

KUBIG 2023-1

겨울방학 자연어처리(NLP) 분반

WEEK 1

CONTENTS



스터디 OT



친해지길 바라



1주차 강의



Part 1

스터디OT

- 스터디 안내
- 사용 교재, 주차 별 커리큘럼
- 과제 진행 방식

예시) 6주차

전반부 | 제성 - Attention, Transformer 복습 발표

중반부 | BERT / BART 진도

후반부 | BERT / BART 관련 예습 코드 리뷰

예습 과제 코드 공유, 피드백

복습 과제
발표

30분

강의식
스터디

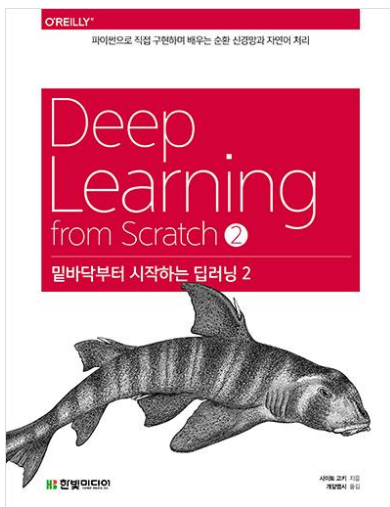
1시간 15분

예습 과제
코드리뷰

15분

매주 한 명씩 돌아가면서 지난주 공부 내용 복습
Ex) 교재 내용 요약, 관련 강의 내용 요약,
관련 논문 소개, 코드 구현 발표

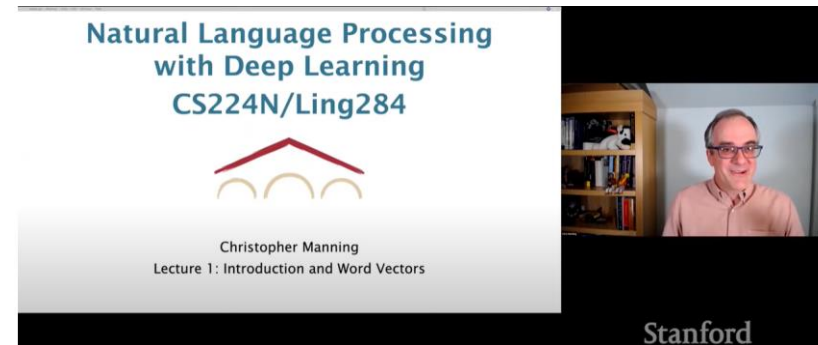
스터디 교재



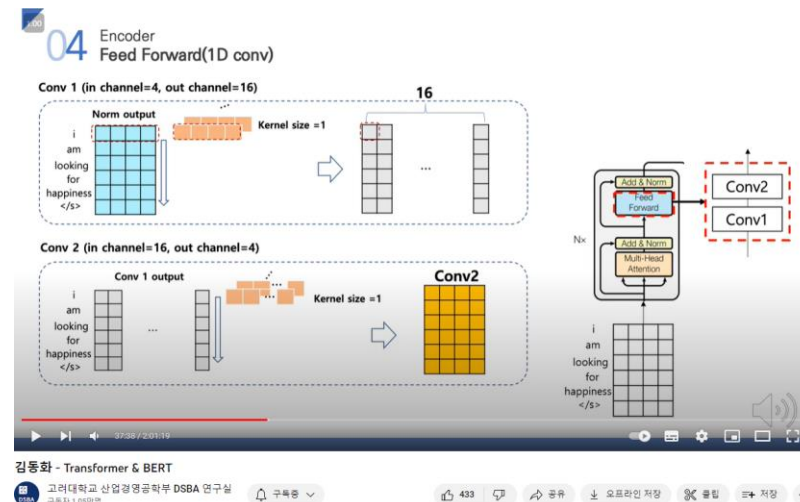
| 밑바닥부터 시작하는 딥러닝 2



| 딥러닝을 이용한 자연어 처리 입문



| Stanford CS224N



| 고려대학교 DSBA 연구실

커리큘럼

주차	복습과제	진도	예습과제
1주차 1/12	<ul style="list-style-type: none">- 지난 주차시에 배운 내용 관련해서 교재 내용 or 논문 or 강의 자료 자유롭게 복습 정리- 2~3회 정도 추가 코딩 구현 과제 나갈 예정 (선택)- 수업 시작 전 복습 과제 발표 (1~2명)- 마감기한 : 세션 시작 전까지	스터디 OT 딥러닝 복습	<ul style="list-style-type: none">- 다음 주차시에 배울 내용과 관련해서 실습 코드 제공- 그대로 클론 코딩 해봐도 좋고 실습 파일에 주어진 데이터 이외에의 데이터를 갖고와 다양한 시도를 해봐도 좋습니다- 마감기한 : 수요일 오후 6시까지
2주차 1/19		텍스트 전처리 언어 모델	
3주차 1/26		워드 임베딩 : Word2Vec GloVe ELMo	
4주차 2/2		순환 신경망 : RNN LSTM	
5주차 2/9		Attention Transformer	
6주차 2/16		BERT BART	
7주차 2/23		GPT	
쿠빅 콘테스트 3/2			



Part 2

찬해지길 바라

자기소개

김태영

엄기영

김희준

이영노

민윤기

임청수

박민규

최규빈

박종혁

하예은

반민정

홍여빈

기본 인적사항
이름 / 학과 / 나이 / 사는 곳
쿠빅 기수

가보자고



NLP에 분반에 들어오게 된 계기
(관련 프로젝트 경험 / 관심 있는 분야 ...)
+ NLP 이해도 (상-중-하)

취미
(유튜브 / 산책 /
되도록 겹치지 않고
다양하게
이야기해주세요 😊)

! 오늘의 TMI !



Part 3

1주차강의

- 딥러닝 복습

머신러닝

Example : Housing Price Prediction



House (X_n)

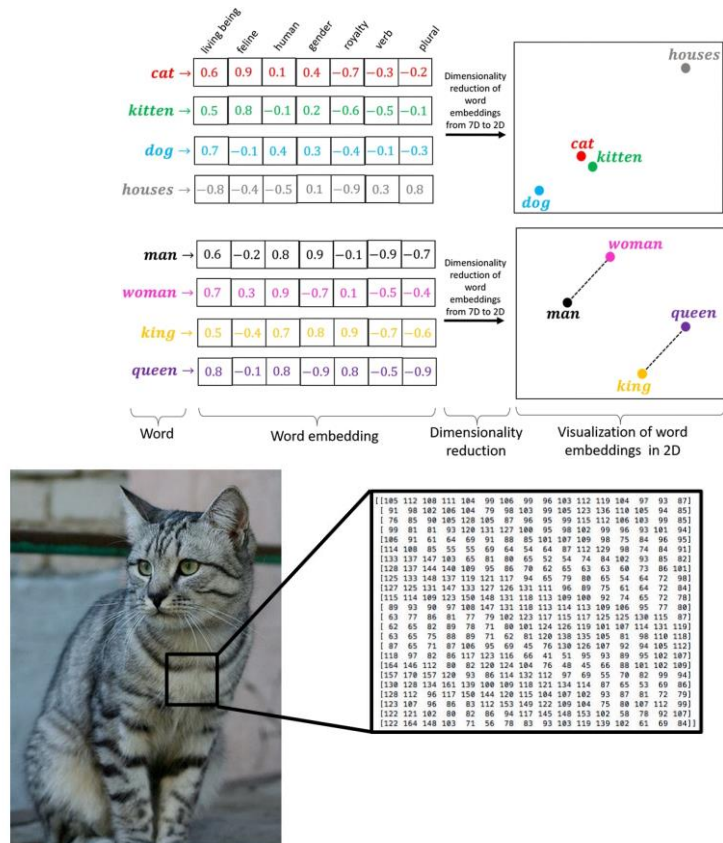
size | age | # room | location | ...



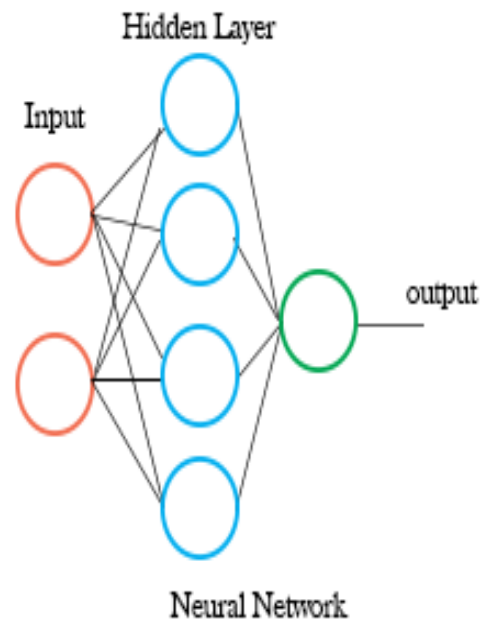
W



Price (Y_n)



Data

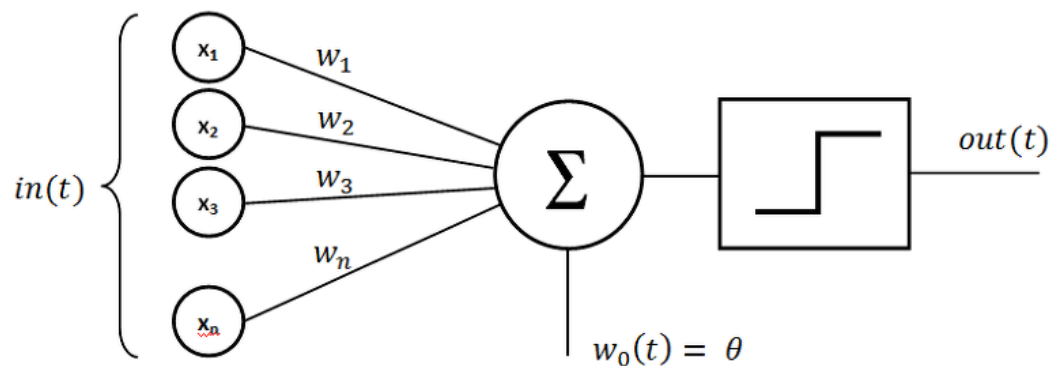


Algorithms



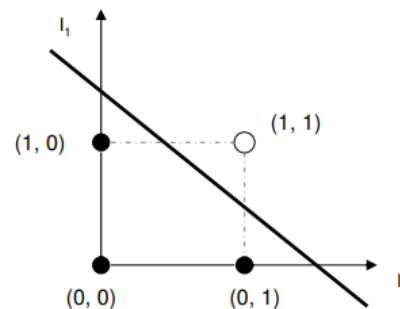
Computation

Perceptron

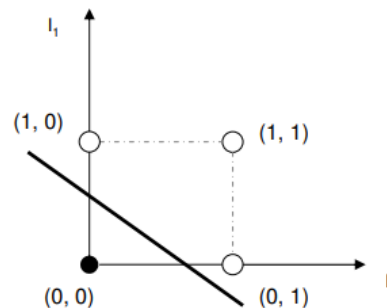


- Inputs : $X = [X_1 \ X_2 \ X_3 \ .. \ X_n]$
- Outputs : Y
- Weights : $W = [W_1 \ W_2 \ W_3 \ .. \ W_n]$
- Bias : W_0 (b)

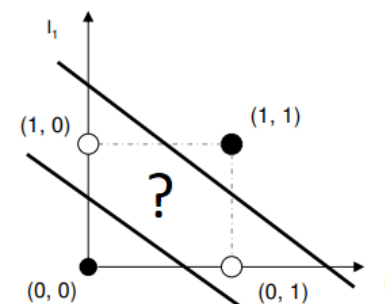
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1

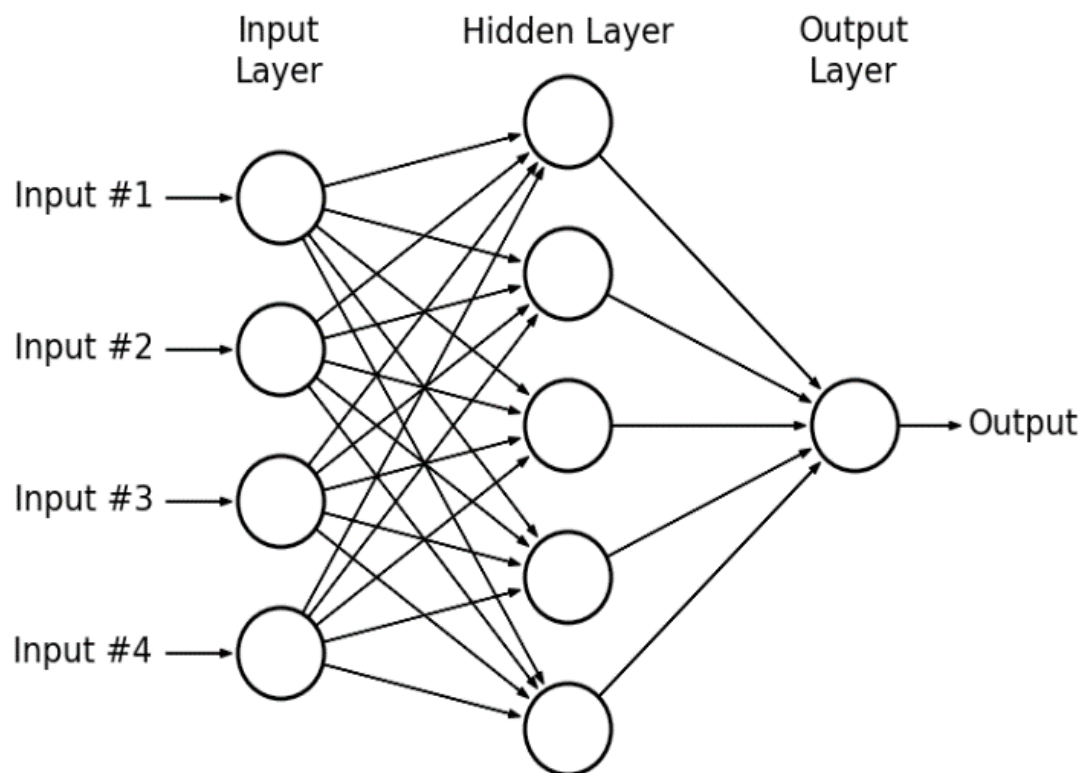


XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



⇒ Perceptron layer 하나만으로는 복잡한 문제를 해결할 수 없다

Mutli-Layer Perceptron



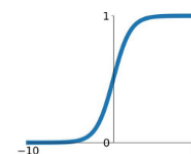
2-layer NN : $f = W_2 \circ g(W_1 X_n)$

<https://paperswithcode.com/methods/category/activation-functions>

Activation Functions

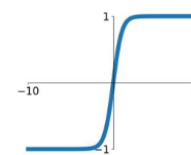
Sigmoid

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



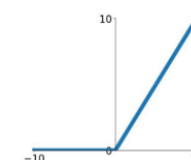
tanh

$$\tanh(x)$$



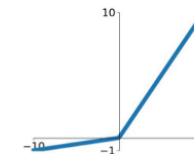
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

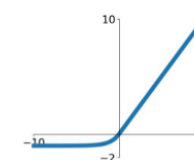


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



"Non-linearity"

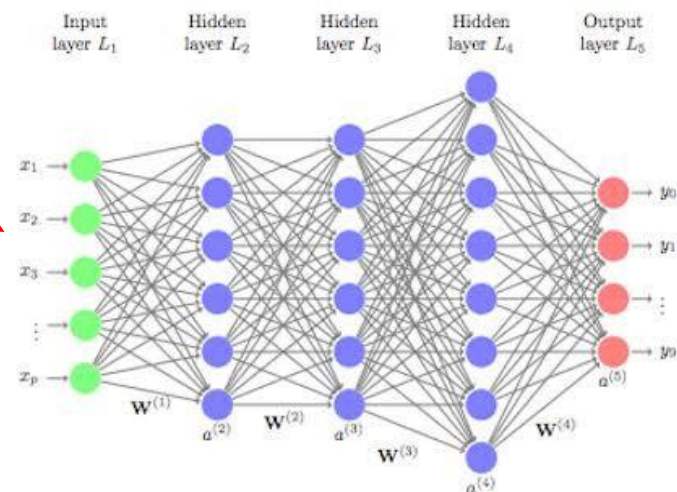
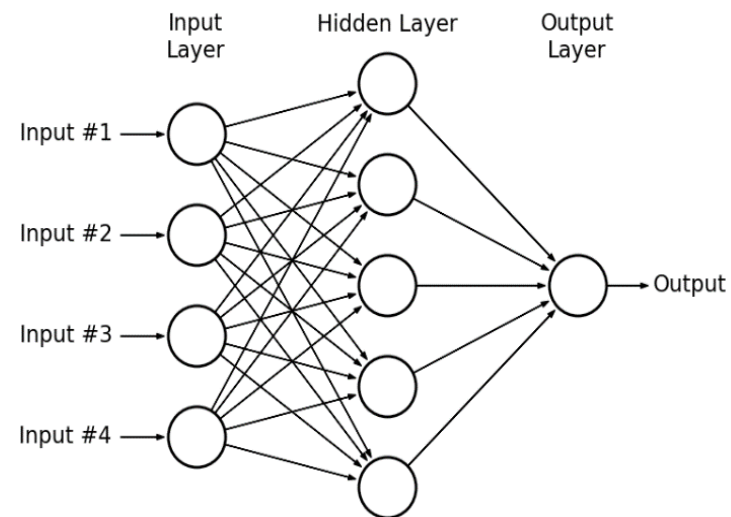
Neural Network
 ~ Fully Connected Networks
 ~ Multi-Layer Perceptron
 ~ Feed Forward Network

Parameter Update

$$\operatorname{argmin}_W \mathbf{L}(w) = \text{loss}(f_W(X_n), y_n)$$

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} L(W)$$

- △ 매우 복잡한 행렬 계산 포함
- △ Loss 값 업데이트 시 재계산량 매우 많음



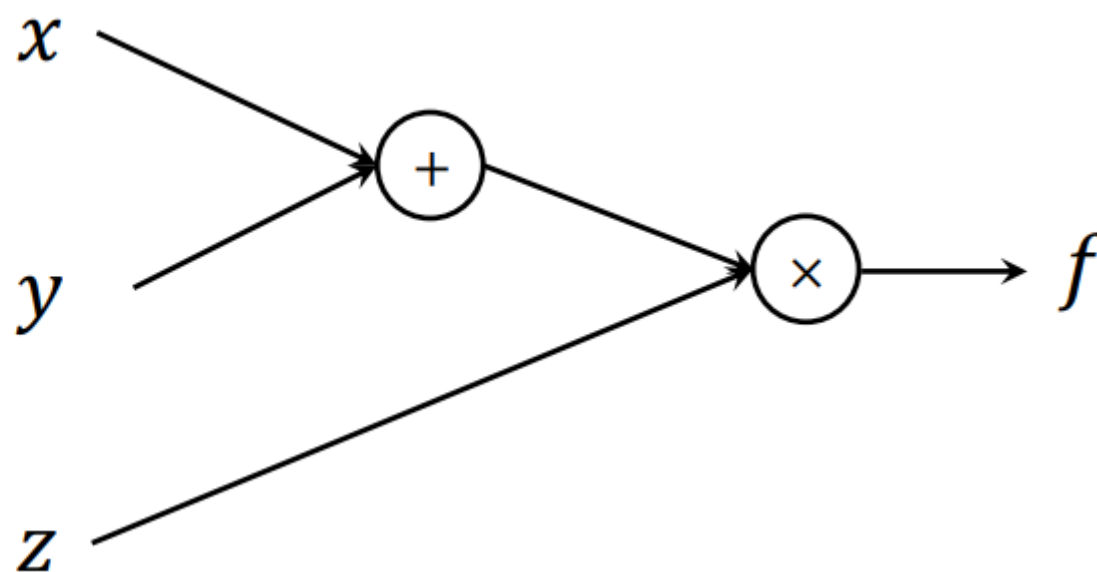
Backpropagation

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

e.g.,

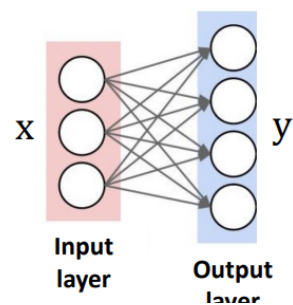
$$x = -2, y = 5, z = -4$$

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

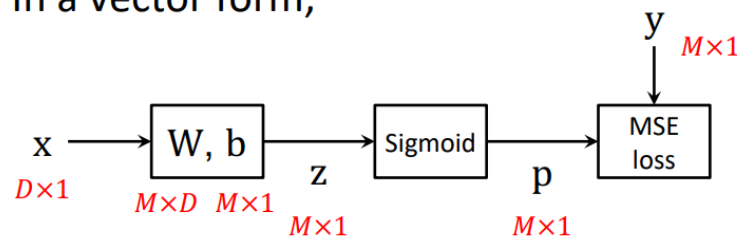


Backpropagation (Try it!)

(Practice) 1-Layer Neural Net with MSE Regression Loss



In a vector form,



$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}, W = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_M^T \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1D} \\ w_{21} & w_{22} & \cdots & w_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{MD} \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}$$

- Linear projection: $z = Wx + b$
- Activation function:

$$p = g(z) = \frac{1}{1 + \exp(-z)}$$

- Loss function: $L = \|y - p\|^2$

$$\frac{\partial L}{\partial p} =$$

$$\frac{\partial L}{\partial z} =$$

$$\frac{\partial L}{\partial W} =$$

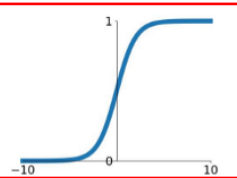
$$\frac{\partial L}{\partial b} =$$

Activation Function

Activation Functions

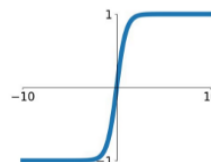
Sigmoid

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



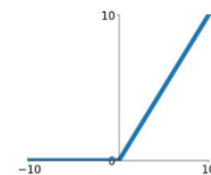
tanh

$$\tanh(x)$$



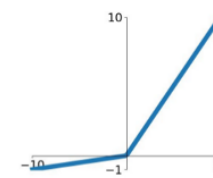
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

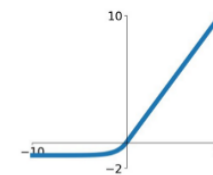


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



△ Gradient Vanishing

-> saturated nerouns kill the gradients

△ Non zero centered

-> always all positive or all negative

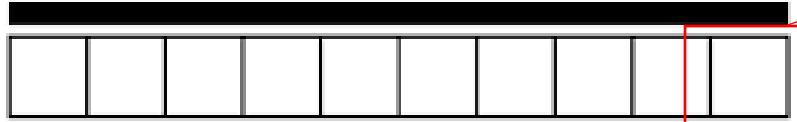
△ exponential -> expensive computation

Batch Normalization

▷ Mini-batch ~ 계산 효율성

Omyo_blog

1Epoch | 1 번의 학습



Batch_size = 100 DataSet = 1000

1Epoch = 10(iteration) x 100(batch_size)
따라서, 총 10번의 기계학습이 실행되었으며,
Iteration 기준 100번의 기계학습이 실행되었다.

<https://m.blog.naver.com/cdi098/222048256037>

- ⇒ easier to train
- ⇒ improves gradient flow
- ⇒ higher learning rate, faster convergence

하나의 배치 내의 평균값과 분산값을 활용하여 정규화 진행

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

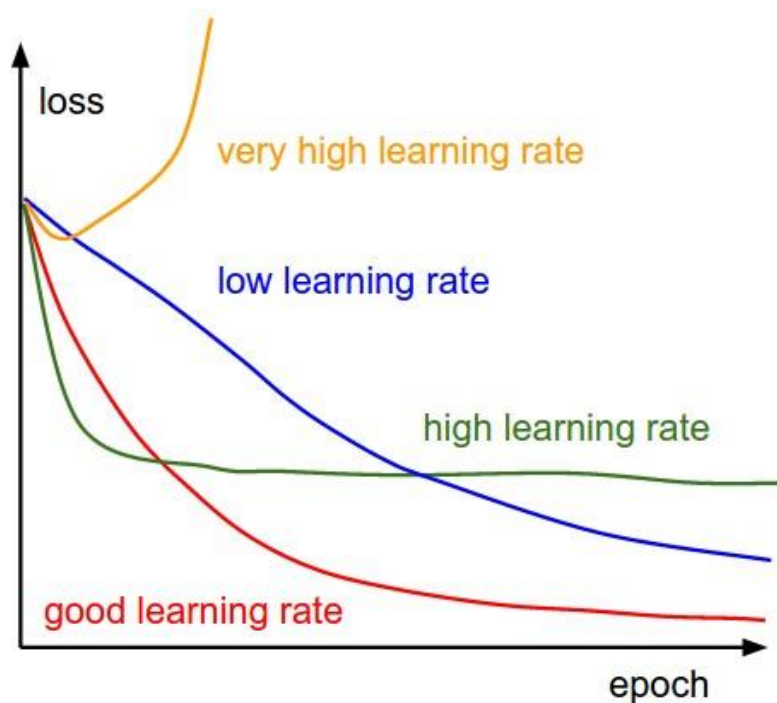
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

*감마와 베타는 추가적인 학습을 통해 얻는 스케일 파라미터

Learning Rate



Linear : $\alpha_+ = \alpha(1 - t/T)$

Inverse Sqrt : $\alpha_+ = \alpha/\sqrt{t}$

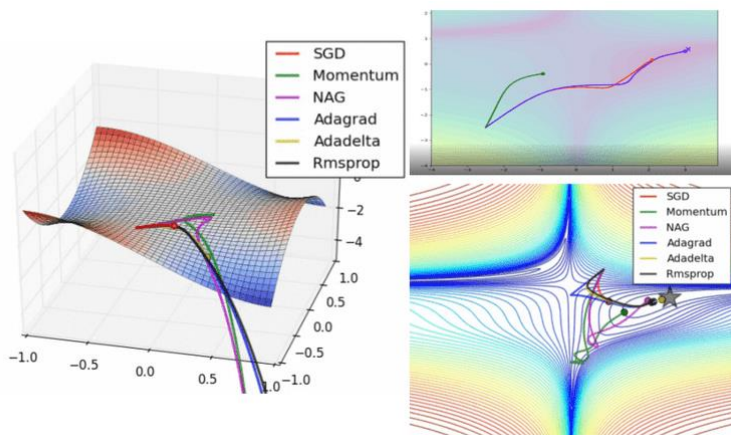
큰 값에서 점차 줄여나가며 찾기

Hyperparameter Optimization

- Single validation rather than multiple folds
- Random search > grid search
- Search on log-scale for learning rate and regularization strength

Optimization

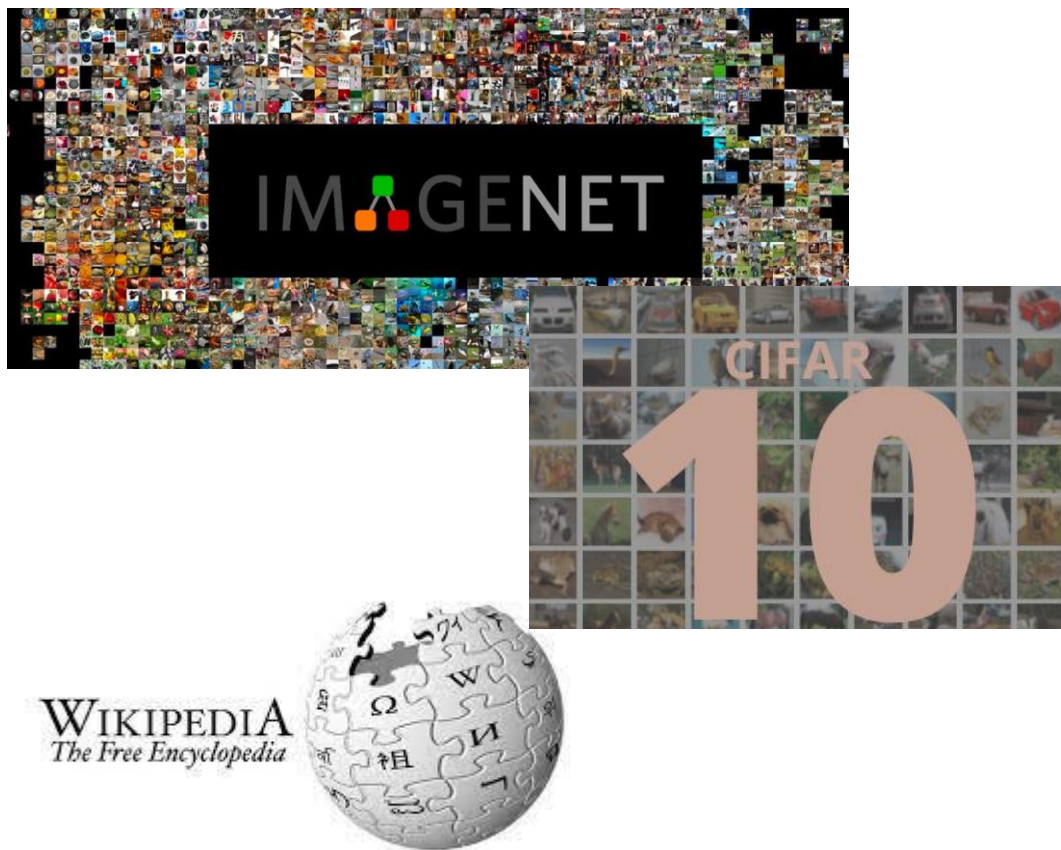
- SGD + Momentum
- Nesterov Momentum
- AdaGrad
- RMSProp
- Adam
- BFGS



Test Error Improvement

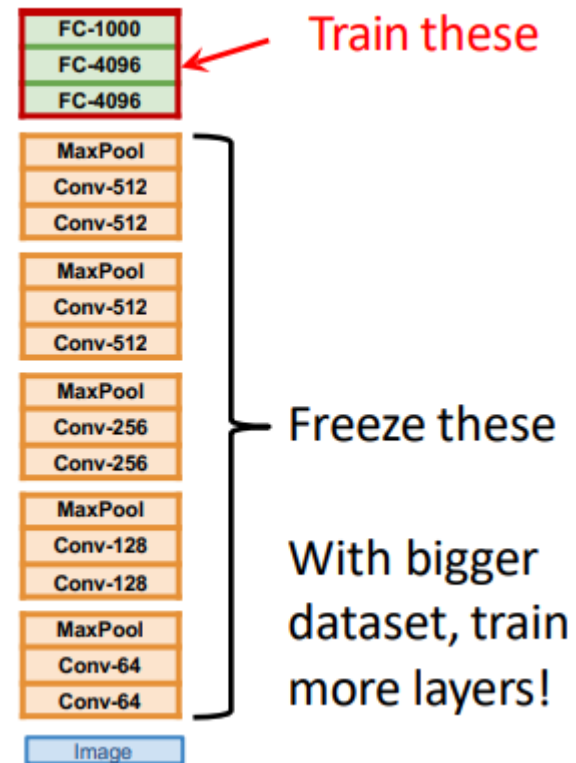
- 1) Early Stopping : 과적합이 발생하기 전, 즉 validation set의 accuracy가 감소하기 직전에 학습 멈춤
- 2) Model Ensemble
- 3) Regularization
 - Dropout : 무작위로 뉴런을 0으로 설정 (prob = 0.5)
 - DropConnect : 무작위로 weight를 제거
 - Fractional Pooling (max pooling ..)
 - Cutout, MixUp : test시에는 온전한 이미지 활용

Transfer Learning



Pre-train

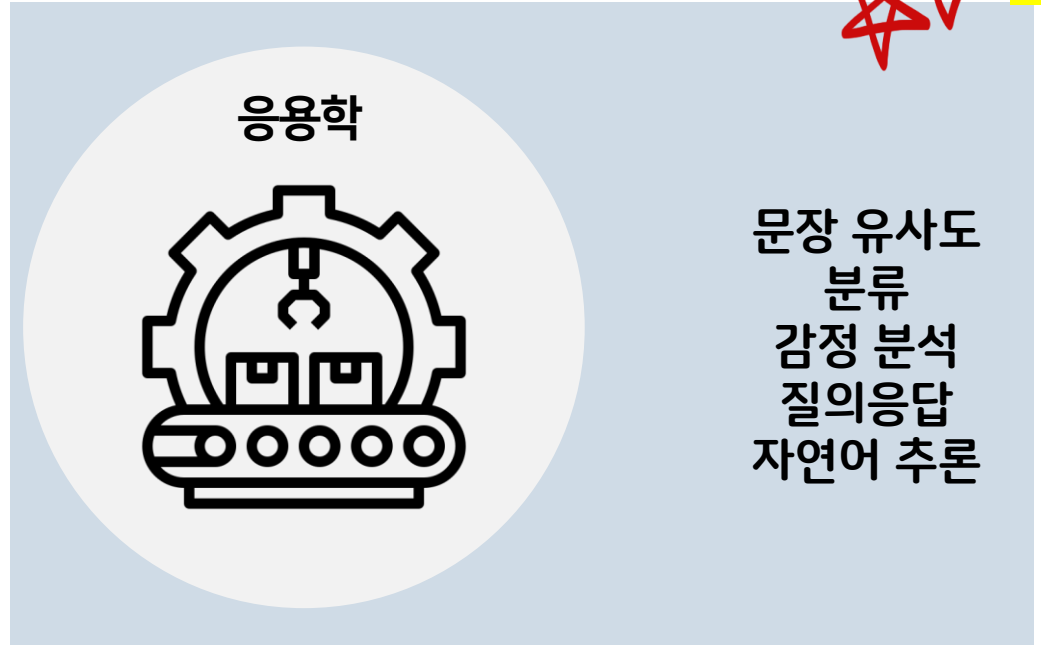
+ 내가 원하는 데이터 =>



Fine-tuning

~~코딩~~

형태소 분석
문법
음성학
형태론



문장 유사도
분류
감정 분석
질의응답
자연어 추론

자연어 처리 Natural Language Processing



**질문은 언제나 늘 환영입니다!
같이 공유하고 고민해봐요 :)**



**이런저런 아이디어가 떠오르면
맘먹고 도전해보세요!**



내가 공부한 건 꼭 기록해보기!

복습과제

○ pytorch tutorial 4가지 task

https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html

○ <밑바닥부터 시작하는 딥러닝2>

Ch1. 신경망 복습 - 따라 구현 해보기 (선택)

다음주 진도

○ <딥러닝을 이용한 자연어처리 입문>

Ch2. 텍스트 전처리

Ch3. 언어 모델

○ <밑바닥부터 시작하는 딥러닝2>

Ch3. word2vec

예습과제

○ IMDB 영화 리뷰 감정 분석



수고하셨습니다!

Contact

15기 김제성

✉ rlawptjd1409@korea.ac.kr

☎ 010-2609-5046