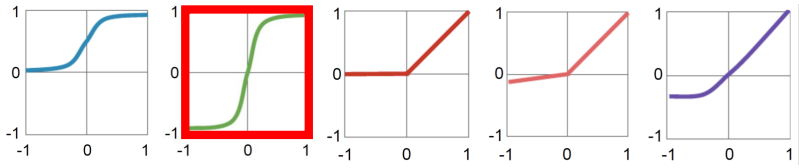
편향 이동 (분산이 계속 커져) → 출력이 0 or 1
→ Vanishing Gradient

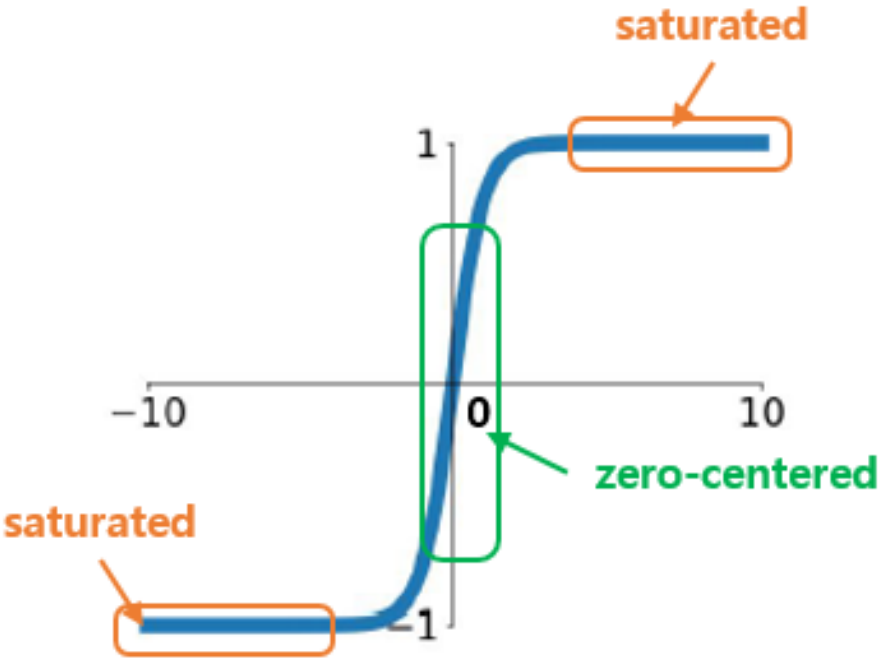
https://excelsior-cjh.tistory.com/177

Challenge of Learning

❑ Activation Functions

- Hyperbolic Tangent





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

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

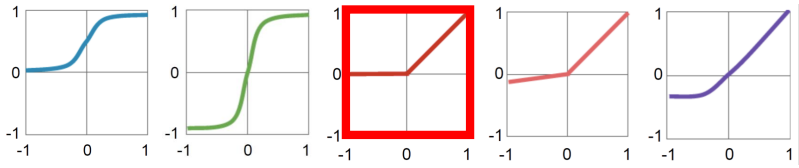
$$= 2\sigma(2x) - 1$$

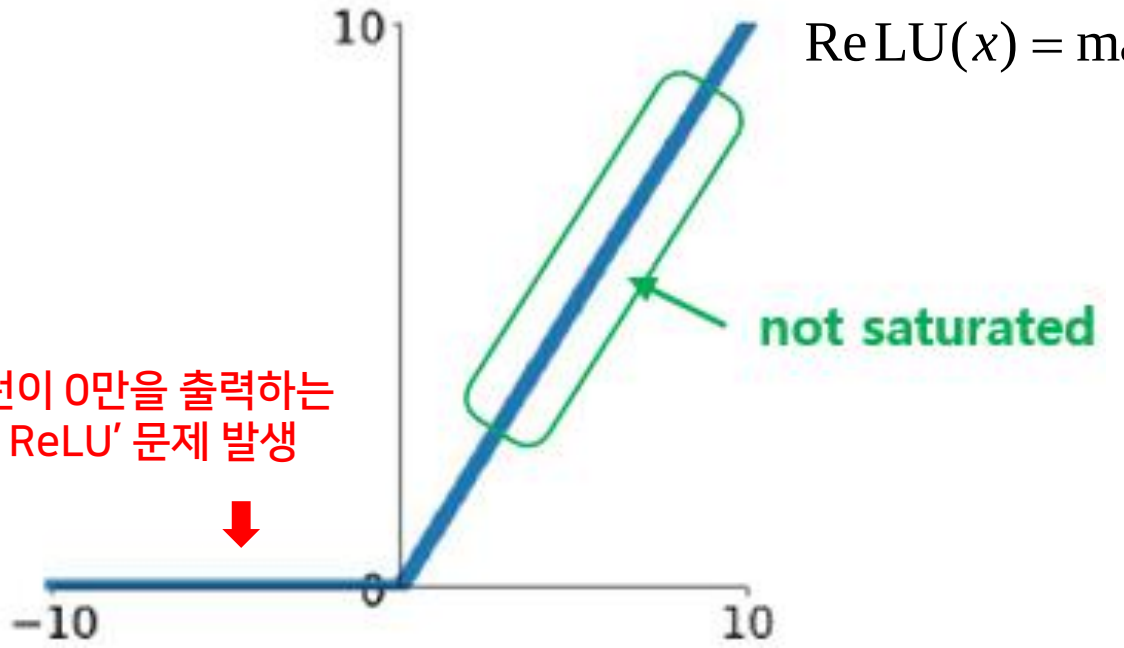
https://excelsior-cjh.tistory.com/177

Challenge of Learning

❑ Activation Functions

- Rectified Linear Unit





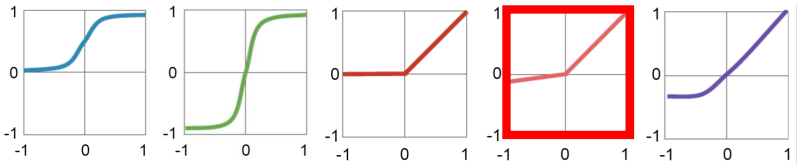
$$\mathrm{ReLU}(x) = \max(0, x)$$

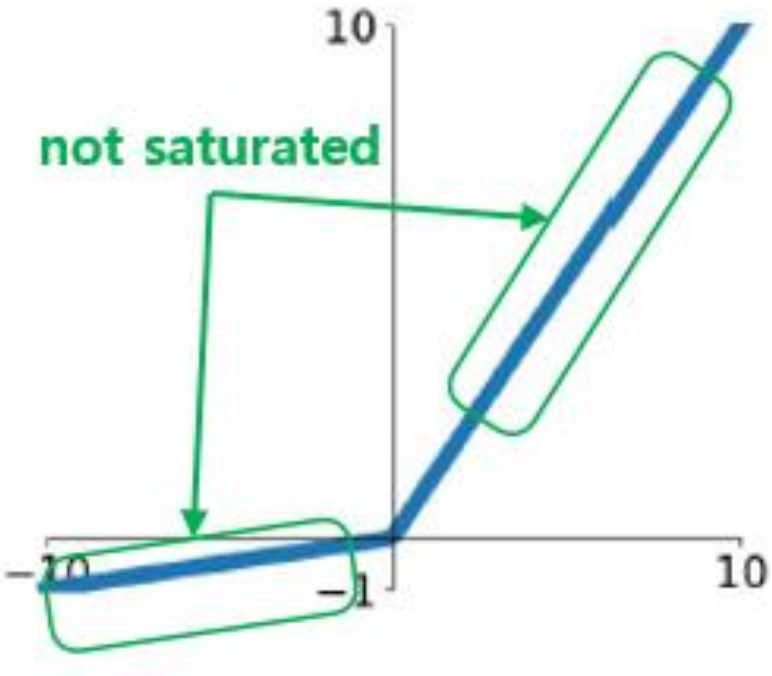
https://excelsior-cjh.tistory.com/177

Challenge of Learning

❑ Activation Functions

- LeakyReLU & PReLU





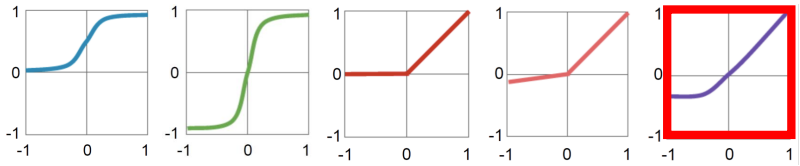
$$\text{Leaky ReLU}_{\alpha}(x) = \max(\alpha x, x)$$

PReLU(Parametric ReLU)는 Leaky ReLU와 식이 동일하지만, LeakyReLU에서 하이퍼파라미터인 α를 가중치 매개변수와 마찬가지로 α의 값도 학습되도록 역전파에 의해 α의 값이 변경되는 함수. PReLU는 대규모 이미지 데이터셋에서는 ReLU보다 성능이 좋았지만, 소규모 데이터셋에는 오버피팅 될 위험

https://excelsior-cjh.tistory.com/177

Challenge of Learning

❑ Activation Functions



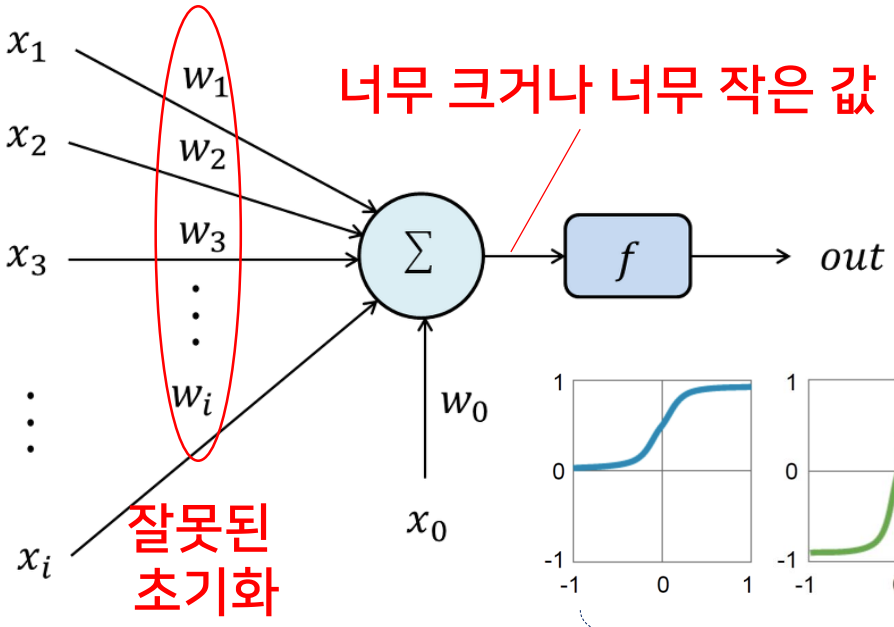
- Exponential Linear Unit

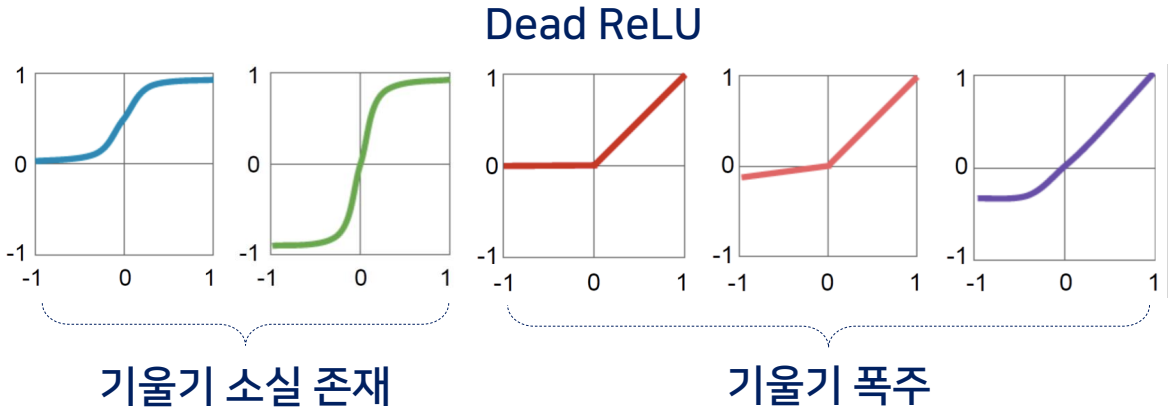
$$ELU_{\alpha}(x) = \begin{cases} \alpha(\exp(x) - 1) & if \ x < 0 \\ x & if \ x \geq 0 \end{cases}$$

- x < 0 일 때 ELU 활성화 함수 출력의 평균이 0에 가까워지기 때문에 편향 이동이 감소하여 그래디언트 소실 문제를 줄여준다. 하이퍼파라미터인 α는 x가 음수일 때 ELU가 수렴할 값을 정의하며 보통 1로 설정
- x < 0 이어도 그래디언티가 0이 아니므로 죽은(dead) 뉴런을 만들지 않는다.
- α = 1일 때 ELU는 x = 0 에서 급격하게 변하지 않고 모든 구간에서 매끄럽게 변하기 때문에 경사하강법에서 수렴속도가 빠르다.

Challenge of Learning

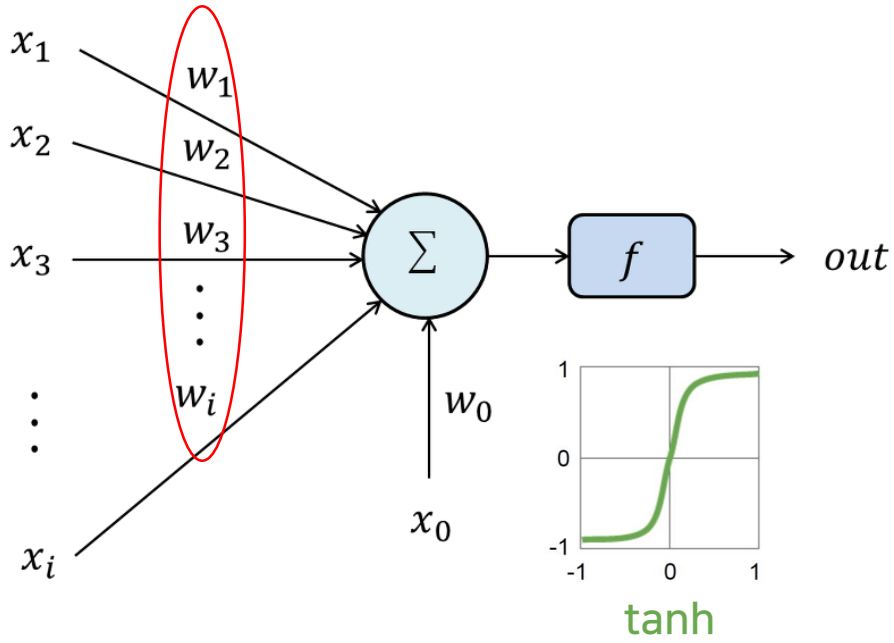
❑ 가중치 초기화 (Weight Initialization)

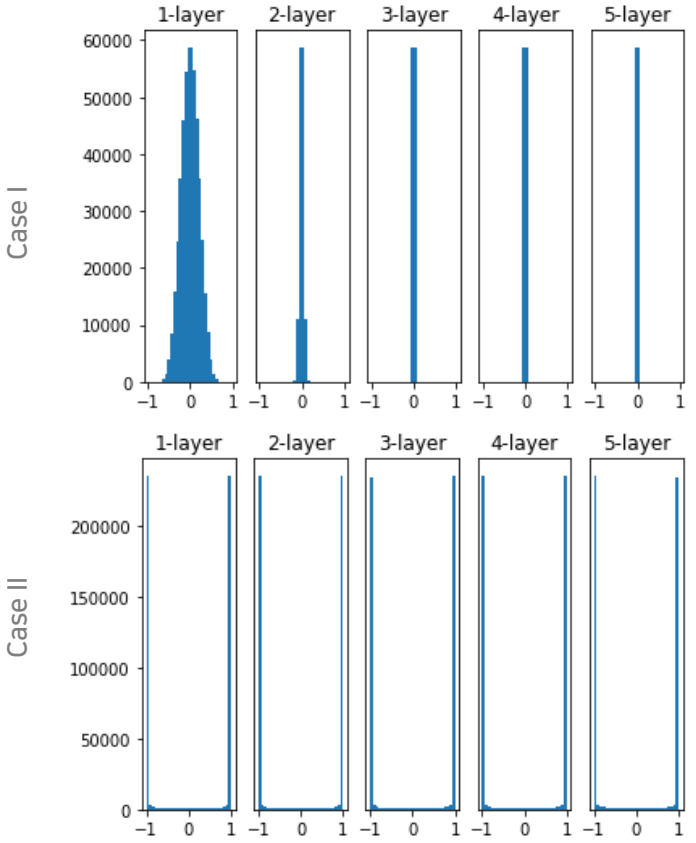




Challenge of Learning

❑ 가중치 초기화 (Weight Initialization)





Case I : (μ, σ) = (0, 0.01)

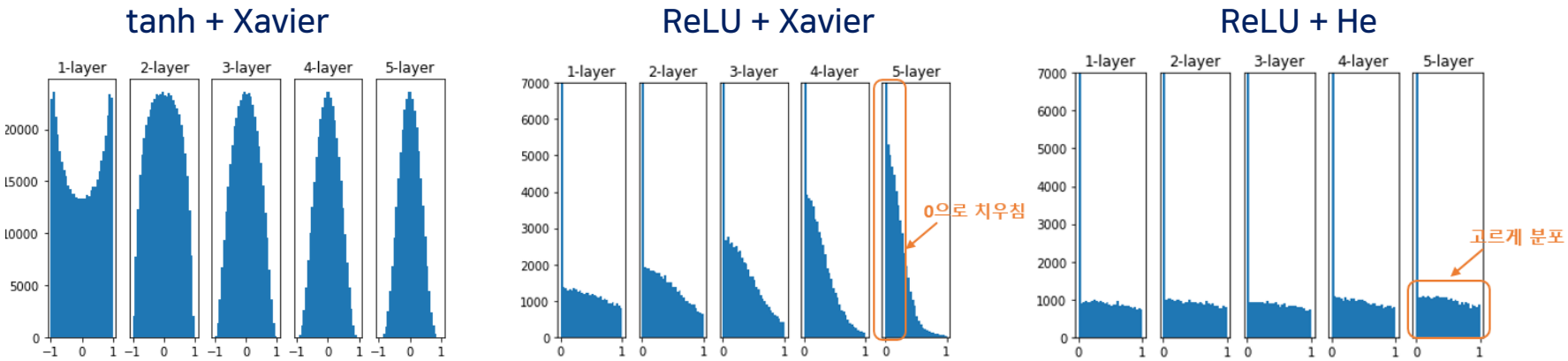
Case II : (μ, σ) = (0, 1)

https://excelsior-cjh.tistory.com/177

Challenge of Learning

❑ 가중치 초기화 (Weight Initialization)

- Xavier 초기화 (tf.contrib.Xavier_initializer)
  - 신경망 깊이가 문제 (기울기 소실/폭주)를 일으킴 ? → 너무 크지도 너무 작지도 않게 레이어 수로 나누자!
- He 초기화 (tf.keras.initializers.he_xxx) : ReLU에서도 잘 되게 해보자

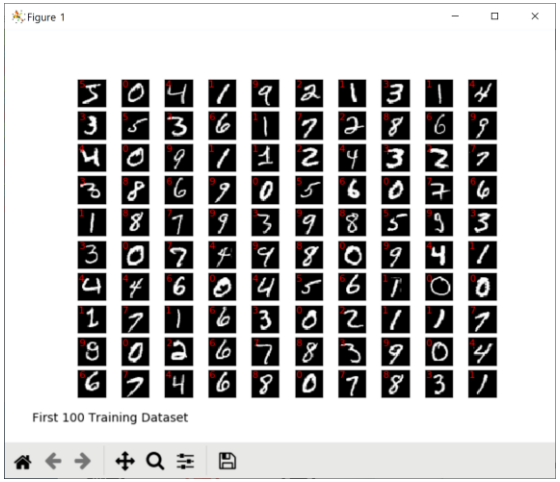


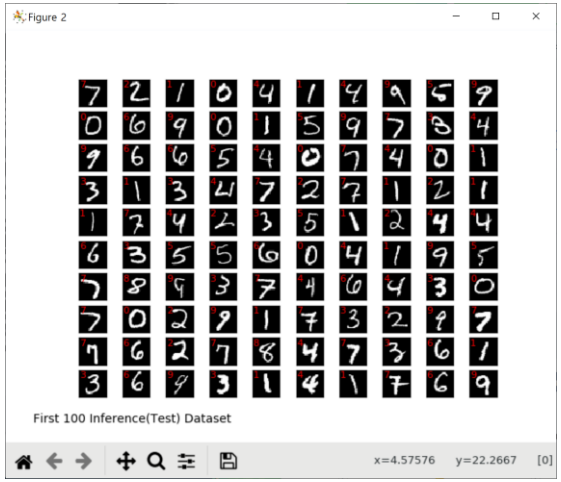
https://excelsior-cjh.tistory.com/177

Optimizer of Learning

❑ MNIST (손글씨-숫자 인식) Data Set

파일	목적
train-images-idx3-ubyte.gz	학습 셋 이미지 - 55000개의 트레이닝 이미지, 5000개의 검증 이미지
train-labels-idx1-ubyte.gz	이미지와 매칭되는 학습 셋 레이블
t10k-images-idx3-ubyte.gz	테스트 셋 이미지 - 10000개의 이미지
t10k-labels-idx1-ubyte.gz	이미지와 매칭되는 테스트 셋 레이블

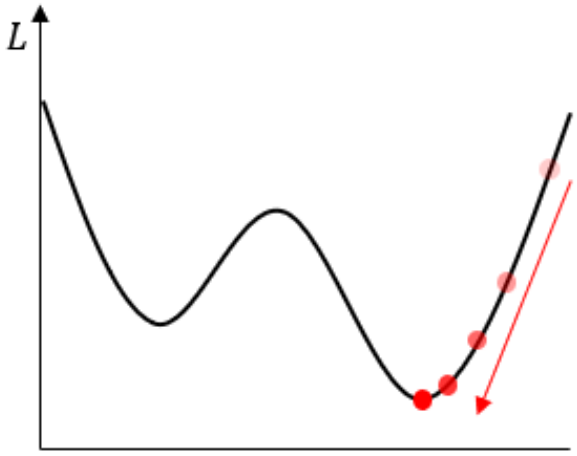




Optimizer of Learning

❑ MNIST (손글씨-숫자 인식) Data Set

파일	목적
train-images-idx3-ubyte.gz	학습 셋 이미지 - 55000개의 트레이닝 이미지, 5000개의 검증 이미지
train-labels-idx1-ubyte.gz	이미지와 매칭되는 학습 셋 레이블
t10k-images-idx3-ubyte.gz	테스트 셋 이미지 - 10000개의 이미지
t10k-labels-idx1-ubyte.gz	이미지와 매칭되는 테스트 셋 레이블

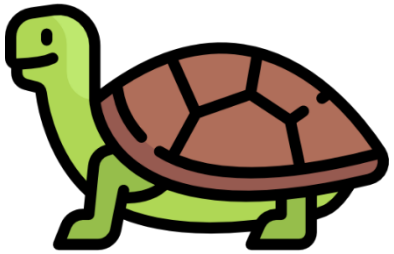


Normal Gradient Descent 기반 학습

55000개 데이터를 넣고 한 Step

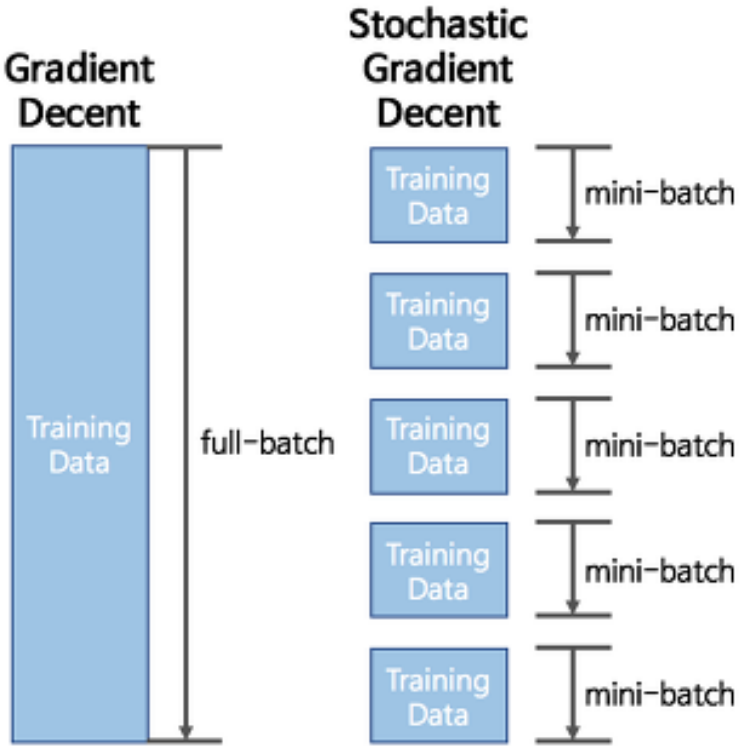
55000개 데이터를 넣고 한 Step

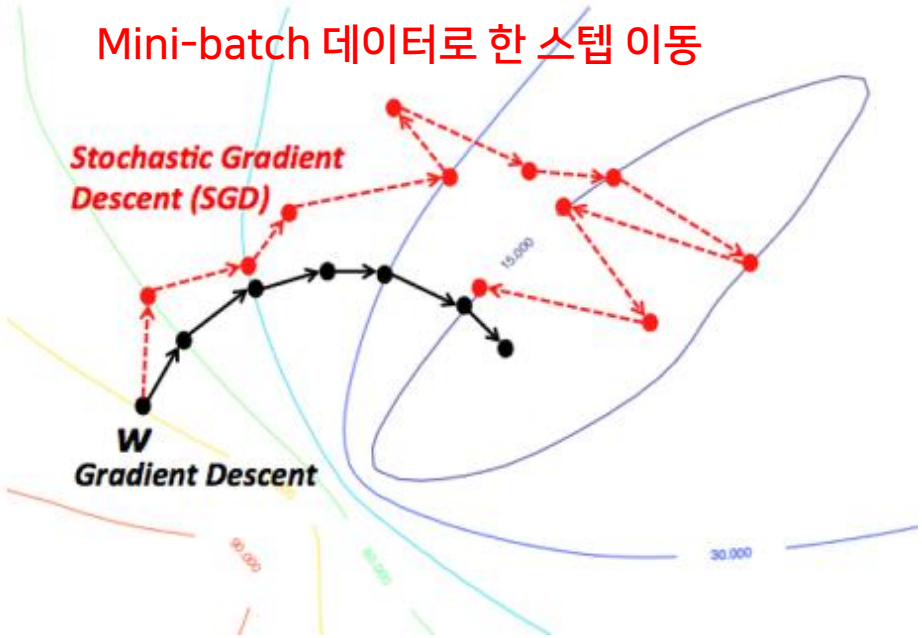
55000개 데이터를 넣고 한 Step



Optimizer of Learning

❑ Stochastic Gradient Descent (SGD)





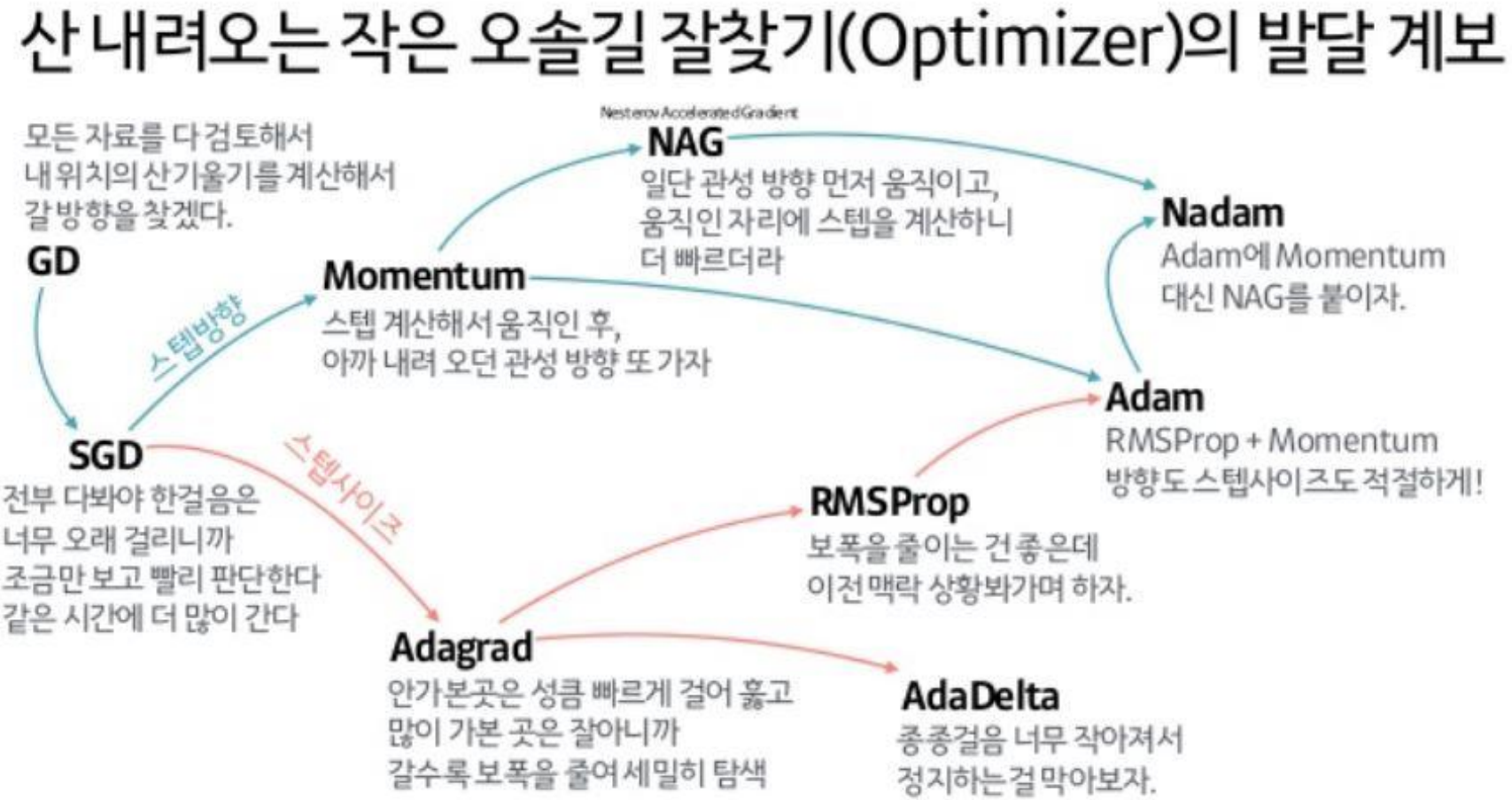
Optimizer of Learning

❑ More than SGD (Stochastic Gradient Descent)

- SGD
  - 경사하강법은 무작정 기울어진 방향으로 이동하는 방식이기 때문에 탐색경로가 비효율적이어서 한참을 탐색
- 다른 Optimizer들
  - Momentum / Nesterov Accelerated Gradient (NAG)
  - Adaptive Gradient (Adagrad)
  - RMSProp
  - Adaptive Delta (AdaDelta)
  - Adaptive Moment Estimation (Adam)

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning



Optimizer of Learning

❑ Learning Notation

- $\theta = \theta - \eta \nabla_\theta J(\theta)$

θ : Parameter η : Learning Rate $J(\theta)$: Loss Function

$\nabla_\theta J(\theta)$: Gradient of Loss

❑ Momentum

- Gradient Descent를 통해 이동하는 과정에 일종의 '관성'을 주는 것
- Time step t 에서의 이동벡터

$$\theta = \theta - v_t$$

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

얼마나 momentum을 줄 것인지에 대한 momentum term으로, 보통 0.9 정도의 값을 사용

$$= \eta \nabla_\theta J(\theta)_t + \gamma \eta \nabla_\theta J(\theta)_{t-1} + \gamma^2 \eta \nabla_\theta J(\theta)_{t-2} + \cdots$$

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning

❑ Learning Notation

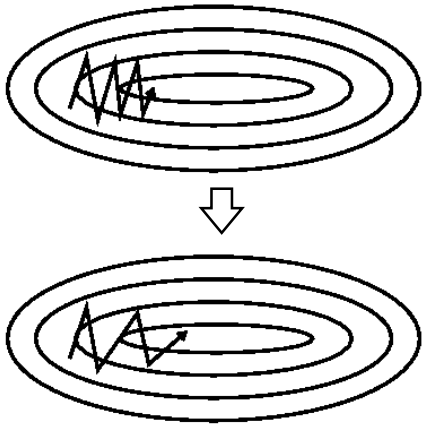
- $\theta = \theta - \eta \nabla_\theta J(\theta)$

θ : Parameter η : Learning Rate $J(\theta)$: Loss Function

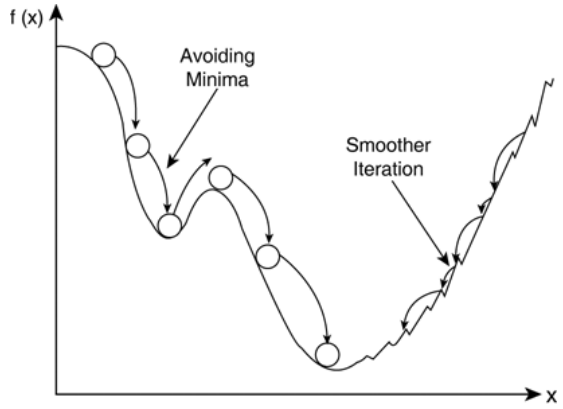
$\nabla_\theta J(\theta)$: Gradient of Loss

❑ Momentum

- Gradient Descent를 통해 이동하는 과정에 일종의 '관성'을 주는 것



방향을 좀 더 잘 찾고



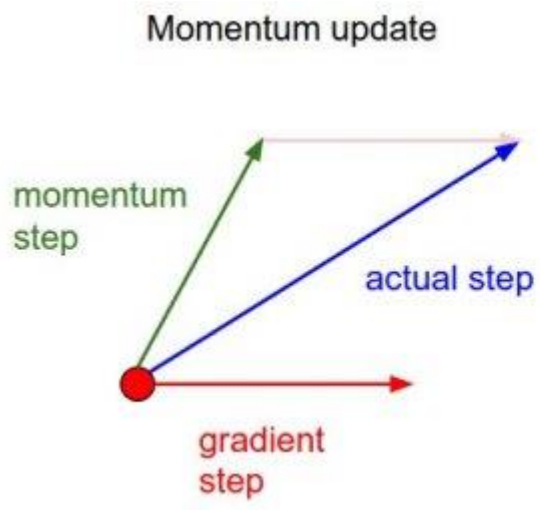
Local Minima에서 탈출 가능

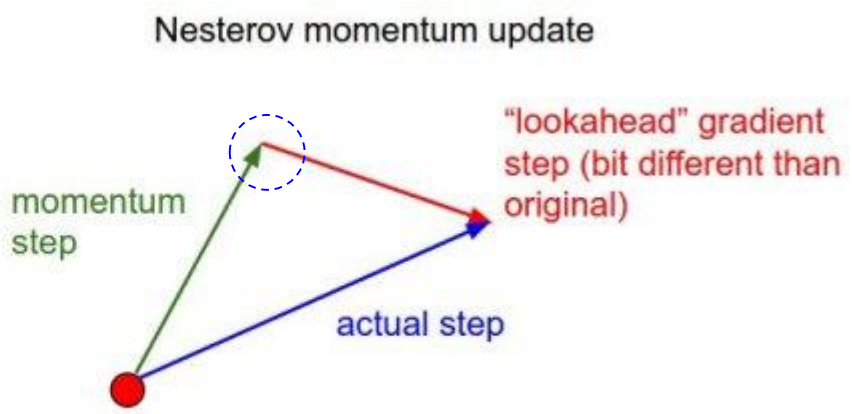
But,
과거에 이동했던 양을 변수별로 저장해야하므로 변수에 대한 메모리가 기존의 두배

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning

❑ Nesterov Accelerated Gradient (NAG)





$$\theta = \theta - v_t$$

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

현재 위치에서의 기울기와 모멘텀 스텝을 독립적으로 계산하고 합침

$$\theta = \theta - v_t$$

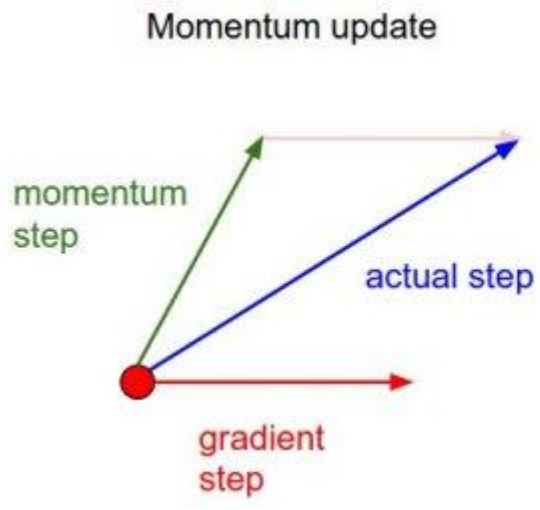
$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta - \gamma v_{t-1})$$

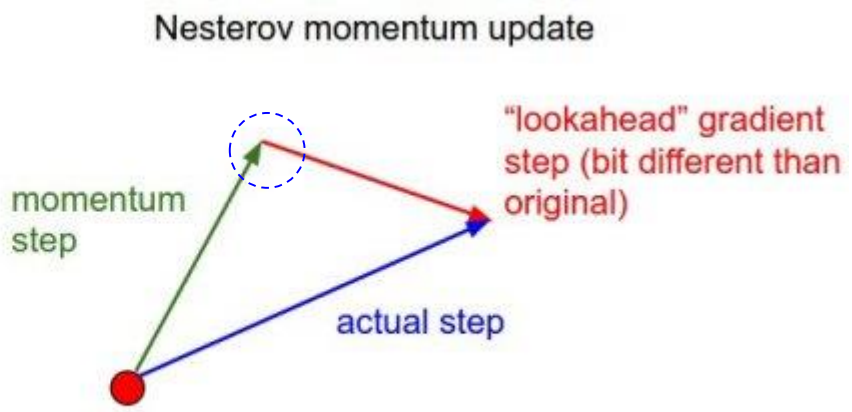
모멘텀 스텝을 먼저 이동했다고 생각한 후 그 자리에서의 기울기를 구해서 기울기 스텝을 이동

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning

❑ Nesterov Accelerated Gradient (NAG)





멈춰야 할 시점에서도 관성에 의해 훨씬 멀리 갈수도 있음

일단 모멘텀으로 이동을 반정도 한 후 어떤 방식으로 이동해야 할 지를 결정

(모멘텀 방식의 빠른 이동 + 적절한 시점에서 제동)

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning

❑ Adaptive Gradient (Adagrad)

- 자주 등장하거나 변화를 많이 한 변수들의 경우 최적점에 가까이 있을 확률이 높음

'지금까지 많이 변화했던 변수들은 step size를 작게 하고,
지금까지 많이 변화하지 않은 변수들은 step size를 크게 하자'

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \grave{o}}} \cdot \nabla_\theta J(\theta_t)$$

보통 adagrad에서 step size로는 0.01

ϵ : 0 나누기 방지 (10^{-4}~10^{-8})

$$G_t = G_{t-1} + [\nabla_\theta J(\theta_t)]^2$$

Neural Network의 parameter가 k개라고 할 때, G_t는 k 차원의 벡터

$$*element - wise$$

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning

❑ Adaptive Gradient (Adagrad)

- 자주 등장하거나 변화를 많이 한 변수들의 경우 최적점에 가까이 있을 확률이 높음

'지금까지 많이 변화했던 변수들은 step size를 작게 하고,
지금까지 많이 변화하지 않은 변수들은 step size를 크게 하자'

- 장점 : 학습 속도 담금질 (Step Size Decay) 필요 없어짐
- 단점 : 학습을 계속하면 Step Size가 너무 줄어 안 움직임
  - 개선한 알고리즘 : RMSProp & AdaDelta

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning

❑ RMSProp (제프리 힌튼 교수가 제안)

Adagrad

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \grave{o}}} \cdot \nabla_\theta J(\theta_t)$$

$$G_t = G_{t-1} + [\nabla_\theta J(\theta_t)]^2$$

$$*element - wise$$

RMSProp

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \grave{o}}} \cdot \nabla_\theta J(\theta_t)$$

$$G_t = \gamma G_{t-1} + (1-\gamma)[\nabla_\theta J(\theta_t)]^2$$

$$*element - wise$$

G_t가 무한정 커지지 않으면서,
최근 변화량의 상대적 크기 차이 유지

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

Optimizer of Learning

❑ Adaptive Moment Estimation (Adam)

RMSProp + Momentum

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \dot{o}}} \cdot \nabla_\theta J(\theta_t)$$

$$G_t = \gamma G_{t-1} + (1-\gamma)[\nabla_\theta J(\theta_t)]^2$$

$$\theta = \theta - v_t$$

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

Adam

$$\theta = \theta - \frac{\eta}{\sqrt{\hat{v}_t + \dot{o}}} \hat{m}_t$$

$$m_t = \beta_1 m_{t-1} - (1-\beta_1)\nabla_\theta J(\theta)$$

$$v_t = \beta_2 v_{t-1} - (1-\beta_2)[\nabla_\theta J(\theta)]^2$$

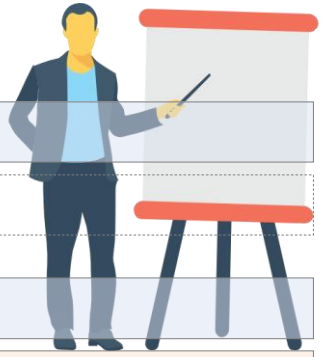
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

보통 β_1 로는 0.9, β_2로는 0.999, ϵ으로는 10^{-8} 정도의 값을 사용

Adam에서는 m과 v가 처음에 0으로 초기화되어 있기 때문에 학습의 초반부에서는 m_t,v_t가 0에 가깝게 bias 되어있을 것이라고 판단하여 이를 unbiased 하게 만들어주는 작업을 거침

http://shuuki4.github.io/deep%20learning/2016/05/20/Gradient-Descent-Algorithm-Overview.html

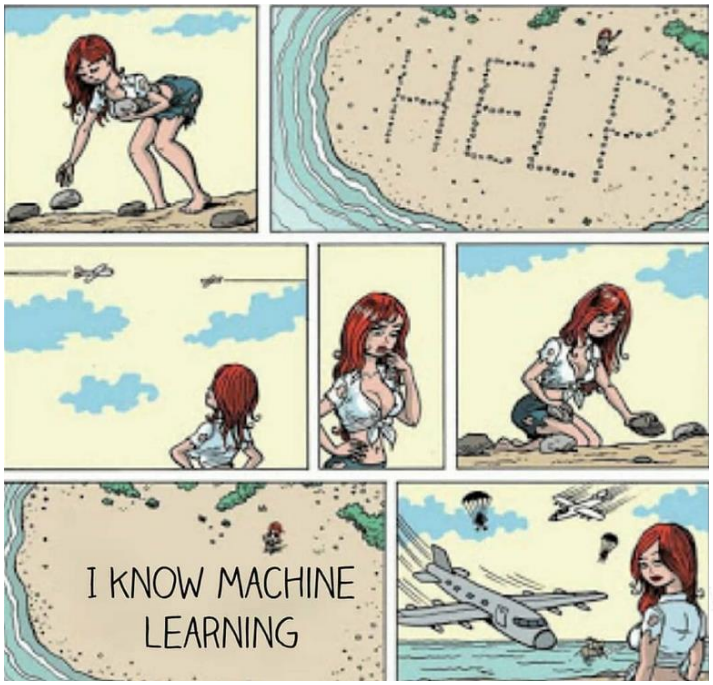
Learning : Human & Machine

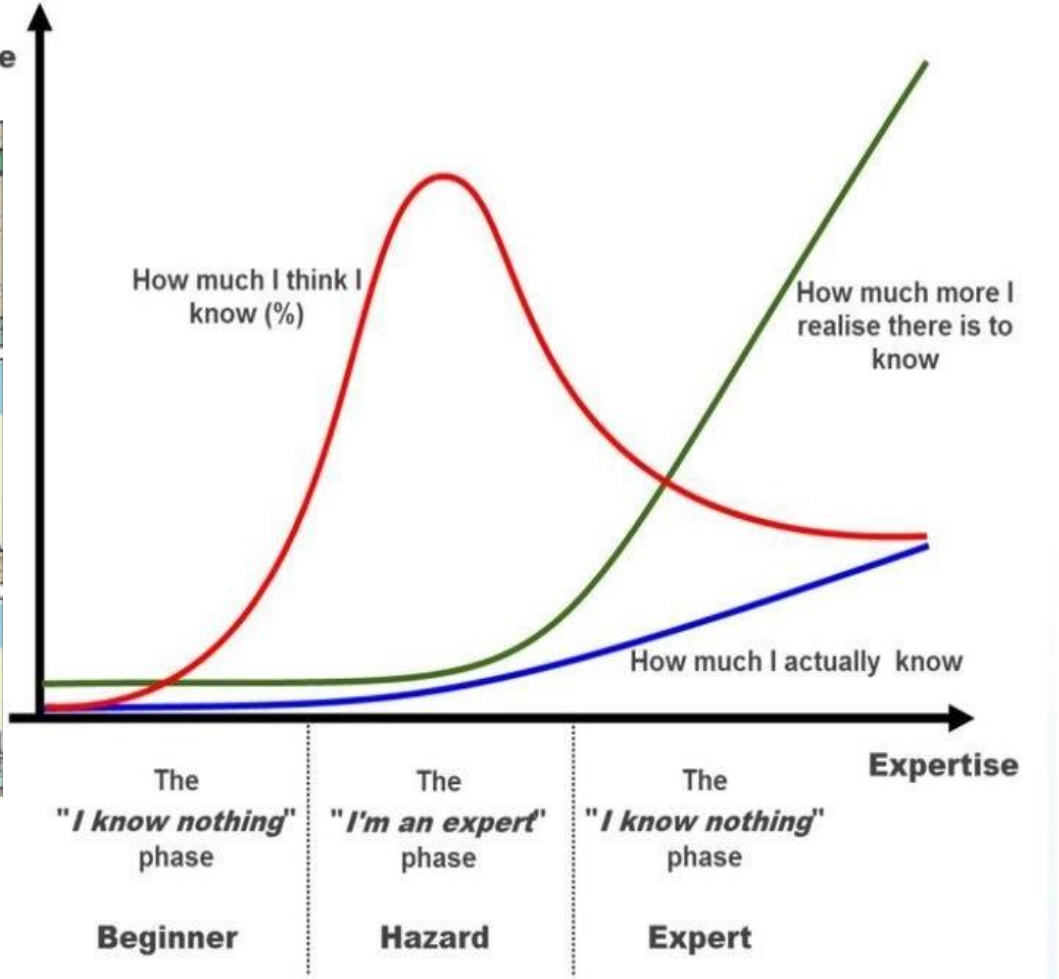


Human	Machine	Supervised Learning
Human	Machine	Augmented Supervised Learning
Human	Machine	Semi-Supervised Learning
Human	Machine	Reinforcement Learning
Human	Machine	Unsupervised Learning

https://deeplearning.mit.edu/

Knowledge versus Expertise





https://deeplearning.mit.edu/

Outline

- WSL (Window Subsystem for Linux) 기반 개발 환경
  - WSL 설치 및 Setup 파일 설명
- **딥러닝 기초 및 실습**
  - TensorFlow 기반 실습
- 딥러닝 하드웨어 가속기 연구 동향

실습

❑ Git Clone

- 프로젝트 폴더 다운

```
[user - 2019.10.02 XX:XX:XX]
/home/user: git clone https://github.com/woongchoi84/project
```

❑ Jupyter Lab 실행

```
[user - 2019.10.02 XX:XX:XX]
/home/user: jupyter lab
```

Linux Basic

❑ Basic Navigation

```
$> pwd
/home/woong
$> ls
setup
$> la
.bash_history   .config     .jupyter   .pyLib                        .vimrc
.bash_logout    .dbus       .keras     .python_history               setup
.bashrc         .gitconfig  .local     .sudo_as_admin_successful
.bashrc_add     .gvfs       .mozilla   .vim
.cache          .ipython    .profile   .viminfo
$> ll
total 44
drwxr-xr-x 1 woong woong   512 Jul  4 13:03 ./
drwxr-xr-x 1 root  root    512 Jun 15 15:37 ../
-rw------- 1 woong woong  9503 Jul  4 13:59 .bash_history
-rw-r--r-- 1 woong woong   220 Jun 15 15:37 .bash_logout
...
```

Print Working Directory - ie. Where are we currently.

List the contents of a directory.

for hidden contents

for permission

Linux Basic

❑ Basic Navigation

```
$> pwd
/home/woong
$> ls
setup
      # 절대경로 (absolute path)
$> cd /home/woong/setup
      # 상대경로 (relative path)
$> cd setup
```

Change Directories - ie. move to another directory.

- Everything is a file under Linux (Even directories)
- Linux is an extensionless system (Files can have any extension they like or none at all)
- Linux is case sensitive (Beware of silly typos)

- /etc - Stores config files for the system.
- /var/log - Stores log files for various system programs. (You may not have permission to look at everything in this directory. Don't let that stop you exploring though. A few error messages never hurt anyone.)
- /bin - The location of several commonly used programs (some of which we will learn about in the rest of this tutorial.
- /usr/bin - Another location for programs on the system.

Linux Basic

❑ Basic Navigation

pwd	Where am I in the system.
ls [path]	Perform a listing of the given path or your current directory. Common options: -l, -h, -a
cd [path]	Change into the given path or into your home directory.
Path	A description of where a file or directory is on the filesystem.
Absolute Path	One beginning from the root of the file system (eg. /etc/sysconfig).
~ (tilde)	Used in paths as a reference to your home directory (eg. ~/Documents).
. (dot)	Used in paths as a reference to your current directories parent directory (eg. ../bin).
TAB completion	Start typing and press TAB. The system will auto complete the path. Press TAB twice and it will show you your alternatives.
file [path]	Find out what type of item a file or directory is.
Spaces in names	Put whole path in quotes (") or a backslash (₩) in front of spaces.
Hidden files and directories	A name beginning with a . (dot) is considered hidden.

https://ryanstutorials.net/linuxtutorial/cheatsheet.php

Linux Basic

❑ Permissions

rwx	r (read) w (write) x (execute)
owner	a single person who owns the file. (typically the person who created the file but ownership may be granted to some one else by certain users)
group	every file belongs to a single group
others	everyone else who is not in the group or the owner.
ls -l [path]	View the permissions of a file or all items in a directory.
chmod <permissions> <path>	Change permissions. Permissions can be either shorthand (eg. 754) or longhand (eg. g+x).

```
$> ll
total 44
drwxr-xr-x 1 woong woong   512 Jul  4 13:03 ./
drwxr-xr-x 1 root  root    512 Jun 15 15:37 ../
-rw------- 1 woong woong  9503 Jul  4 13:59 .bash_history
-rw-r--r-- 1 woong woong   220 Jun 15 15:37 .bash_logout
...
```

The first character identifies the file type. If it is a dash (-) then it is a normal file. If it is a d then it is a directory.

The following 3 characters represent the permissions for the owner.
The following 3 characters represent the permissions for the group.
Finally the last 3 characters represent the permissions for others

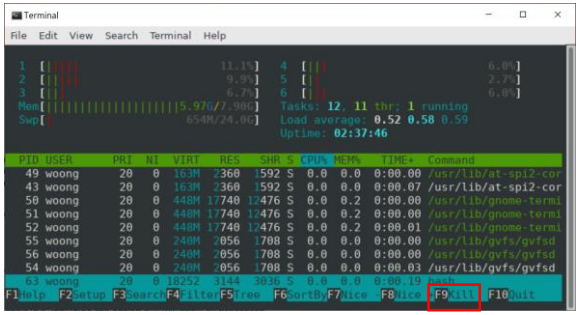
https://ryanstutorials.net/linuxtutorial/cheatsheet.php

Linux Basic

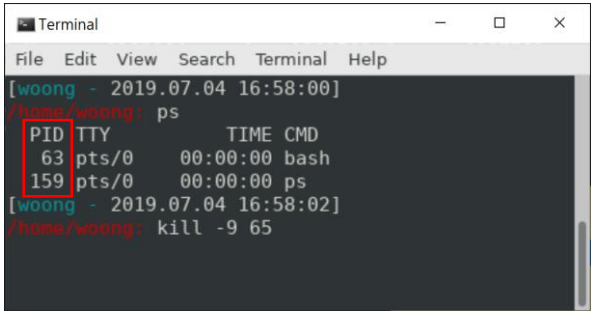
❑ Process Management

CTRL + C	Cancel the currently running process.
kill <process id>	Cancel the given process. Include the option -9 to kill a stubborn (완고한) process.
ps	Obtain a listing of processes and their id's. Including the option aux will show all processes.
CTRL + Z	Pause the currently running process and put it in the background.
jobs	See a list of current processes in the background.
fg <job number>	Move the given process from the background to the foreground.
top	View real-time data about processes running on the system.

```
$> htop
```



Kill 하고 싶은 프로세스로 이동 (키보드 화살표) 후 F9 로 Kill 가능



https://ryanstutorials.net/linuxtutorial/cheatsheet.php

Linux Basic

❑ Filters & Useful Commands

cat [path]	Show the first n lines.
head [-n] [path]	Show the first n lines.
tail [-n] [path]	Show the last n lines.
sort [-options] [path]	Sort lines in a given way.
nl [-options] [path]	Print line numbers before data.
wc [-options] [path]	Print a count of lines, words and characters.
cut [-options] [path]	Cut the data into fields and only display the specified fields.
sed <expression> [path]	Do a search and replace on the data.
grep	Search for a given pattern.
uniq [options] [path]	Remove duplicate lines.
du -sh ./*	Find the size of every directory in your current directory.
df -h	Display how much disk space is used and also free.
basename -s .jpg -a *.jpg | xargs -n1 -i cp {}.jpg {}_original.jpg	Make a copy of every jpg image file in the current directory and rename adding _original.
find /home -mtime -1	Find all files in the given directory (and subdirectories) which have been modified in the last 24 hours.
shutdown -h now	Shutdown the system. (Replace -h with -r for reboot.)

Filters allow us to process and format data in interesting ways.

https://ryanstutorials.net/linuxtutorial/cheatsheet.php

Linux Basic

❑ File Manipulation

mkdir <directory name>	Create a directory
touch <file name>	Create a blank file
cp <source> <destination>	Copy the source file to the destination.
mv <source> <destination>	Move the source file to the destination
rm <path>	Remove a file or directory. Common options: -r, -f
ln –s file link	Create symbolic link link to file
tar -cvf [file.tar] [path]	Create tar archive file (c – Creates a new .tar archive file, v – Verbosely show the .tar file progress, f – File name type of the archive file)
tar -zcvf [file.tar.gz] [path]	Create tar.gz archive file
tar -xvf [file.tar] [path]	Untar tar archive file
tar -zxvf [file.tar.gz] [path]	Uncompress tar.gz archive file

https://ryanstutorials.net/linuxtutorial/cheatsheet.php

Linux Basic

❑ 셔뱅 (shebang)

셔뱅(shebang)은 해시 기호와 느낌표(#!)로 이루어진 문자 시퀀스로, 스크립트의 맨 처음에 온다. 샤-뱅(sha-bang), 해시뱅(hashbang), 파운드-뱅(pound-bang), 해시-플링(hash-pling), 크런치뱅(crunchbang)이라고도 한다.

유닉스 계열 운영 체제에서 셔뱅이 있는 스크립트는 프로그램으로서 실행되며, 프로그램 로더가 스크립트의 첫 줄의 나머지 부분을 인터프리터 지시자(interpreter directive)로 구문 분석한다. 즉, 지정된 인터프리터 프로그램이 대신 실행되어 스크립트의 실행을 시도할 때 처음 사용되었던 경로를 인수로서 넘겨주게 된다. 이를테면 스크립트의 경로가 *path/to/script*이고 다음의 줄로 시작한다면:

```
#! /bin/bash
```

프로그램 로더는 프로그램 */bin/bash*를 대신 실행하되 *path/to/script*를 첫 번째 인수로 넘겨준다.

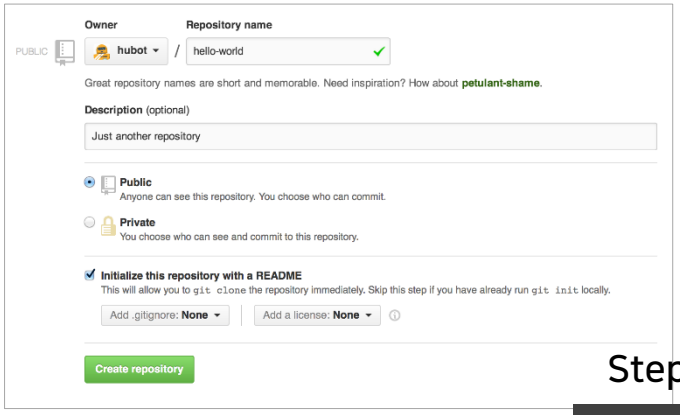
```
$> which python3
/usr/bin/python3
      # python code 파일 (source_code.py) 첫줄에 '#! /usr/bin/python' 추가 후
$> chmod +x <source_code.py>
$> ./<source_code.py>
```

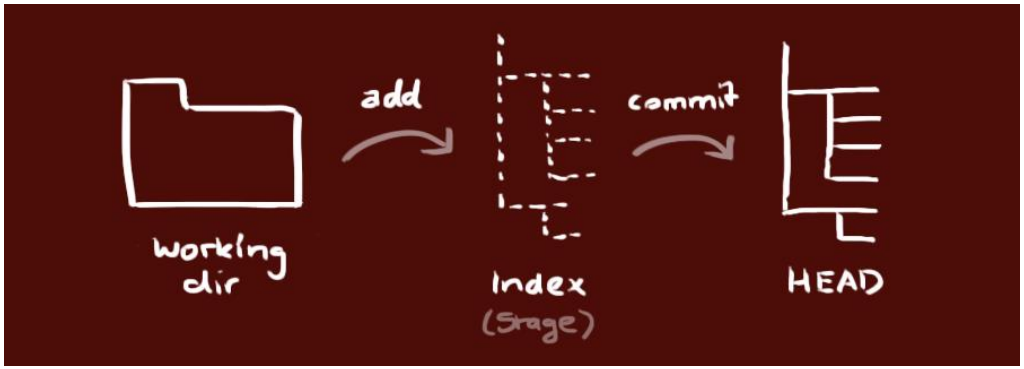
출처 : 위키백과

GitHub

❑ GitHub에 Code 올리기

Step 1. Sign-up & Create a Repository





Step 2. 초기화, Repository 할당, Commit, and Push

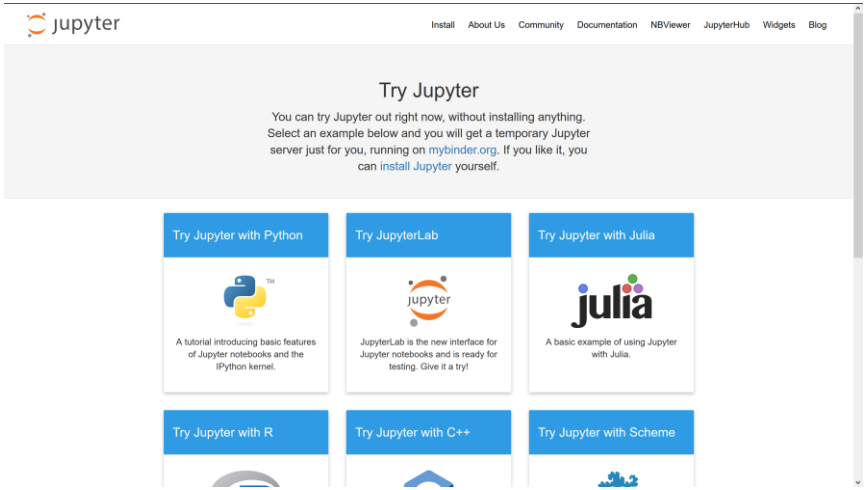
```
$> git config --global user.name "woongchoi84"
$> git config --global user.email "woongchoi84@gmail.com"
$> git init
$> git remote add origin https://github.com/woongchoi84/setup
$> git add .
$> git commit -m "test"
$> git push origin master
```

https://github.com/
https://github.com/KennethanCeyer/tutorial-git

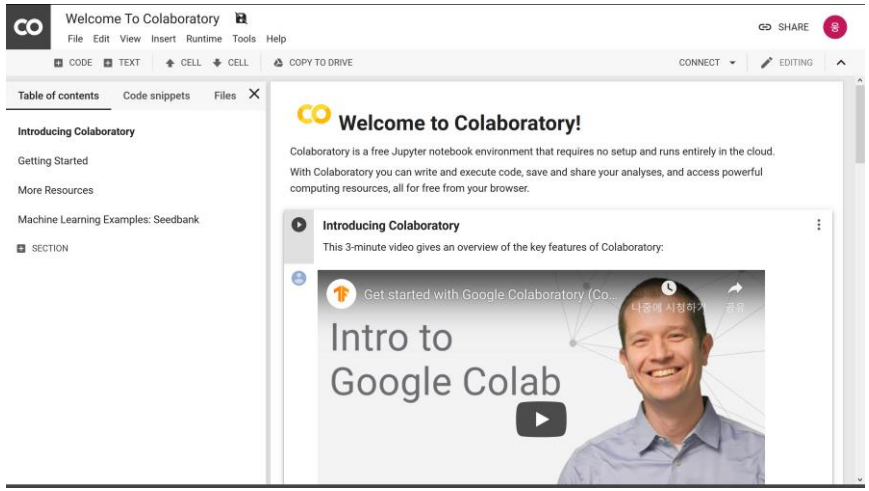
Web-based Programs

❑ Try Jupyter & Google Colaboratory

- https://jupyter.org/try
- https://colab.research.google.com/notebooks/welcome.ipynb



Can't use tensorflow.
Just for brief learning!



We can use tensorflow.
Google Colab =
Google Drive + Jupyter Notebook

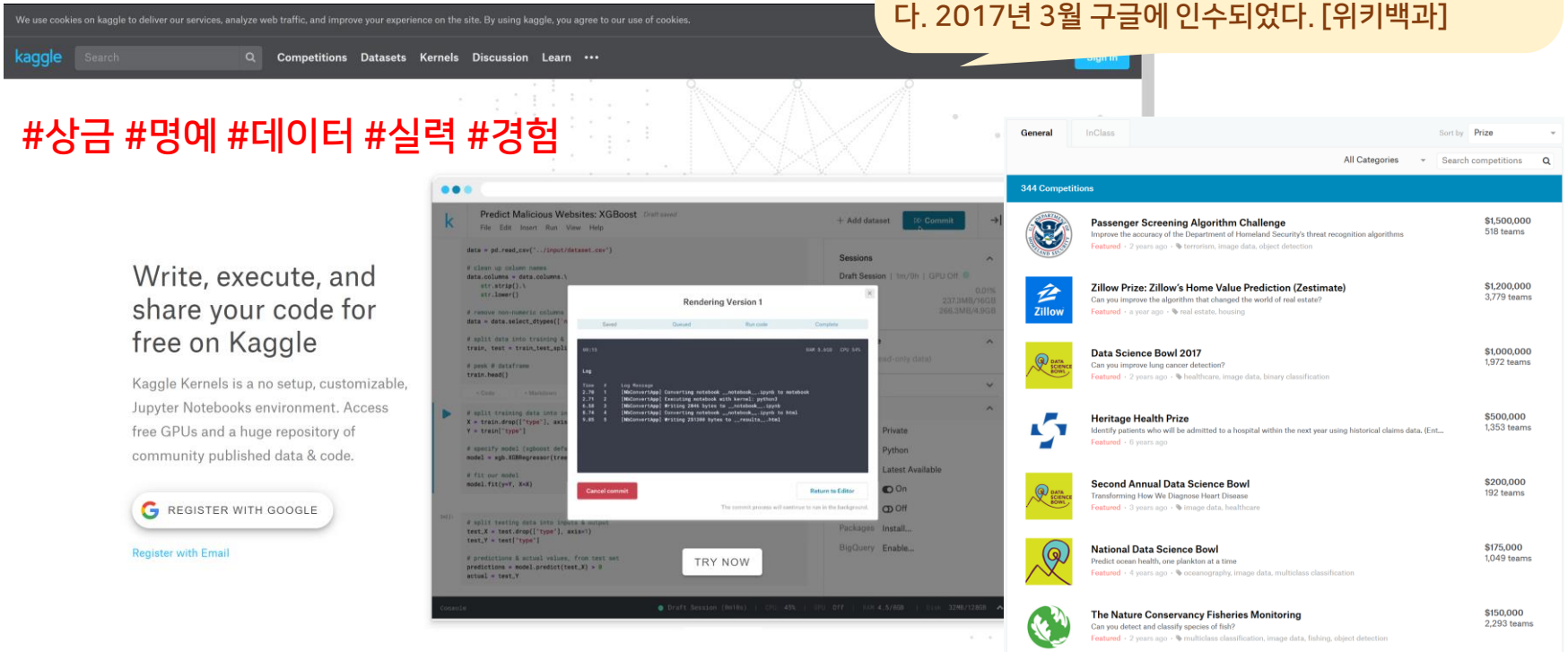
Web-based Programs

❑ Kaggle

- https://www.kaggle.com/

캐글은 2010년 설립된 예측모델 및 분석 대회 플랫폼이다. 기업 및 단체에서 데이터와 해결과제를 등록하면, 데이터 과학자들이 이를 해결하는 모델을 개발하고 경쟁한다. 2017년 3월 구글에 인수되었다. [위키백과]

#상금 #명예 #데이터 #실력 #경험



Google Colaboratory

❑ Hardware Spec.

Untitled1.ipynb ☆

File Edit View Insert Runtime Tools Help

COMMENT SHARE 응

CODE TEXT CELL CELL

RAM Disk EDITING

```
[4]  !cat /etc/issue.net
     !sed -n '5p' /proc/cpuinfo
     !head -1 /proc/meminfo
     !df -h | egrep 'overlay|sda'
     !nvidia-smi | head -10
```

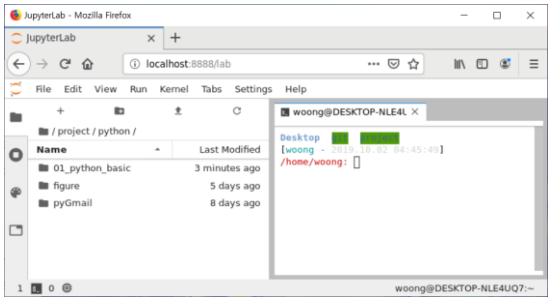
```
Ubuntu 18.04.2 LTS
model name      : Intel(R) Xeon(R) CPU @ 2.30GHz
MemTotal:       13335276 kB
overlay         359G   30G  311G   9% /
/dev/sda1       365G   39G  327G  11% /opt/bin
Fri Jul  5 06:24:36 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.67       Driver Version: 410.79       CUDA Version: 10.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   52C    P8    16W /  70W |      0MiB / 15079MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

Jupyter Lab

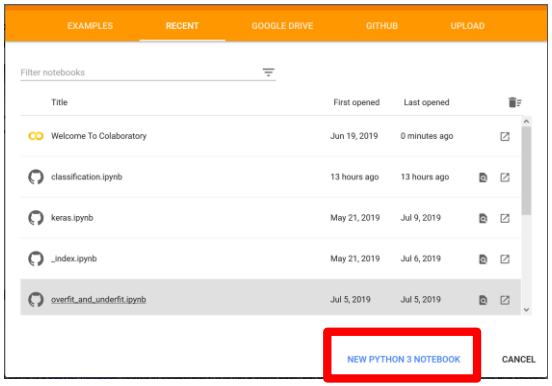
❑ Jupyter Lab (or Jupyter Notebook) 실행

- WSL (Window Sub-system for Linux)
  - 실행 : xming + ubuntu 실행
  - Jupyter Lab 실행

    [user – 2019.10.02 XX:XX:XX]
    /home/user: jupyter lab

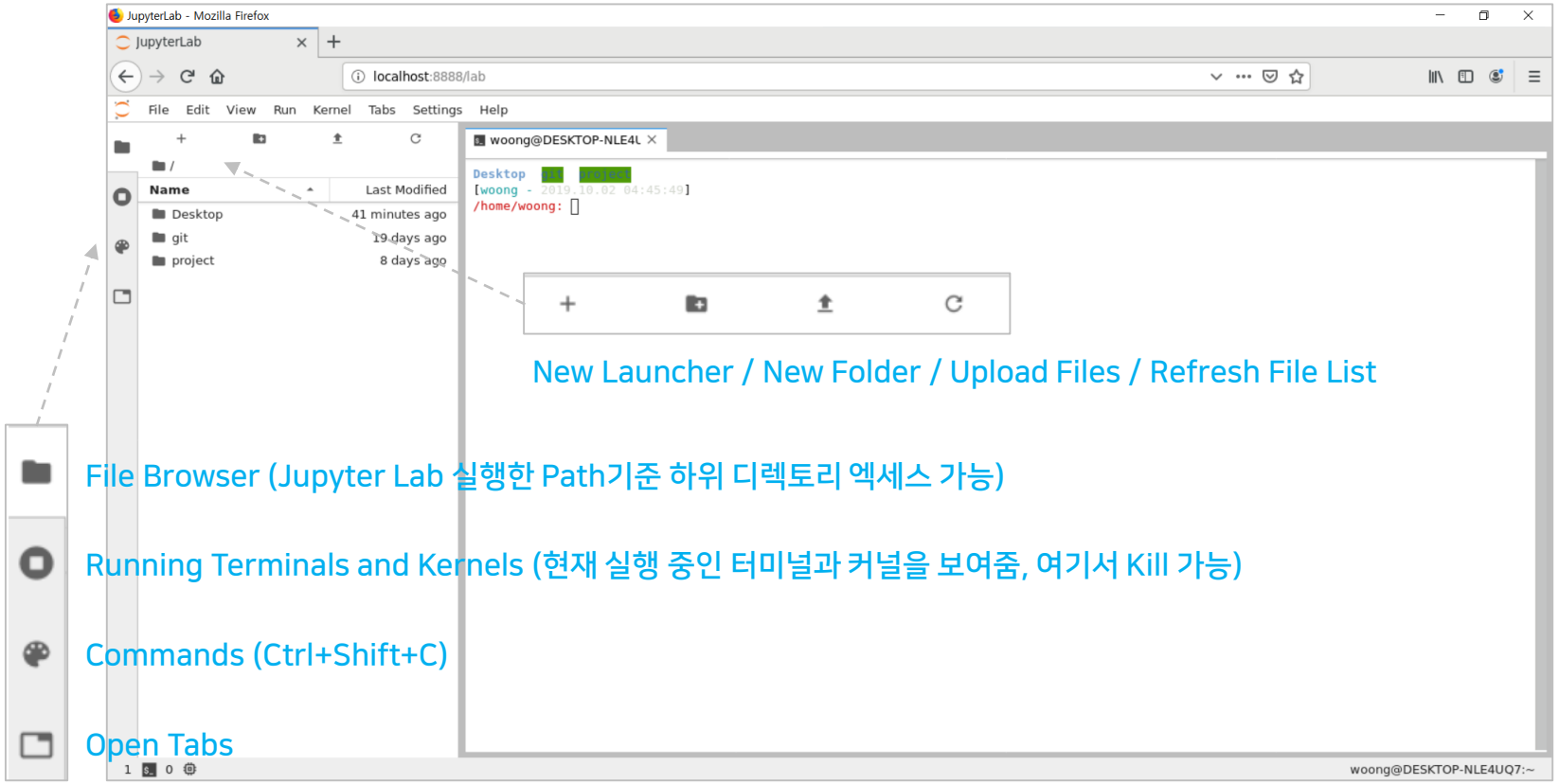


- WSL에 문제가 있는 경우
  - 구글 로그인 → Google Colab 검색 후 접속
  - 'NEW PYTHON3 NOTEBOOK' 클릭



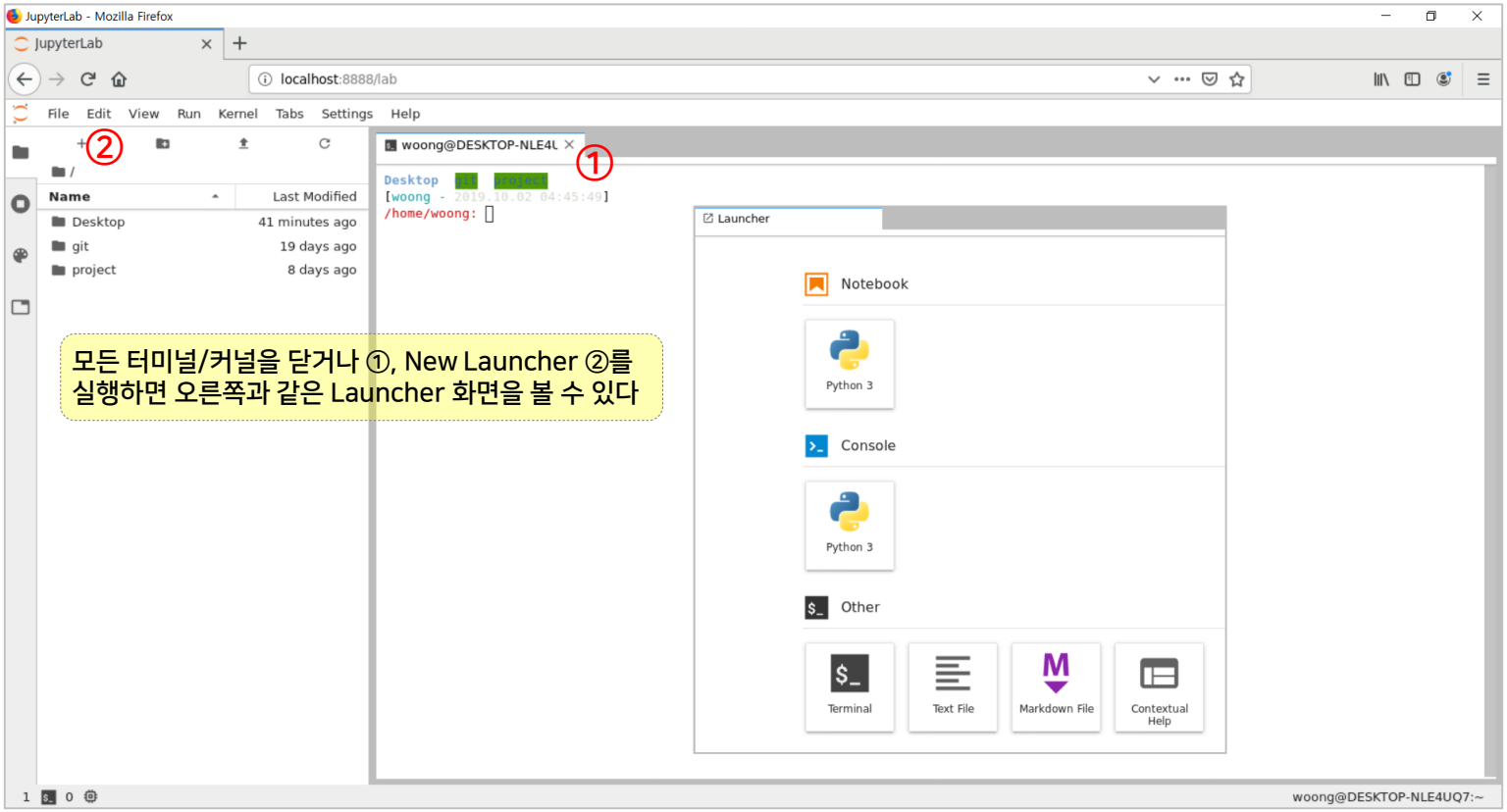
Jupyter Lab

❑ Jupyter Lab : Interface



Jupyter Lab

❑ Jupyter Lab : Interface



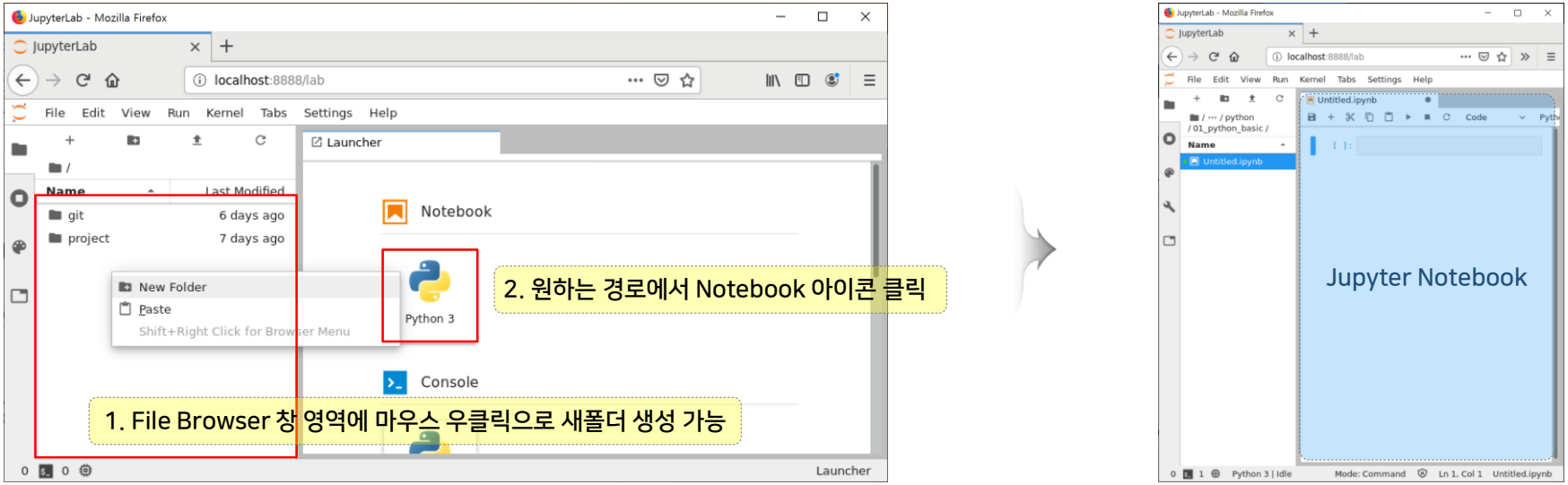
Python Programming

❑ Reference Site

- https://www.programiz.com/python-programming/tutorial

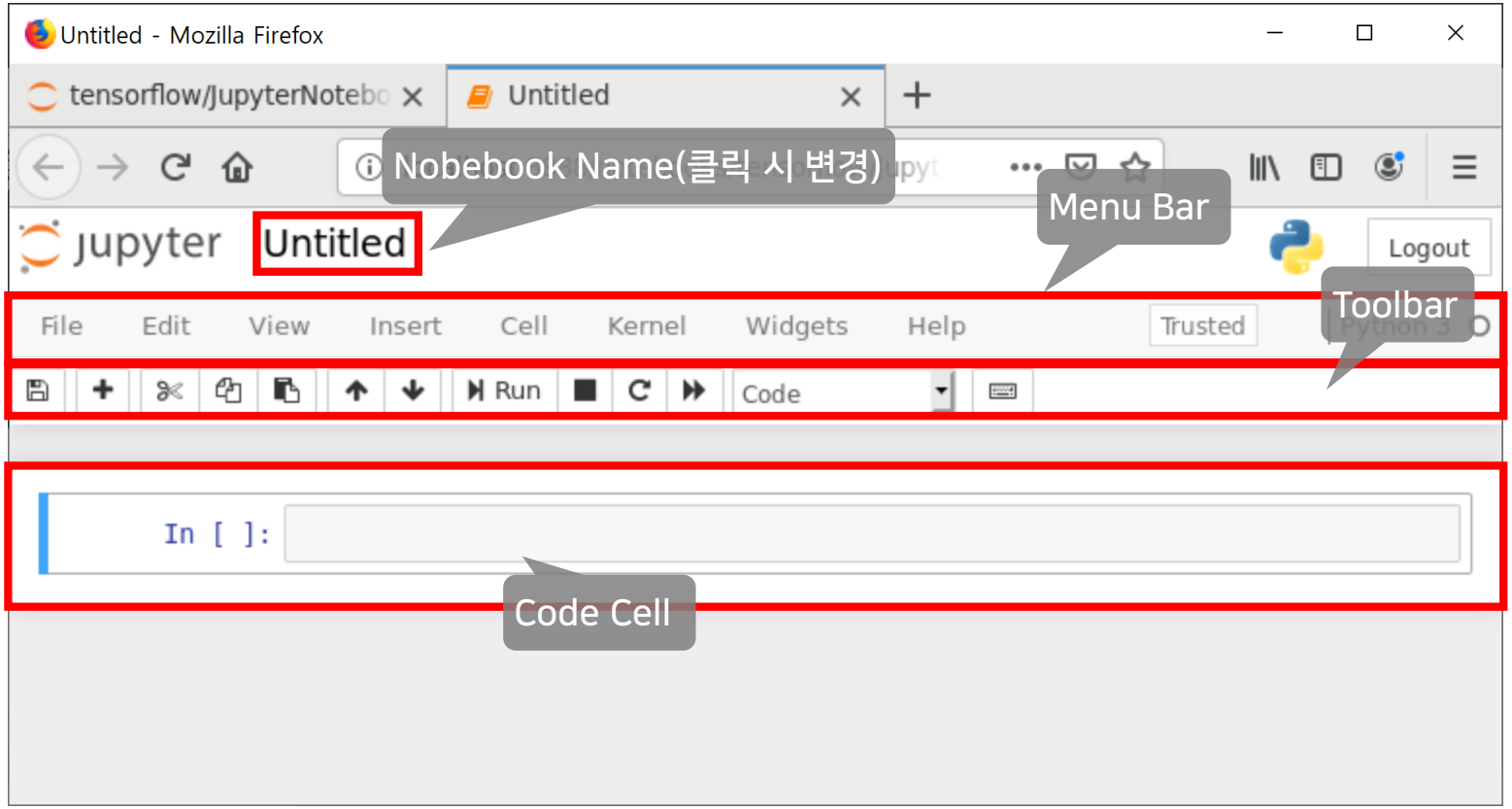
❑ Let's Start

- 자신이 원하는 디렉토리에서 연습 코드를 실행



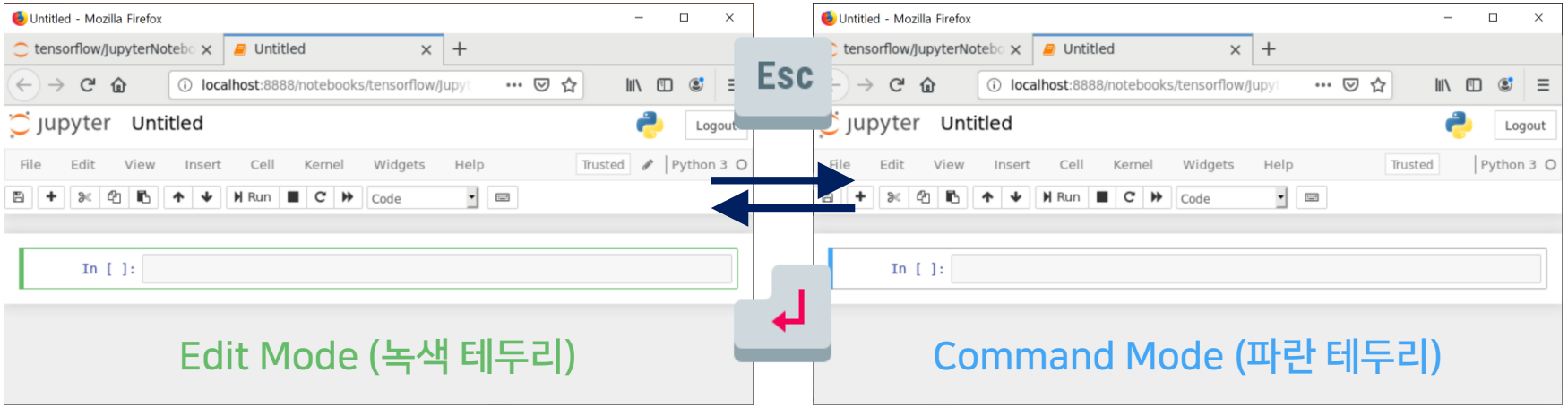
Jupyter Notebook

❑ 인터페이스



Jupyter Notebook

❑ 인터페이스



- 현재 Cell 실행 : Shift + Enter

- Navigator : 화살표 키
- 위에 Cell 삽입 : a / 아래에 Cell 삽입 : b
- Cell 삭제 : dd

Notebooks consist of a linear sequence of (code/markdown/raw) cells

Jupyter Notebook

❑ Keyboard Short Cut : Command Mode

F	find and replace	Shift-J	extend selected cells below
Ctrl-Shift-F	open the command palette	A	insert cell above
Ctrl-Shift-P	open the command palette	B	insert cell below
Enter	enter edit mode	X	cut selected cells
P	open the command palette	C	copy selected cells
Shift-Enter	run cell, select below	Shift-V	paste cells above
Ctrl-Enter	run selected cells	V	paste cells below
Alt-Enter	run cell and insert below	Z	undo cell deletion
Y	change cell to code	D,D	delete selected cells
M	change cell to markdown	Shift-M	merge selected cells
R	change cell to raw	Ctrl-S	Save and Checkpoint
1	change cell to heading 1	S	Save and Checkpoint
2	change cell to heading 2	L	toggle line numbers
3	change cell to heading 3	O	toggle output of selected cells
4	change cell to heading 4	Shift-O	toggle output scrolling of selected cells
5	change cell to heading 5	H	show keyboard shortcuts
6	change cell to heading 6	I,I	interrupt the kernel
K	select cell above	0,0	restart the kernel (with dialog)
Up	select cell above	Ctrl-V	Dialog for paste from system clipboard
Down	select cell below	Esc	close the pager
J	select cell below	Q	close the pager
Shift-K	extend selected cells above	Shift-L	toggles line numbers
Shift-Up	extend selected cells above	Shift-Space	scroll notebook up
Shift-Down	extend selected cells below	Space	scroll notebook down

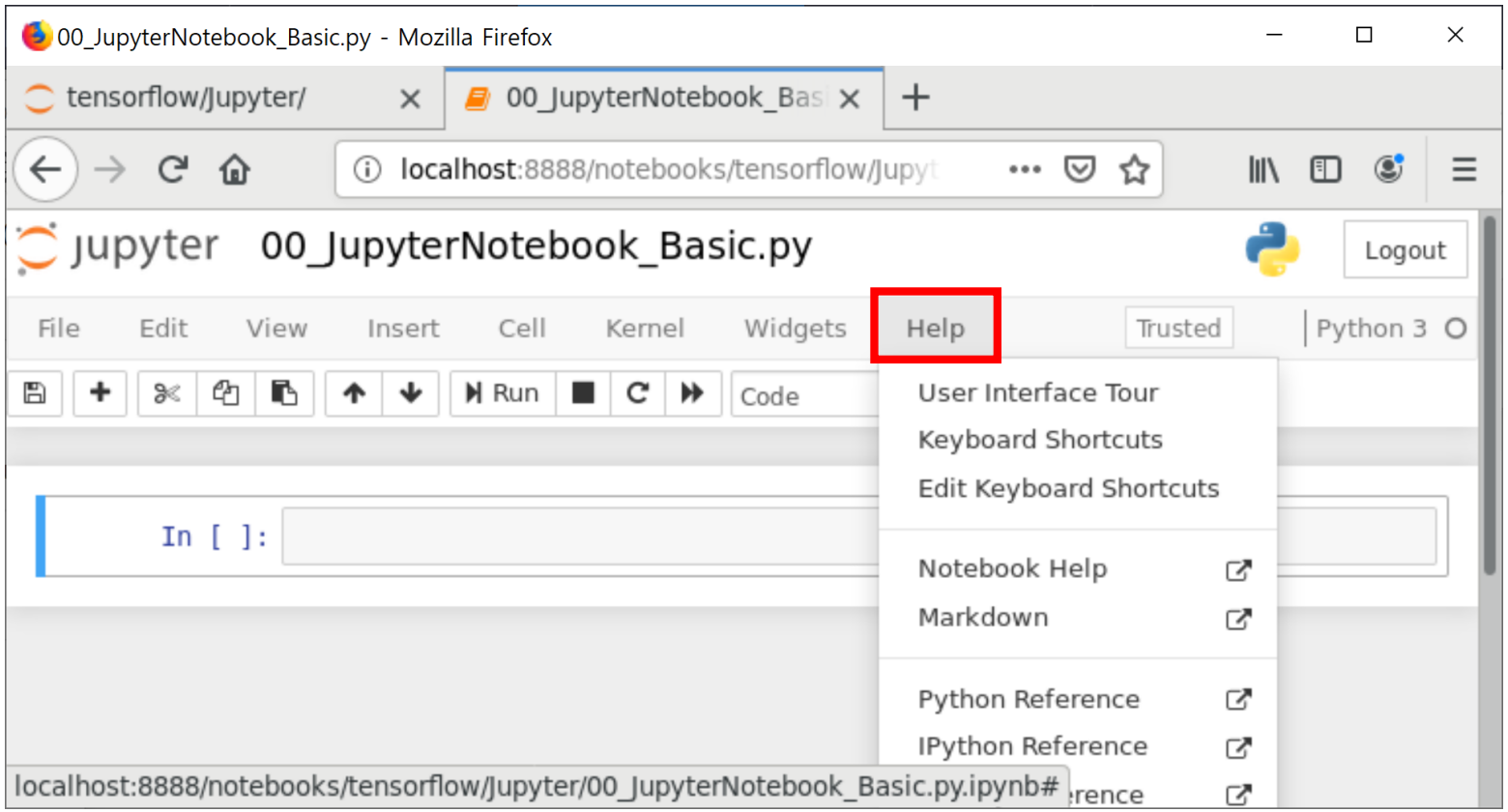
Jupyter Notebook

❑ Keyboard Short Cut : Edit Mode

Tab	code completion or indent	Ctrl-Backspace	delete word before
Shift-Tab	tooltip	Ctrl-Delete	delete word after
Ctrl-]	indent	Ctrl-Y	redo
Ctrl-[	dedent	Alt-U	redo selection
Ctrl-A	select all	Ctrl-M	enter command mode
Ctrl-Z	undo	Ctrl-Shift-F	open the command palette
Ctrl-/	comment	Ctrl-Shift-P	open the command palette
Ctrl-D	delete whole line	Esc	enter command mode
Ctrl-U	undo selection	Shift-Enter	run cell, select below
Insert	toggle overwrite flag	Ctrl-Enter	run selected cells
Ctrl-Home	go to cell start	Alt-Enter	run cell and insert below
Ctrl-Up	go to cell start	Ctrl-Shift-Minus	split cell at cursor
Ctrl-End	go to cell end	Ctrl-S	Save and Checkpoint
Ctrl-Down	go to cell end	Down	move cursor down
Ctrl-Left	go one word left	Up	move cursor up
Ctrl-Right	go one word right		

Jupyter Notebook

❑ More Details : Help Menu



Window Files → Linux

❑ 윈도우 C:₩ 의 경로

- 윈도우 파일 → Linux 복사는 가능
- Linux 파일 → 윈도우 복사는 불가능

```
[user - 2019.10.02 XX:XX:XX]
/home/user: cd /mnt/c
```

❑ GitHub

- GitHub 레포지토리 → Linux

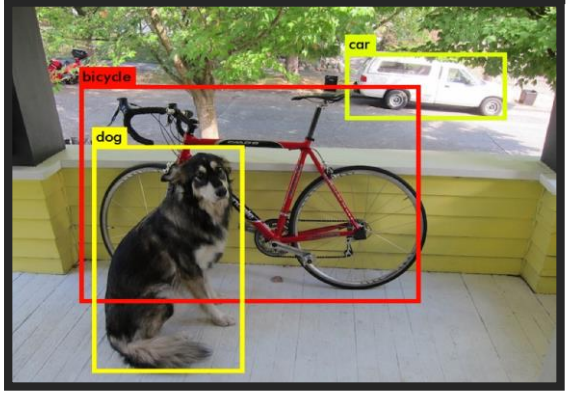
```
[user - 2019.10.02 XX:XX:XX]
/home/user: git clone github repository address
```

Outline

- WSL (Window Subsystem for Linux) 기반 개발 환경
  - WSL 설치 및 Setup 파일 설명
- 딥러닝 기초 및 실습
  - TensorFlow 기반 실습
- **딥러닝 하드웨어 가속기 연구 동향**

CNN Applications

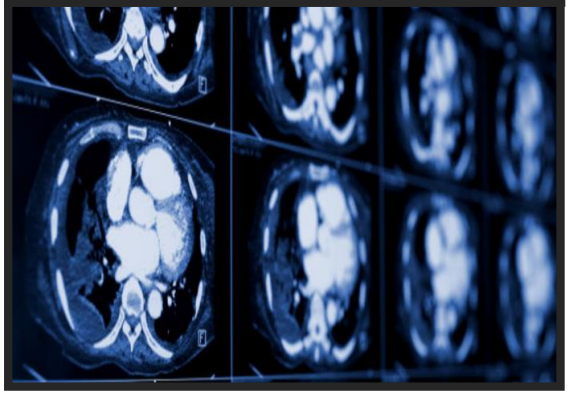
Image Process



Autonomous Machines

Security & Defense

Medical

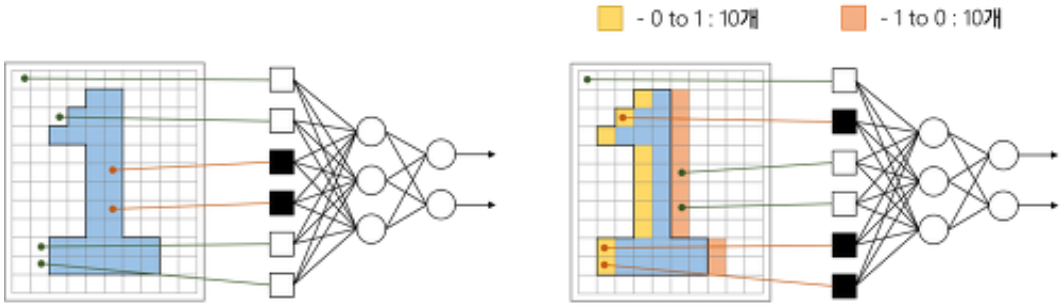


Game

Convolutional Neural Network

❑ MLP(Multi-Layer Perceptron)의 문제점

- Whole Feature에 대해서 인식



한 칸만 움직였는데,
변하는 값이 20개

- 부분적 Feature에 대해 인식하게 할 순 없을까?

1단계 : 가로, 동그라미, 세모, 부드러움

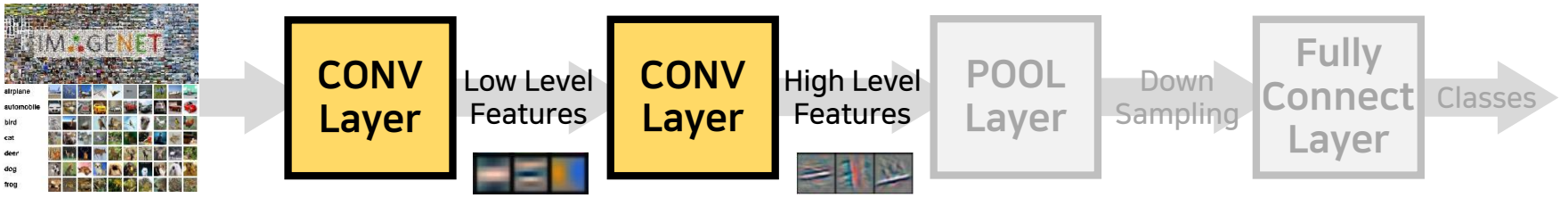
2단계 :눈, 코, 귀, 발

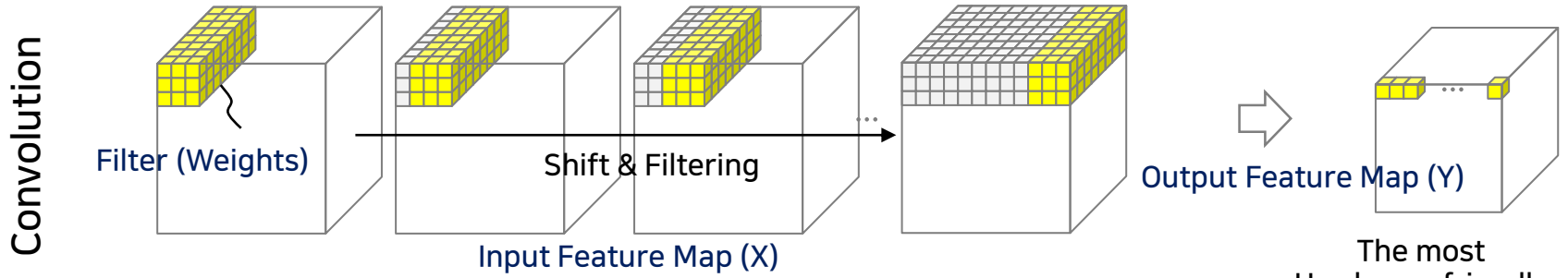
3단계 :고양이!

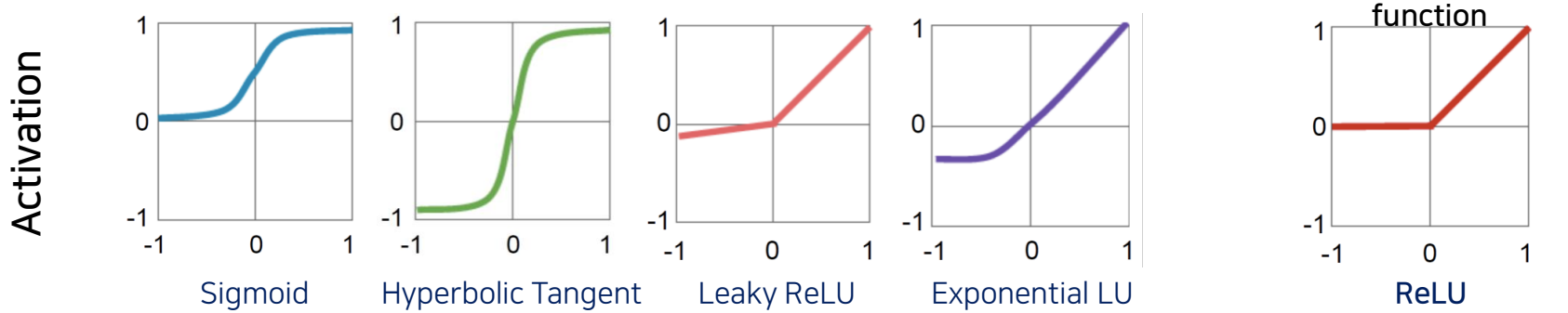
CNN!

https://gomguard.tistory.com/188

Convolutional Neural Network

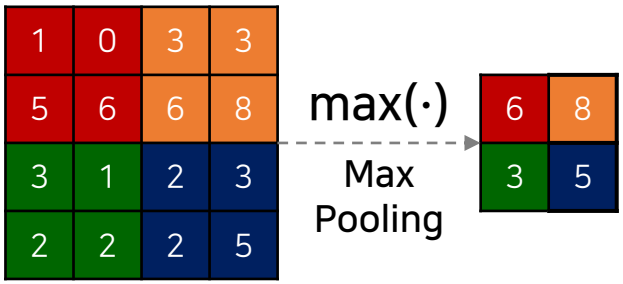


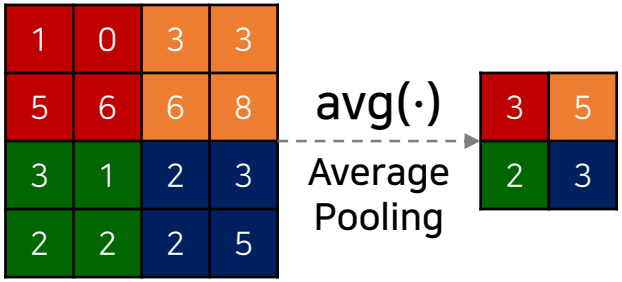




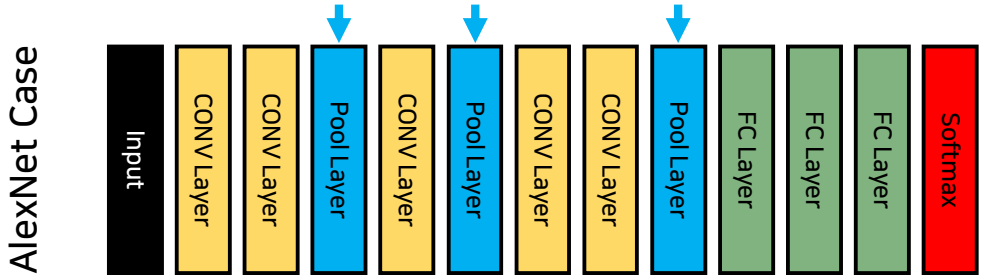
Source : MIT Tutorial

Convolutional Neural Network





Location of Pooling Layers

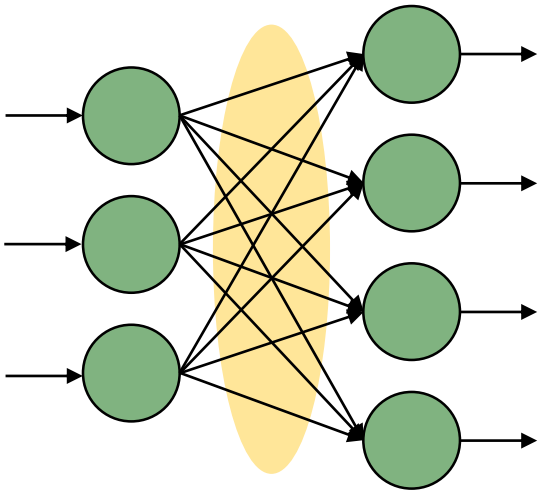


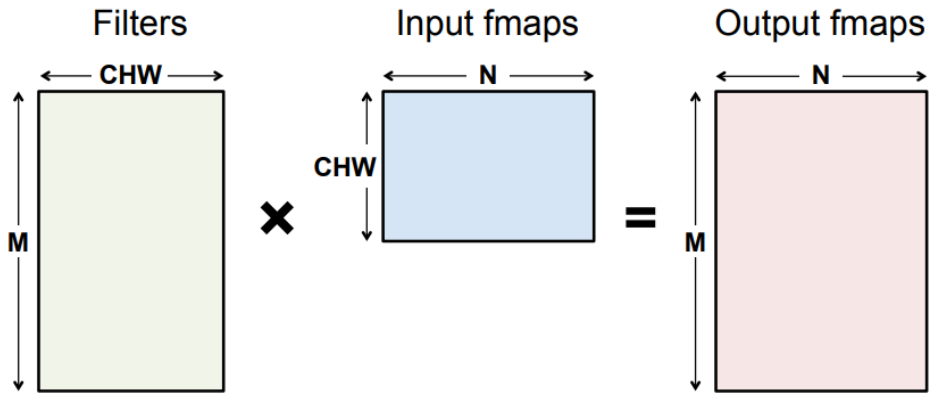
- Reduce resolution of each channel independently
- Increase translation-invariance and noise-resilience

Source : MIT Tutorial

Convolutional Neural Network

Fully-connected



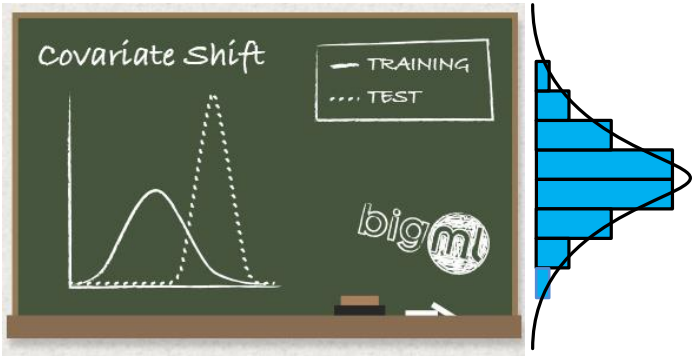


- Fully-connected layer account for more than 90% of total number of parameters, dominating memory and energy
- Simple matrix multiplication

Source : MIT Tutorial

And Others …

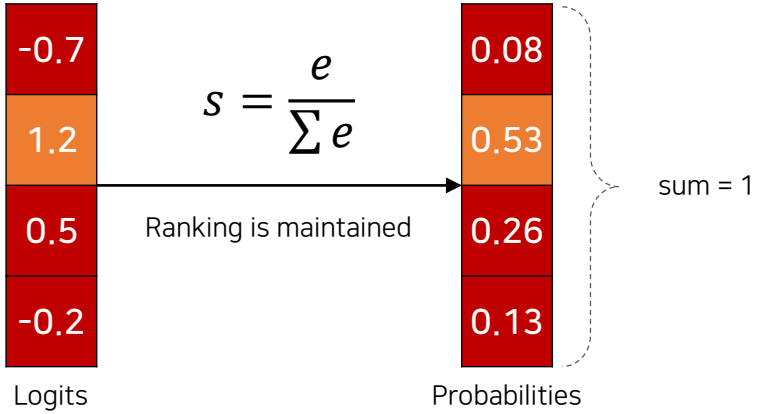
Normalization



[Sergey Ioffe et al., ICML 2015]

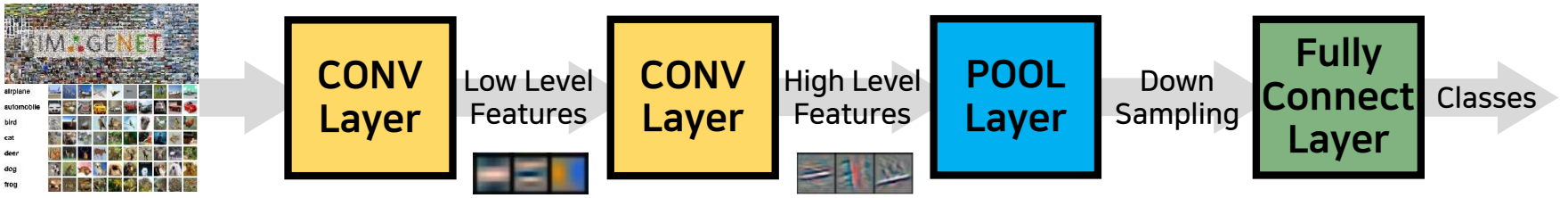
- Pre-processing to balance between the training and inference (accuracy highly relies on these procedure)

Softmax



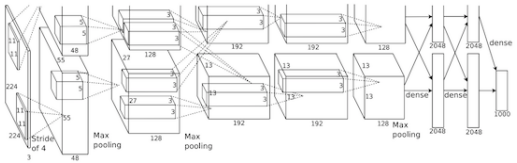
- Not essential when the difference between each class is not seriously importance

Advantage of CNN



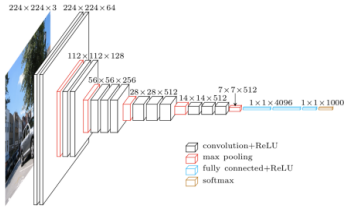
- MLP 의 한계점 극복
  - MLP는 1차원 입력 데이터만 받음
    - 이미지는 (RGB) 3 차원 데이터
- 파라미터 (Weight & Bias)를 줄이고 오버피팅 방지
- 풀링 레이어 등으로 노이즈에 내성
- 기존 Training 알고리즘 유지

Various CNN Configurations



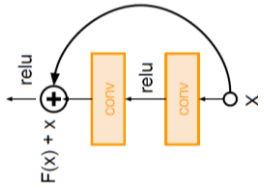
AlexNet [1]

Layer	Filter Size	# Filters	# Channels	Stride
1	11X11	96	3	4
2	5X5	256	96	1
3	3X3	384	256	1
4	3X3	384	384	1
5	3X3	256	384	1
6	Fully-Connected Layer			
7	Fully-Connected Layer			
8	Fully-Connected Layer			



VGG-16 [2]

Layer	Filter Size	# Filters	# Channels	Stride
1	3X3	64	3	1
2	3X3	64	64	1
3	3X3	128	64	1
4	3X3	128	128	1
5	3X3	256	128	1
6	3X3	256	256	1
7	3X3	256	256	1
8	3X3	512	256	1
9	3X3	512	512	1
10	3X3	512	512	1
11	3X3	512	512	1
12	3X3	512	512	1
13	3X3	512	512	1
14	Fully-Connected Layer			
15	Fully-Connected Layer			
16	Fully-Connected Layer			



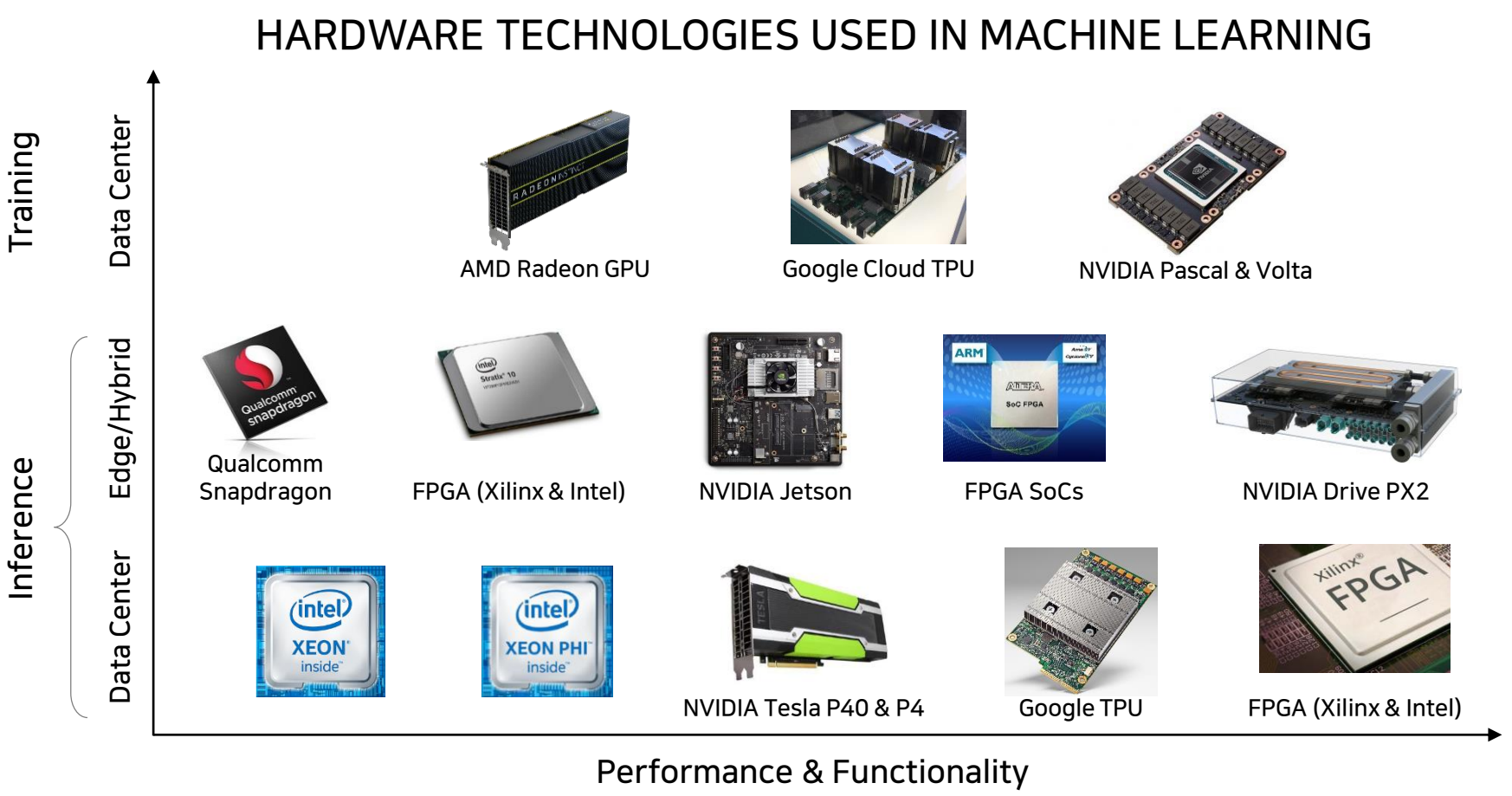
ResNet-50 [3]

Layer	Filter Size	# Filters	# Channels	Stride
1	7X7	64	3	2
2	1X1	256	64	1
3	1X1	64	64	1
4	3X3	64	64	1
5	1X1	256	64	1
	Addition Layer			
6	1X1	64	64	1
7	3X3	64	64	1
8	1X1	256	64	1
	Addition Layer			
9	1X1	64	256	1
10	3X3	64	64	1
11	1X1	256	64	1
	Addition Layer			
12	1X1	512	64	2
13	1X1	128	64	2
14	3X3	128	128	1
15	1X1	512	128	1
	Addition Layer			
16	1X1	128	512	1
17	3X3	128	128	1
18	1X1	512	128	1
	Addition Layer			
19	1X1	128	512	1

⋮

[1] [Krizhevsky et al. NIPS 2012]
[2] [Simonyan and Zisserman, ICLR 2015]
[3] [He et al., CVPR 2016]

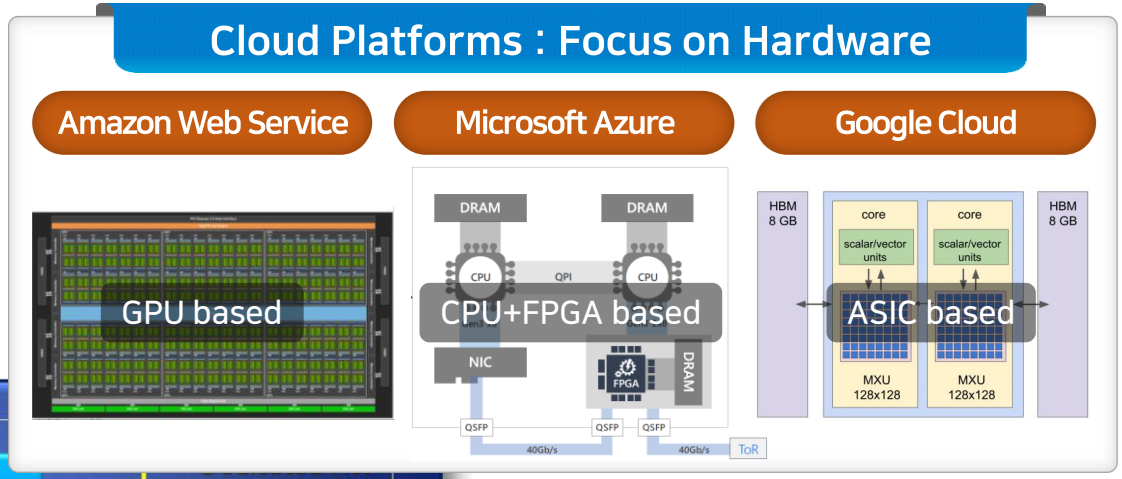
Hardware Technologies

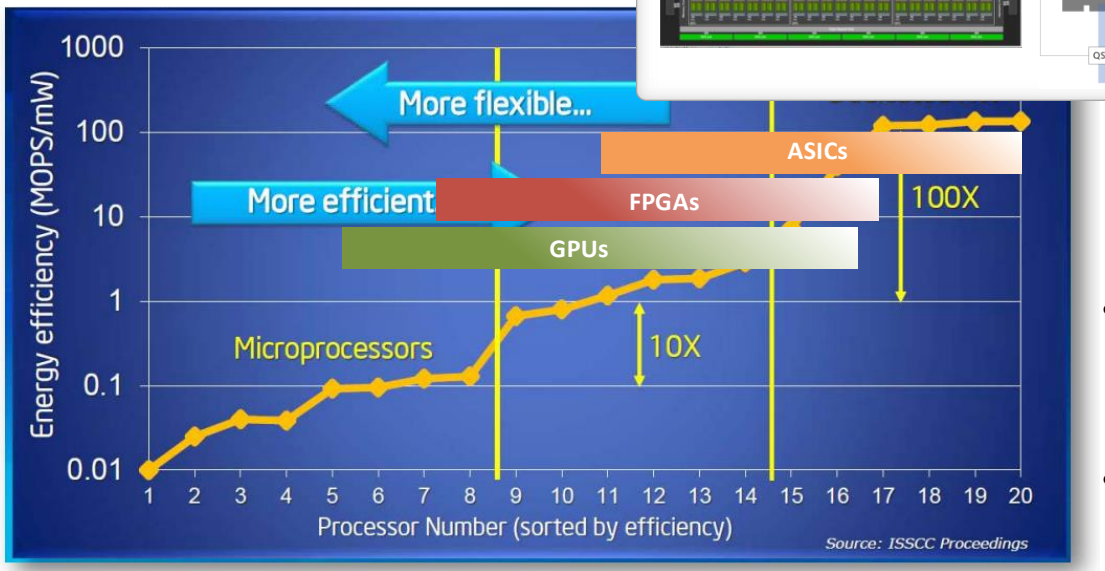


Source: https://tanjo.ai/contents/2323923

DNN on “Cloud Platforms”

SKT, 자일링스 AI 가속기 데이터센터에 적용…"성능 5배↑"
https://www.mk.co.kr/news/business/view/2018/08/513572/
2018. 8. 16. - 회로 변경 가능한 FPGA 반도체 탑재…향후 협력 확대. SK텔레콤은 스(Xilinx)의 칩세트를 탑재한 인공지능(AI) 가속기를 데이터 …

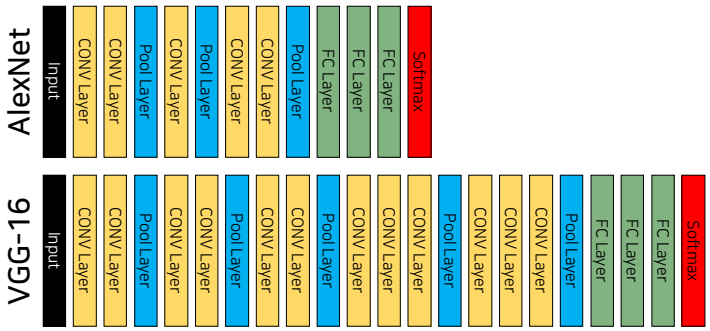




Source: Bob Broderson, Berkeley Wireless group

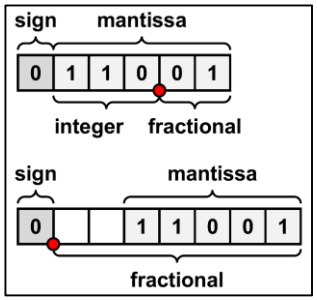
- Accelerator is more efficient in terms of power and energy consumption
- Trade off between Energy Efficiency and Flexibility

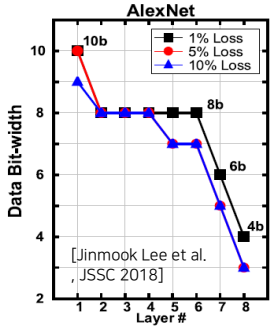
Need Reconf. Accelerator



In different Network

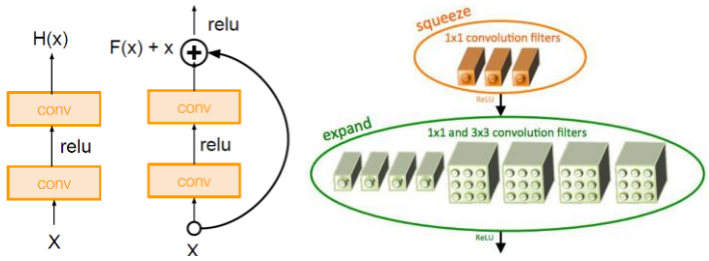
- Different number of layers
- Different number of filters / channels





In different Quantization

- Different bit-width of different layers

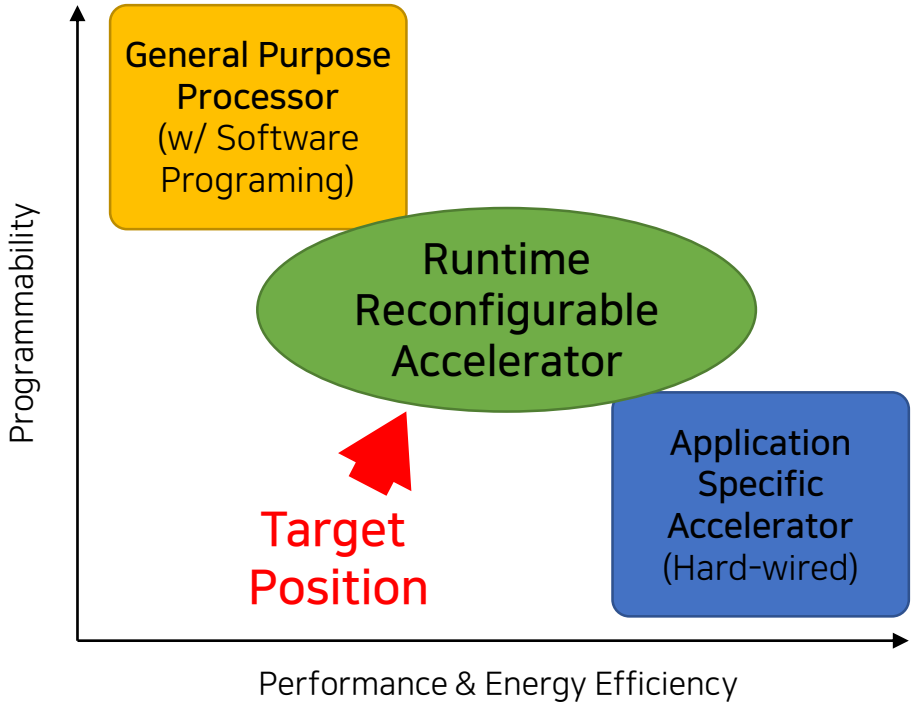


In different Architecture

- Different algorithmic structures

Reconfig. Vs Energy Efficiency

Dynamically Reconfigurable DNN Accelerator

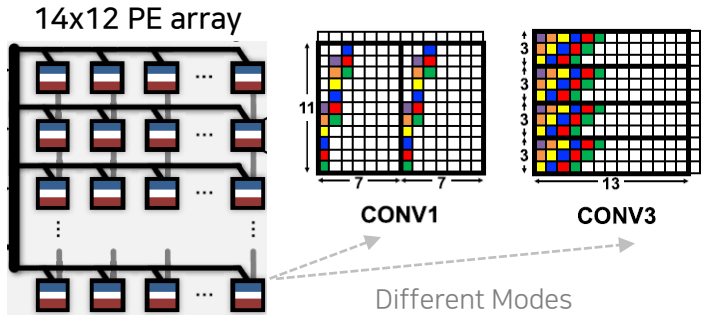


High-level Reconfiguration

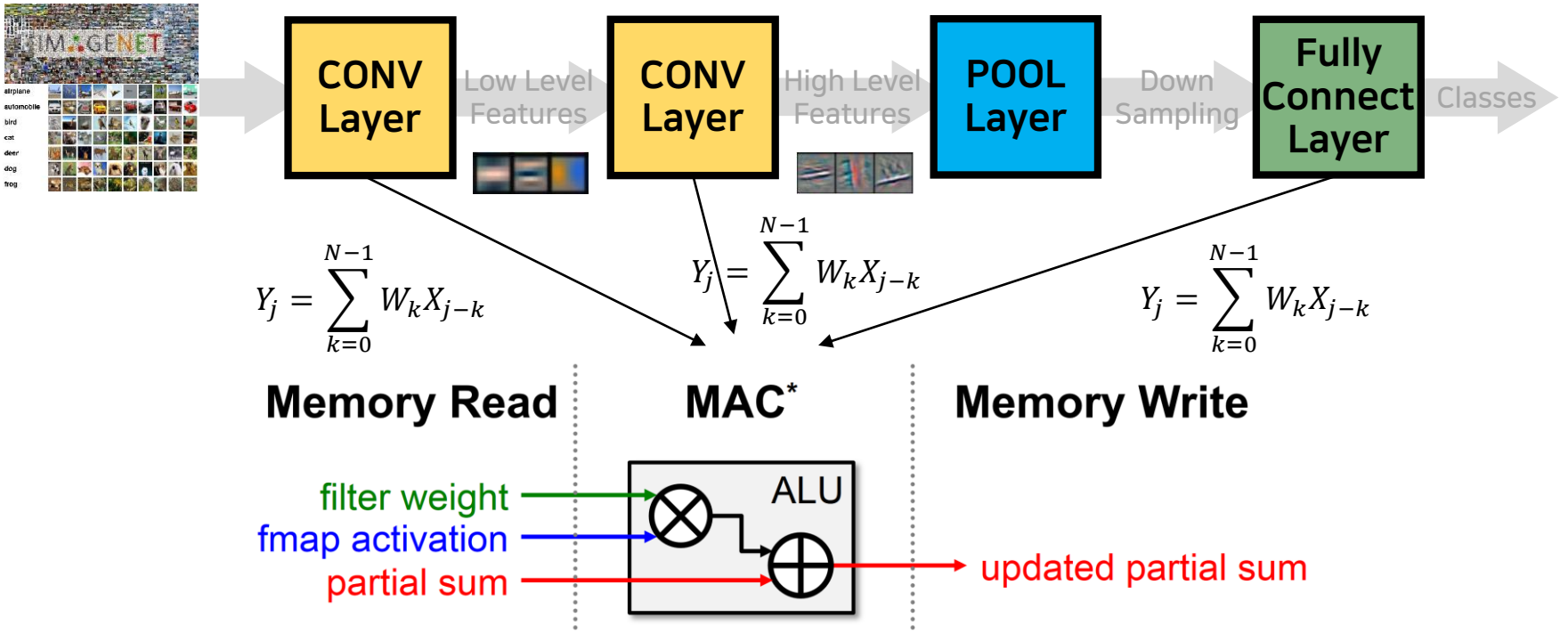
```
VLOAD   $3, $0, #100          // load input vector from address (100)
MLOAD   $4, $2, #300          // load weight matrix from address (300)
MMV     $7, $1, $4, $3, $0    // Wx
VAV     $8, $1, $7, $5        // tmp=Wx+b
VEXP    $9, $1, $8            // exp(tmp)
VAS     $10, $1, $9, #1       // 1+exp(tmp)
VDV     $6, $1, $9, $10       // y=exp(tmp)/(1+exp(tmp))
VSTORE  $6, $1, #200          // store output vector to address (200)
```

Use Instruction Set Architecture

Low-level Reconfiguration

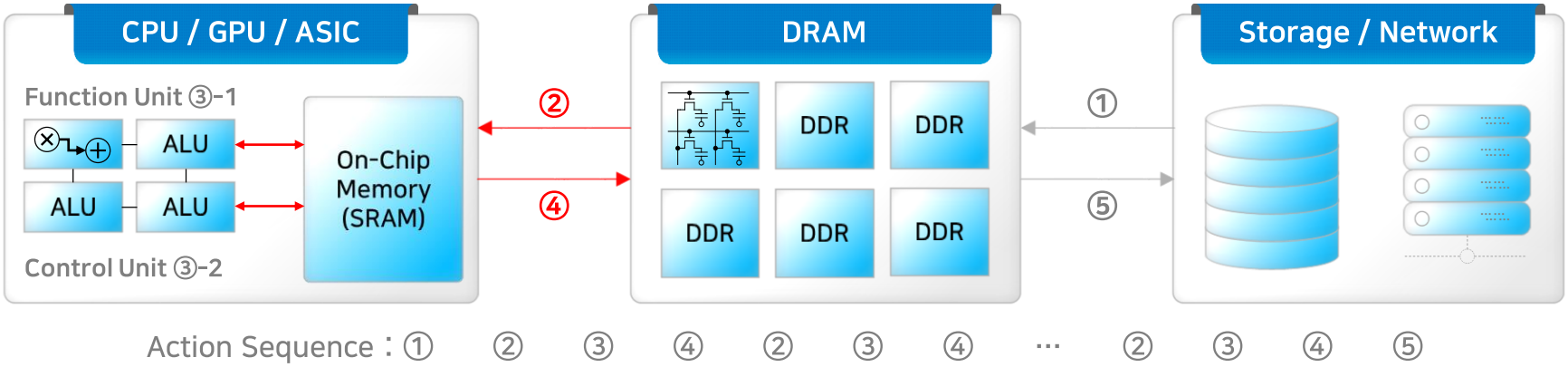


Main Operation in CNN

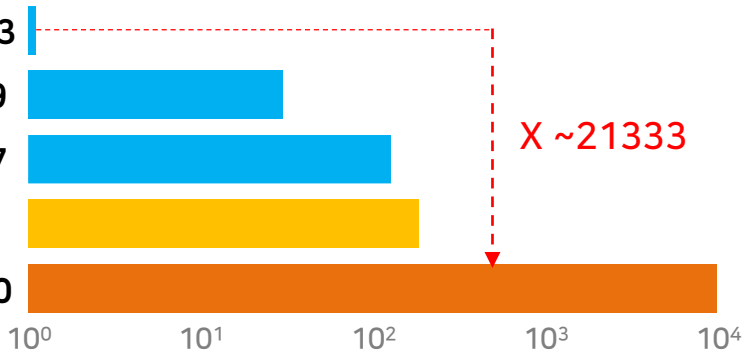


Architecture	Weight Size	Ifmap Size	# Multiply-Adds	Top-1 Accuracy
AlexNet	238 MB	1.6 MB	724 M	57.10 %
VGG-16	528 MB	34.8 MB	15.5 B	70.50 %
ResNet-50	99 MB	37.5 MB	3.9 MB	75.20 %

Data-Centric CNN



Operation	Energy (pJ) & Relative Energy Cost
8bit Add	0.03
32bit FP Add	0.9
32bit FP Mult	3.7
32bit SRAM Read	5
32bit DRAM Read	640

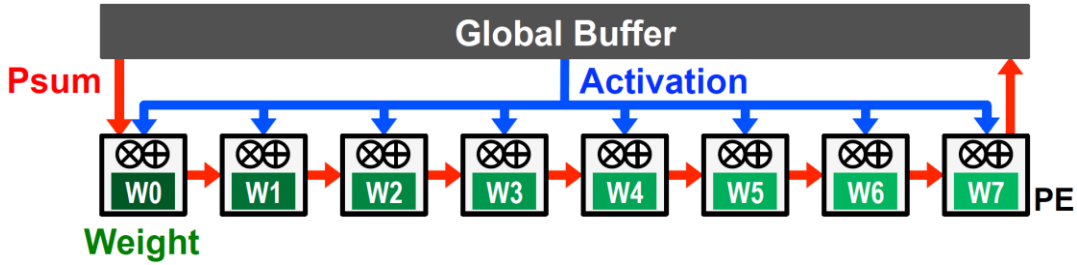


[M Horowitz et al., ISSCC 2014]

Accelerator Design

- Maximize Data Reuse
- Reduction: Computation Size
- Reduction: Computation Number
- Processing-in-memory

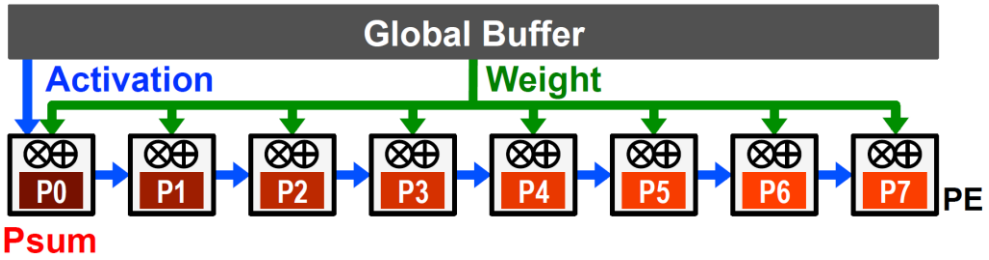
Maximize Data Reuse : Data Flow



[nn-X (NeuFlow), *CVPRW* 2014] [Park, *ISSCC* 2015] [ISAAC, *ISCA* 2016] [PRIME, *ISCA* 2016]

Weight Stationary

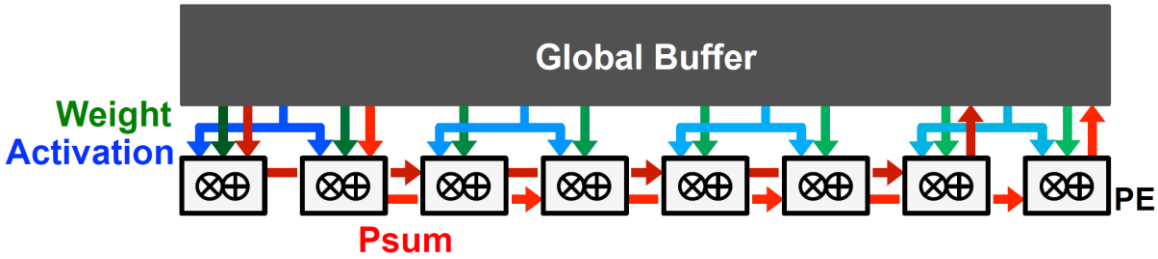
- Maximize weight reuse
- Broadcast activation
- Accumulate pSUMs spatially



[Peemen, *ICCD* 2013] [ShiDianNao, *ISCA* 2015] [Gupta, *ICML* 2015] [Moons, *VLSI* 2016]

Output Stationary

- Maximize pSUM reuse
- Broadcast weight
- Reuse activation spatially



[DianNao, *ASPLOS* 2014] [DaDianNao, *MICRO* 2014] [Zhang, *FPGA* 2015] [TPU, *ISCA* 2017]

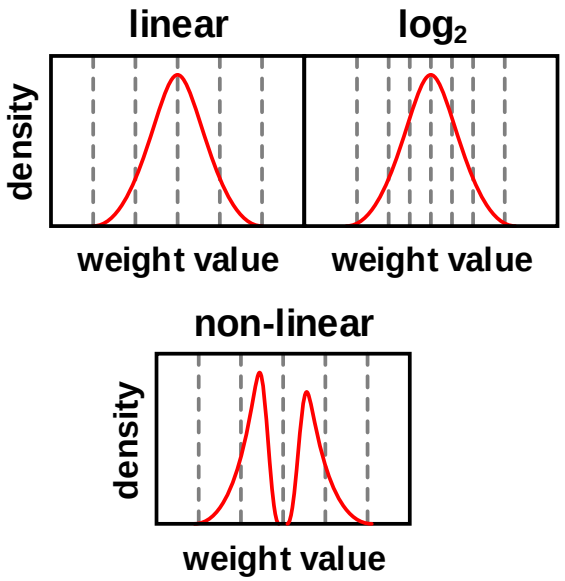
No Local Reuse

- Use a large global buffer
- Reduce DRAM access
- Multicast activation & weight
- Accumulate pSUMs spatially

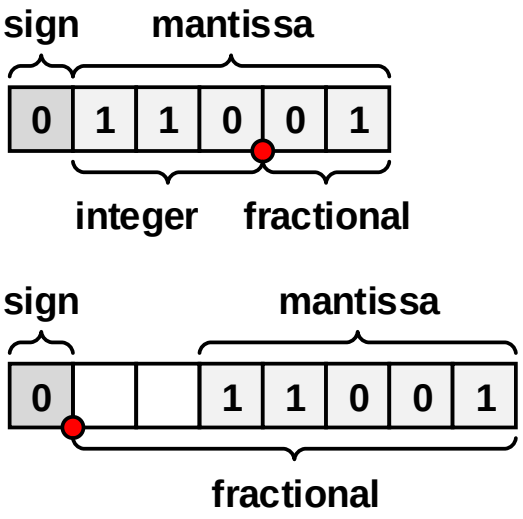
Source : MIT Tutorial

Reduction: Computation Size

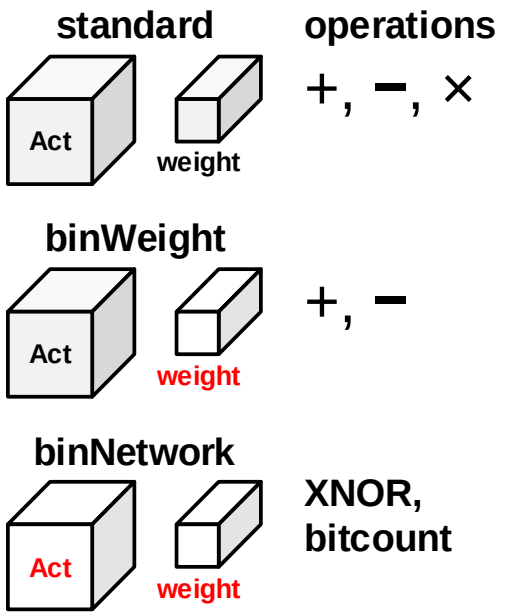
Non-linear Quantization



Dynamic Fixed Point



Binary Neural Network



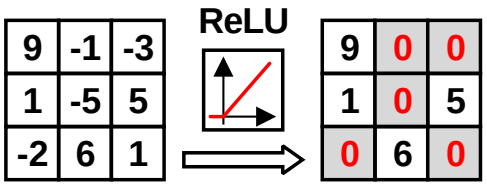
- Directly reduced the memory & PEs
- Trade-off : Bit-width ↔ Accuracy

Processing-in-Memory

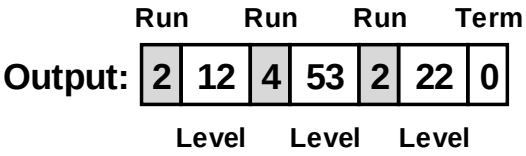
[Ref.] Woong Choi, Kyungrak Choi, and Jongsun Park, "Low Cost Convolutional Neural Network Accelerator Based on Bi-directional Filtering and Bit-width Reduction", IEEE Access, vol. 6, pp. 14734-14746, Mar. 2018.

Reduction: Computation Number

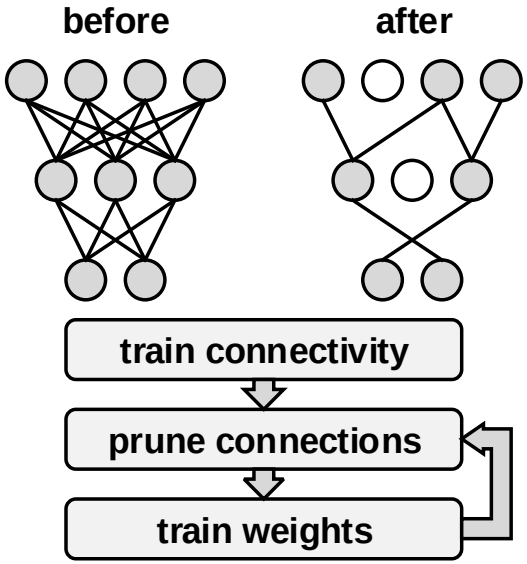
Activation Sparsity



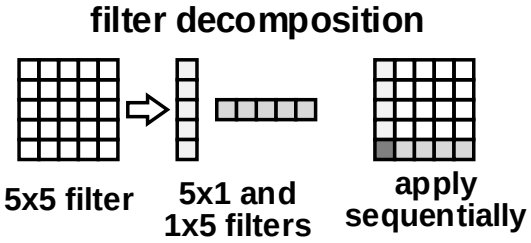
Input: 0,0,12,0,0,0,0,53,0,0,22, ...

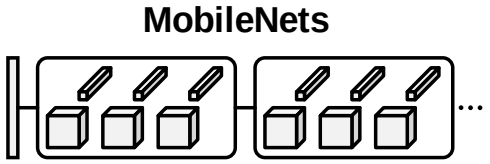


Network Pruning



Compact Architecture

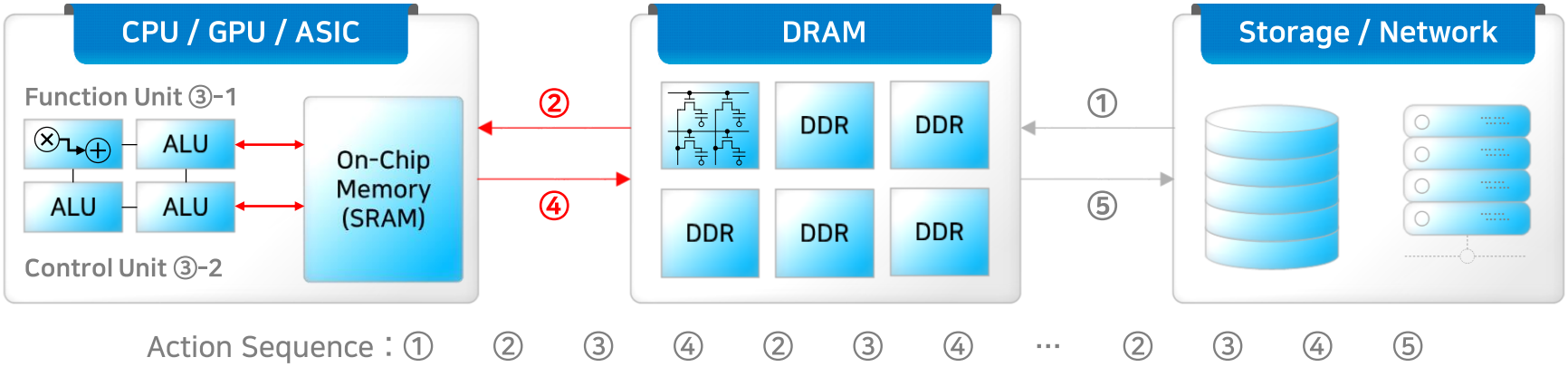


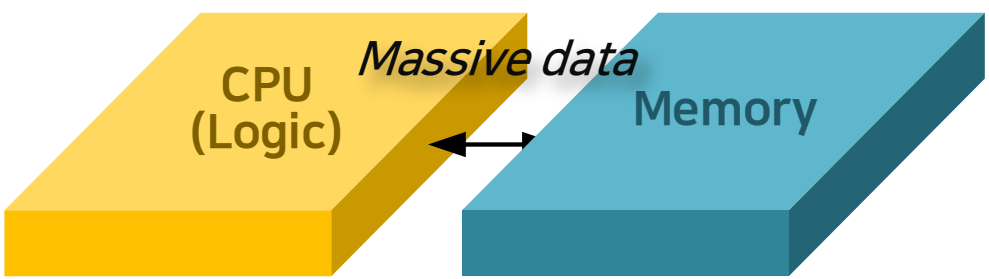


- Reduced Activation → Gating or skipping cycle & memory access
- Accuracy loss depending on the techniques

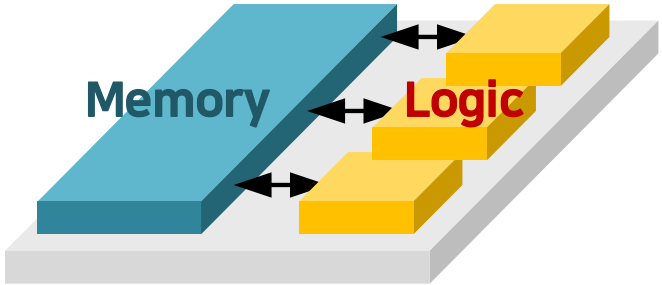
[Ref.] Woong Choi, Kyungrak Choi, and Jongsun Park, "Low Cost Convolutional Neural Network Accelerator Based on Bi-directional Filtering and Bit-width Reduction", IEEE Access, vol. 6, pp. 14734-14746, Mar. 2018.

Processing in Memory





von Neumann architecture

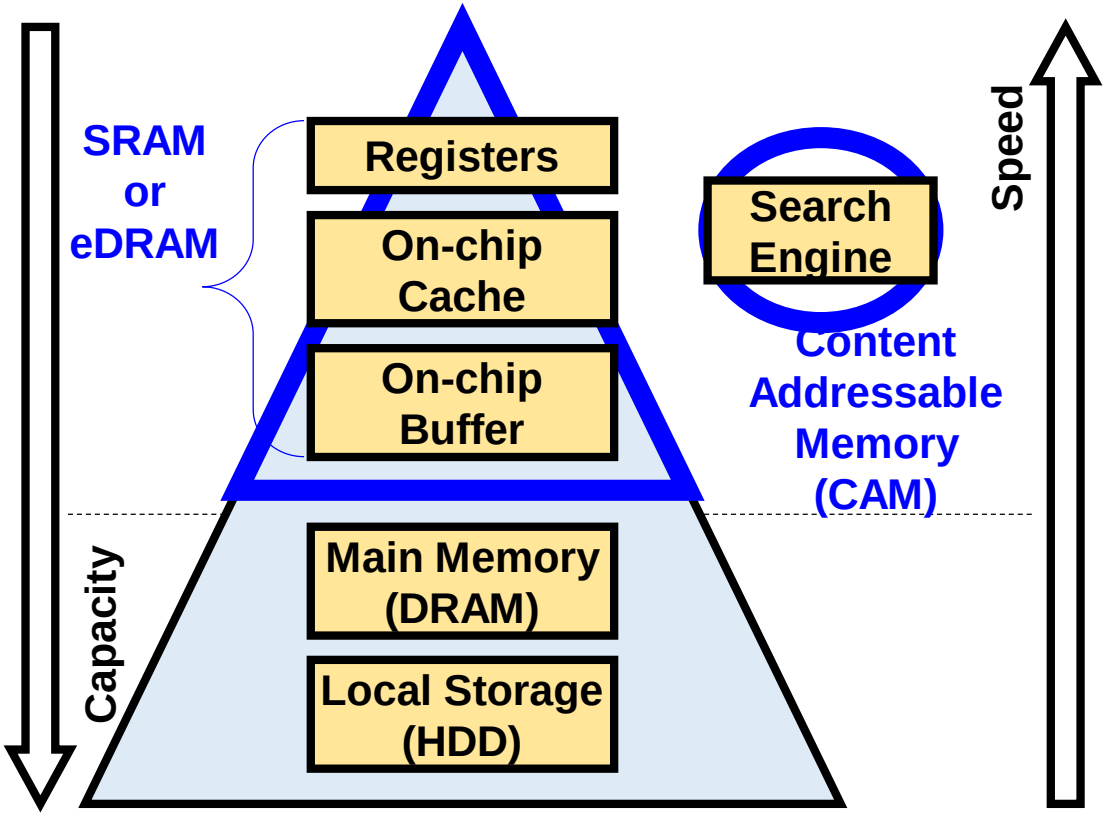


Processing-in-memory

- PIM: A technique that performs simple logic within a memory device to reduce the amount of data being passed to the processor.

Processing in Memory

❑ Memory Hierarchy



w/o Embedded Mem.

- Limited # of I/O (Bandwidth)
- Power expensive inter-chip comm.

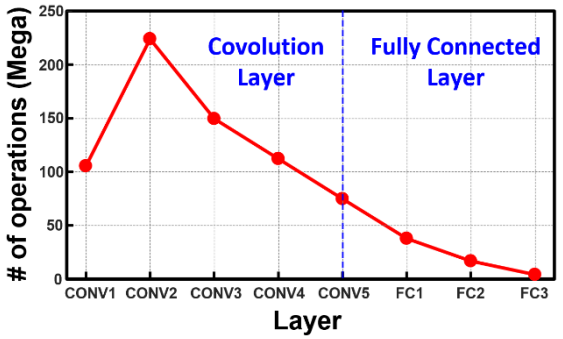
w/ Embedded Mem.

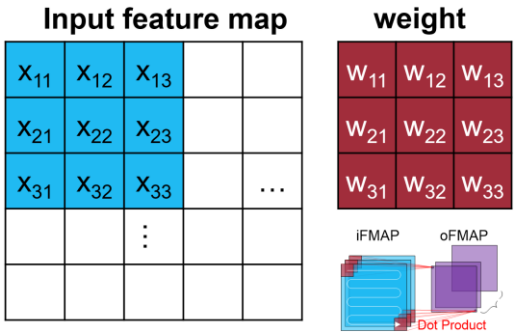
- Less power
- Higher performance

Processing in Memory

❑ Convolutional Operation

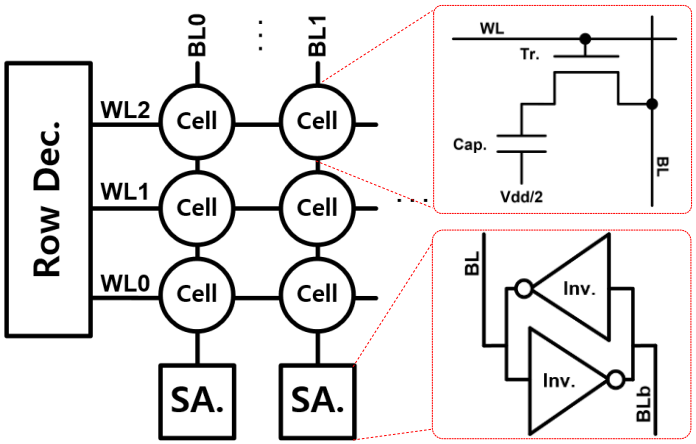
- > ~90% of overall operation
- Non-binary Net (Mult. & Add.)
  - $x_{11}*w_{11}+ \cdots x_{33}*w_{33}$
- Binary Net (XNOR & Pop Count)
  - popcount($x_{11} \otimes w_{11}, \cdots, x_{33} \otimes w_{33}$)



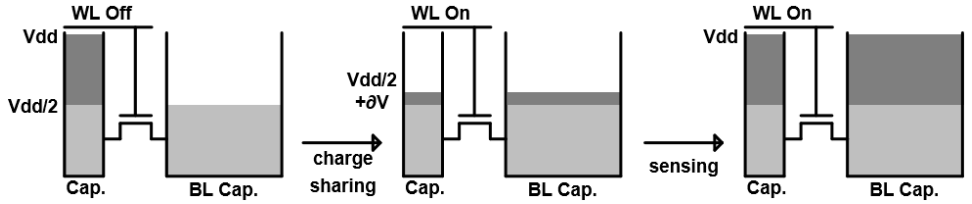


Processing in Memory

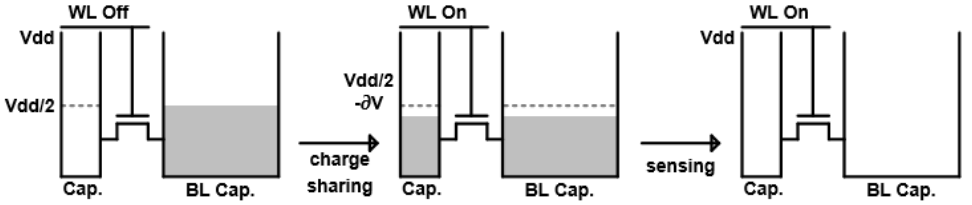
❑ Basic DRAM Operation : Read → Write-Back



DRAM Array Architecture



DRAM read operation: sensing "1"



DRAM read operation: sensing "0"

- DRAM consists of Cell(1T1C) array
- WL on → charge sharing btw. cell cap. and Bit-line(BL) cap. → sensing

[Vivek Seshadri et al., MICRO 2017]

Processing in Memory

❑ DRAM-based PIM : AND, OR Operation

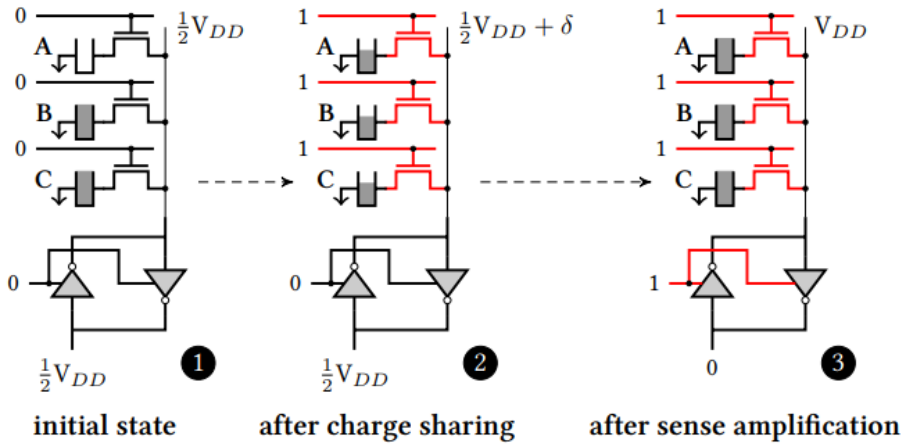


Figure 4: Triple-row activation

R = A *OR* B

A	B	C
1	1	1
1	0	1
0	1	1
0	0	1

Charge Sharing

R
1
1
1
0

R = A *AND* B

A	B	C
1	1	0
1	0	0
0	1	0
0	0	0

Charge Sharing

R
1
0
0
0

DRAM PIM: OR/AND Operation Table

- AND/OR operation : 3 WLs on → charge sharing → sensing
- Majority function for A,B, and C input
- A & B when C = 1, A || B when C = 0

[Vivek Seshadri et al., MICRO 2017]

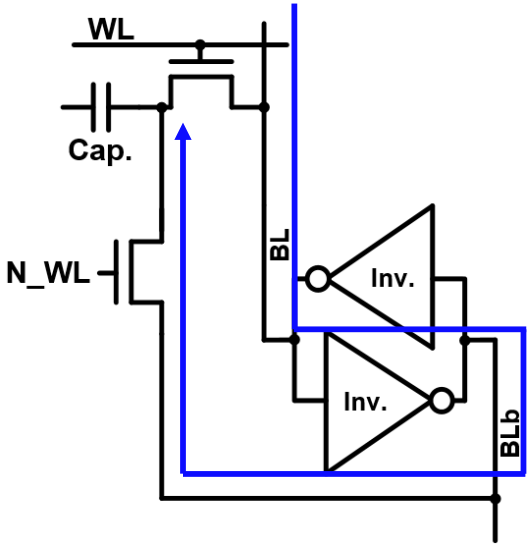
Processing in Memory

❑ DRAM-based PIM : NOT Operation (Inverting)

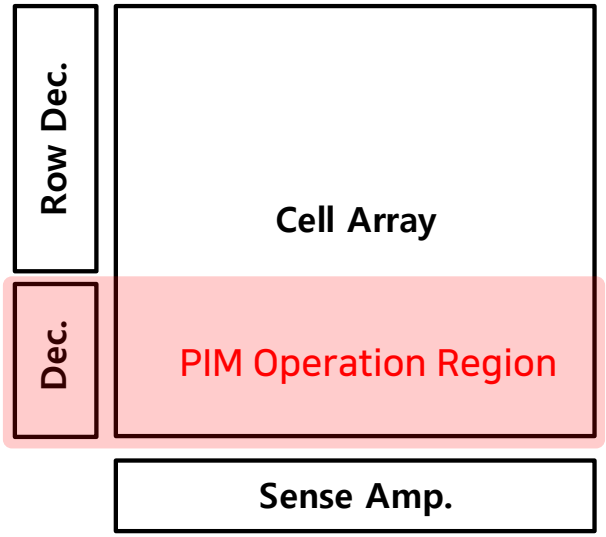
Read

Inverting

Write-back



Special 2T1C Cell & Column Line



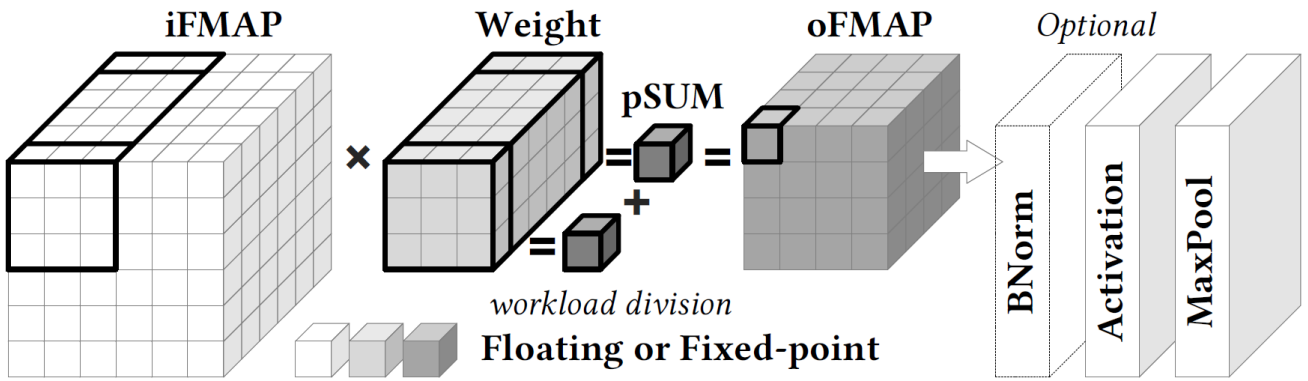
DRAM 2T1C Row & Specific PIM Region

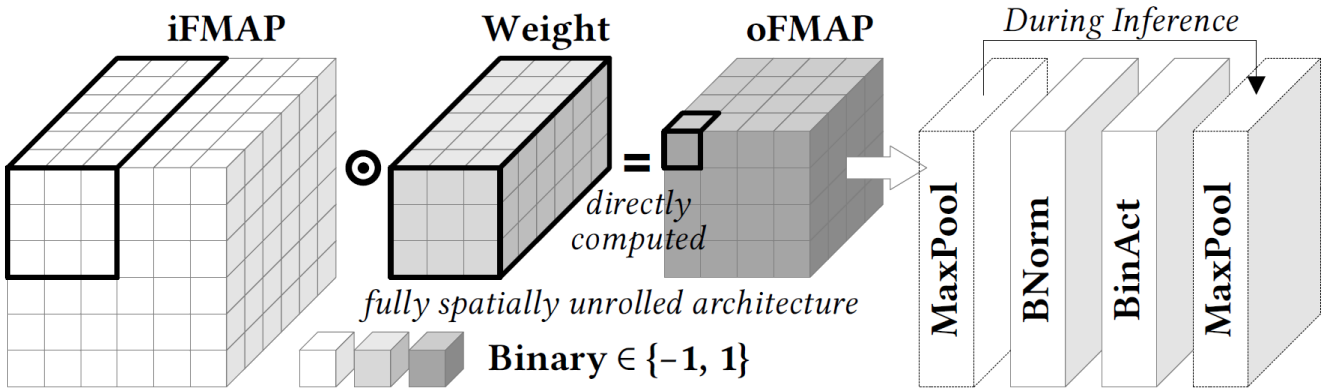
- 2T1C Cell : Read → Inverting → Write-Back
- Separated decoder to reduce hardware overhead
- Throughput x32, energy x35 compared to DDR3 in AND/OR/NOT

[Vivek Seshadri et al., MICRO 2017]

Processing in Memory

❑ Binary Neural Network





Floating or Fixed Point Operation

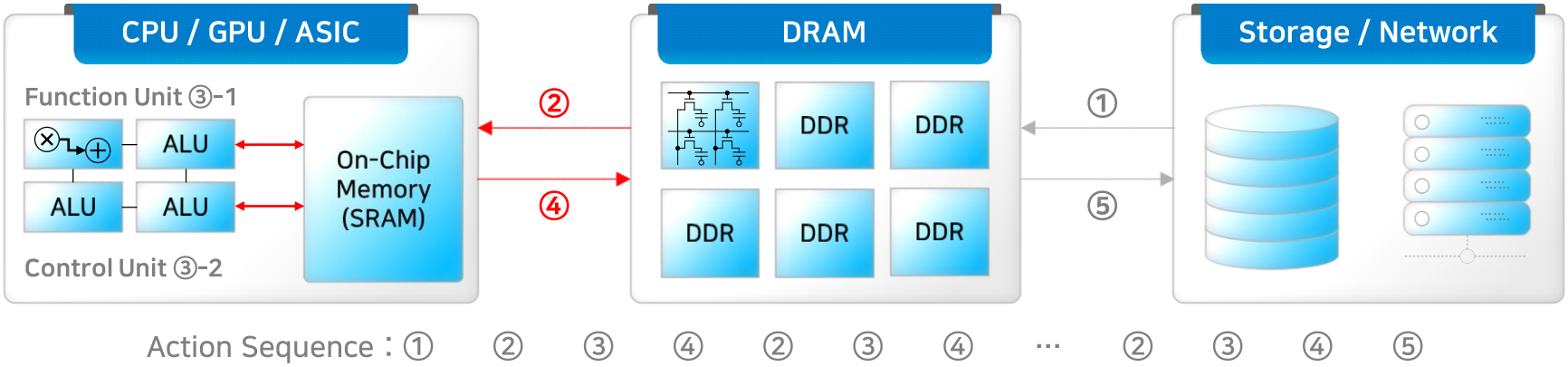
Binary (Bit-wise) Operation (Hardware-friendly)

Explosive PIM Researches

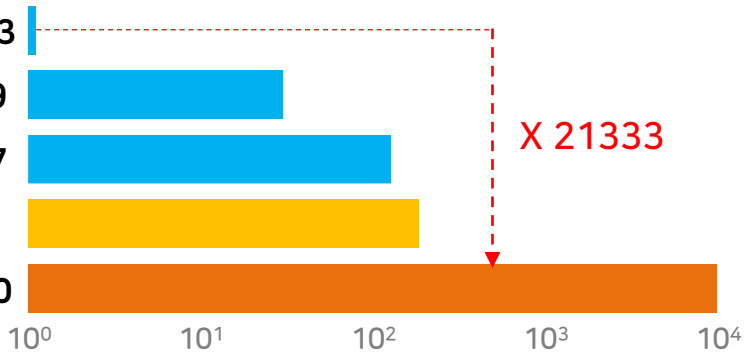
[1] M. Courbariaux et al. 2016. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1. (2016). arXiv:1602.02830.

[2] W. Tang et al. 2017. Wang, "How to Train a Compact Binary Neural Network with High Accuracy?. in 2017 AAAI, 2625-2631.

Data-Centric CNN



Operation	Energy (pJ) & Relative Energy Cost
8bit Add	0.03
32bit FP Add	0.9
32bit FP Mult	3.7
32bit SRAM Read	5
32bit DRAM Read	640



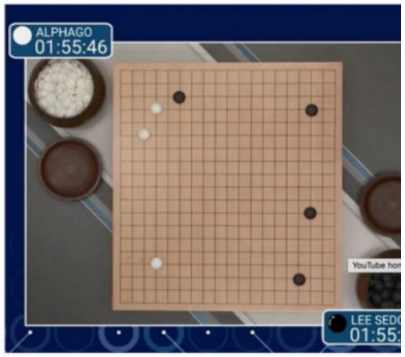
[M Horowitz et al., ISSCC 2014]

Accelerator Design

- Maximize Data Reuse
- Reduction: Computation Size
- Reduction: Computation Number
- Processing-in-memory

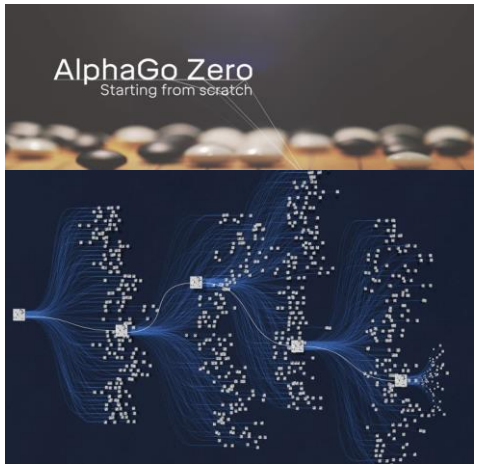
AI Hardware vs Human

❑ Energy Discrepancy



AlphaGO
1202 CPUs, 176 GPUs,
100+ Scientists.

$5*10^4$ W



Alphago Zero

$1\sim2*10^3$ W

VS.

20 W

- Where does this inefficiency come from? Algorithm, Architecture, Circuits, Device, and Materials

Q & A

woongchoi@sookmyung.ac.kr