KSBi-BIML 2024

Bioinformatics & Machine Learning(BIML) Workshop for Life and Medical Scientists

생명정보학 & 머신러닝 워크샵 (오프라인)

Introduction to Deep Learning

이상근 _ 고려대학교

KSBI KOREAN SOCIETY FOR BIOINFORMATICS | 한국생명정보학회

본 강의 자료는 한국생명정보학회가 주관하는 BIML 2024 워크샵 오프라인 수업을 목적으로 제작된 것으로 해당 목적 이외의 다른 용도로 사용할 수 없음을 분명하게 알립니다.

이를 다른 사람과 공유하거나 복제, 배포, 전송할 수 없으며 만약 이러한 사항을 위반할 경우 발생하는 **모든 법적 책임은 전적으로 불법 행위자 본인에게 있음을 경고**합니다.

KSBi-BIML 2024

Bioinformatics & Machine Learning(BIML)
Workshop for Life and Medical Scientists

안녕하십니까?

한국생명정보학회가 개최하는 동계 교육 워크샵인 BIML-2024에 여러분을 초대합니다. 생명정보학 분야의 연구자들에게 최신 동향의 데이터 분석기술을 이론과 실습을 겸비해 전달하고자 도입한 전문 교육 프로그램인 BIML 워크샵은 2015년에 시작하여 올해로 벌써 10년 차를 맞이하게 되었습니다. BIML 워크샵은 국내 생명정보학 분야의 최초이자 최고 수준의 교육프로그램으로 크게 인공지능과 생명정보분석 두 개의 분야로 구성되어 있습니다. 올해 인공지능 분야에서는 최근 생명정보 분석에서도 응용이 확대되고 있는 다양한 인공지능 기반 자료모델링 기법들에 대한 현장 강의가 진행될 예정이며, 관련하여 심층학습을 이용한 단백질구조예측, 유전체분석, 신약개발에 대한 이론과 실습 강의가 함께 제공될 예정입니다. 또한 단일세포오믹스, 공간오믹스, 메타오믹스, 그리고 롱리드염기서열 자료 분석에 대한 현장 강의는 많은 연구자의 연구 수월성 확보에 큰 도움을 줄 것으로 기대하고 있습니다.

올해 BIML의 가장 큰 변화는 최근 연구 수요가 급증하고 있는 의료정보자료 분석에 대한 현장 강의를 추가하였다는 것입니다. 특히 의료정보자료 분석을 많이 수행하시는 의과학자 및 의료정보 연구자들께서 본 강좌를 통해 많은 도움을 받으실 수 있기를 기대하고 있습니다. 또한 다양한 생명정보학 분야에 대한 온라인 강좌 프로그램도 점차 증가하고 있는 생명정보 분석기술의 다양화에 발맞추기 위해 작년과 비교해 5강좌 이상을 신규로 추가했습니다. 올해는 무료 강좌 5개를 포함하여 35개 이상의 온라인 강좌가 개설되어 제공되며, 연구 주제에 따른 연관된 강좌 추천 및 강연료 할인 프로그램도 제공되며, 온라인을 통한 Q&A 세션도 마련될 예정입니다. BIML-2024는 국내 주요 연구중심 대학의 전임 교원이자 각 분야 최고 전문가들의 강의로 구성되었기에 해당 분야의 기초부터 최신 연구 동향까지 포함하는 수준 높은 내용의 강의가 될 것이라 확신합니다.

BIML-2024을 준비하기까지 너무나 많은 수고를 해주신 운영위원회의 정성원, 우현구, 백대현, 김태민, 김준일, 김상우, 장혜식, 박종은 교수님과 KOBIC 이병욱 박사님께 커다란 감사를 드립니다. 마지막으로 부족한 시간에도 불구하고 강의 부탁을 흔쾌히 허락하시고 훌륭한 현장 강의와 온라인 강의를 준비하시는데 노고를 아끼지 않으신 모든 강사분들께 깊은 감사를 드립니다.

2024년 2월

한국생명정보학회장 이 인 석

강의 시간표

DAY1 : 2월 24일 (토)

시간	강 의 (자연과학대학 28동 101호)
12:30-12:50	등록
12:50-13:00	공지사항 전달
13:00-14:30	**의료빅데이터/인공지능 총론** 김헌성 교수(가톨릭대학교)
14:30-14:45	휴식
14:45-16:15	**의료영상 인공지능의 이해 및 의료영상 레이블링 실습** 백서연 교수(연석대학교)
16:15-16:30	휴식
16:30-18:00	**의료 정보처리 자동화 실습 / 독자적인 어플리케이션 만들기** 김선근 대표(원닥 주식회사), 서사도 조교

시간	강 의 (자연과학대학 28동 102호)
12:30-12:50	등록
12:50-13:00	공지사항 전달
13:00-14:20	**EMR 데이터를 활용한 머신러닝 기반 예후예측:** **Decision Tree-based Models + EMR 샘플 데이터 실습 (MIMIC sample dataset)** 고태훈 교수(가톨릭대학교)
14:20-14:40	휴식
14:40-16:00	**Chest X-ray 영상을 활용한 딥러닝 기반 폐질환 진단:** **Convolutional Neural Network + 의료영상 샘플 데이터 실습 (NIH Chest X-ray14)** 고태훈 교수(가톨릭대학교)
16:00-16:20	휴식
16:20-17:40	**심전도 데이터를 활용한 딥러닝 기반 부정맥 탐지: Recurrent Neural Network +** **Transformer + 심전도 샘플 데이터 실습 (MIT-BIH Arrhythmia Database)** 고태훈 교수(가톨릭대학교)

DAY1 : 2월 26일 (월)

시간	강 의 (자연과학대학 28동 101호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-10:50	**DNN (이론)** 이상근 교수(고려대학교)
10:50-11:00	휴식
11:00-12:10	**CNN (이론)** 이상근 교수(고려대학교)
12:10-13:40	점심
13:40-15:10	**RNN, ChatGPT, XAI (이론)** 이상근 교수(고려대학교)
15:10-15:20	휴식
15:20-16:50	**CNN/RNN 모델 구조 정의, 학습 알고리즘 적용, 성능 평가, 시각화 방법 (Tensorflow 실습)** 이정현 조교, 한성민 조교

시간	강 의 (자연과학대학 28동 102호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-11:00	**Best practice for single-cell data analysis** 박종은 교수(KAIST)
11:00-11:10	휴식
11:10-12:40	**Practice1: Scanpy basic workflow** 정성민 조교, 고용준 조교
12:40-14:10	점심
14:10-15:30	**Public database, data integration, reference mapping, multiomics** 박종은 교수(KAIST)
15:30-15:40	휴식
15:40-16:50	**Practice2: Advanced single-cell analysis (siVI universe)** 정성민 조교, 고용준 조교

DAY1 : 2월 27일 (화)

시간	강 의 (자연과학대학 28동 101호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-10:50	**AI-based protein structure prediction** **- Intro to protein structure prediction** **- Early AI-based approaches** **- AlphaFold and RoseTTAFold** 백민경 교수(서울대학교)
10:50-11:00	휴식
11:00-12:10	**단백질 구조 예측 실습** **- ColabFold를 활용한 단백질 구조 및 상호작용 예측** **- Tips &Tricks for better structure modeling** 백민경 교수(서울대학교)
12:10-13:40	점심
13:40-15:10	**AI-based protein design** **- Intro to protein design** **- Protein backbone design using RFdiffusion** **- Protein sequence design using ProteinMPNN** 백민경 교수(서울대학교)
15:10-15:20	휴식
15:20-16:50	**단백질 디자인 실습** **- RFdiffusion 및 ProteinMPNN의 활용법 실습** 백민경 교수(서울대학교)

시간	강 의 (자연과학대학 28동 102호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-11:00	**Introduction to Single-cell biology** 최정민 교수(고려대학교)
11:00-11:10	휴식
11:10-12:40	**i. Unsupervised Spatial transcriptome analysis** **ii. Tumor Boundary Determination in Spatial Transcriptomics** 유광민 조교, 이문영 조교
12:40-14:10	점심
14:10-15:30	**i. Deconvolution Analysis Using Single-cell RNA Sequencing and Spatial Transcriptomics** **ii. Cell-Cell Interaction Analysis in Spatial Transcriptomics** 김지현 조교, 최승지 조교
15:30-15:40	휴식
15:40-16:50	**i. Open Chromatin Region Analysis and Biological Interpretation of Using scATAC-seq Dataset** **ii. Construction of Gene Regulatory Networks Based on Integrated Analysis of scATAC-seq and scRNA-seq Datasets** 천하림 조교, 이호진 조교

DAY1 : 2월 28일 (수)

시간	강 의 (자연과학대학 28동 101호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-11:00	**Introduction to Transformers (이론)** 전민지 교수 (고려대학교)
11:00-11:10	휴식
11:10-12:40	**Introduction to Transformers (실습)** 봉현수 조교, 임우택 조교
12:40-14:10	점심
14:10-15:40	**Deep learning in Bioinformatics** 노미나 교수(한양대학교)
15:40-15:50	휴식
15:50-17:20	**Deep learning model을 이용한 실습** 박예솔 조교

시간	강 의 (자연과학대학 28동 102호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-10:50	**마이크로바이옴 기본 이론** 이선재 교수(GIST)
10:50-11:00	휴식
11:00-12:10	**16S rRNA amplicon seq. - DADA2** 조준우 조교, 백재우 조교
12:10-13:40	점심
13:40-14:40	**최신 메타지놈 분석 기법의 현황** 이선재 교수(GIST)
14:40-14:50	휴식
14:50-16:50	**Shotgun metagenome 분석 (Linux)** 조준우 조교, 백재우 조교

DAY1 : 2월 29일 (목)

시간	강 의 (자연과학대학 28동 101호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-10:50	**화학정보학 기초(Cheminformatics) / 약물특성 및 약물다움(druglikeness)** **Molecular Notations &Descriptors / AI 신약개발을 위한 Databases** **AI 신약개발을 위한 Programming 기초** 김동섭 교수(KAIST)
10:50-11:00	휴식
11:00-12:10	**Google Colab에 RDKit 설치 / 화합물 정보 읽기 실습** **Bioactivity database 검색 및 정보 읽기 실습** **Molecular descriptor (fingerprint) 생성 및 similarity 계산 실습** 정수재 조교, 나민주 조교
12:10-13:40	점심
13:40-15:10	**AI 신약개발을 위한 기계학습법 기초 / QSAR 모델링 기초 / AI 신약개발을 위한 딥러닝 모델** **Virtual screening (ligand-based, structure-based) 및 de novo design** 김동섭 교수(KAIST)
15:10-15:20	휴식
15:20-16:50	**QSAR modeling 전체 과정 실습/ 화합물의 Bioactivity 예측 모델 개발** **Virtual screening 과정을 통한 신약후보물질 발굴 실습** 정수재 조교, 나민주 조교

시간	강 의 (자연과학대학 28동 102호)
09:00-09:20	등록
09:20-09:30	공지사항 전달
09:30-11:00	**Single cell multiomics 이론 / Gene regulatory network 이론** 김준일 교수(숭실대학교)
11:00-11:10	휴식
11:10-12:40	**Seurat/Signac, ArchR, TENET+ 실습** 김현규 조교, 정회빈 조교
12:40-14:10	점심
14:10-15:40	**롱리드 시퀀싱 소개 및 유전체 조립 실습** 김준 교수(충남대학교)
15:40-15:50	휴식
15:50-17:20	**변이 분석 및 시각화 실습** 김준 교수(충남대학교)

강의개요

Introduction to Deep Learning

딥러닝은 영상 처리, 시계열 예측 등 다양한 분야에서 다량의 데이터를 기반으로 분류 등 문제를 해결하기 위한 기계학습 기법입니다. 본 과정에서는 기계학습과 딥러닝에 대한 개념적 이해를 바탕으로, 최근 많이 활용되고 있는 CNN (Convolutional Neural Network), RNN (Recurrent Neural Network)의 구조와 활용 방법에 대해 소개합니다. 또한 최근 각광받고 있는 생성형 AI인 ChatGPT 기술과 설명 가능한 인공지능 (eXplainable AI, XAI) 기술에 대해 간단히 소개하려 합니다. 본 과정은 각 기법의 직관적이면서도 수학적 이해를 통해 수강생이 각 기법의 동작 원리와 장단점에 대해 파악할 수 있도록 하는 것을 목표로 합니다. 또한 구글의 딥러닝 소프트웨어인 Tensorflow를 이용한 실습을 통해 딥러닝 기법 적용을 위한 기초 소양을 다지고자 합니다.

강의는 다음의 내용을 포함한다:

- 기계학습 및 딥러닝의 기초
- DNN (Deep Neural Network), CNN (Convolutional Neural Network), RNN (Recurrent Neural Network) 이해
- ChatGPT 기술의 개요, XAI 기법 소개

* 참고강의교재: Deep learning, Goodfellow, Bengio & Courville, MIT Press, 2016

* 교육생준비물:
  노트북 (메모리 8GB 이상, 디스크 여유공간 30GB 이상), 구글 크롬 웹 브라우저
  실습 시 구글 Colaboratory 사용 예정 (설치 필요 없음, 구글 개인 계정 생성 필수)
  https://colab.research.google.com/notebooks/welcome.ipynb

* 강의 난이도: 초급~중급

* 강의: 이상근 교수 (고려대학교 정보보호대학원)

Curriculum Vitae

Speaker Name: Sangkyun Lee, Ph.D.

▶ **Personal Info**

Name	Sangkyun Lee
Title	Associate professor
Affiliation	Korea University

▶ **Contact Information**

Address	145, Anam-ro, Seongbuk-gu, Seoul, 02841, Korea
Email	sangkyun@korea.ac.kr
Phone Number	02-3290-4890

Research Interest

Secure AI, AI model compression, XAI, AI for Security

Educational Experience

2003	B.S., Seoul National University
2005	M.S., Seoul National University
2011	Ph.D., University of Wisconsin-Madison, USA

Professional Experience

2011-2014	Post-doc Researcher, SFB 876, TU Dortmund University, Germany
2015-2017	Principal Investigator, SFB 876, TU Dortmund University, Germany
2017-2019	Assistant Professor, Department of Computer Science, Hanyang University ERICA
2020-2021	Assistant Professor, School of Cybersecurity, Korea University
2022-current	Associate Professor, School of Cybersecurity, Korea University

Selected Publications (5 maximum)

1. Anomaly Candidate Extraction and Detection for Automatic Quality Inspection of Metal Casting Products using High-Resolution Images, Byeonggil Jung, Heegon You, Sangkyun Lee, J. Manuf. Syst., 2023
2. Libra-CAM: An Activation-Based Attribution Based on the Linear Approximation of Deep Neural Nets and Threshold Calibration, Sangkyun Lee, Sungmin Han, IJCAI, 2022
3. Model Stealing Defense against Exploiting Information Leak Through the Interpretation of Deep Neural Nets, Jeonghyun Lee, Sungmin Han, Sangkyun Lee, IJCAI, 2022
4. Hunt for Unseen Intrusion: Multi-Head Self-Attention Neural Detector, Seongyun Seo, Sungmin Han, Janghyeon Park, Shinwoo Shim, Han-Eul Ryu, Byoungmo Cho, and Sangkyun Lee, IEEE Access, 2021

KSBI | 한국생명정보학회

고려대학교 KOREA UNIVERSITY

AIR LAB

Introduction to Deep Learning

Introduction to ML & DNN

고려대학교 정보보호대학원 인공지능연구실
이상근

BIML 2024

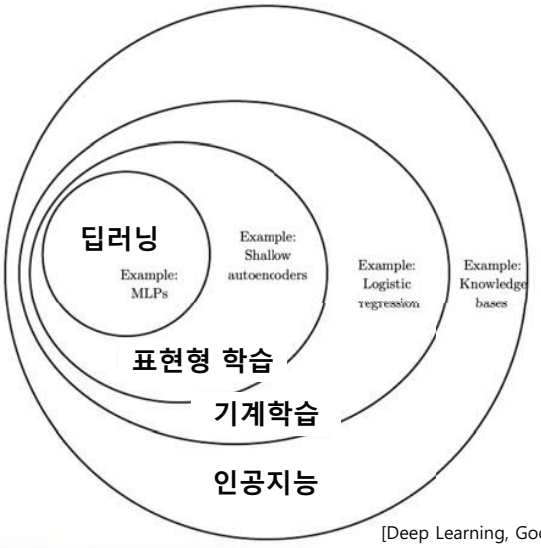
Machine Learning

- Arthur Lee Samuel (1901~1990, 1959)
  - A pioneer in AI
  - AI: a field of study that gives computers the ability to learn without being explicitly programmed

- Vladimir Vapnik (1936~)
  - The father of ML
  - Statistical Learning Theory (Wiley, 1998)

기계학습

AI: 학습이나 문제해결 등, 인간의 인지와 관련된 기능을 모사하는 SW/HW



[Deep Learning, Goodfellow et al., 2016]

인공지능 AI
: 전문가 시스템, Cybernetics

기계학습 Machine Learning
: SVM, 로지스틱 회귀, decision trees, …

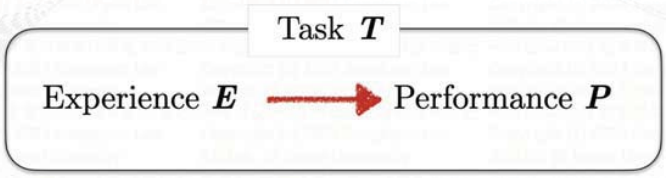
표현형 학습 Representation Learning

딥러닝 Deep Learning
: 자연어 처리, computer vision, …

Machine Learning

- Tom Mitchell (1997)

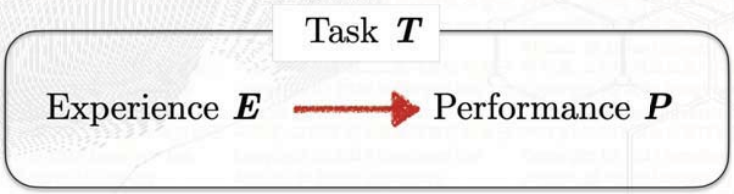
  "A computer program is said to learn from experience *E* w.r.t. some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."



Side: AGI (Artificial General Intelligence)

- No limitation on the task T

Machine Learning



Task T	Experience E	Performance P
• Classification	• Supervised Learning	• Training Error
• Regression	• Unsupervised Learning	• Test Error
• Machine translation	• Semi-Supervised	• Generalization Error
• Outlier detection	• Self-Supervised	.
• Synthesis	• Reinforcement Learning	.
.	.	.
.	.	
.	.	

Experience = Data

- Supervised Learning: X (input), Y (output)
- Unsupervised Learning: X (input), no Y
- Semi-supervised Learning: (X1,Y1) and X2
- Self-supervised Learning: X → (X', Y')

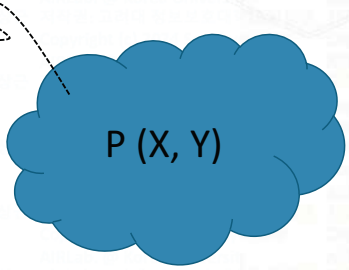
In Statistical Learning…

Sampling (i.i.d)

i : independent
i.d.: identical distribution

Data Set $\{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$

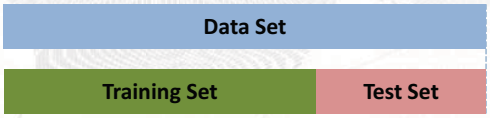
A set consists of m samples

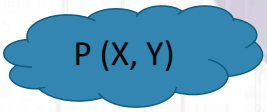


Probability distribution
(usually, unknown)

Note:
- There can be many possible data sets
- We're dealing with just one set of samples

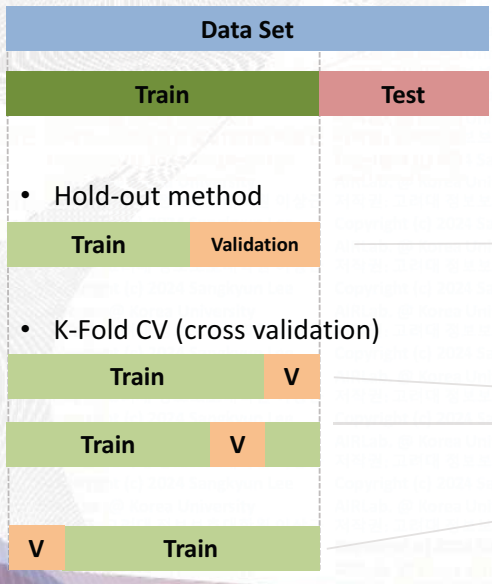
Performance



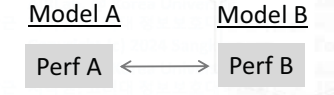


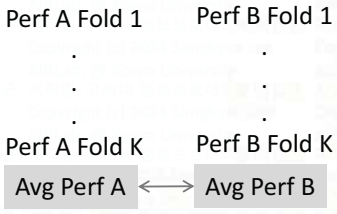
- **Training Error (Rate)** : error on the training set $\frac{1}{|\text{tr}|}\sum_{i\in\text{tr}}\mathbf{1}[y^{(i)} \neq f_w(x^{(i)})]$
- **Test Error (Rate)** : error on the test set $\frac{1}{|\text{tt}|}\sum_{i\in\text{tt}}\mathbf{1}[y^{(i)} \neq f_w(x^{(i)})]$
- **Generalization Error**: error on the **all possible** data $\mathbb{E}_{(X,Y)}\left[\mathbf{1}[Y \neq f_w(X)]\right]$

Model Selection

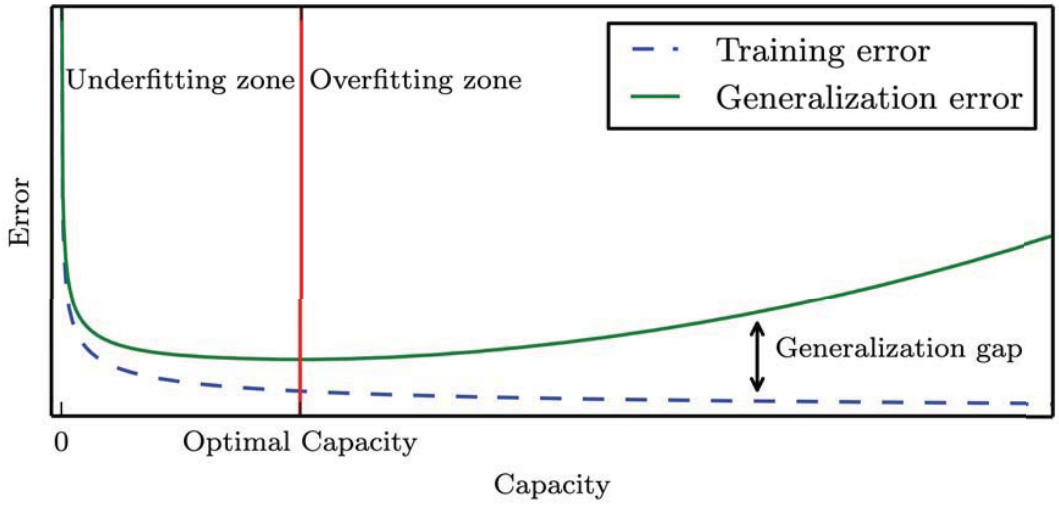


Model selection is a part of hyper-parameter tuning





Generalization & Capacity



Statistical Learning

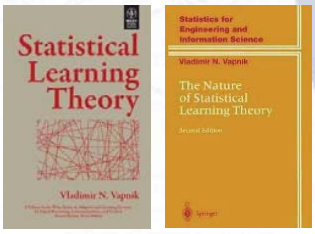
A dataset consists of
n samples (x, y)

(Input, Label) Space

Unknown Probability Distribution: D(X,Y)

Predictor (Hypothesis) $h : X \to Y$ $h \in \mathcal{H}$ Hypothesis space

Loss $\ell_h : X \times Y \to \mathbb{R}$ e.g. set of linear functions, etc.



Risk and Empirical Risk

A dataset consists of n samples (x, y)

Predictor (Hypothesis) $h : X \to Y, h \in \mathcal{H}$

Loss $\ell_h : X \times Y \to \mathbb{R}$

Risk $$r(h) := \mathbb{E}_{(X,Y)\sim D}[\ell_h(X, Y)]$$

Empirical Risk $$\hat{r}(h) := \frac{1}{n}\sum_{i=1}^{n} \ell_h(x_i, y_i)$$

PAC (Probably Approximately Correct) Learning

PAC Bound : $$\mathbb{P}(|r(h^*) - r(\hat{h})| \leq \epsilon) \geq 1 - \delta$$

i) Finite hypothesis space: $|\mathcal{H}| = k$

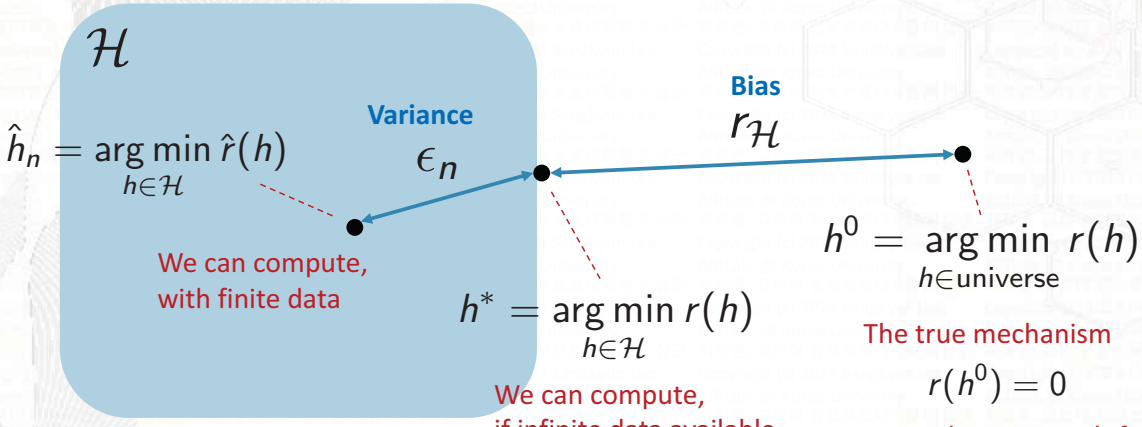
$$r(\hat{h}) \leq \underbrace{\left(\min_{h\in\mathcal{H}} r(h)\right)}_{\text{Bias}(\mathcal{H})} + \underbrace{2\sqrt{\frac{1}{2n}\log\frac{2k}{\delta}}}_{\text{Variance }(\mathcal{H})}$$

Bias-Variance Tradeoff

ii) Infinite hypothesis space: $\text{VC}(\mathcal{H}) = d$ Vapnik-Chervonenkis Dimension

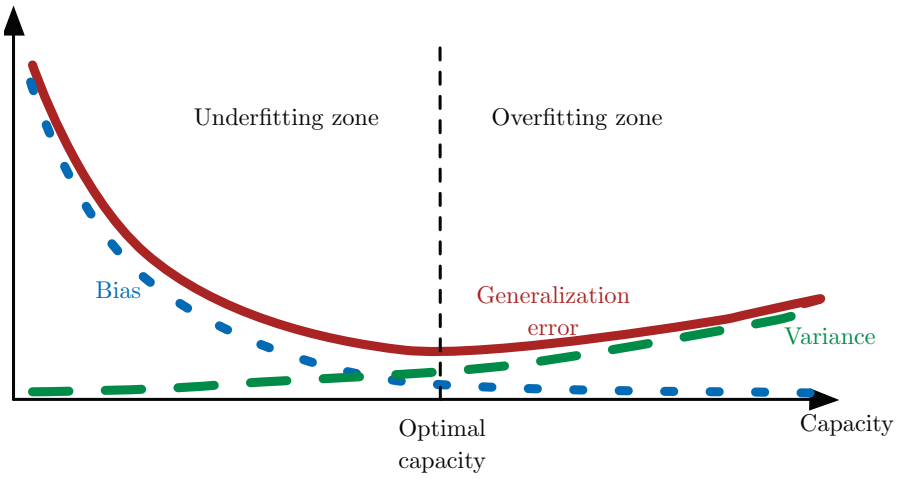
$$r(\hat{h}) \leq \left(\min_{h\in\mathcal{H}} r(h)\right) + \mathcal{O}\left(\sqrt{\frac{d}{n}\log\frac{n}{d} + \frac{1}{n}\log\frac{1}{\delta}}\right)$$

PAC (Probably Approximately Correct) Learning

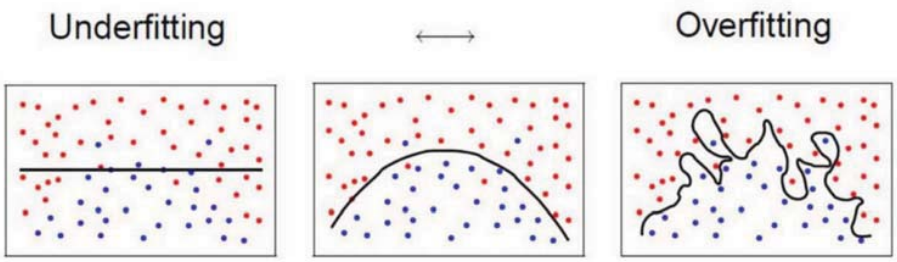


Mere humans can't fathom
- Infinite data required
- Infinite computer required

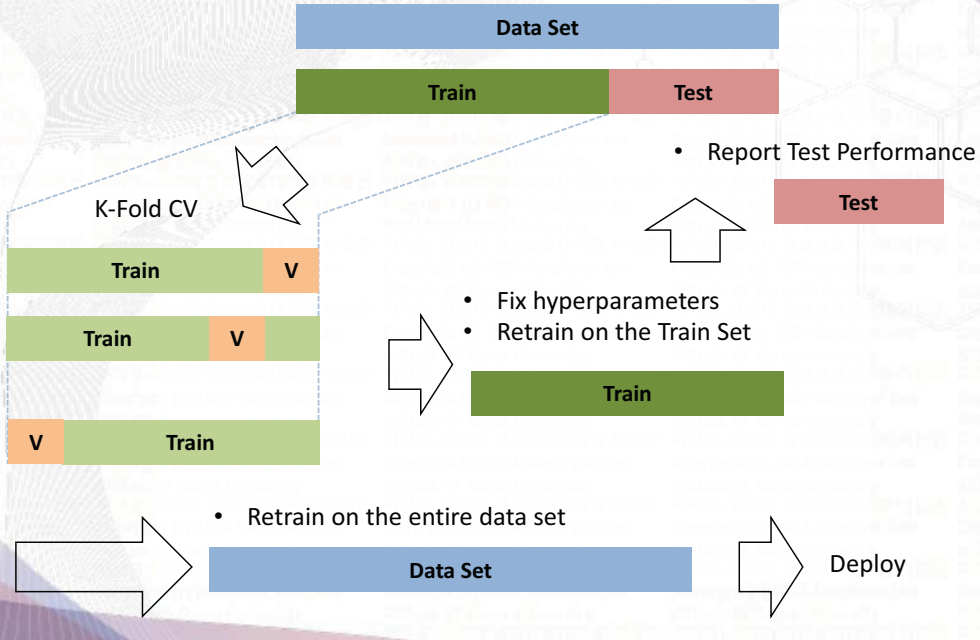
Bias-Variance Tradeoff



Overfitting Issue



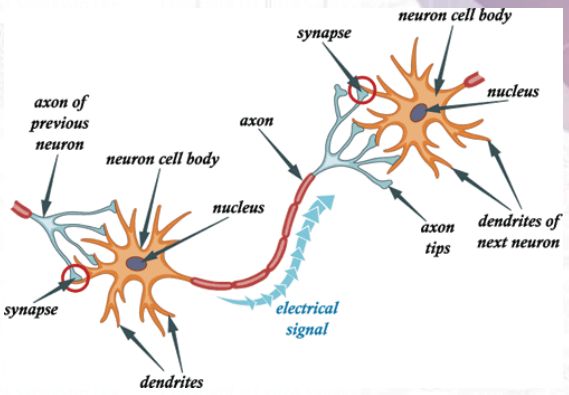
ML Development Cycle



Neuron

Camillo Golgi

Santiago Ramón y Cajal



Nobel prize 1906
Structure of nerve system

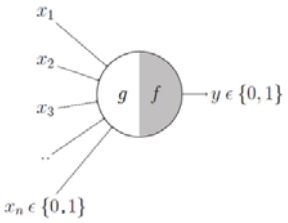
In Human Brain…

- Neurons: 약 100억
- 약 7000 synapse per neuron
  (Total 100조 이상)

Neural Net

Neuron [McCulloch & Pitts, 1943]

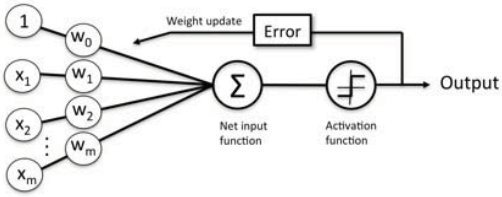
- The first computational model of a neuron



Walter Pitts, 1954 MIT

Perceptron [Rosenblatt, 1957]

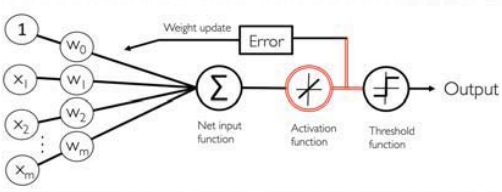
- The first neural net



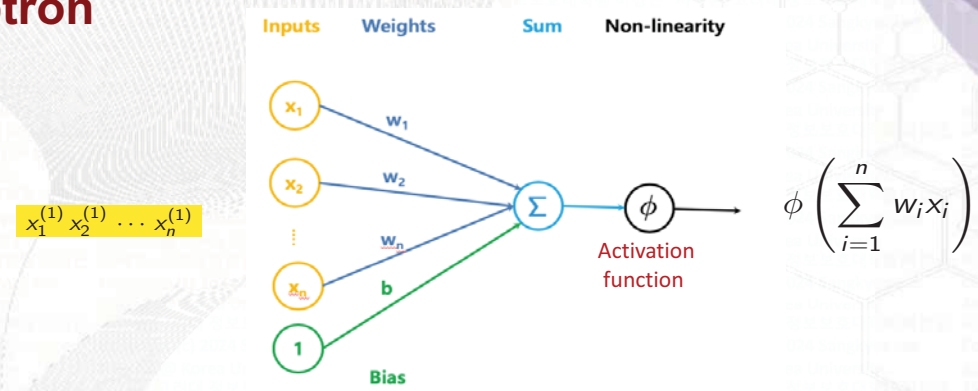
Frank Rosenblatt

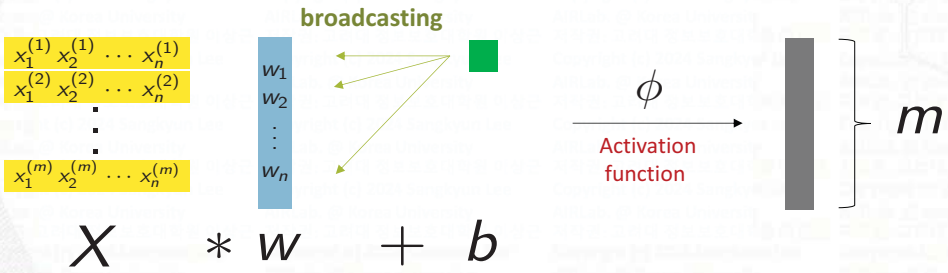
ADALINE [Widrow & Hoff, 1960]

- Adaptive Linear Element



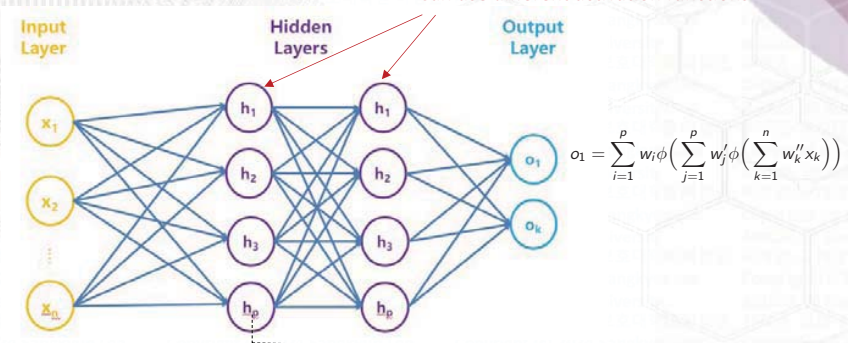
Perceptron

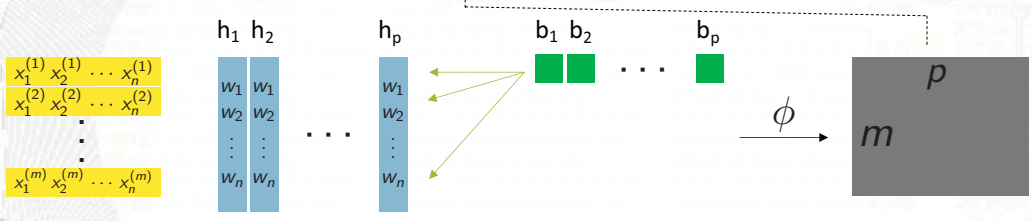




$$X * w + b$$

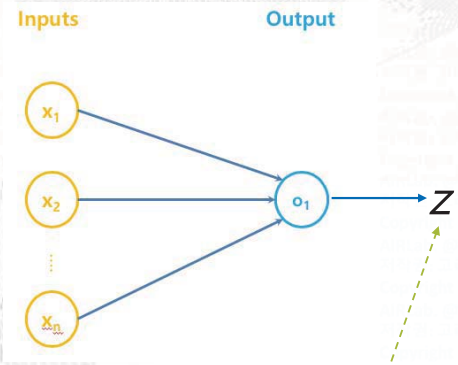
MLP (Multi-Layer Perceptron)
DNN (Deep Neural Network)





$$X * W + b$$

Output Transformation



Output without an activation
Called "logit"

Regression:

- $y \in \mathbb{R}$
- Use z as it is

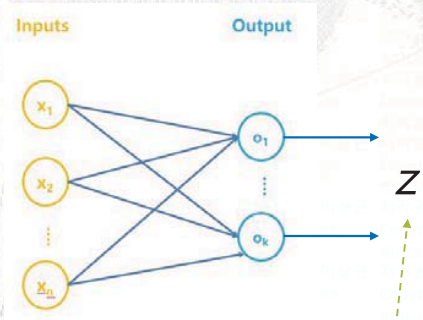
Classification:

- $y \in \{0, 1\}$
- Use

$$\mathbb{P}(Y = 1) = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

Sigmoid function

Output Transformation



Output without an activation
Called "logit"

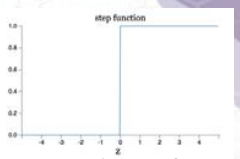
Classification:

- $y \in \{1, 2, \ldots, k\}$
- Use

$$\mathbb{P}(Y = i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)}$$

Softmax(z)$_i$

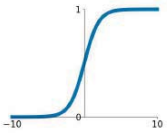
Activation Functions



Alternatives to the step function

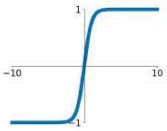
Sigmoid

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



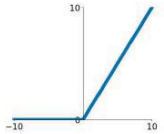
tanh

$\tanh(x)$



ReLU

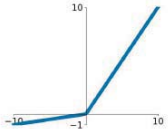
$\max(0, x)$



(Rectifier Linear Unit)

Leaky ReLU

$\max(0.1x, x)$

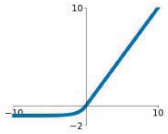


Maxout

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

ELU

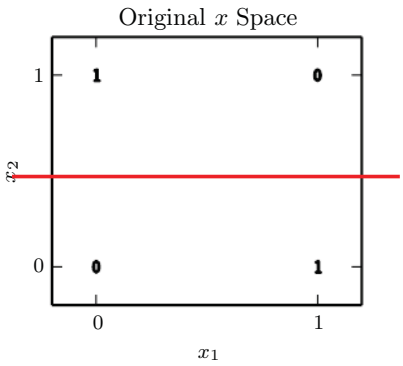
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

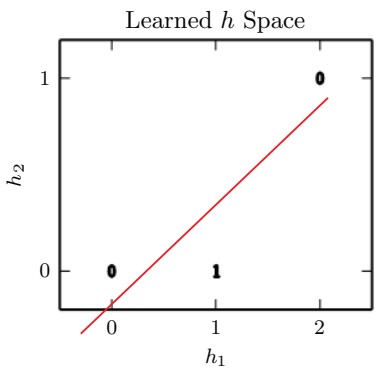


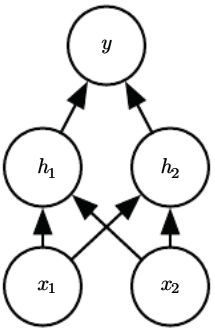
(Exponential Linear Unit) $\alpha > 0$

Benefit of Depth

- Solving XOR problem







Benefit of Depth

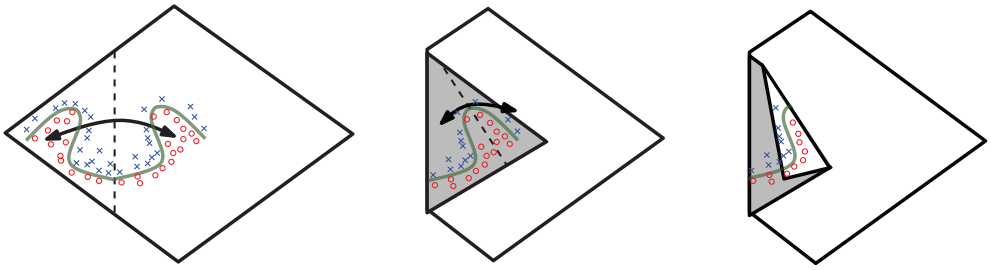
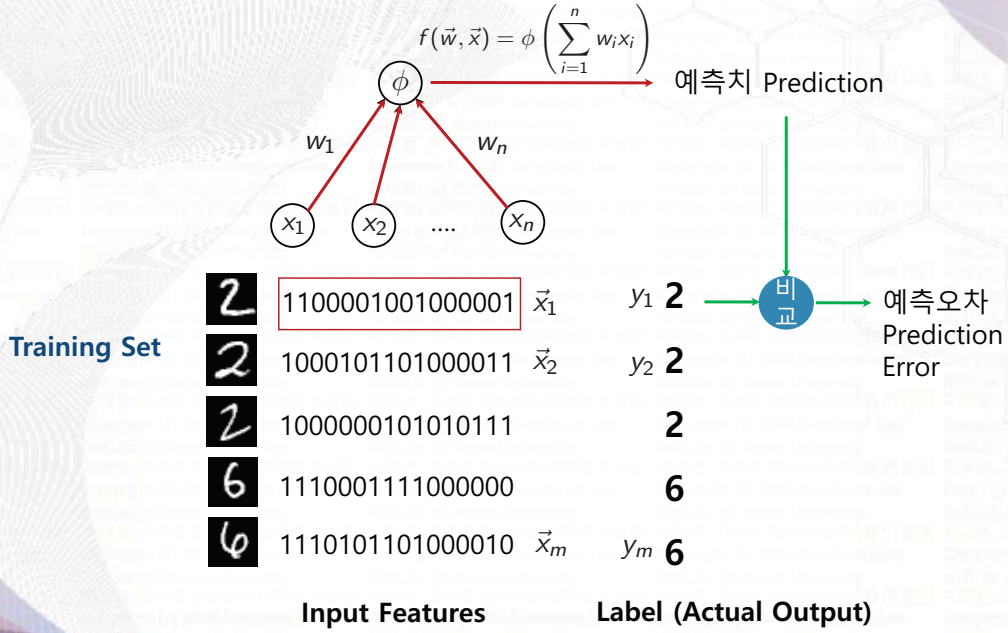


Figure 6.5: An intuitive, geometric explanation of the exponential advantage of deeper rectifier networks formally shown by Pascanu *et al.* (2014a) and by Montufar *et al.* (2014). *(Left)* An absolute value rectification unit has the same output for every pair of mirror points in its input. The mirror axis of symmetry is given by the hyperplane defined by the weights and bias of the unit. A function computed on top of that unit (the green decision surface) will be a mirror image of a simpler pattern across that axis of symmetry. *(Center)* The function can be obtained by folding the space around the axis of symmetry. *(Right)* Another repeating pattern can be folded on top of the first (by another downstream unit) to obtain another symmetry (which is now repeated four times, with two hidden layers).

Training



Training = Numerical Optimization

- Training (학습): 주어진 training 데이터에서 예측오차를 최소화하는 최적 기계학습 파라미터 $\vec{w}^* = (w_1^*, \ldots, w_n^*)$의 값을 찾는 문제

m: Training 데이터 포인트 수

$$\vec{w}^* \in \arg\min_{\vec{w} \in \mathbb{R}^n} J(\vec{w}) = \frac{1}{m} \sum_{i=1}^{m} \ell(y_i, f(\vec{w}, \vec{x}_i))$$

n: 학습 모델의 파라미터 수 (최적화 문제의 차원)

Loss function

- n 또는 m 이 큰 경우 (Big Data), 또는 loss function이 다루기 어려운 경우 (e.g. 미분불가), 효율적인 **수치 최적화** 알고리듬이 필요함

Training Problem

$$\min_{w \in \mathbb{R}^p} \sum_{i=1}^{n} \ell(w; x_i, y_i)$$

Task	Labels	Loss function
Regression	$y_i \in \mathbb{R}$	$\ell(w; x_i, y_i) = (y_i - f(w, x_i))^2$
Classification (Binary)	$y_i \in \{0, 1\}$	$-\ell(w; x_i, y_i) = y_i \log f(w, x_i) + (1 - y_i) \log(1 - f(w, x_i))$
Classification (Multi-Class)	$y_i \in \{1, \ldots, K\}$	$-\ell(w; x_i, y_i) = \sum_{k=1}^{K} I_{y_i=k} \log \text{softmax}(f(w, x_i))_k$

Likelihood Function

Likelihood

$$\mathbb{P}(o_1, o_2, \ldots, o_n; \theta)$$

- **Joint probability observations under the model**
- **Probability that the model has generated the observations**
- **A function in θ**

MLE (Maximum Likelihood Estimation)

- A coin toss problem in elementary school:
  - Assume a fair dice: *P(X = i) = 1/6, i = 1,2,…,6*
  - Toss the dice 10 times, where each toss is independent
  - What is the probability of the event, $\{3, 6, 1, 4, 2, 2, 4, 5, 6, 3\}$?

- Given the observations: $\{3, 6, 1, 4, 2, 2, 4, 5, 6, 3\}$
  - What will be a good guess of *P(X = i)* ?

MLE (Maximum Likelihood Estimation)

- Given observations: $\{o_1, o_2, \ldots, o_n\}$ $\quad o_i \overset{i.i.d.}{\sim} \mathbb{P}(O;\theta)$
- Likelihood function $\quad L(\theta) = \mathbb{P}(o_1, o_2, \ldots, o_n;\theta)$
  - Joint probability of observations under the model (parameter: θ)
  - Probability that the model has generated the observations
  - A function in θ
- MLE: find the θ that maximizes the likelihood

$$\max_{\theta} \; L(\theta) \qquad \min_{\theta} \; -LL(\theta) = -\log L(\theta)$$

Negative log likelihood (NLL)

  - MLE is *efficient*: given n examples, MLE is the most accurate procedure to estimate the parameters

MLE for Binary Classification

Conditional Bernoulli model of labels:

$$P(Y=1|X=x;w) = \sigma(f(w,x_i)) = \frac{1}{1+\exp(-f(w,x_i))}$$
$$P(Y=0|X=x;w) = 1-\sigma(f(w,x_i))$$

(Conditional) Log likelihood function:

i.i.d

$$\log P(y_1,\ldots,y_n|x_1,\ldots,x_n;w) = \log \prod_{i=1}^{n} P(y_i|x_i;w)$$
$$= \log \prod_{i=1}^{n} P(y_i=1|x_i;w)^{y_i} P(y_i=0|x_i;w)^{1-y_i}$$
$$= \sum_{i=1}^{n} \{y_i \log \sigma(f(w,x_i)) + (1-y_i)\log(1-\sigma(f(w,x_i)))\}$$

MLE for Multi-Class Classification

Softmax function: $$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}, \quad i = 1, 2, \ldots, k$$

$$P(Y = k|x) = \text{softmax}(f(w, x))_k$$

Log likelihood function:

$$= \log \prod_{i=1}^{n} P(y_i = 1|x_i; w)^{I_{y_i=1}} \ldots P(y_i = K|x_i; w)^{I_{y_i=K}}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} I_{y_i=k} \log \text{softmax}(f(w, x_i))_k$$

Training Problem (Classification)

$$\max_{\theta} \log \mathbb{P}(o_1, o_2, \ldots, o_n; \theta)$$

Minimization of Negative Log Likelihood function:

$$-\min_{\theta} -\log \mathbb{P}(o_1, o_2, \ldots, o_n; \theta)$$

Stochastic Gradient Descent (SGD)

$$\vec{w}^* \in \arg\min_{\vec{w}\in\mathbb{R}^n} J(\vec{w}) = \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, f(\vec{w}, \vec{x}_i))$$

- Initialize w randomly
- For N epochs
  - For a random training example $J_i(w) = \ell(y_i, f(\vec{w}, \vec{x}_i))$
  - Compute stochastic (sub)gradient of loss: $\frac{\partial J_i(w)}{\partial w}$
  - Update w: $w = w - \eta \frac{\partial J_i(w)}{\partial w}$

Learning rate

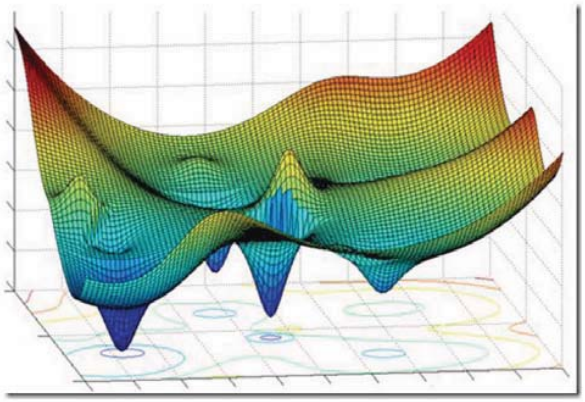
Mini-Batches

Use a small subset of examples per update, to reduce variance of gradient estimates

- Initialize w randomly
- For N epochs
  - For minibatch samples $J_i(w) = \ell(y_i, f(\vec{w}, \vec{x}_i))$
  - Compute stochastic (sub)gradient of loss: $\frac{\partial J(w)}{\partial w} \approx \frac{1}{B}\sum_{i}^{B} \frac{\partial J_i(w)}{\partial w}$
  - Update w: $w = w - \eta \frac{\partial J(w)}{\partial w}$

DNN Objective Functions

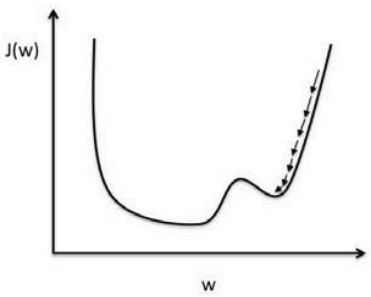
Objective functions of DNN are typically nonconvex, with lots of local minimizers and saddle points



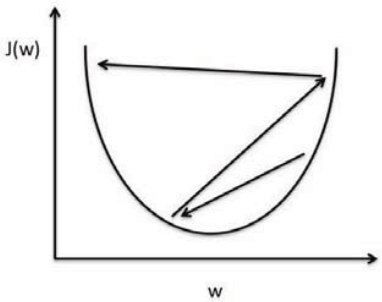
Learning Rate is Important

$$w = w - \eta \frac{\partial J(w)}{\partial w}$$

How to choose the learning rate?



Small learning rate: Many iterations until convergence and trapping in local minima.



Large learning rate: Overshooting.

Adaptive Learning Rate

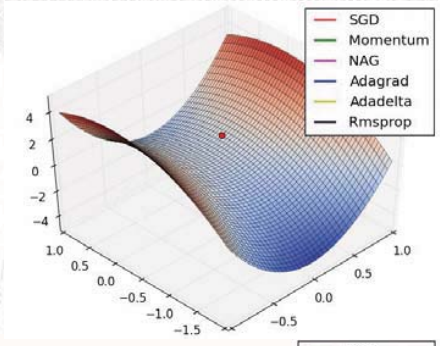
Learning rates can be chosen adaptively to:

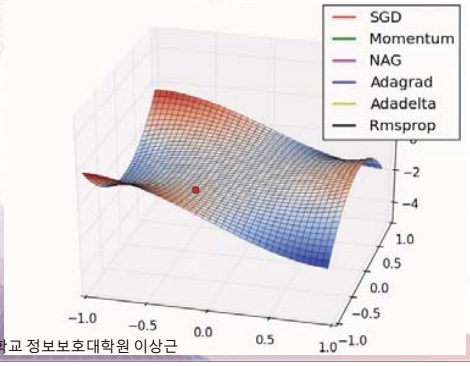
- How large the gradient is
- How fast learning is happening
- Magnitude of particular weights
- ….

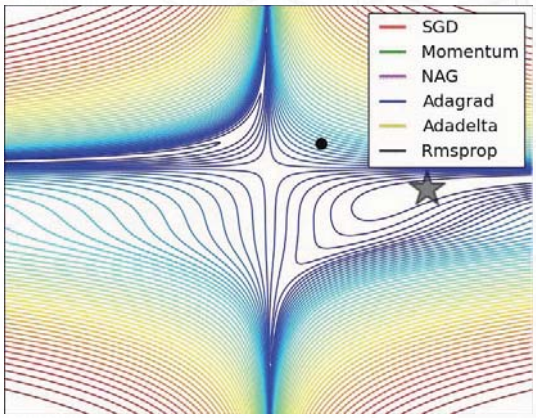
Adaptive learning rate algorithms:

ADAM, Momentum, NAG, Adagrad, Adadelta, RMSProp, …

Adaptive Learning Rate Algorithms







Regularization to Avoid Overfitting

Dropout: in training, randomly set some activations to zero

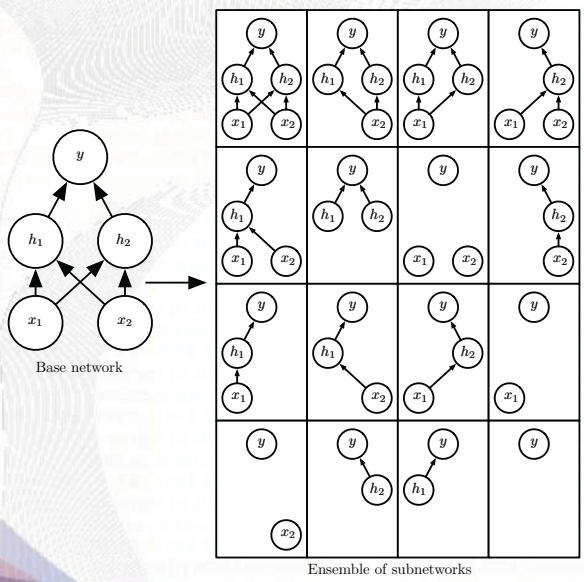
- Typically drop 50% of activations in layers
- Forces the network not to rely on small set of nodes

Early Stopping:

Weight regularization

$$\vec{w}^* \in \arg\min_{\vec{w}\in\mathbb{R}^n} J(\vec{w}) = \frac{1}{m}\sum_{i=1}^{m} \ell(y_i, f(\vec{w}, \vec{x}_i)) + \lambda \|w\|_2^2$$

Dropout [Srivastava et al., 2014]



Inexpensive but powerful method of regularization

Dropout trains the ensemble consisting of all sub-networks that can be formed by removing non-output units from an underlying base network

In each step of the SGD, a different binary mask is sampled to apply to all input and hidden units

Large networks are preferred to apply dropout

ML / DL Platforms (Python)

- ML : scikit-learn
- DL

Caffe (UC Berkeley) → Caffe2 (Facebook)

Torch (NYU / Facebook) → PyTorch (Facebook)

Theano (U Montreal) → TensorFlow (Google)

Paddle (Baidu)

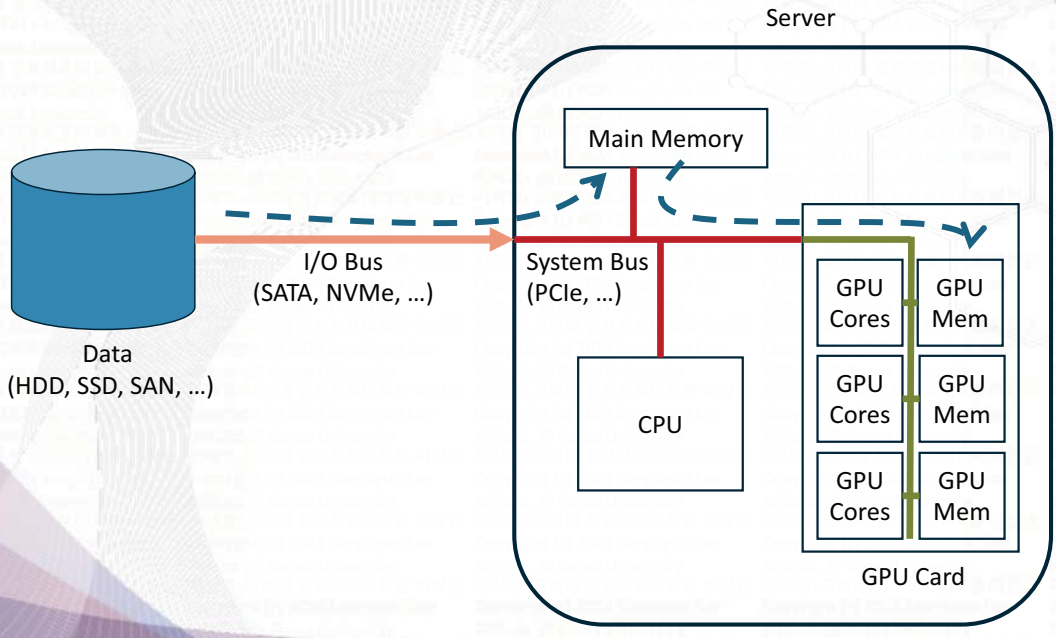
CNTK (Microsoft)

MXNet (Amazon)
Developed by U Washington, CMU, MIT, Hong Kong U, etc but main framework of choice at AWS

And others...

- Home-brewed ML / DL toolkits?

DL Pipeline in Computers



Turing Award 2018

Yoshua Bengio is a Professor at the University of Montreal, and the Scientific Director of both Mila (Quebec's Artificial Intelligence Institute) and IVADO (the Institute for Data Valorization). He is Co-director (with Yann LeCun) of CIFAR's Learning in Machines and Brains program. Bengio received a Bachelor's degree in electrical engineering, a Master's degree in computer science and a Doctoral degree in computer science from McGill University.

Geoffrey Hinton is VP and Engineering Fellow of Google, Chief Scientific Adviser of The Vector Institute and a University Professor Emeritus at the University of Toronto. Hinton received a Bachelor's degree in experimental psychology from Cambridge University and a Doctoral degree in artificial intelligence from the University of Edinburgh. He was the founding Director of the Neural Computation and Adaptive Perception (later Learning in Machines and Brains) program at CIFAR.

Yann LeCun is Silver Professor of the Courant Institute of Mathematical Sciences at New York University, and VP and Chief AI Scientist at Facebook. He received a Diplôme d'Ingénieur from the Ecole Superieure d'Ingénieur en Electrotechnique et Electronique (ESIEE), and a PhD in computer science from Université Pierre et Marie Curie.

Q/A

KSBI | 한국생명정보학회

고려대학교 KOREA UNIVERSITY

AI LAB

Introduction to Deep Learning

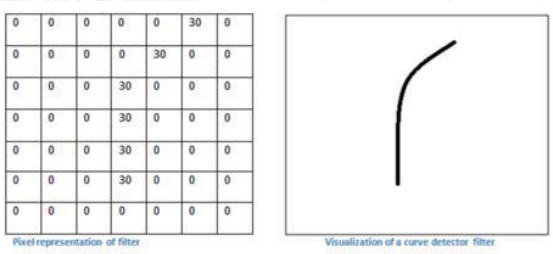
Convolutional Neural Networks

고려대학교 정보보호대학원 인공지능연구실
이상근

BIML 2024

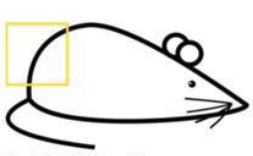
Convolution

필터 Filter
(커널 Kernel)



학습대상
이미지

Convolution



Visualization of the filter on the image

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

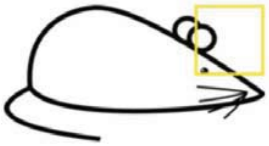
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Convolution



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

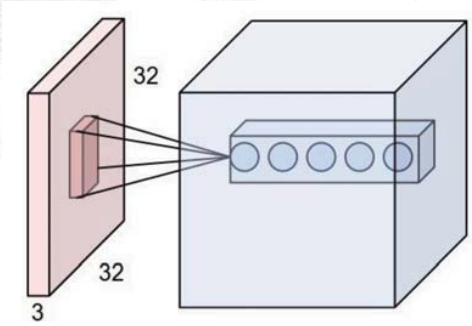
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

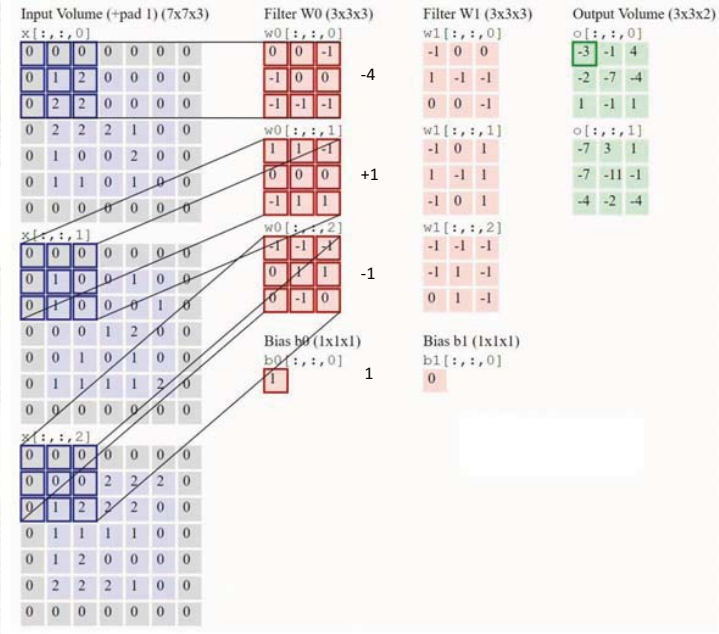
Multiplication and Summation = 0

Terminology

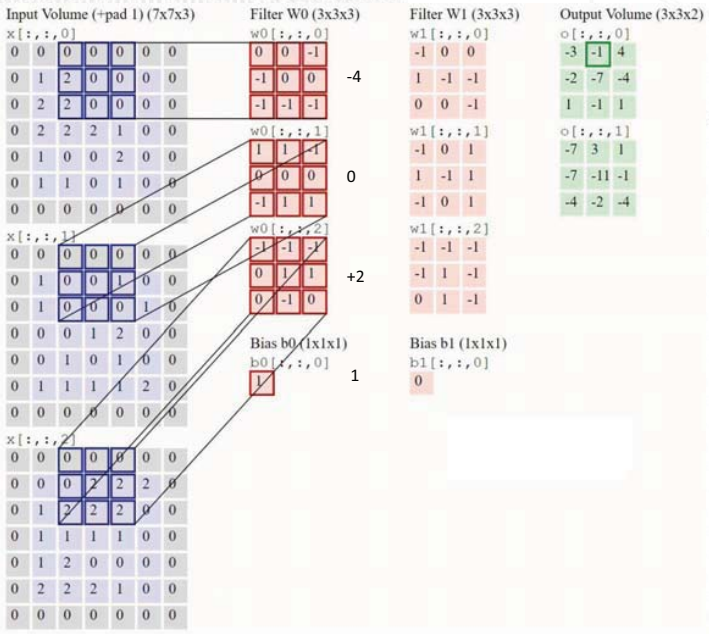


- **Depth:** number of filters
- **Stride:** filter step size (when we "slide" it)
- **Padding:** zero-pad the input

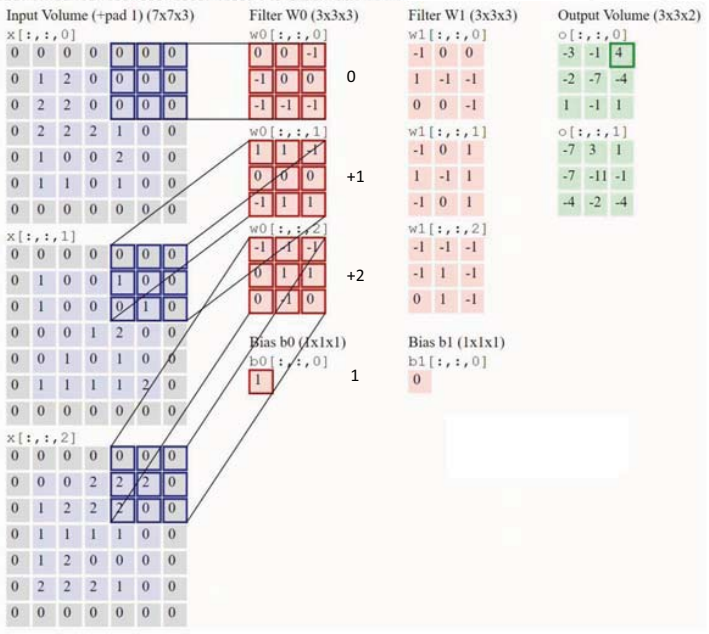
Convolution (Padding 1, Stride 2)



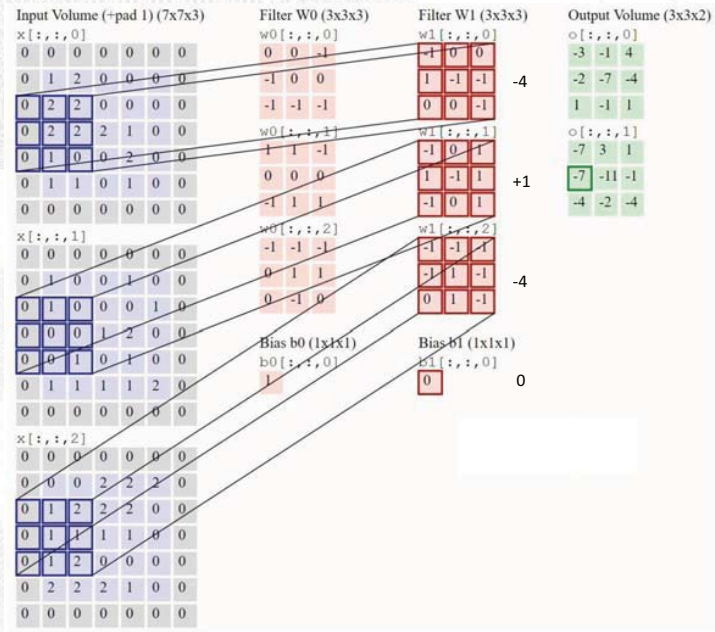
Convolution (Padding 1, Stride 2)



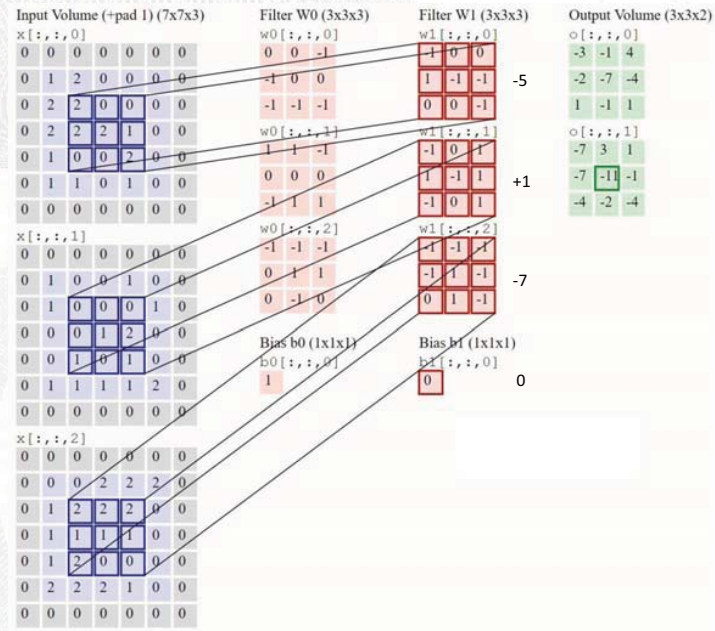
Convolution (Padding 1, Stride 2)



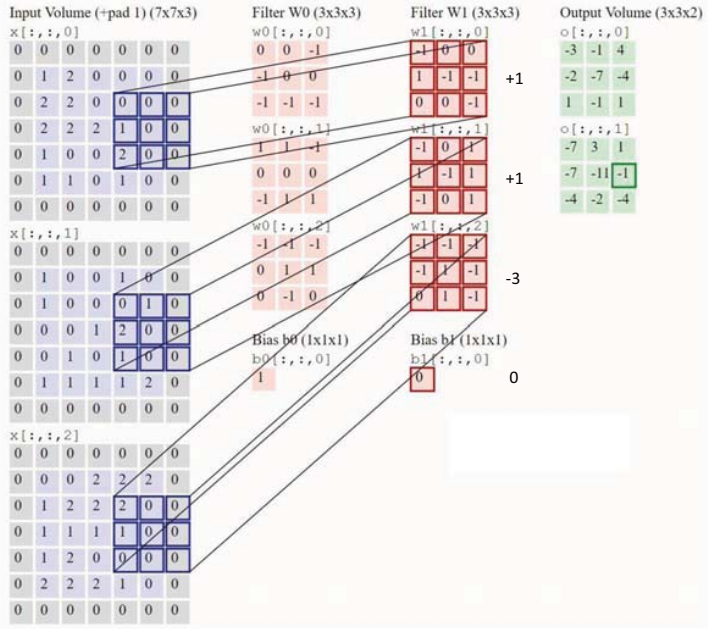
Convolution (Padding 1, Stride 2)



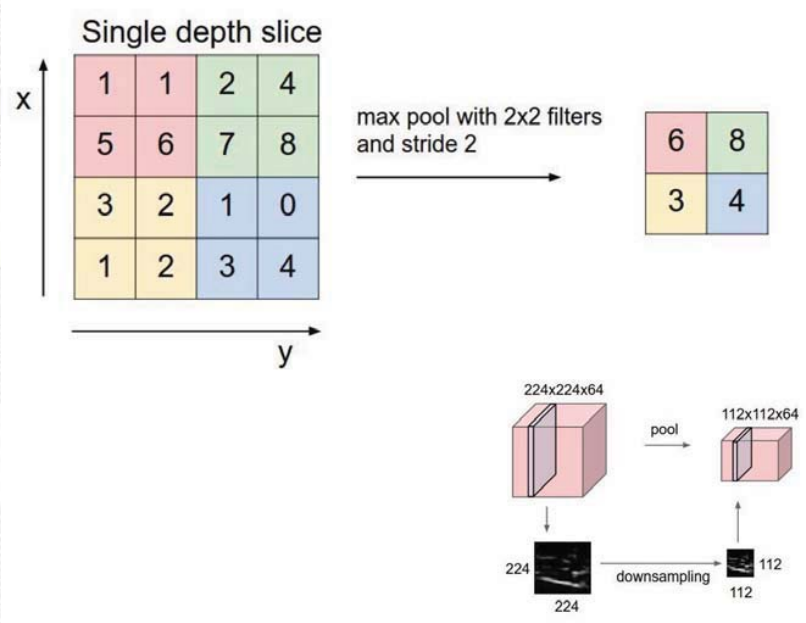
Convolution (Padding 1, Stride 2)



Convolution (Padding 1, Stride 2)

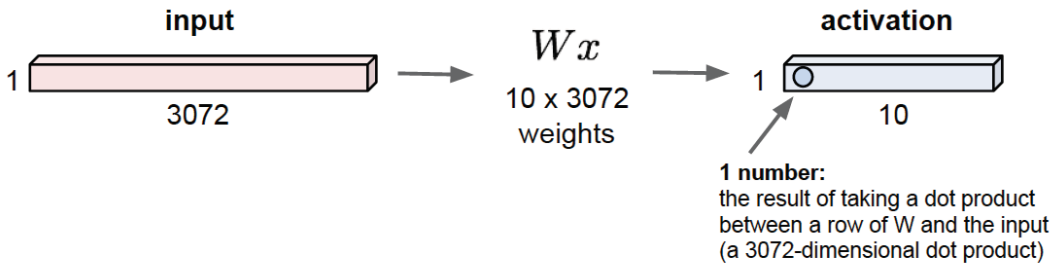


Pooling

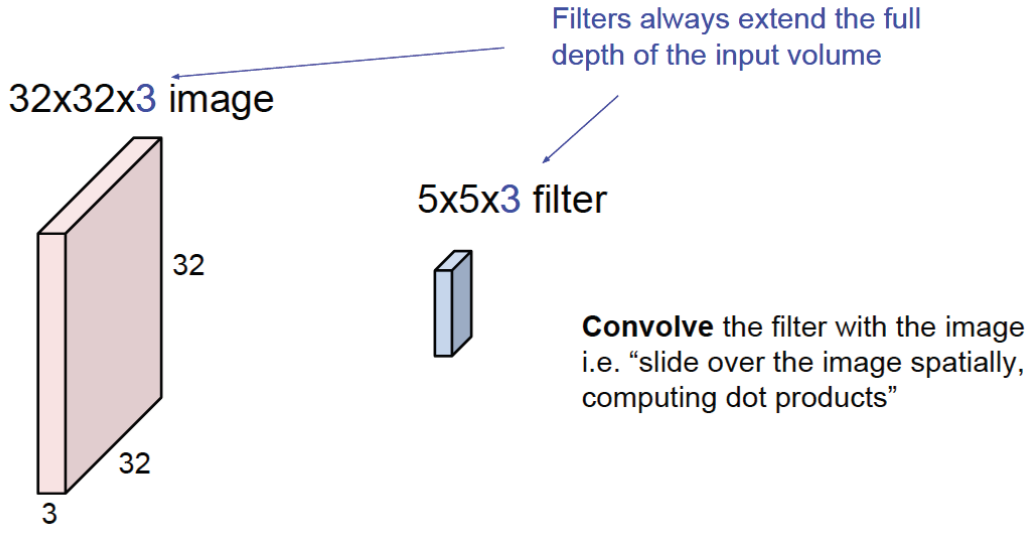


FC (Fully Connected) Layer

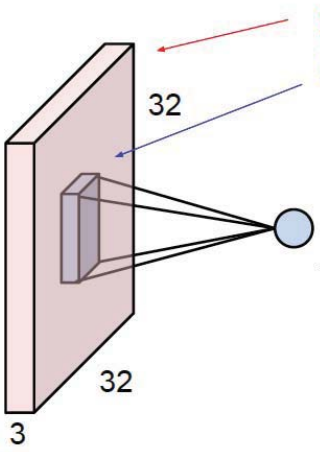
32x32x3 image -> stretch to 3072 x 1



Convolution Layer



Convolution Layer

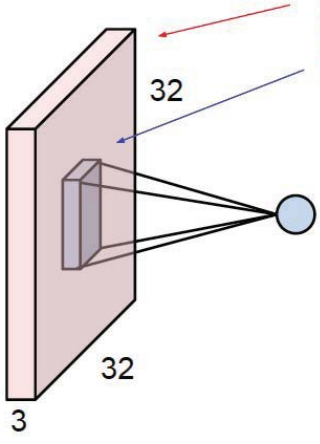


32x32x3 image
5x5x3 filter w

1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

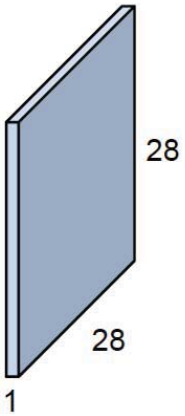
Convolution Layer



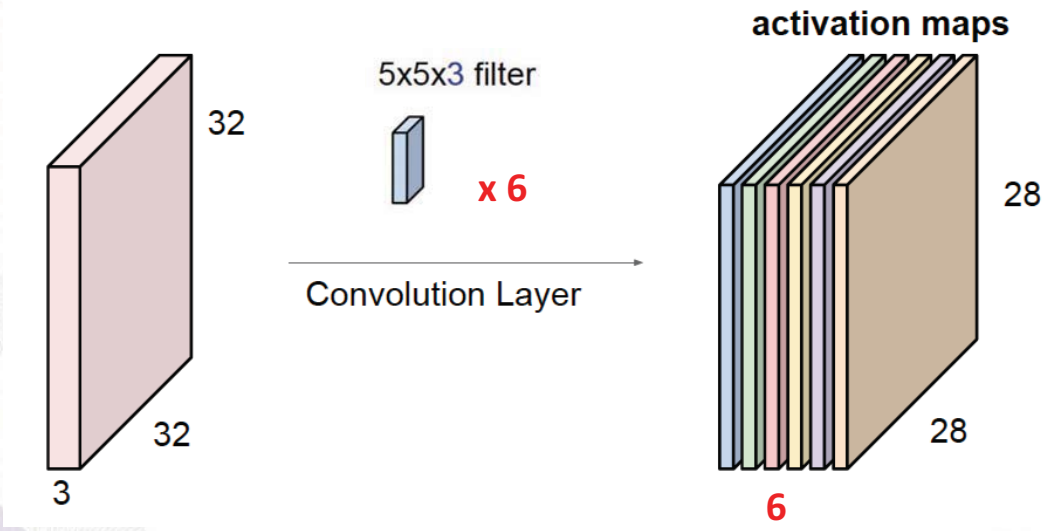
32x32x3 image
5x5x3 filter

convolve (slide) over all spatial locations

activation map

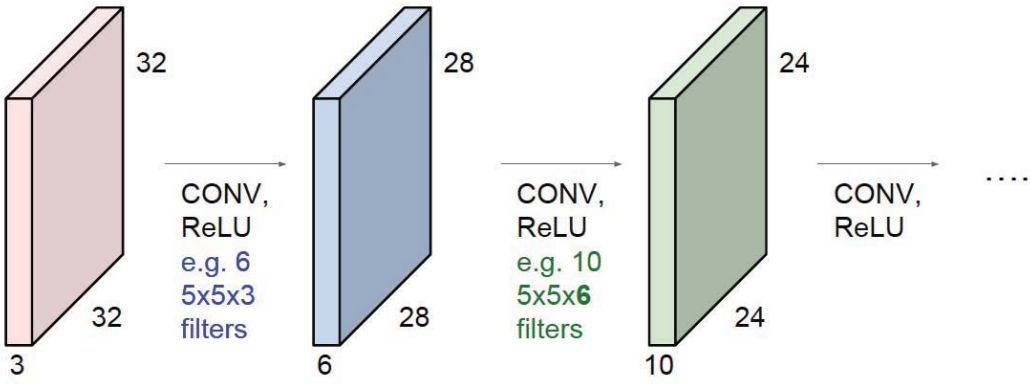


Convolution Layer

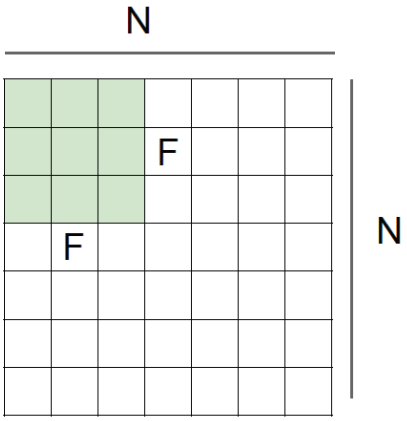


We stack these up to get a “new image” of size 28x28x6!

CNN



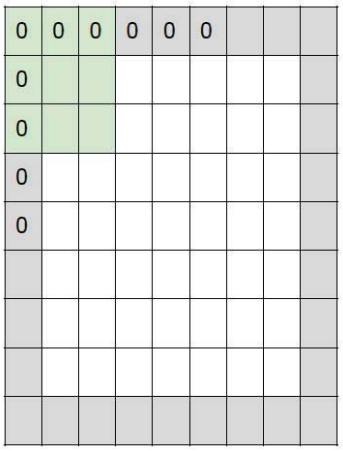
Output Size



Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33 :\

Output Size with Zero-Padding



e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

7x7 output!
in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with (F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

Output Size

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Output volume size: ?

Output Size

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Output volume size:
(32+2*2-5)/1+1 = 32 spatially, so
32x32x10

No. of Parameters

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Number of parameters in this layer?

No. of Parameters

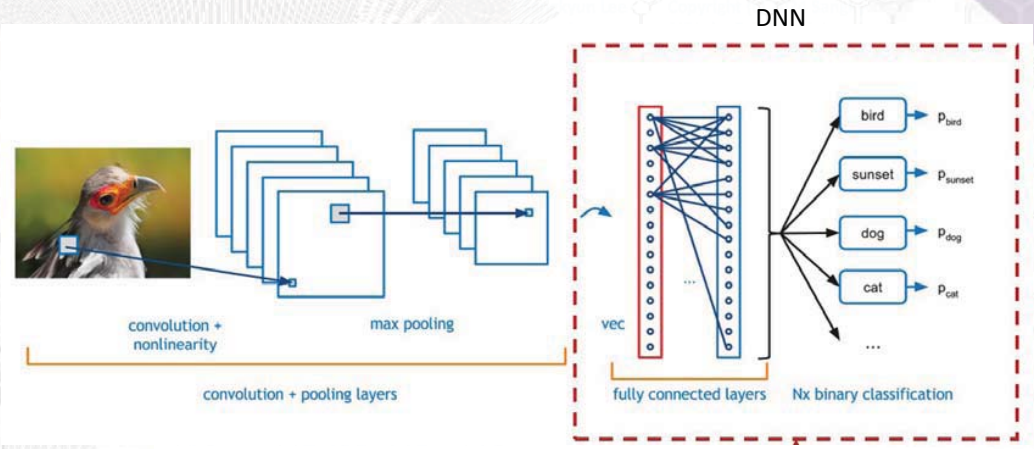
Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Number of parameters in this layer?
each filter has 5*5*3 + 1 = 76 params
=> 76*10 = **760**

Spatial Dim & No. Parameters

- Input volume: W1 x H1 x D1
- Filter (kernel)
  - No of filters K
  - Spatial extent F
  - Stride S
  - Amount of zero padding P
- Output volume: W2 x H2 x D2
  - W2 = (W1-F+2P)/S+1
  - H2 = (H1-F+2P)/S+1
  - D2 = K
- With weight sharing,
  - (F x F x D1) x K weights
  - K biases

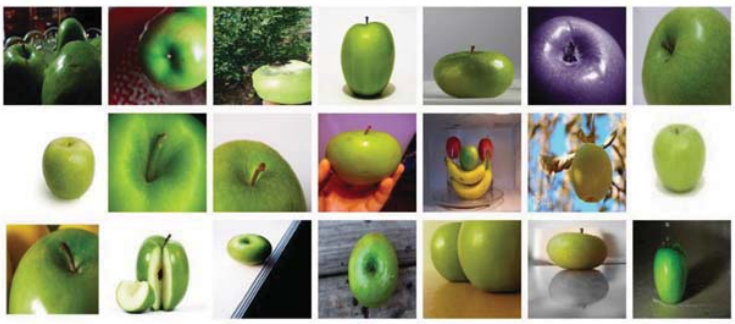
CNN Architecture



응용에 따라 Convolution 부분이나 DNN 구조를 바꿀 수 있음

ImageNet Data

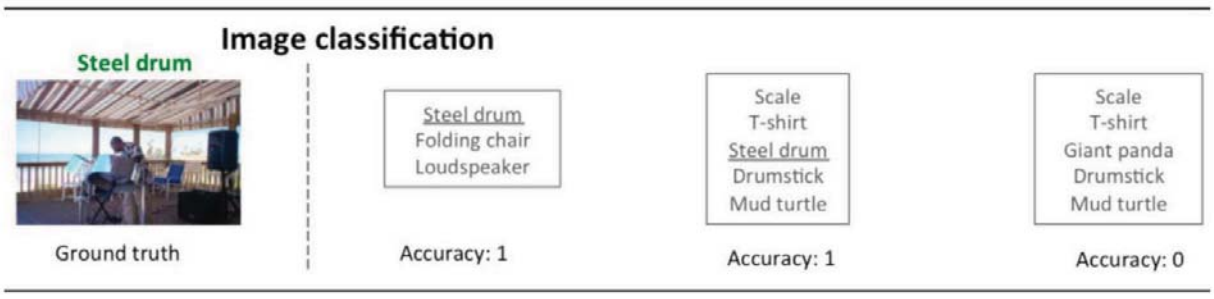
- Dataset of 14+ million images of 21,841 categories
- Category “Fruit”: 180,000 images
  - 1206 Granny Smith apples



- ILSVRC : ImageNet Large Scale Visual Recognition Challenge

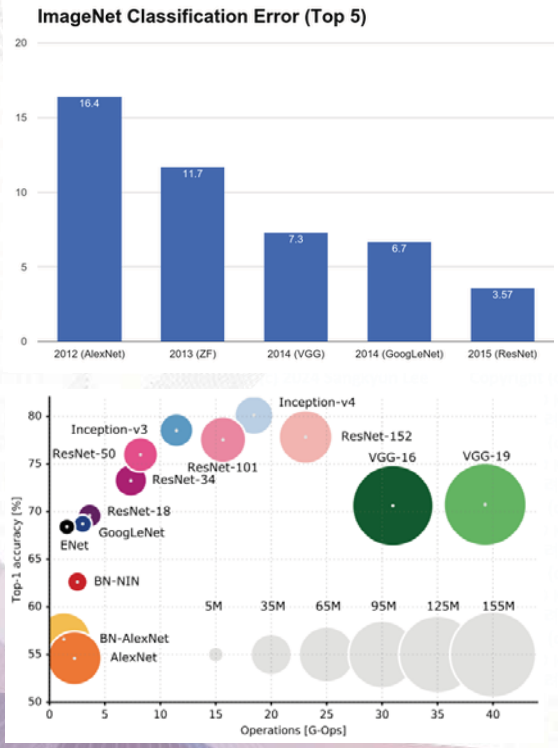
ILSVRC Challenge

- Top 5 error rate:
  - Can make 5 guesses to get the correct label



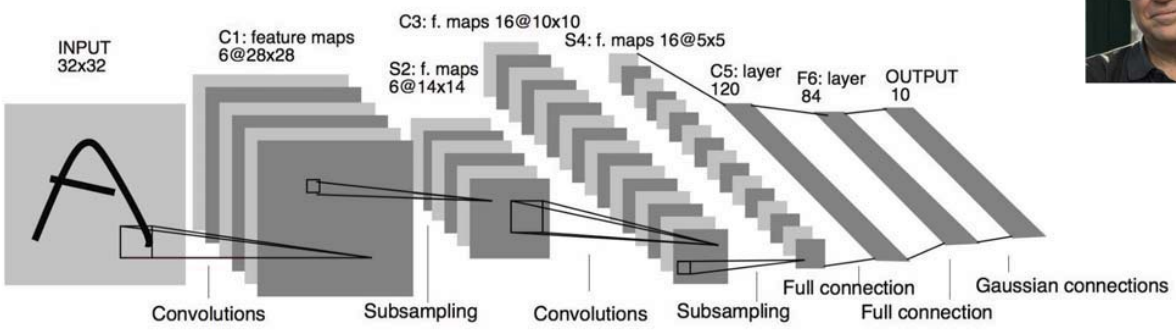
- Human annotation: binary (“apple” or “not apple”)

ILSVRC Challenge



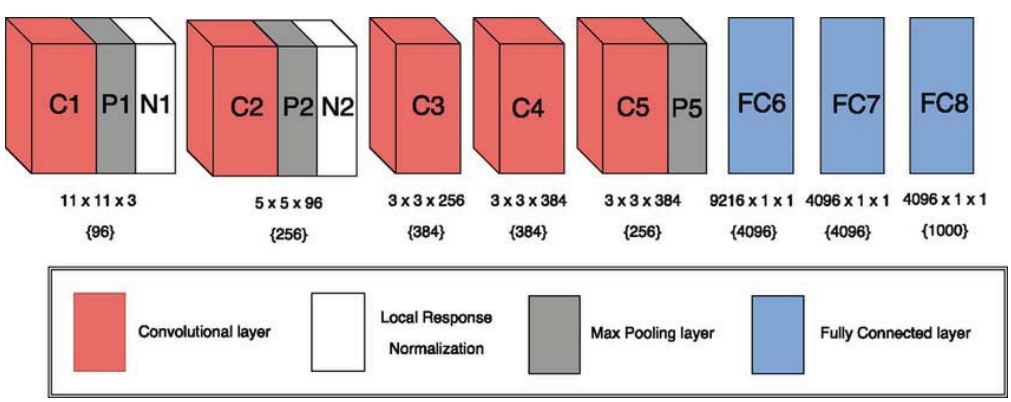
- **AlexNet (2012): First CNN for ILSVRC (16.4%)**
  - 8 layers
  - 61 million parameters
- **ZFNet (2013): 16.4% to 11.7%**
  - 8 layers
  - More filters, denser stride.
- **VGGNet (2014): 11.7% to 7.3%**
  - Beautifully uniform:
    - 3x3 conv, stride 1, pad 1, 2x2 max pool
  - 16 layers
  - 138 million parameters
- **GoogleNet (2014): 11.7% to 6.7%**
  - Inception module
  - 22 layers
  - 5 million parameters
    - (throw away FC layers)
- **ResNet (2015): 6.7% to 3.57%**
  - More layers = better performance
  - 152 layers
- **CUImage (2016): 3.57% to 2.99%**
  - Ensemble of 6 models

LeNet5 (1994)



- The very first CNN, by Yann LeCun (1994)
- No GPU computation, non-linearity = sigmoid / tanh
- Insight:
  - Image features are distributed across the entire image
  - Convolutions with learnable features

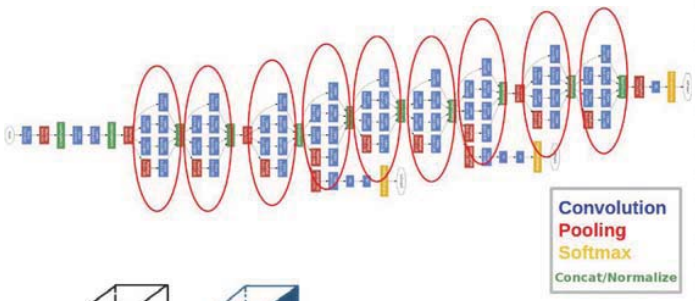
AlexNet (2012)

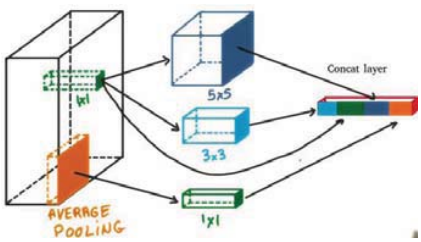


- Extension of LeNet5 to learn more complex objects and object hierarchy
- ReLU, dropout, overlapping max pooling, NVIDIA GTX 580

Krizhevsky et al. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

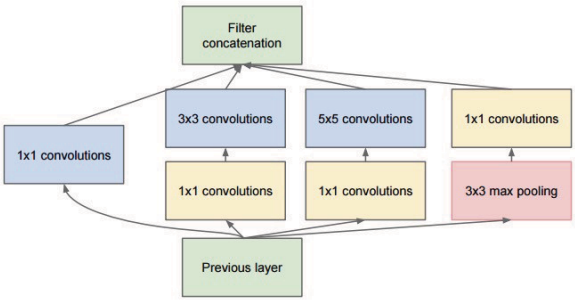
GoogLeNet (2014)





Goal: reduce no. of parameters by going deeper

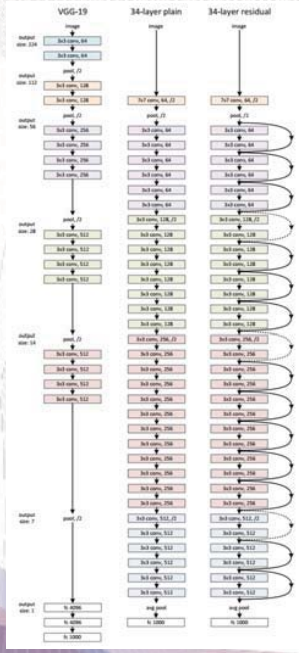
"Inception" module:



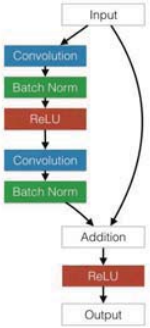
Szegedy et al. "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

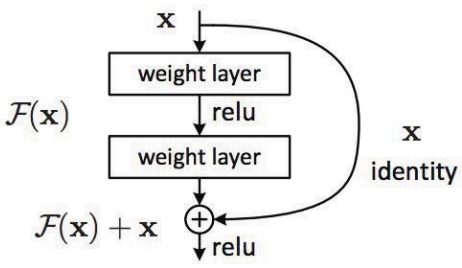
1x1 Convolution

ResNet (2015)

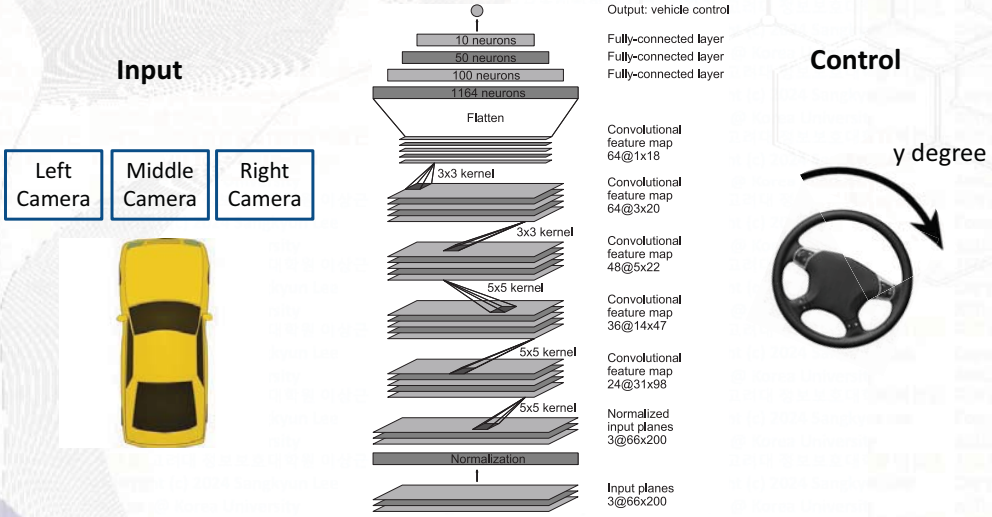


He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.



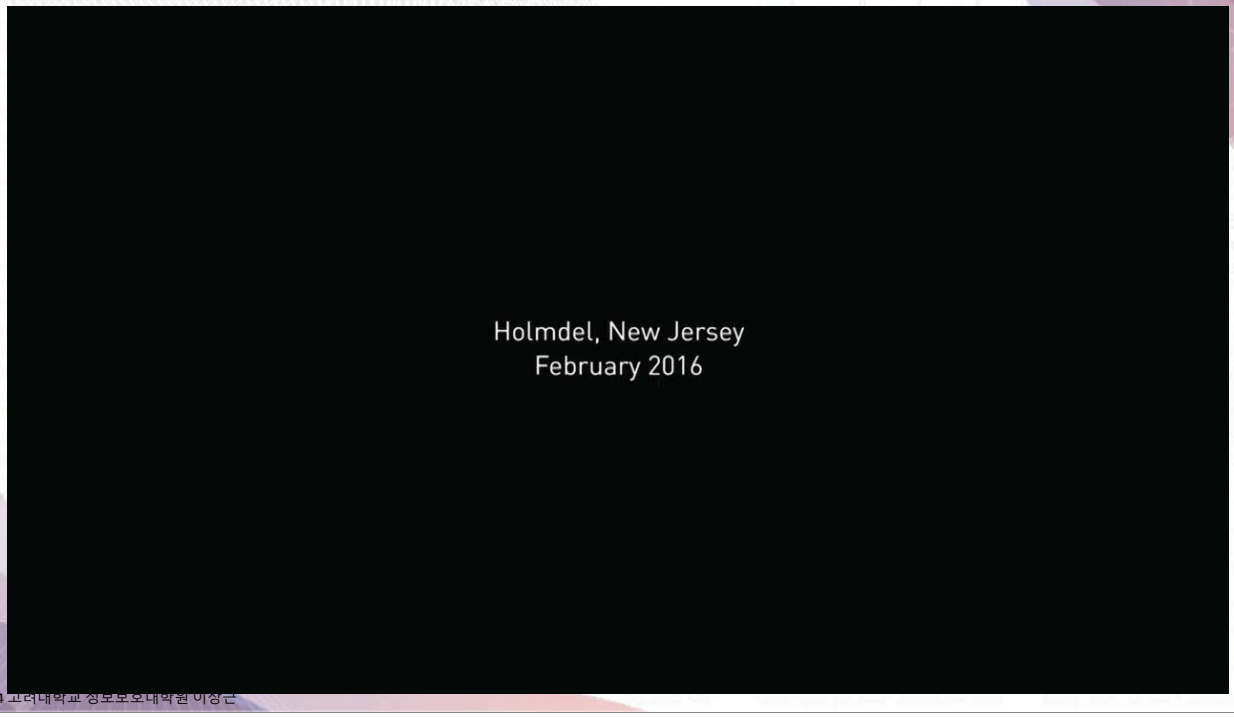


NVIDIA End-to-End CNN (2016)

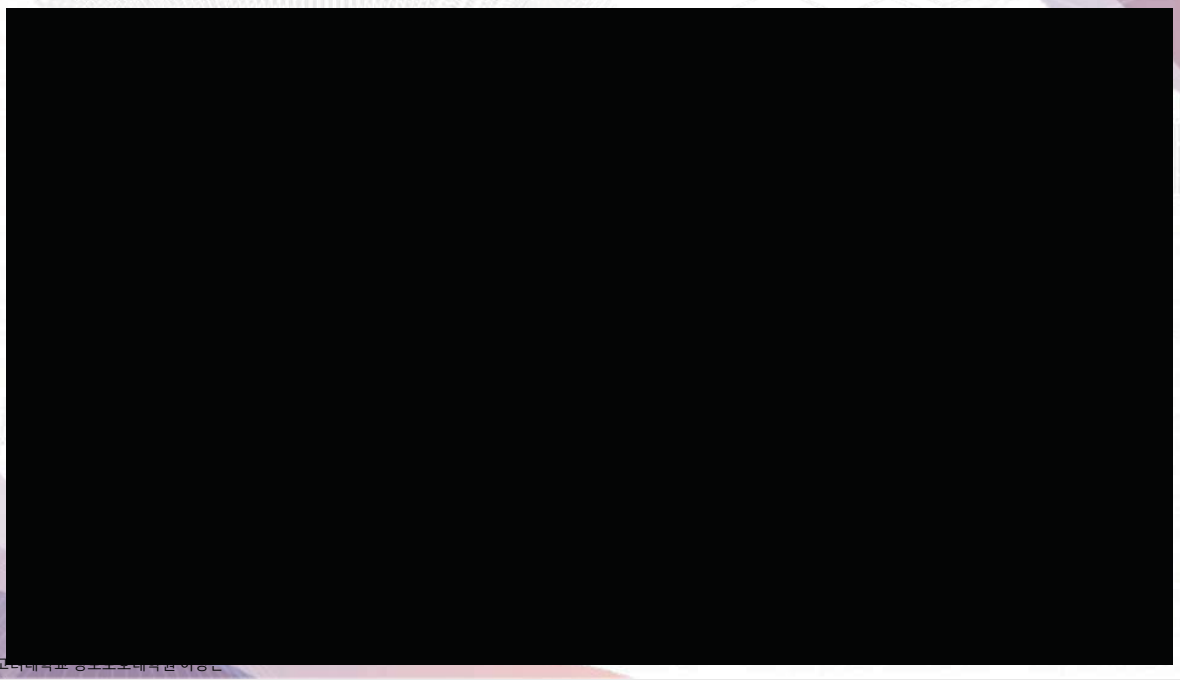


# of parameters: 250,000

NVIDIA Self-Driving (2016)

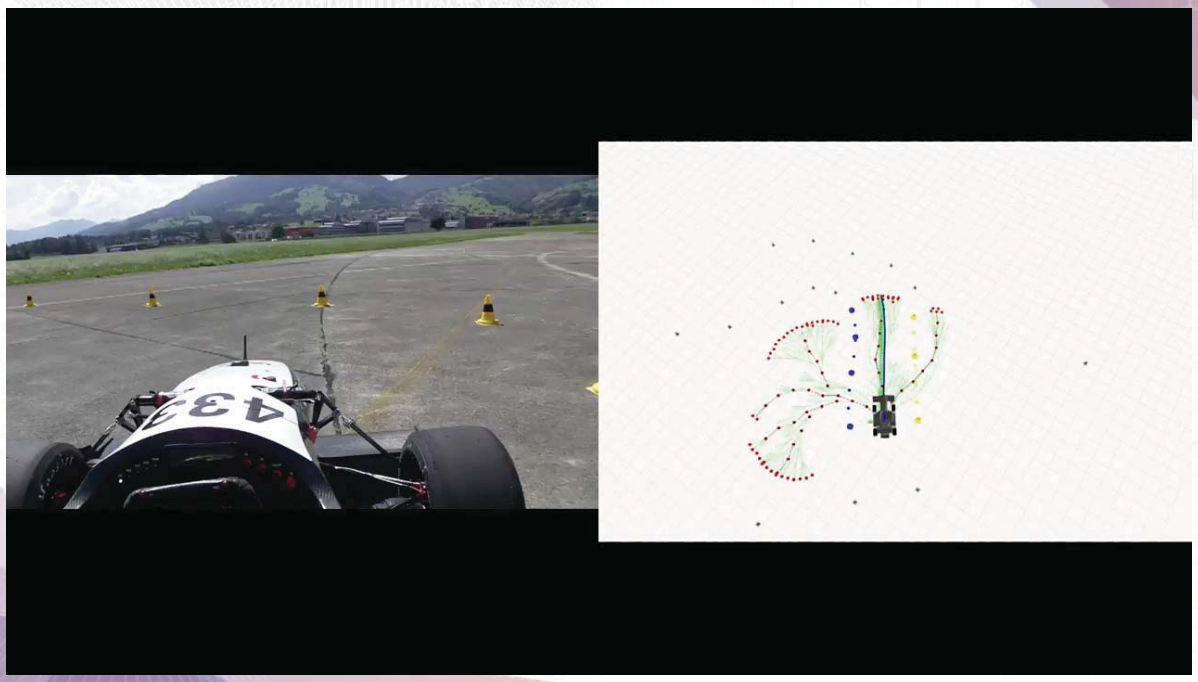


자율주행 시스템 (NVIDIA CES 2017)



DEVBOT by Roborace

Formula Students 2017 (Füela)



Q/A

KSBI | 한국생명정보학회

고려대학교 KOREA UNIVERSITY

AI LAB

Introduction to Deep Learning

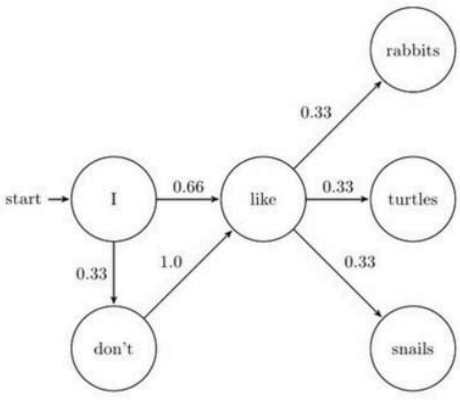
Recurrent Neural Networks + ChatGPT, XAI

고려대학교 정보보호대학원 인공지능연구실
이상근

BIML 2024

Sequence Modeling

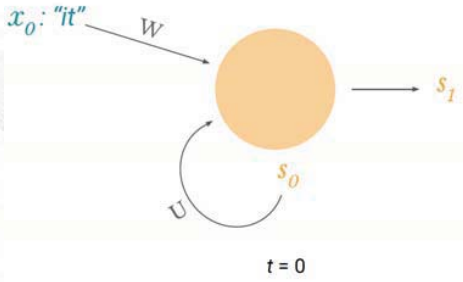
Markov models:



- Markov assumption: each state depends only on the last state
- We cannot model long-term dependencies:
  - In France, I had a good time and I learnt some of the _____ language

RNN

RNN hidden nodes “remember” their previous state:

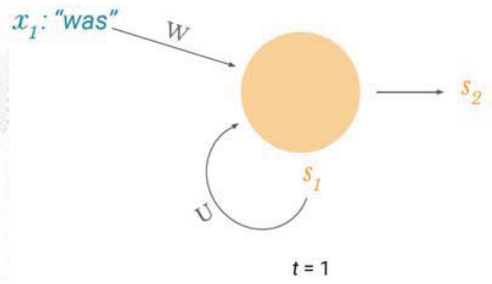


x_0 : vector representing first word
s_0 : cell state at $t = 0$ (some initialization)
s_1 : cell state at $t = 1$

$s_1 = tanh(Wx_0 + Us_0)$

RNN

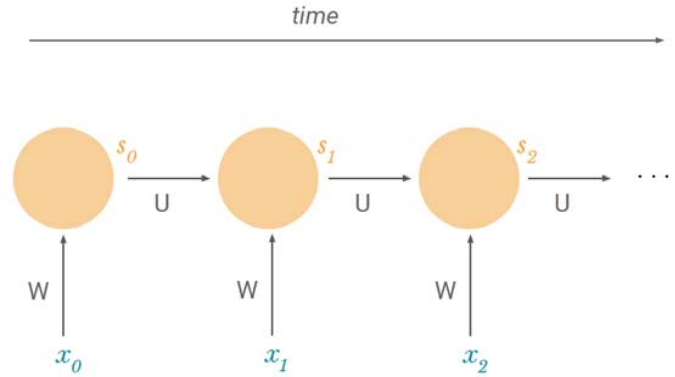
RNN hidden nodes “remember” their previous state:



x_1 : vector representing second word
s_1 : cell state at $t = 1$
s_2 : cell state at $t = 2$

$s_2 = tanh(Wx_1 + Us_1)$

Unfolding RNN Hidden States



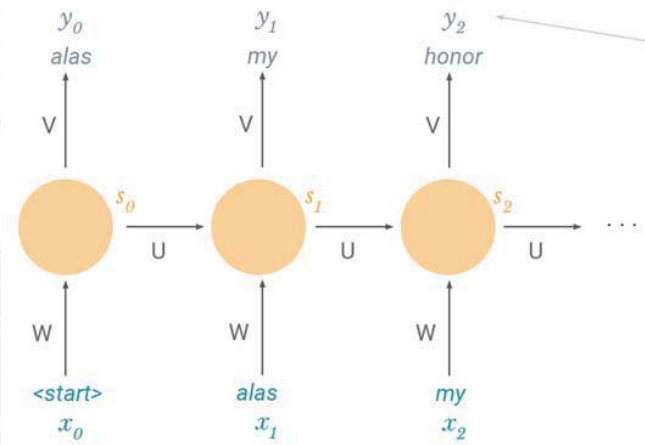
s_n can contain information from all previous states

Language Modeling

all the works of shakespeare → *language model* →

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.

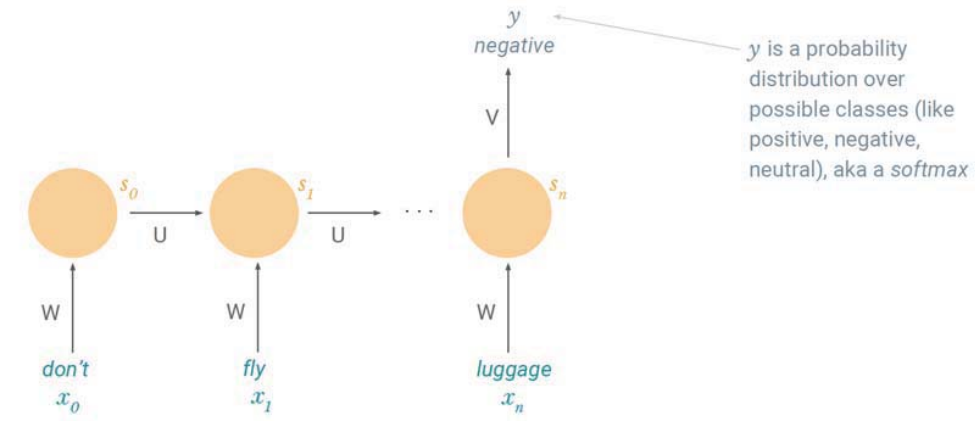
Language Modeling



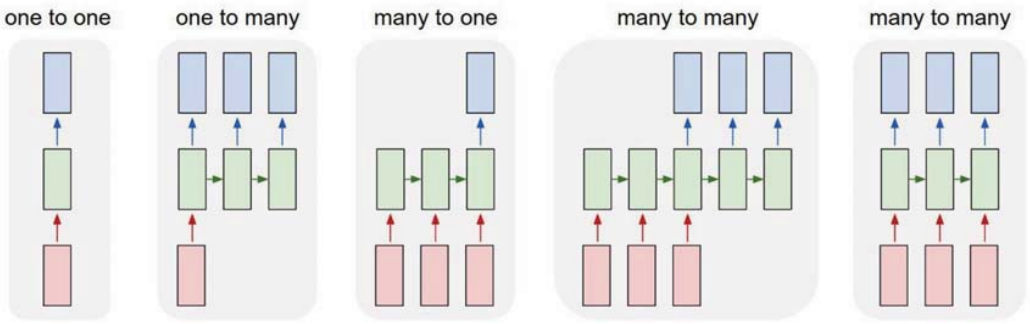
y_i is actually a probability distribution over possible next words, aka a *softmax*

Sentiment Analysis

Sentiment Analysis

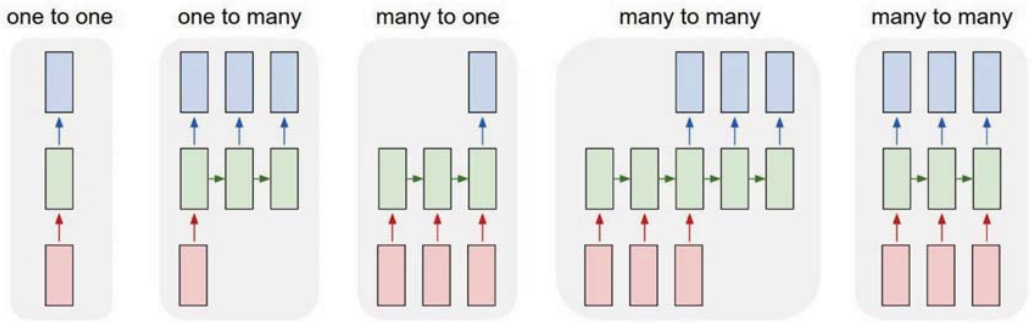


RNN: Model Sequences



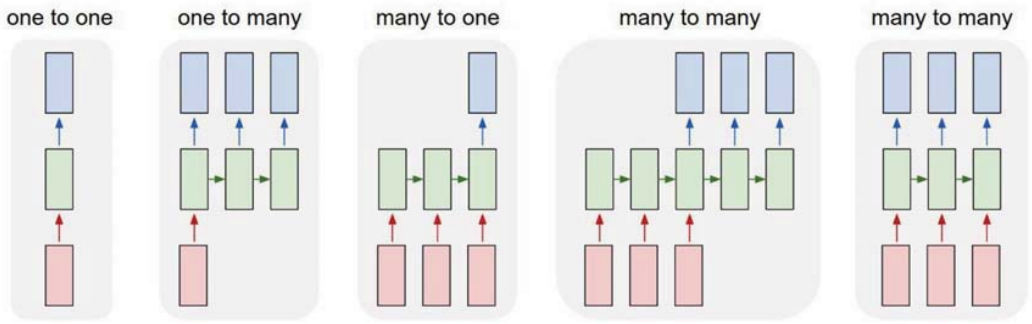
e.g. **Image Captioning**
image -> sequence of words

RNN: Model Sequences



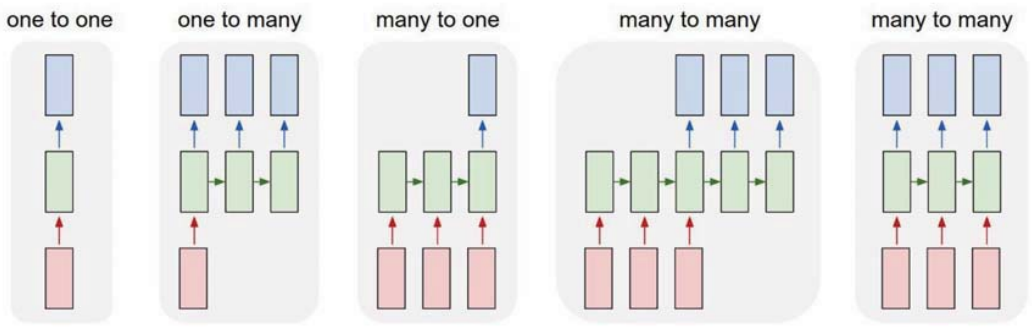
e.g. **Sentiment Classification**
sequence of words -> sentiment

RNN: Model Sequences



e.g. **Machine Translation**
seq of words -> seq of words

RNN: Model Sequences



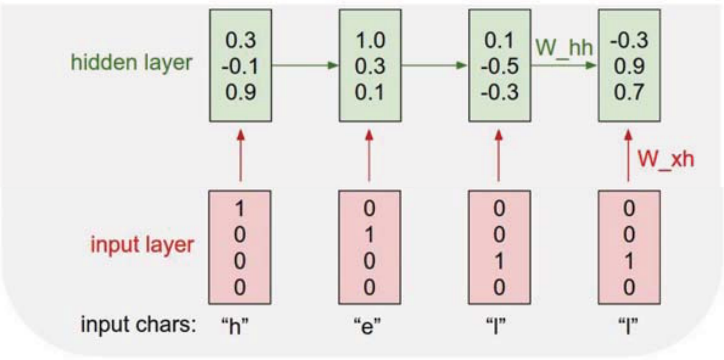
e.g. **Video classification on frame level**

Ex. Character-Level Language Modelling

Vocaburary:
[h, e, l, o]

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

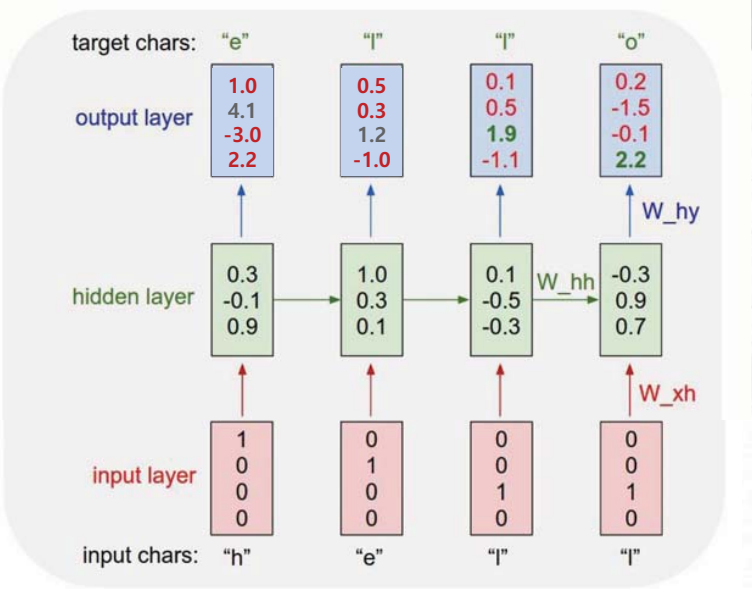
Training
seqex: “hello”



Ex. Character-Level Language Modelling

Vocaburary:
[h, e, l, o]

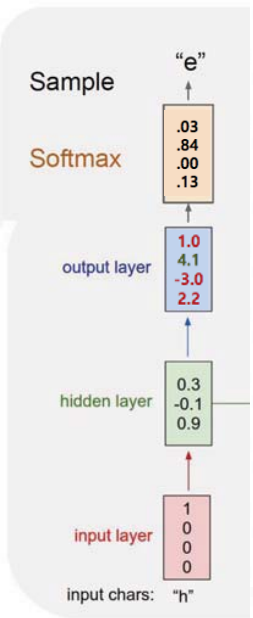
Training sequence exa
"hello"



Ex. Character-Level Language Modelling

Vocaburary:
[h, e, l, o]

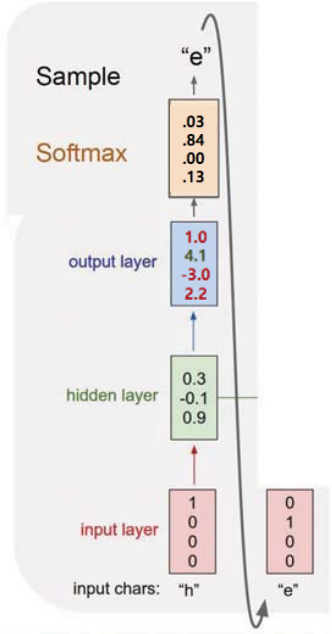
Test-time: sample one character at a time, feeding back to the model



Ex. Character-Level Language Modelling

Vocaburary:
[h, e, l, o]

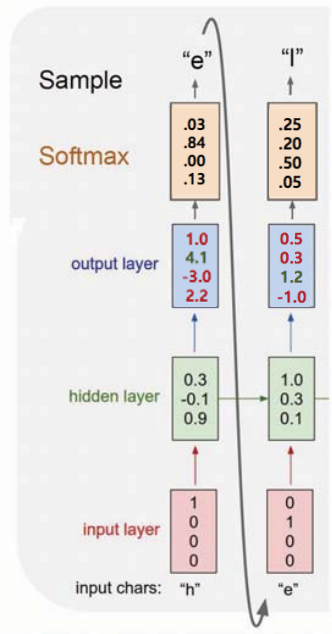
Test-time: sample one character at a time, feeding back to the model



Ex. Character-Level Language Modelling

Vocaburary:
[h, e, l, o]

Test-time: sample one character at a time, feeding back to the model



Ex. Character-Level Language Modelling

Vocaburary:
[h, e, l, o]

Test-time: sample one character at a time, feeding back to the model

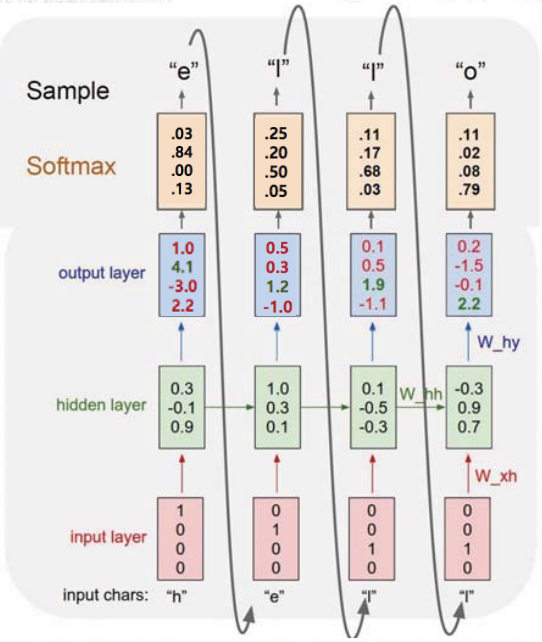


Image Captioning

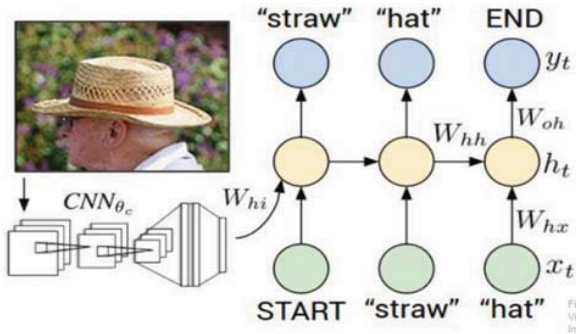


Figure from Karpathy et a, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015; figure copyright IEEE, 2015. Reproduced for educational purposes.

Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

Image Captioning

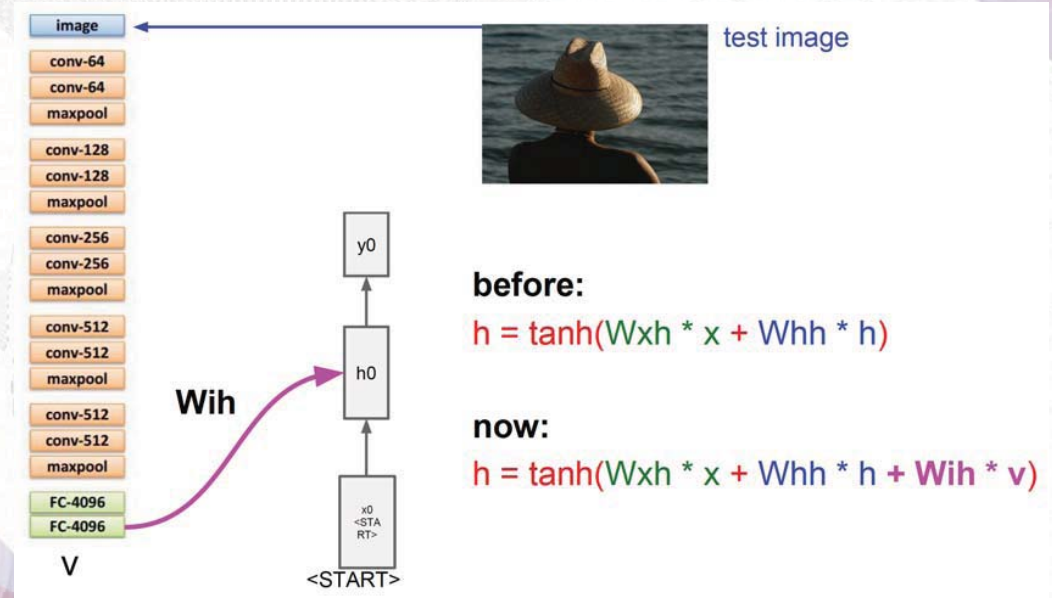


Image Captioning

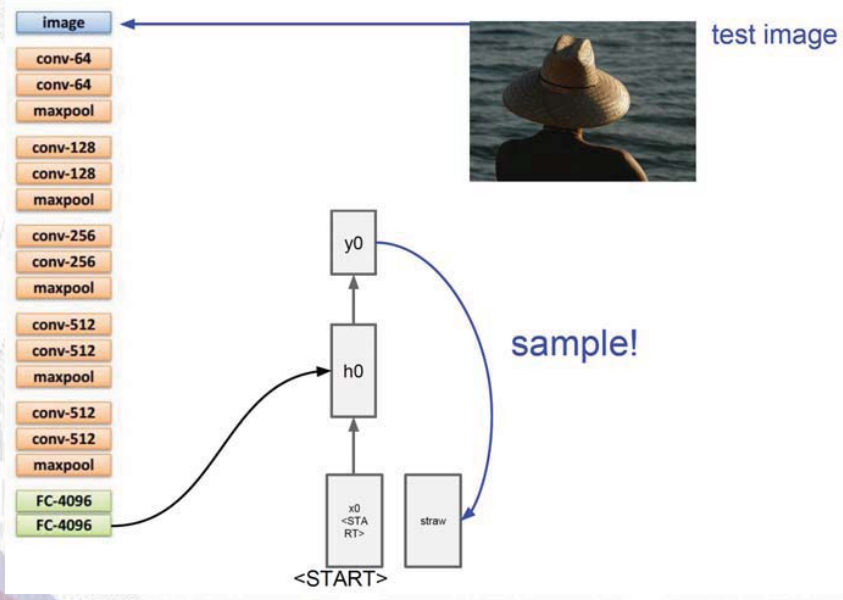


Image Captioning

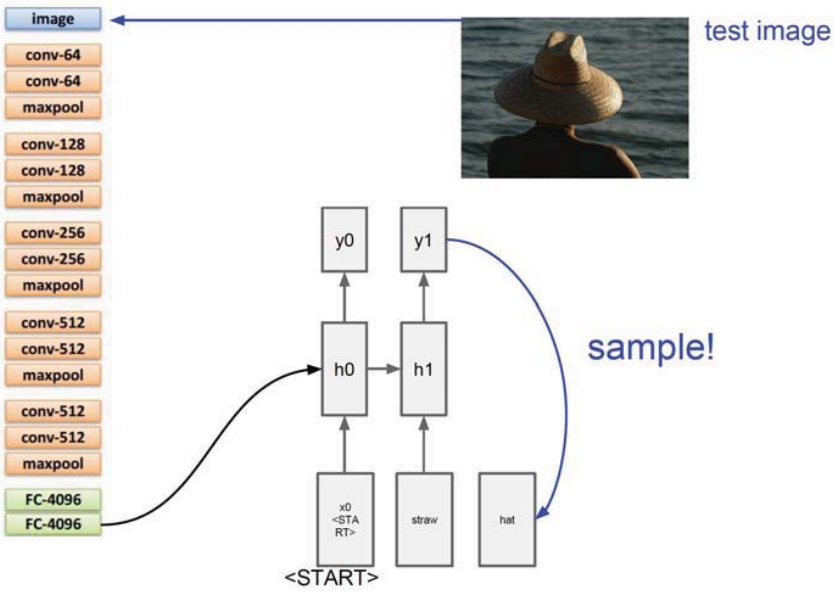


Image Captioning

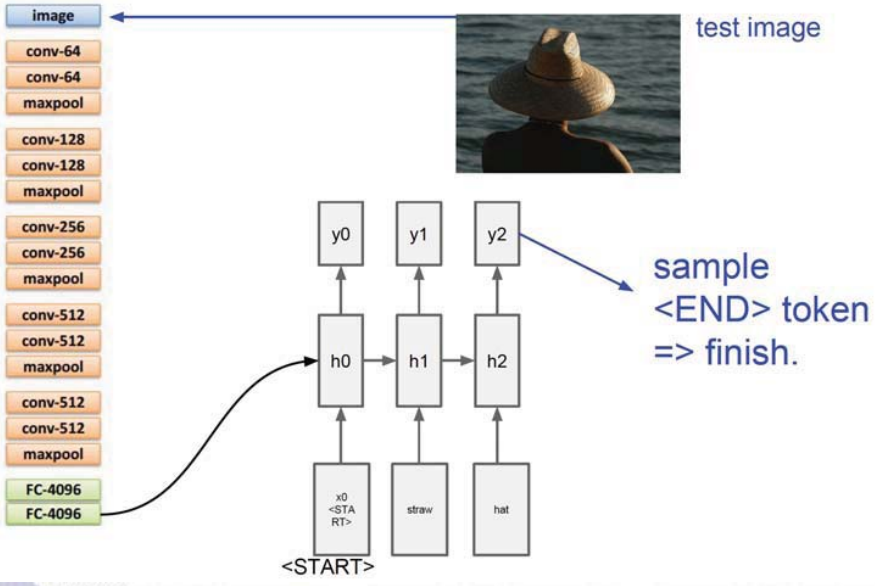


Image Captioning: Examples

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

A tennis player in action on the court

Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

Image Captioning: Failures

A woman is holding a cat in her hand

A person holding a computer mouse on a desk

A woman standing on a beach holding a surfboard

A bird is perched on a tree branch

A man in a baseball uniform throwing a ball

Visual Question Answering

Q: What endangered animal is featured on the truck?

A: A bald eagle.
A: A sparrow.
A: A humming bird.
A: A raven.

Q: Where will the driver go if turning right?

A: Onto 24 ¾ Rd.
A: Onto 25 ¾ Rd.
A: Onto 23 ¾ Rd.
A: Onto Main Street.

Q: When was the picture taken?

A: During a wedding.
A: During a bar mitzvah.
A: During a funeral.
A: During a Sunday church service

Q: Who is under the umbrella?

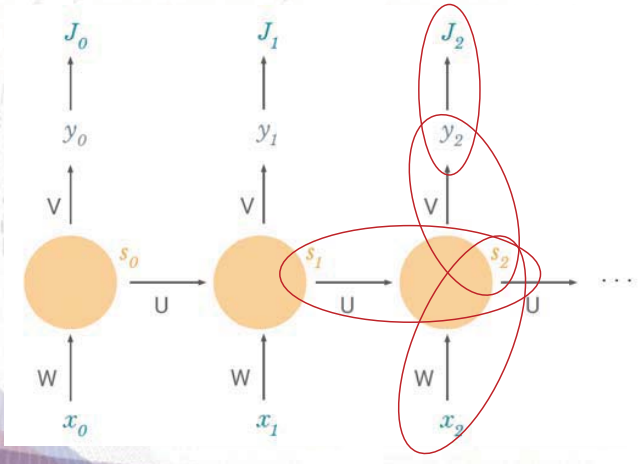
A: Two women.
A: A child.
A: An old man.
A: A husband and a wife.

Agrawal et al, "VQA: Visual Question Answering", ICCV 2015
Zhu et al, "Visual 7W: Grounded Question Answering in Images", CVPR 2016
Figure from Zhu et al, copyright IEEE 2016. Reproduced for educational purposes.

Back Propagation

The sum of stepwise losses: $J(w) = \sum_t J_t(w)$

$$\frac{\partial J}{\partial W} = \sum_t \frac{\partial J_t}{\partial W}$$

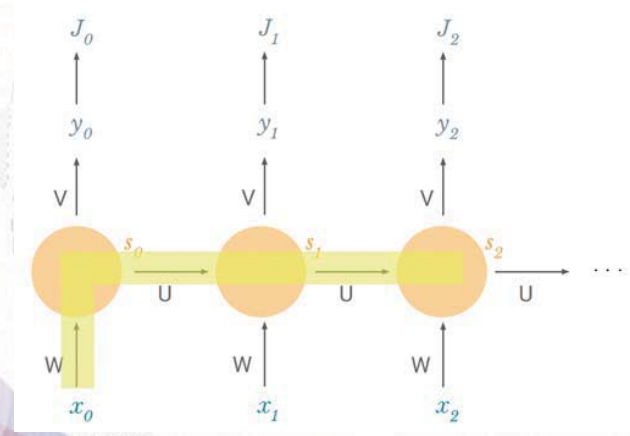


$$\frac{\partial J_2}{\partial W} =$$

$$s_2 = \tanh(Us_1 + Wx_2)$$

s_1 also depends on W !!

Dependency of s_2 on W



$$\frac{\partial s_2}{\partial W} + \frac{\partial s_2}{\partial s_1}\frac{\partial s_1}{\partial W} + \frac{\partial s_2}{\partial s_1}\frac{\partial s_1}{\partial s_0}\frac{\partial s_0}{\partial W}$$

In RNNs with many time steps, you may multiply lots of small numbers during backpropagation

➔ **Vanishing gradient problem**

Problem: Vanishing Gradient

We're multiplying lots of **small numbers**

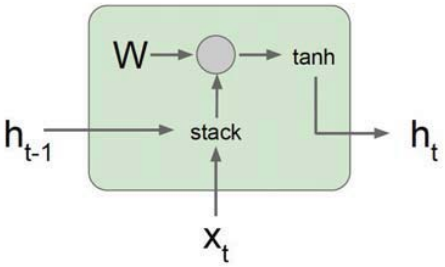
→Errors due to further back timesteps have increasingly **smaller gradients**

→Parameters become biased to capture **short-term dependencies**

Solutions:

- Use special units instead of hidden nodes
- E.g. LSTM (Long Short-Term Memory)

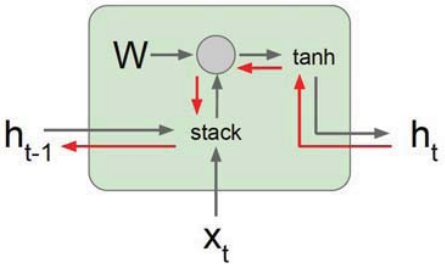
Vanilla RNN Gradient Flow



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$
$$= \tanh\left((W_{hh} \quad W_{hx})\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$
$$= \tanh\left(W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

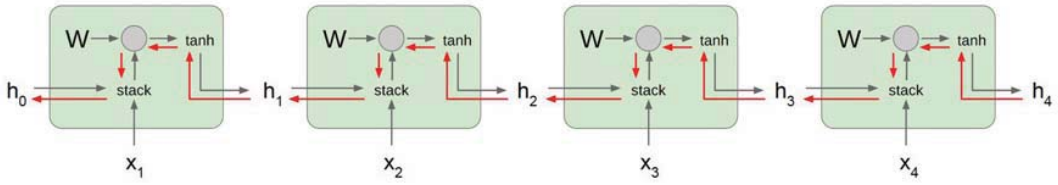
Vanilla RNN Gradient Flow

Backpropagation from h_t to h_{t-1} multiplies by W (actually W_{hh}^T)



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$
$$= \tanh\left((W_{hh} \quad W_{hx})\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$
$$= \tanh\left(W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

Vanilla RNN Gradient Flow



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1: **Exploding gradients**

Largest singular value < 1: **Vanishing gradients**

LSTM [Hochreiter et al., 1997]

f: Forget gate, Whether to erase cell
i: Input gate, whether to write to cell
g: Gate gate (?), How much to write to cell
o: Output gate, How much to reveal cell

$$h_t = \tanh\left(\begin{bmatrix} W_{hh} & W_{xh} \end{bmatrix} \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}\right)$$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \begin{bmatrix} W^i_{hh} & W^i_{xh} \\ W^f_{hh} & W^f_{xh} \\ W^o_{hh} & W^o_{xh} \\ W^g_{hh} & W^g_{xh} \end{bmatrix} \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}$$

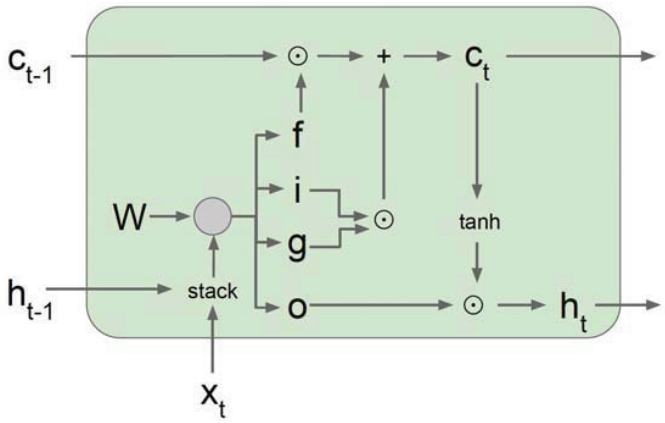
$$h_t = \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

LSTM: Gradient Flow



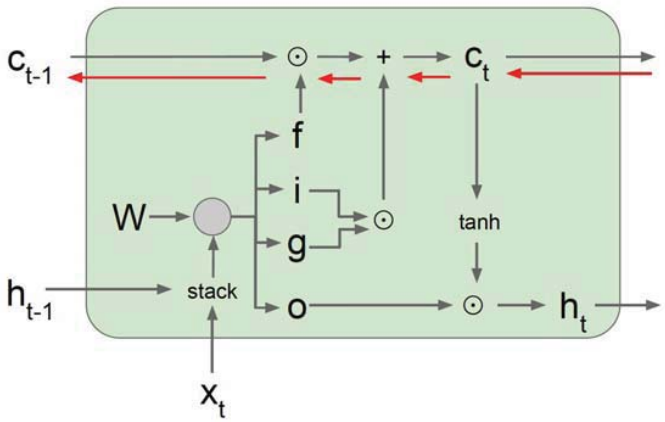
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

LSTM: Gradient Flow

Backpropagation from c_t to c_{t-1} only elementwise multiplication by f, no matrix multiply by W

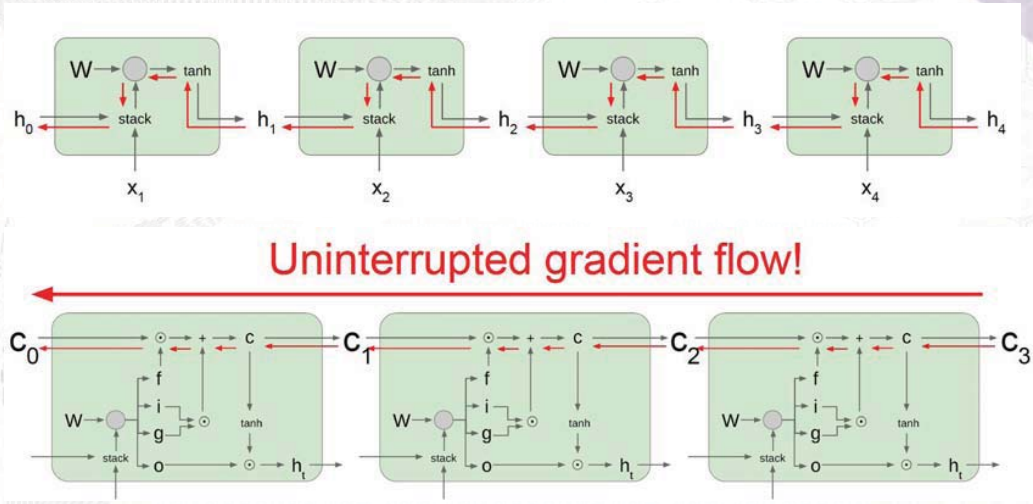


$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Vanilla vs. LSTM: Gradient Flow

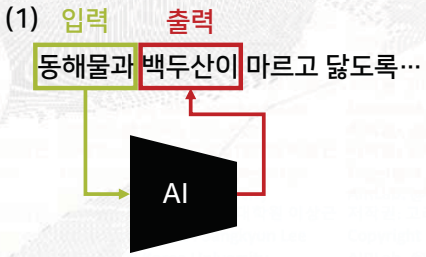


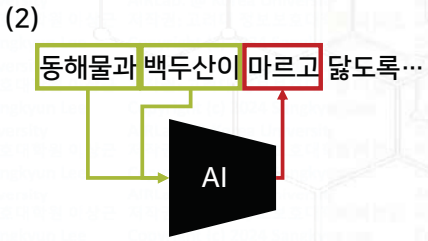
Gated Recurrent Unit

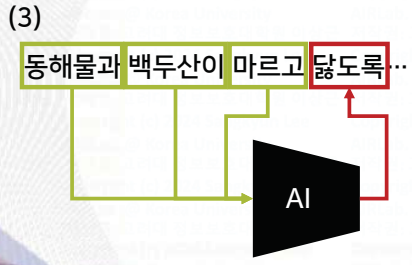
GRU [*Learning phrase representations using rnn encoder-decoder for statistical machine translation*, Cho et al. 2014]

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$
$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$
$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h)$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

NLP AI 모델: 학습



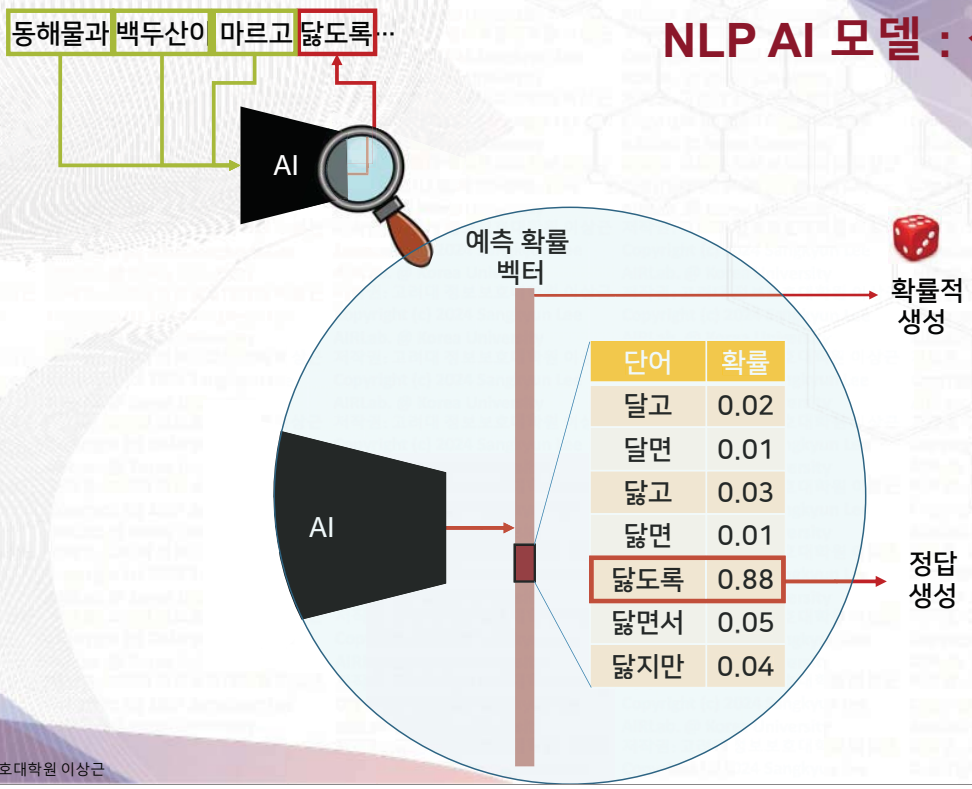




자연어 처리 작업

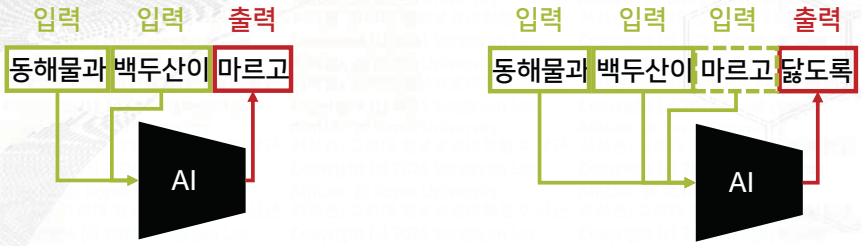
이전 단어 → 다음 단어
이전 구절 → 다음 구절
질문 → 답변
빈칸 채우기
키워드 자동완성
챗봇
…

NLP AI 모델 : 생성

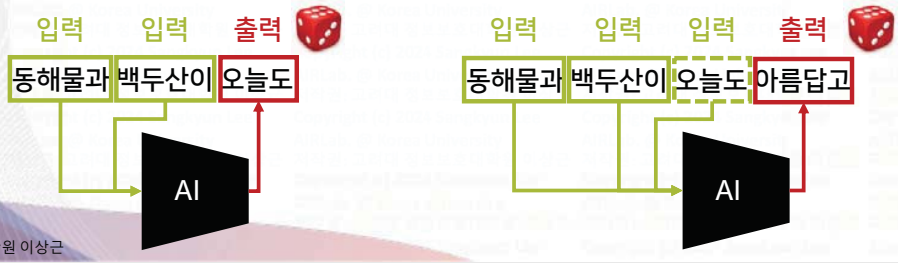


NLP AI 모델 : 출력 피드백을 통한 생성

정답을 생성할 경우 (MLE, maximum likelihood estimation):

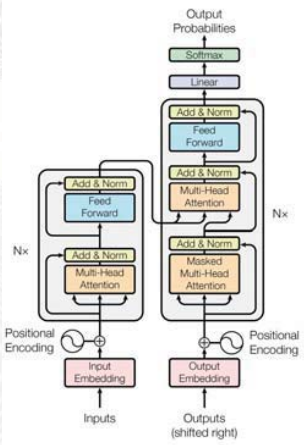


확률적으로 답을 생성할 경우:

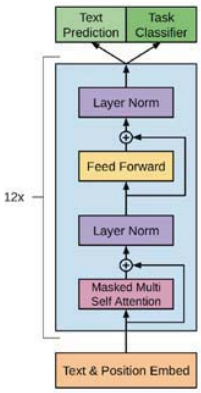


GPT의 심층 신경망 구조

Transformer 구조*



GPT (Generative Pre-trained Transformer) 구조**



어텐션 구조: 기존 RNN 모델에 비해 GPU를 이용한 병렬 학습에 유리

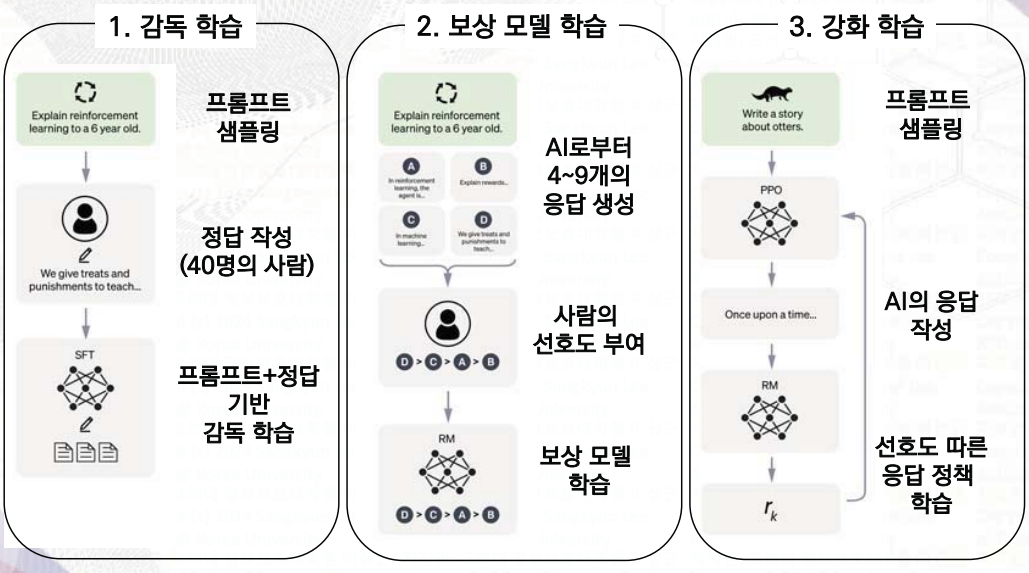
* Attention Is All You Need, NeurIPS (2017)
** Improving Language Understanding by Generative Pre-Training, OpenAI (2018, GPT-1)

ChatGPT의 학습(1): 사전학습

- 다양한 데이터: 책, 웹페이지, 논문 등
- 대용량 데이터: 약 45TB (미국 의회 도서관 4.5배)
- 비감독 학습
- 언어 모델링
- GPU 기반 대규모 병렬 학습

Training language models to follow instructions with human feedback, NeurIPS (2022, InstructGPT)

ChatGPT의 학습(2): Fine-tuning



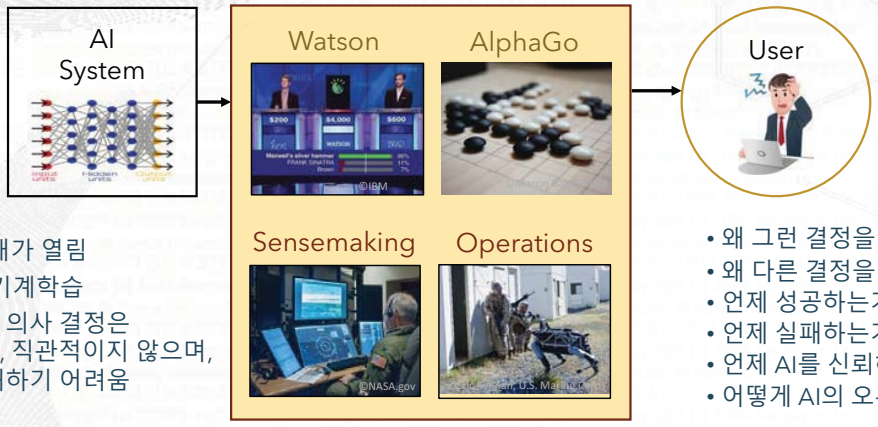
Training language models to follow instructions with human feedback, NeurIPS (2022, InstructGPT)

ChatGPT의 비용

- Fine-tuning을 위한 데이터 생성 비용
  - 사람에 의한 질문 또는 응답 작성, AI 응답에 대한 선호도 평가 비용
- 대규모 병렬 학습
  - 클라우드 컴퓨팅 기반
  - 계산량: NVIDIA A100 80Gb GPU 그래픽카드 1만장 x 수주간
    - A100 (80Gb) GPU 1장: 10,000 $ (총 GPU 비용: 약 1400억원) … now 20,000$
      (Need 5 A100 GPUs just for loading the model & a prompt)
    - H100: x9 faster training, x30 faster inference (on paper), 1장: 44,000$
  - 1회 학습: 지구 ⇄ 달 왕복 (약 70만km) 차량 주행 만큼 CO_2 배출 (sustainability?)
- 응답 생성 비용
  - 사용자 1억명 2개월만에 달성 (인스타그램 2.5년, 틱톡 9개월)
  - 하루 무료 이용자 서비스 비용 약 1.4억원 추산

XAI (eXplainable AI)

- 미국 방위고등연구계획국(DARPA)의 연구 프로그램 (2016~2021)
  - https://www.darpa.mil/program/explainable-artificial-intelligence



- AI 응용 시대가 열림
- 핵심기술: 기계학습
- 기계학습의 의사 결정은 불투명하고, 직관적이지 않으며, 사람이 이해하기 어려움

- 왜 그런 결정을 내렸는가?
- 왜 다른 결정을 내리지 않았는가?
- 언제 성공하는가?
- 언제 실패하는가?
- 언제 AI를 신뢰해도 괜찮은가?
- 어떻게 AI의 오류를 보정할 수 있는가?

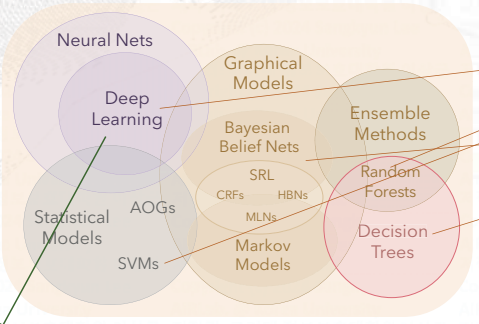
- AI의 의사 결정과 행동의 이유를 설명하지 못하면 AI의 효과적 적용이 제한될 수밖에 없음
- 설명가능한 AI는 AI를 이해하고, 적정 수준까지 신뢰하며, 효과적으로 운영하기 위해 필수적임

XAI 모델 (1): 설명가능한 새로운 딥러닝 모델

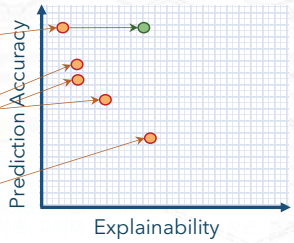
새로운 방법론

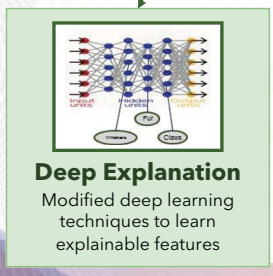
더 나은 설명력과
예측 성능을 갖는
모델의 개발

현재 AI 방법론



설명가능성





Deep Explanation
Modified deep learning techniques to learn explainable features

Example Explanations

Hendricks et al., Generating Visual Explanations, arXiv, 2016 (UC Berkeley)

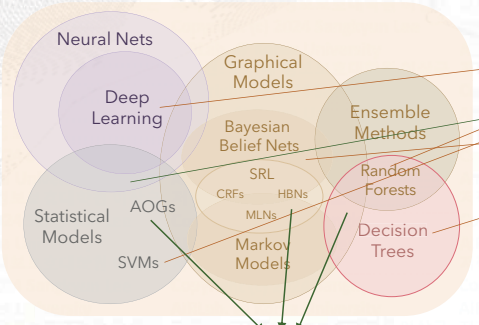
- 새의 종류를 85%의 정확도로 예측
- 이미지의 설명과 종의 정의와 부합하는 부분을 설명

XAI 모델 (2): 설명가능한 향상된 기계학습 모델

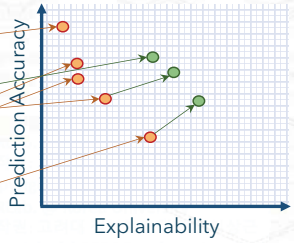
새로운 방법론

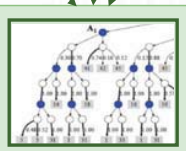
더 나은 설명력과
예측 성능을 갖는
모델의 개발

현재 AI 방법론



설명가능성





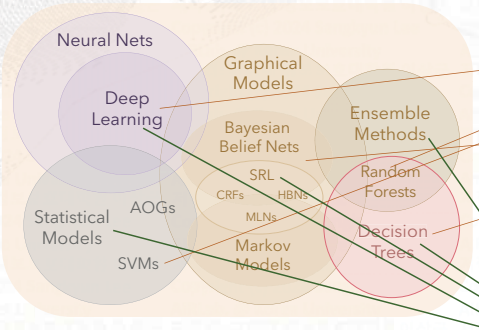
Interpretable Models
Techniques to learn more structured, interpretable, causal models

XAI 모델 (3): 설명 추출 기법

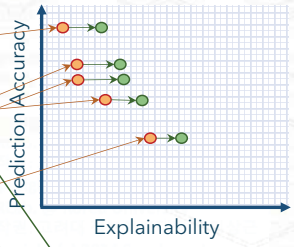
새로운 방법론

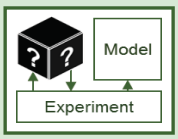
더 나은 설명력과
예측 성능을 갖는
모델의 개발

현재 AI 방법론



설명가능성





Model Induction
Techniques to infer an explainable model from any model as a black box

KOREA UNIVERSITY

IJCAI ECAI VIENNA 22

Libra-CAM: An Activation-Based Attribution Based on the Linear Approximation of Deep Neural Nets and Threshold Calibration

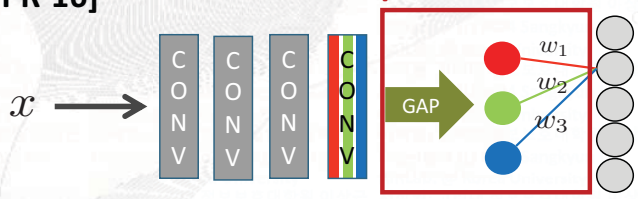
Sangkyun Lee* & Sungmin Han

School of Cybersecurity
Korea University, South Korea

IJCAI-22

CAM and Grad-CAM

❖ **CAM [CVPR`16]**



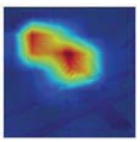
Explicit modification

Trained, Channel-wise

$$s^c(x) = \sum_k \boxed{w_k^c} \sum_{i,j} A_{ij}^k$$

Class Activation Map (CAM)

$$w_1^c + w_2^c + w_3^c =$$



$$I^c_{\text{CAM}}(x) := \sum_k \boxed{w_k^c} A_{ij}^k$$

❖ **Grad-CAM [ICCV`17]**

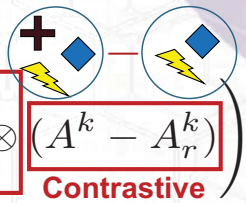
$$I^c_{\text{Grad-CAM}}(x) := \text{ReLU}\left(\sum_k \boxed{w_k^c} A^k(x) \right)$$

Computable, Channel-wise

$$\boxed{w_k^c = \frac{1}{Z} \sum_{i,j} \frac{\partial f_c(x)}{\partial A_{ij}^k}}$$

Libra-CAM: a CAM based on LInear approximation and threshold caliBRAtion

❖ **A single-ref version:**



$$I_r^c(x) := \rho \left(\alpha \sum_k \boxed{\frac{\partial f_c}{\partial A^k}\Big|_x} \otimes \boxed{(A^k - A_r^k)} \right)$$

Element-wise

Contrastive

Scaling to [0,1] range

Can be arbitrarily small > 0
→ minimize approximation error

❖ **Multiple contrastive reference points**

- We can choose any reference without sacrificing the approximation error
- Use references contrastive to the target class c:

$$I^c_{\text{Libra-CAM}}(x) := \frac{1}{R} \sum_{r=1}^{R} I_r^c(x)$$

- A pre-built reference library is used with ref filtering: $f_c(I_r \otimes x) > \gamma$

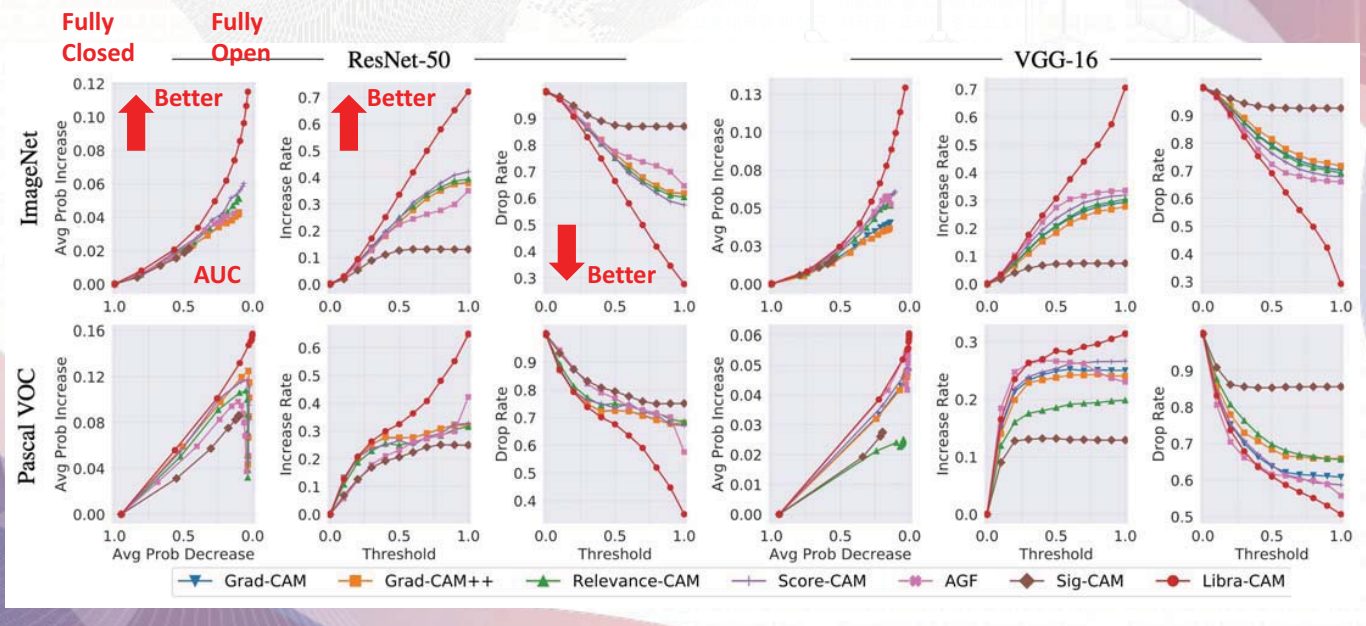
Quality Measures

- Avg Prob Inc (API): $\frac{1}{n}\sum_{i=1}^{n}\frac{(o_i^c - y_i^c)^+}{y_i^c}$
- Avg Prob Drop (APD): $\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i^c - o_i^c)^+}{y_i^c}$
- Inc Rate (IR): $\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(y_i^c < o_i^c)$
- Drop Rate (DR): $\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(y_i^c > o_i^c)$

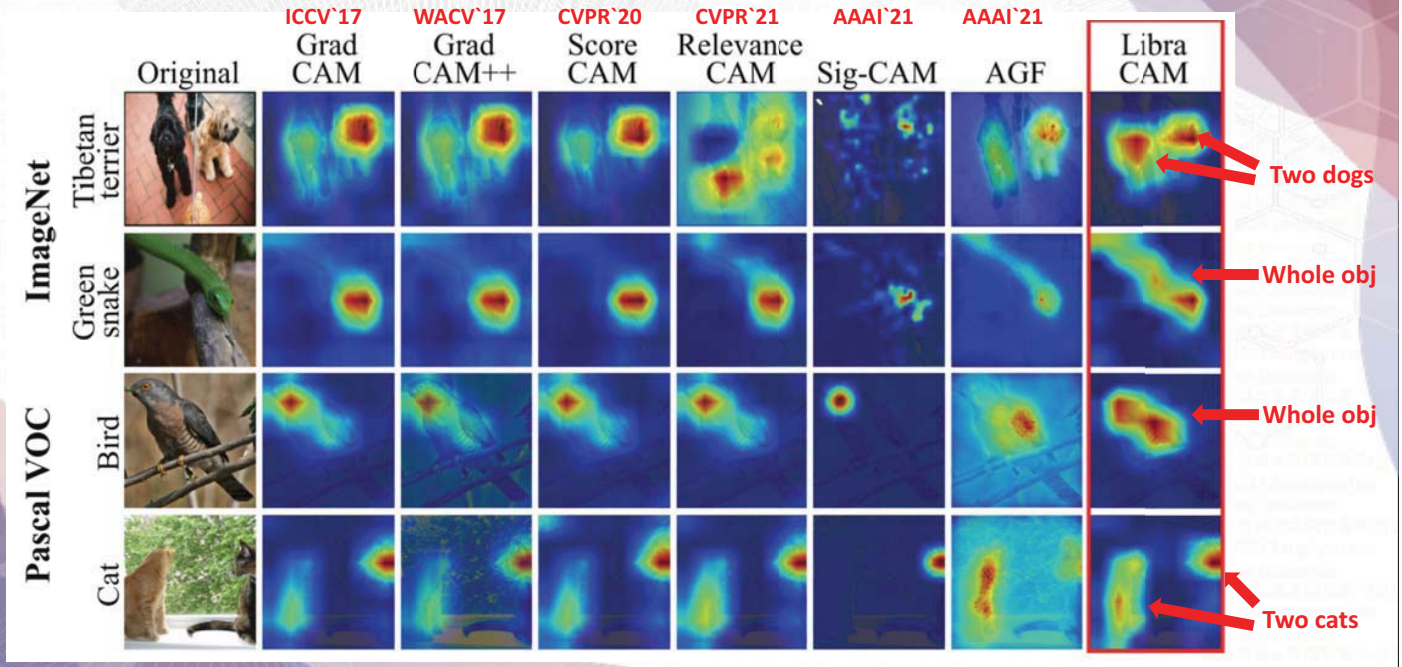
- $y_i^c = f_c(x)$
- $o_i^c = f_c(I^c(x) \otimes x)$

Attribution Quality

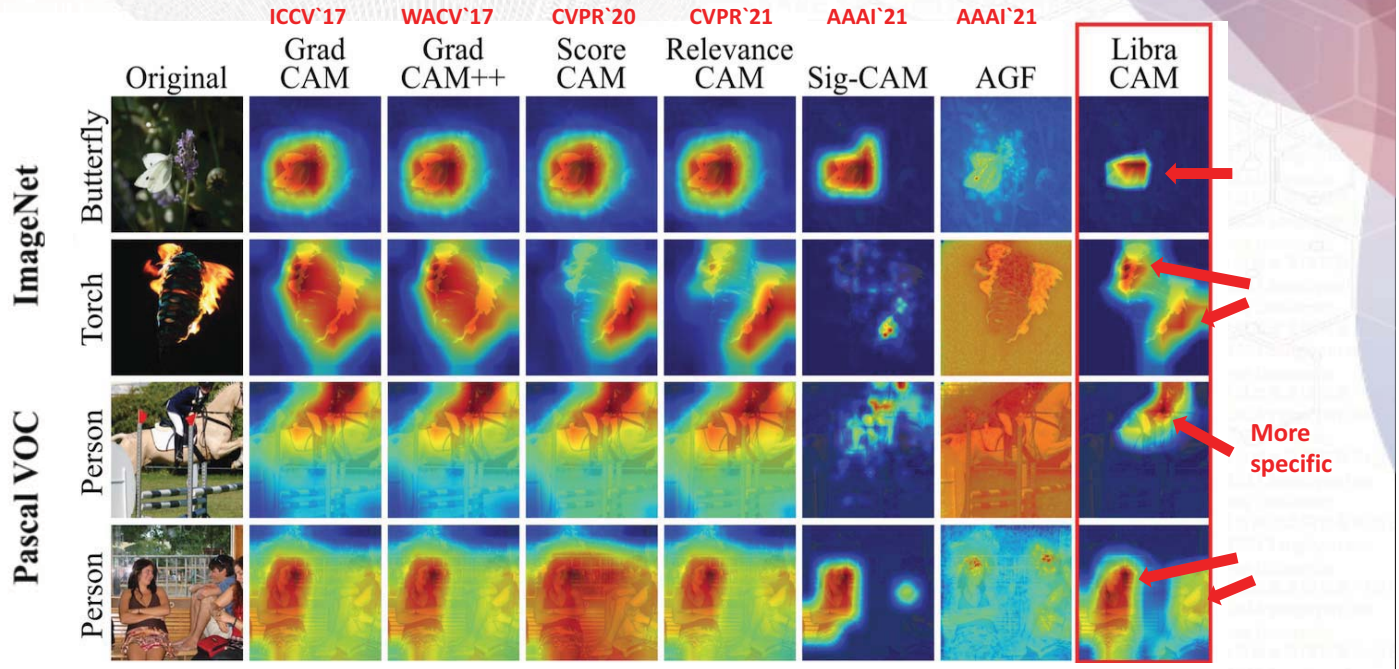
at threshold levels t ∈ [0, 1] with the increment of 0.1 from left to right in all plots.



Qualitative Result (VGG-16)



Qualitative Result (ResNet-50)



Q/A

Model Induction Methods

- Perturbation-Based Methods
  - 입력의 변화에 따른 예측값의 변화로 특정 인자의 중요도를 산출
  - LIME, SHAP, EMP, RISE, XRAI…
- Input Gradient-Based Methods
  - 입력에 대한 출력의 미분치로 입력의 중요도를 산출
  - Guided Backprobagation, SmoothGrad, VarGrad, Integrated Gradients, Guided Integrated Gradients, DeepLIFT, …
- Decomposition Methods
  - 출력에서 보이는 중요도를 입력으로 전달하는 일종의 역전파 알고리즘을 구성
  - LRP, Contrastive LRP, RAP, …
- Activation-Based Methods
  - CNN의 마지막 activation의 민감도를 인자 중요도 산출에 사용
  - CAM, Grad-CAM, Grad-CAM++, Score-CAM, Ablation-CAM, Layer-CAM, …