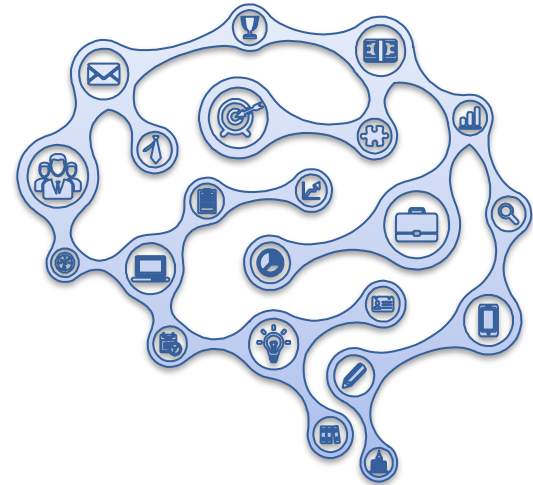


의료 Artificial Intelligence

인공신경망과 딥러닝 (chap11,12)

2022.06.09



오늘 배울 내용 ...

1. 인공신경망과 딥러닝
2. 인공지능 실습
3. mblock 실습

어렵지 않다
쉬운 것도 아니다



인공지능 이론

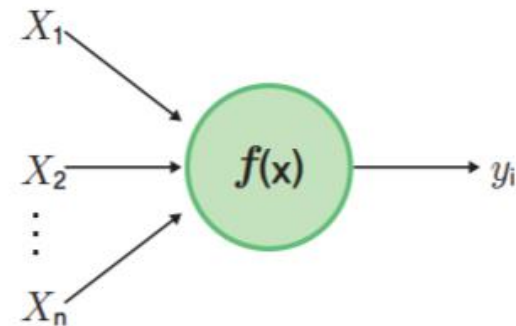
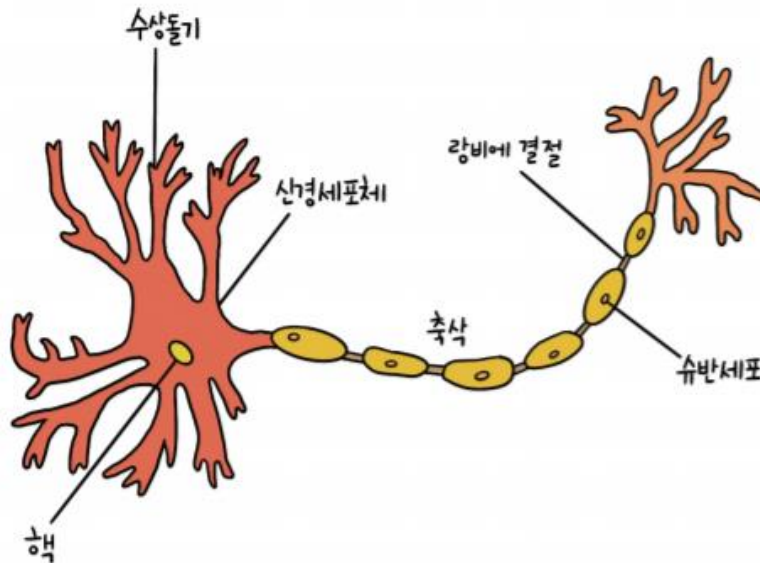
인공 신경망 - 최초 아이디어

- 인간 뇌 신경세포를 복잡한 스위치들이 연결된 네트워크로 표현

: ‘McCulloch-Pitts 뉴런’ 모델

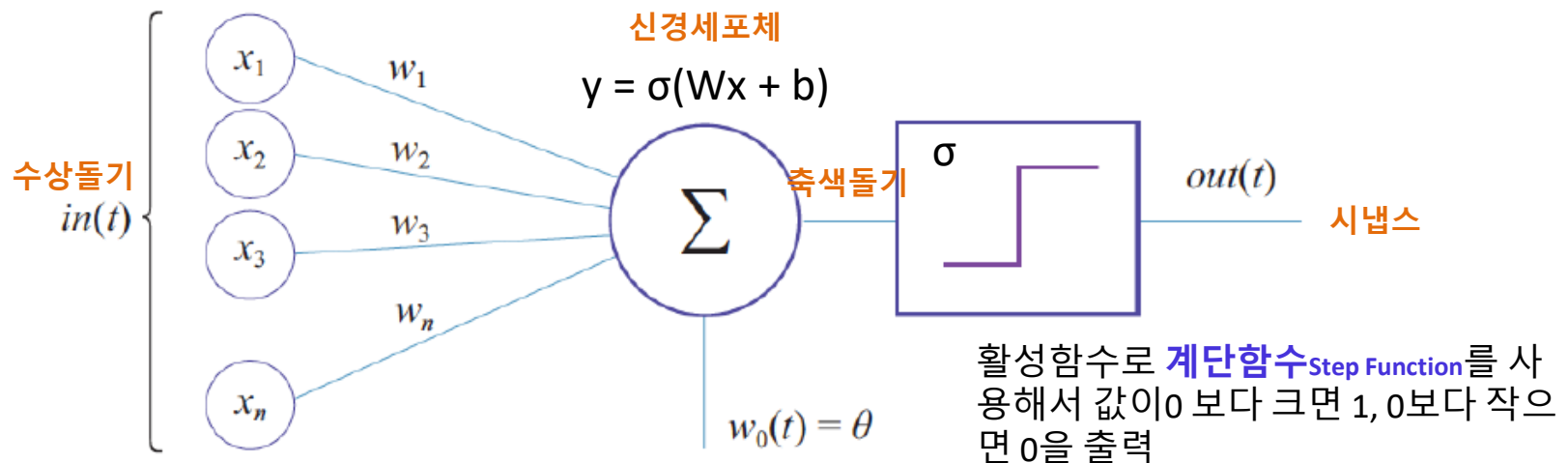
“A logical calculus of the ideas immanent in nervous activity” 논문 (1943)

- “컴퓨터도 인간의 뇌처럼 대량의 병렬처리 연산을 수행하도록 만들면 컴퓨터도 인간이 쉽게 할 수 있는 인지행동을 할 수 있지 않을까?”



인공 신경망 - 퍼셉트론

- 공학적 구현을 최초로 제안한 것은 1958년에 Frank Rosenblatt이 발표한
“The perceptron : A probabilistic model for information storage and organization in the brain” 논문
- 퍼셉트론Perceptron은 생물학적 뉴런을 공학적인 구조로 변형한 그림
입력층Input Layer $in(t)$ 과 출력층Output Layer $out(t)$ 을 가지고 있음
- 퍼셉트론은 입력층에서 인풋데이터 x 를 받고, 이를 가중치 W 와 곱한 후, 이 값에 바이어스Bias b 를 더함.
이 값을 활성화함수 σ 의 입력값으로 대입해서 출력층은 최종적으로 0 또는 1의 값을 출력



인공 신경망 - 가중치

- Perceptron의 가중치(W) : Input의 중요도를 나타낸다
- 입력 데이터 (고려사항 데이터) 수 만큼 가중치(W) 필요

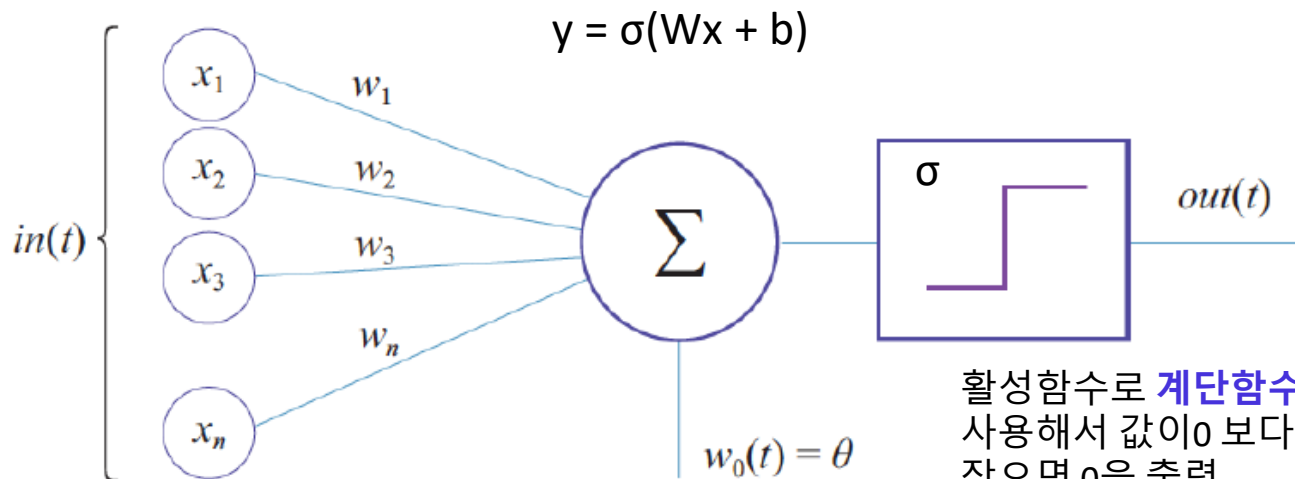
예를 들어, "주말에 집에서 나가 데이트를 할 것인가?"에 대한 의사결정모델을 Perceptron을 이용해서 만든다

이때 의사결정 고려사항은 다음과 같이 가정

1. 날씨가 좋은가? (W1)
2. 이성 친구가 바쁜 일이 없는가? (W2)
3. 데이트 장소가 집에서 가까운가? (W3)

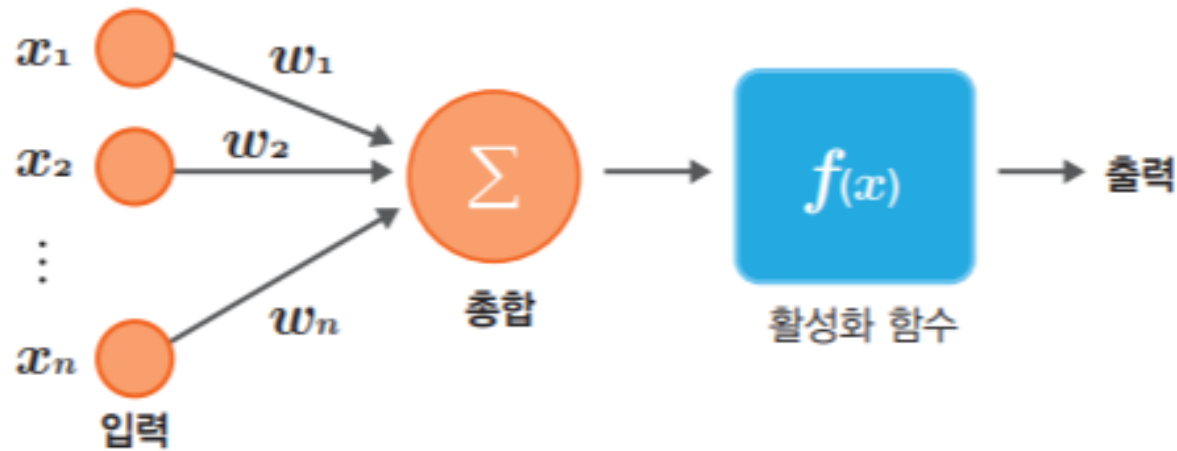
세 가지 고려사항 데이터를 Input(x)으로 넣음. 날씨를 가장 중요하게 고려한다면 W1=6, W2=2, W3=2 가중치 부여

데이트 장소가 집에서 가까운 것을 가장 중요하게 고려한다면 W1=2, W2=2, W3=6 가중치 부여



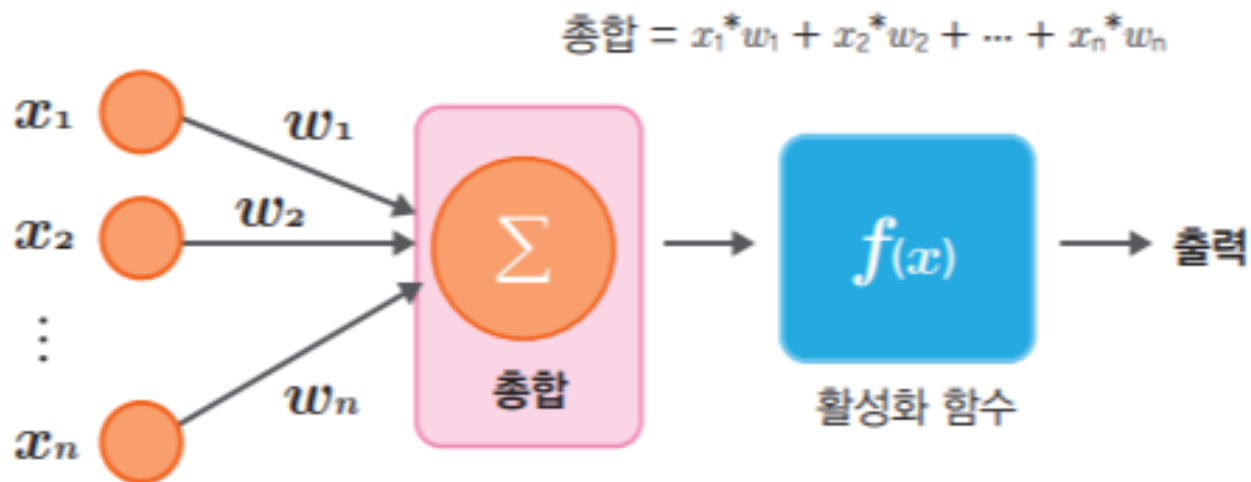
인공 신경망 계산 방법

1. 입력 층에서 데이터 값을 입력 받음
2. 입력의 중요도에 따라 가중 값으로 크기를 변경



인공 신경망 계산 방법

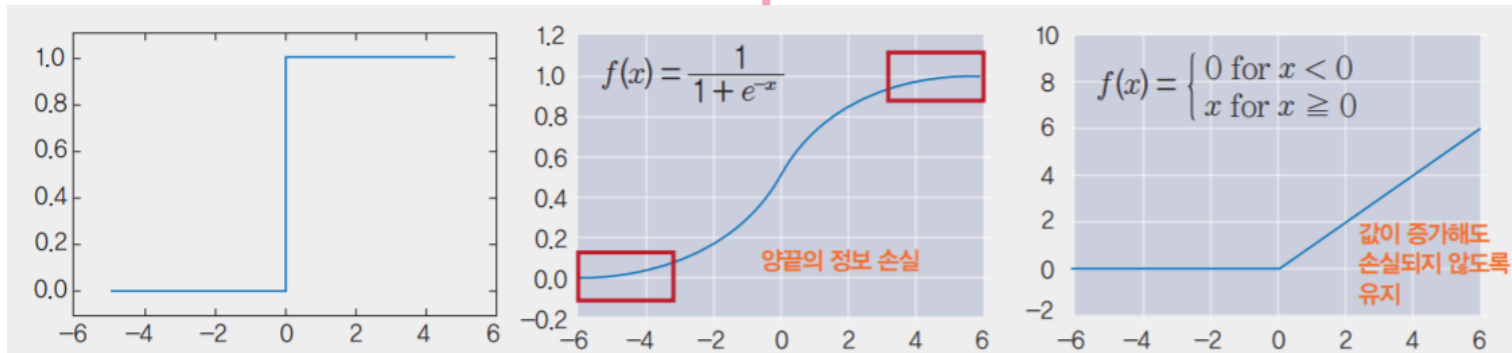
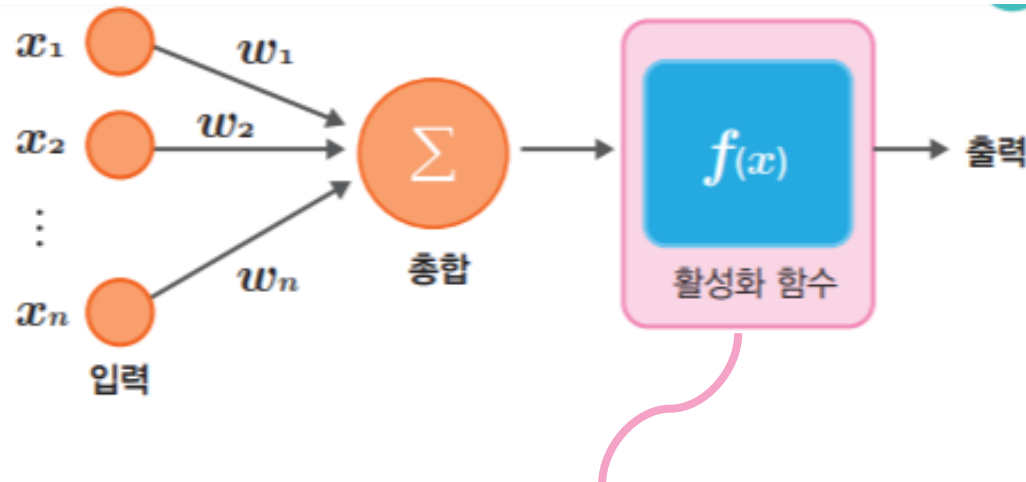
3. 입력 x 와 가중 값 w 를 곱하고 모두 더함



인공 신경망 계산 방법 - 활성화함수

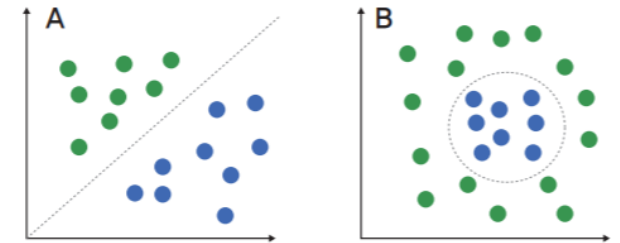
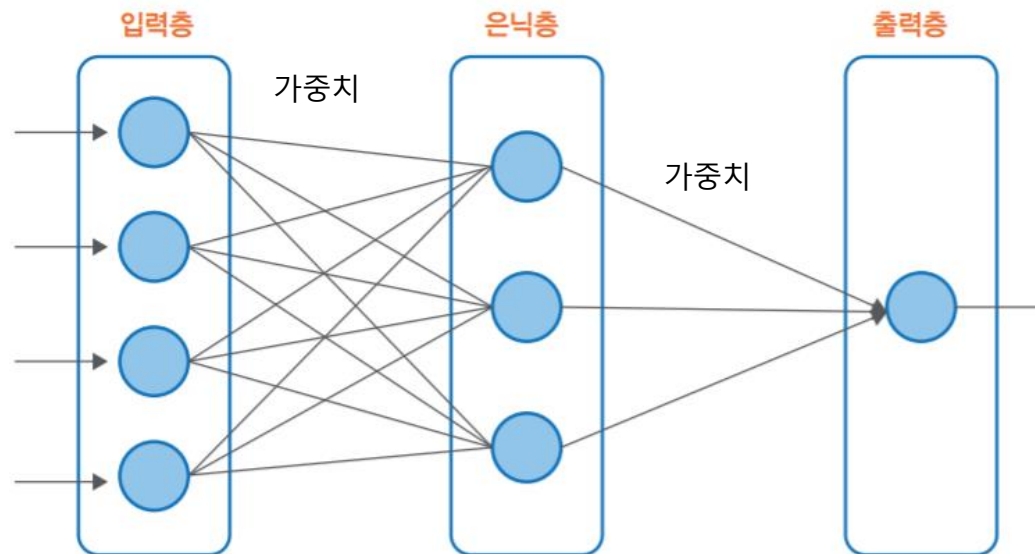
4. 합한 값을 활성화 함수로 출력값을 계산

(계단함수, 시그모이드, ReLU, 탄젠트 함수 등을 활용)



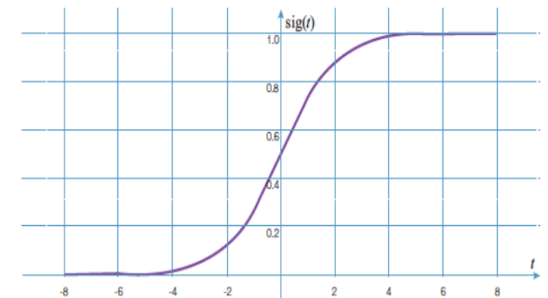
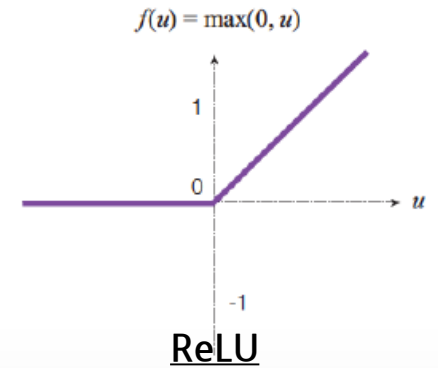
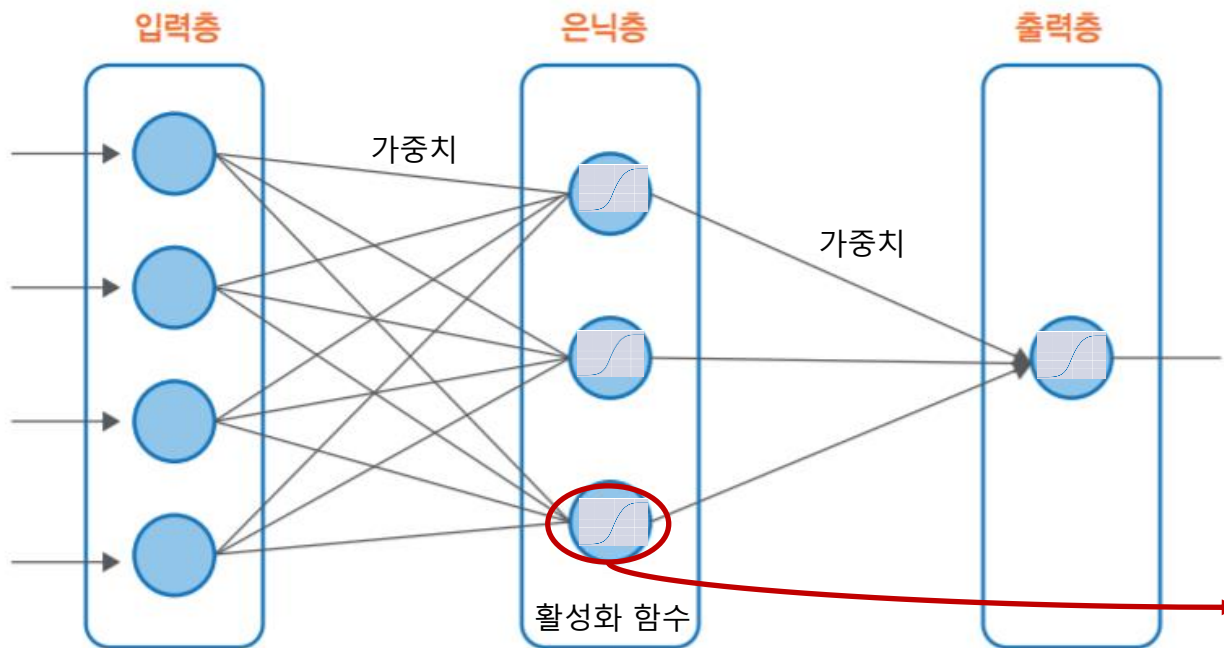
다층 퍼셉트론 (MLP)

- 퍼셉트론을 여러 층 쌓아 올린 다층퍼셉트론 Multi-Layer Perceptron(MLP)
= 인공신경망(Artificial Neural Networks)
- 다층 퍼셉트론의 구조는 입력층Input Layer과 은닉층Hidden Layer, 출력층Output Layer으로 구성
- 은닉층은 데이터의 입출력 과정에서 직접적으로 보이지 않지만 숨겨진 특징을 학습하는 역할을 함



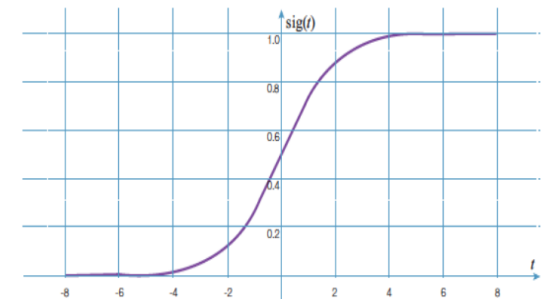
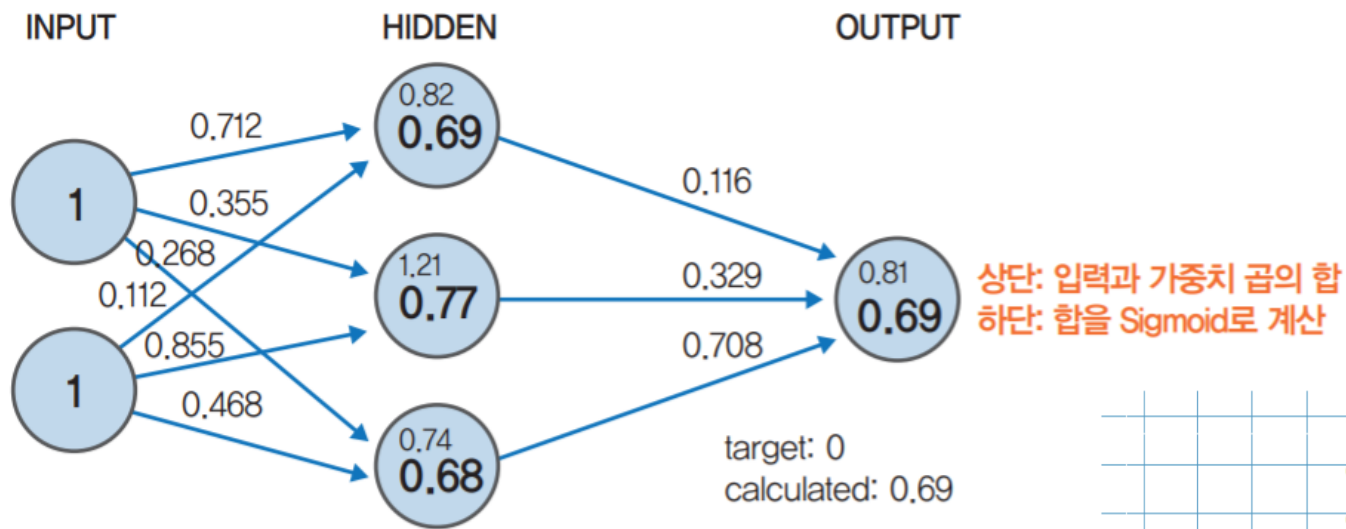
다층 퍼셉트론 (MLP)

- 활성화함수로서 단일 퍼셉트론에서는 계단함수를 사용하여 출력값을 0,1로 이진분류 형태로 만드는 것이 목적이었지만,
MLP 구조에서는 분류기가 비선형적인 Non-Linear 특징을 학습할 수 있도록 만드는 것이 목적
- 활성화함수로 비선형함수 인 시그모이드Sigmoid와 ReLU를 주로 사용
과거에는 sigmoid 함수를 많이 사용했지만,최근에는 ReLU가 딥러닝 학습에 더 적합하다고 알려져서 ReLU를 많이 사용



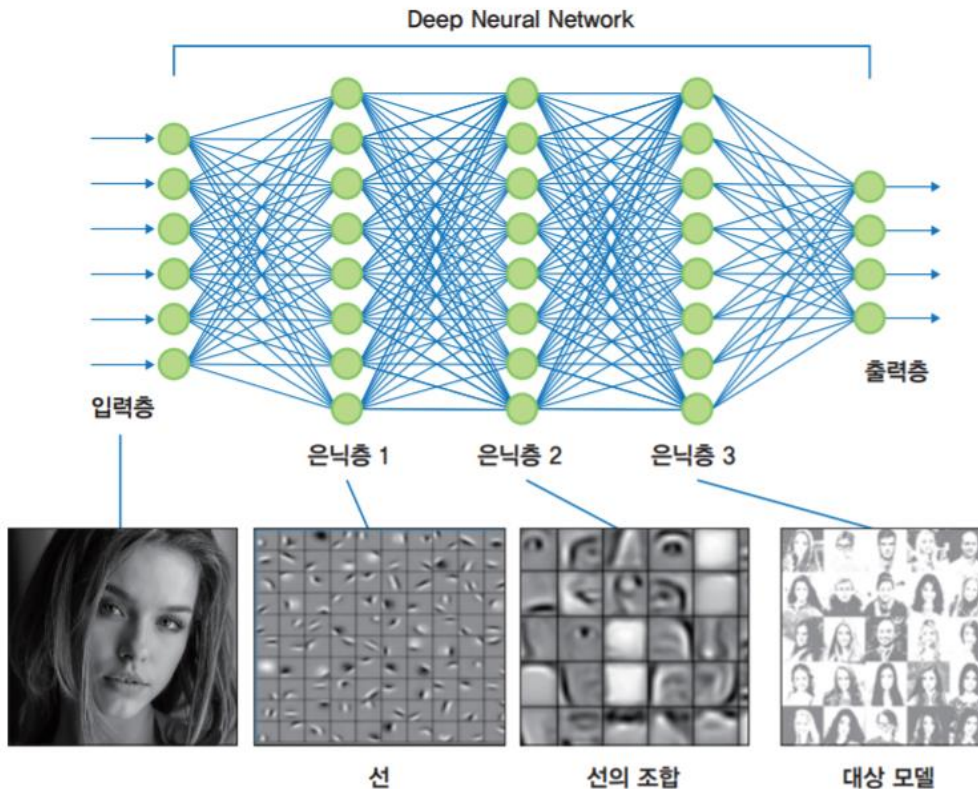
다층 퍼셉트론 계산 방법

여러 개의 입력값과 가중치 곱의 합을 활성화 함수로 계산해 여러 개 뉴런에서 출력



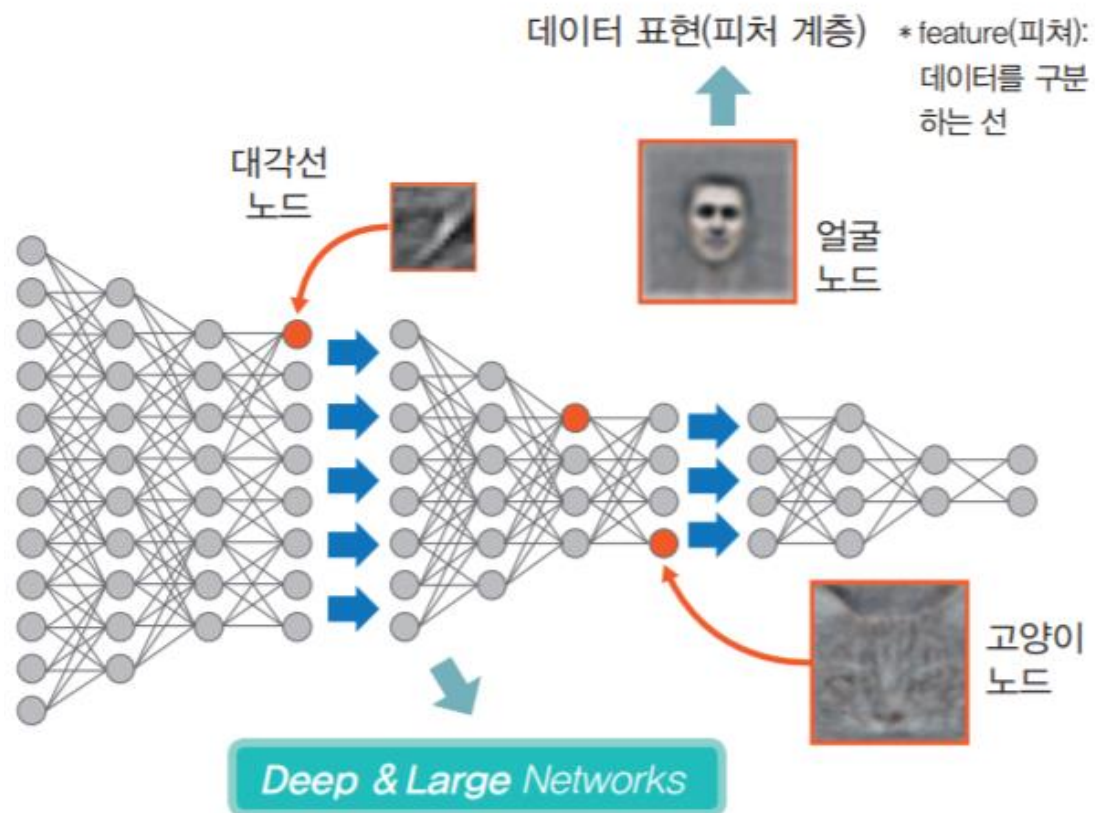
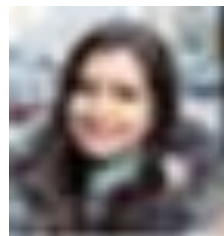
딥러닝 (Deep Learning)

- 딥러닝(Deep Learning) : 머신러닝 기법 중 하나인 인공신경망(Artificial Neural Networks)기법의 은닉층(Hidden Layer)을 깊게 쌓은 구조를 이용해 학습하는 기법
- 딥러닝의 장점 : 데이터의 특징을 단계별로 추상화를 높여 가면서 학습할 수 있음
- 얇은 은닉층은 점,선,면과 같은 추상화 단계가 낮은 특징을 학습하고, 깊은 은닉층은 얼굴의 눈,코,입 등 추상화 단계가 높은 특징을 학습한다. (세부 특징 → 상위 특징(세부 특징의 조합))
- 딥러닝을 사용할 경우 사람처럼(추상화 단계가 높은 특징을 사용해서 판단) 고차원적 인지활동을 수행할 수 있음



기존 신경망과 비교해서
은닉층과 출력층이 2개 이상

딥러닝의 적용



딥러닝의 적용

- 사람과 유사한 수준의 이미지 분류, 영상 인식
- 사람과 유사한 수준의 음성 인식
- 사람과 유사한 수준의 필기 인식
- 향상된 기계 자동 번역
- 향상된 TTS(Text-To-Speech) 변환
- 구글 나우, 아마존 알렉사와 같은 디지털 비서
- 사람과 유사한 수준의 자율주행 능력
- 구글, 바이두(Baidu), 빙(Bing)에서 사용하는 광고 타겟팅
- 향상된 웹 검색 엔진의 결과
- 자연어 질문에 대답하는 능력
- 사람을 능가하는 바둑과 게임 실력



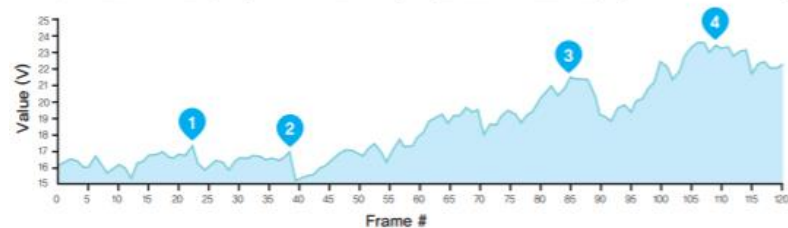
▲ 시리



▲ 자율주행 자동차



▲ 네이버 클로바 디지털 비서



▲ 사람을 능가하는 게임 실력

GPU 성능 향상으로 인한 연산 시간의 단축과 대량으로 쏟아져 나오는 빅데이터를 학습에 이용하게 되면서 획기적인 발전을 이룬

딥러닝과 이미지 인식

- 이미지 속의 사물의 종류를 인식하는 것을 넘어
- 영상과 다중 이미지 속의 상황을 이해하기 시작함



▲ 영상 이해(출처: fredrik Gustafsson, Neural Image Captioning for Intelligent Vehicle-to-Passenger Communication, Department of Electrical Engineering Stanford University)

딥러닝과 이미지 인식

- 딥러닝이 잘 동작하는 문제영역:

이미지나 자연어, 음성 등의 비정형화된 대량의 데이터로부터 인식(Recognition)을 수행하는 문제영역에 잘 동작

- 딥러닝이 잘 동작하지 않는 문제영역:

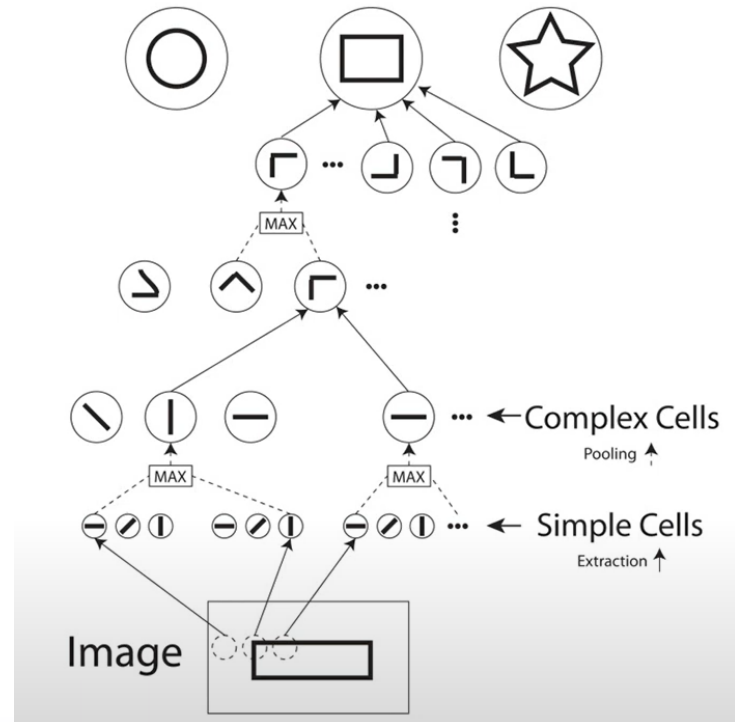
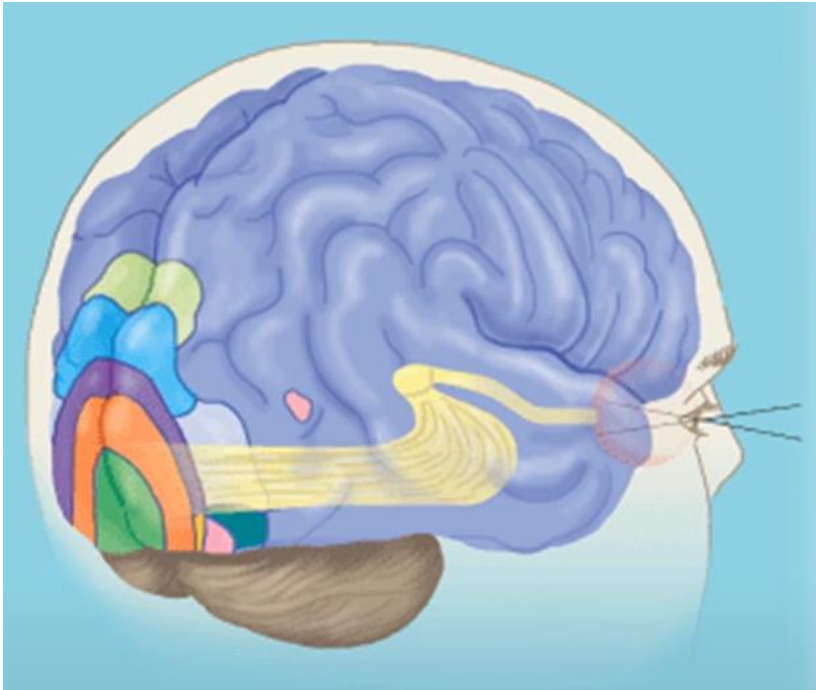
데이터가 부족하거나, 정형화 된 데이터에 대해서는 상대적으로 잘 동작하지 않음.

이런 상황에서는 딥러닝 알고리즘 외에 다른 알고리즘을 사용하는 것을 추천

합성곱 신경망 (Convolution Neural Network)

합성곱 신경망 (Convolution Neural Network)

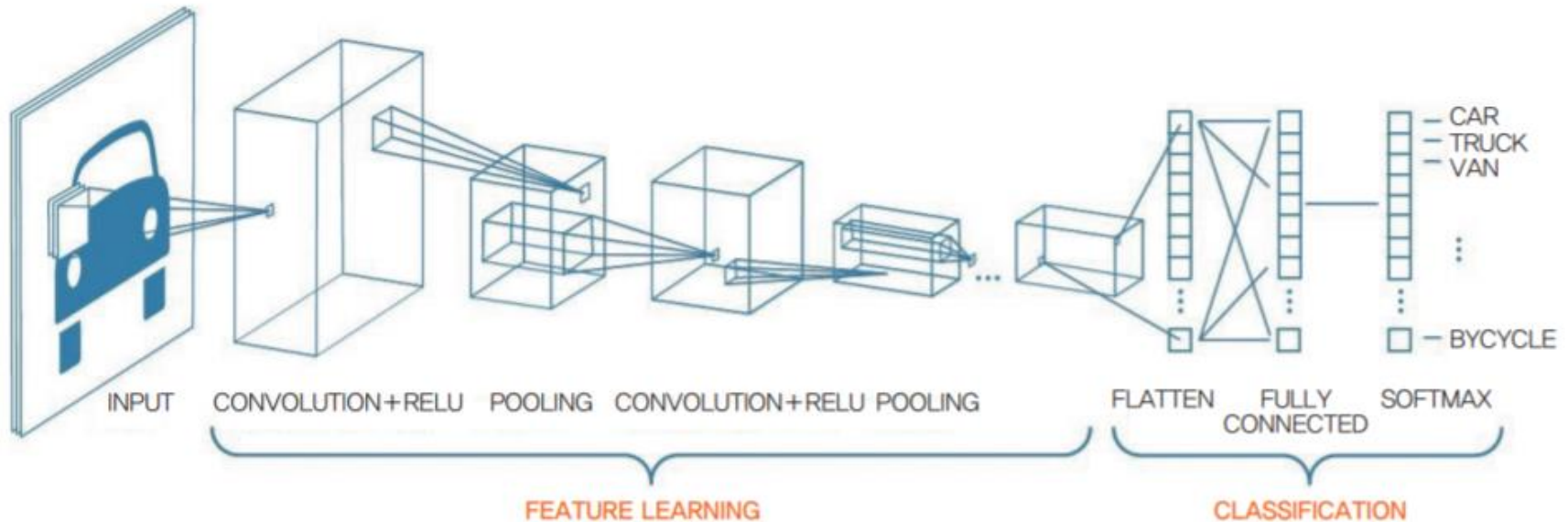
- 뉴런들의 연결 패턴을 구성한 전방향 인공 신경망으로 하위 레이어들에서 전처리를 수행하도록 설계된 딥러닝 신경망 모델
- 동물의 시각 피질의 구조에서 영감을 받아 만들어 짐
 - : 시각자극이 1차 시각피질 → 2차시각 피질 → 3차 시각피질 등을 거치면서 계층적으로 정보 처리 (추상적 특징 처리)



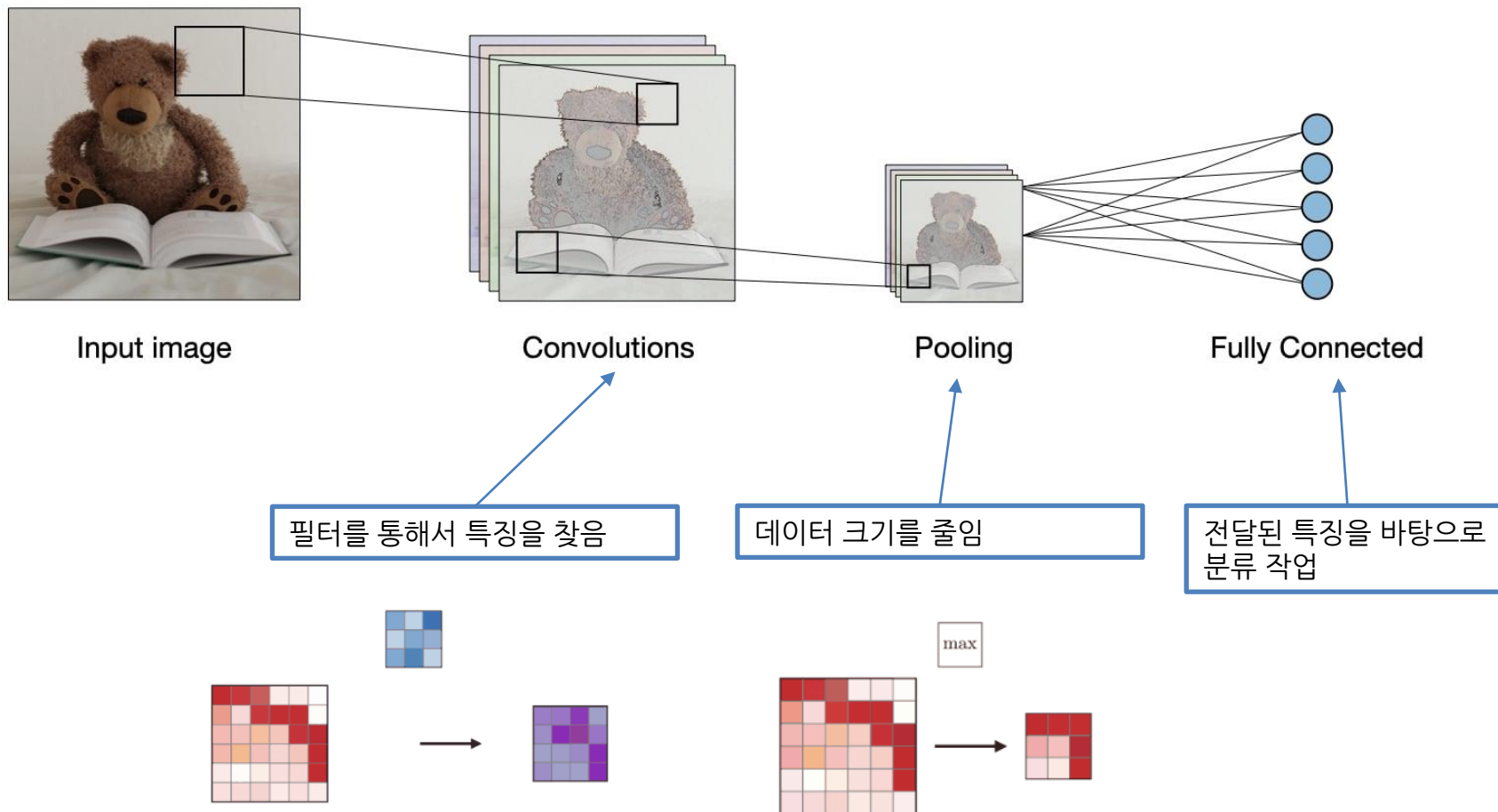
합성곱 신경망 (Convolution Neural Network)

합성곱 신경망 (Convolution Neural Network) :

- 전반부 : 컨볼루션 연산을 수행하여 **특징 추출**
- 후반부 : 특징을 이용하여 **분류**
- * 영상분류, 문자인식 등 인식문제에 높은 성능을 보임



합성곱 신경망 처리 과정



합성곱 신경망 처리 과정

1. Convolutions

컨볼루션 레이어

컨볼루션 연산을 통해 특징들을 추출하는 레이어를 말한다. 입력되는 이미지의 채널 개수만큼 필터가 존재하며 각 채널에 할당된 필터를 적용해 합성곱 계층의 출력 이미지가 생성된다.

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 *

1	0	1
1	2	0
3	0	1

 =

40	

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 *

1	0	1
1	2	0
3	0	1

 =

40	32
26	

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 *

1	0	1
1	2	0
3	0	1

 =

40	32

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 *

1	0	1
1	2	0
3	0	1

 =

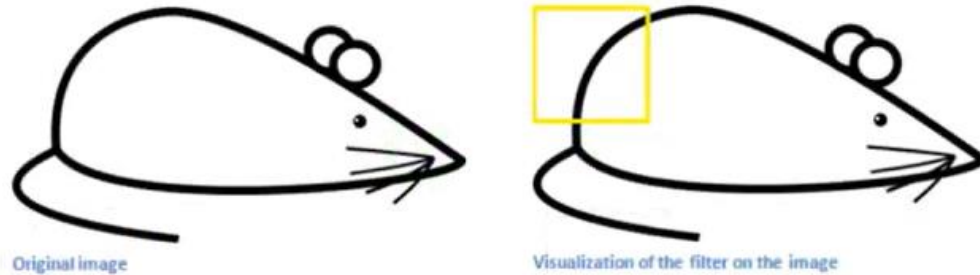
40	32
26	25

▲ 하나의 채널에 대한 합성곱 계층의 동작

합성곱 신경망 처리 과정

필터 : 주어진 데이터에 찾으려는 특징이 데이터에 있는 지 검출

- 필터와 이미지 데이터를 합성곱하여 이미지에 있는 특징을 찾음



필터



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

*

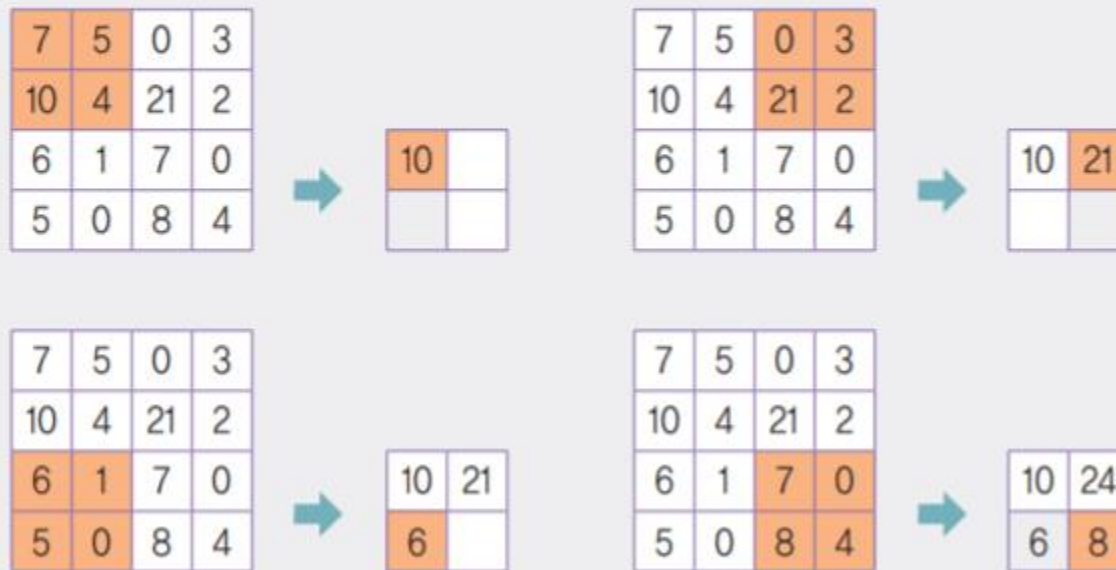
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

합성곱 신경망 처리 과정

2. Pooling(down sampling)

풀링 레이어(Pooling layer)

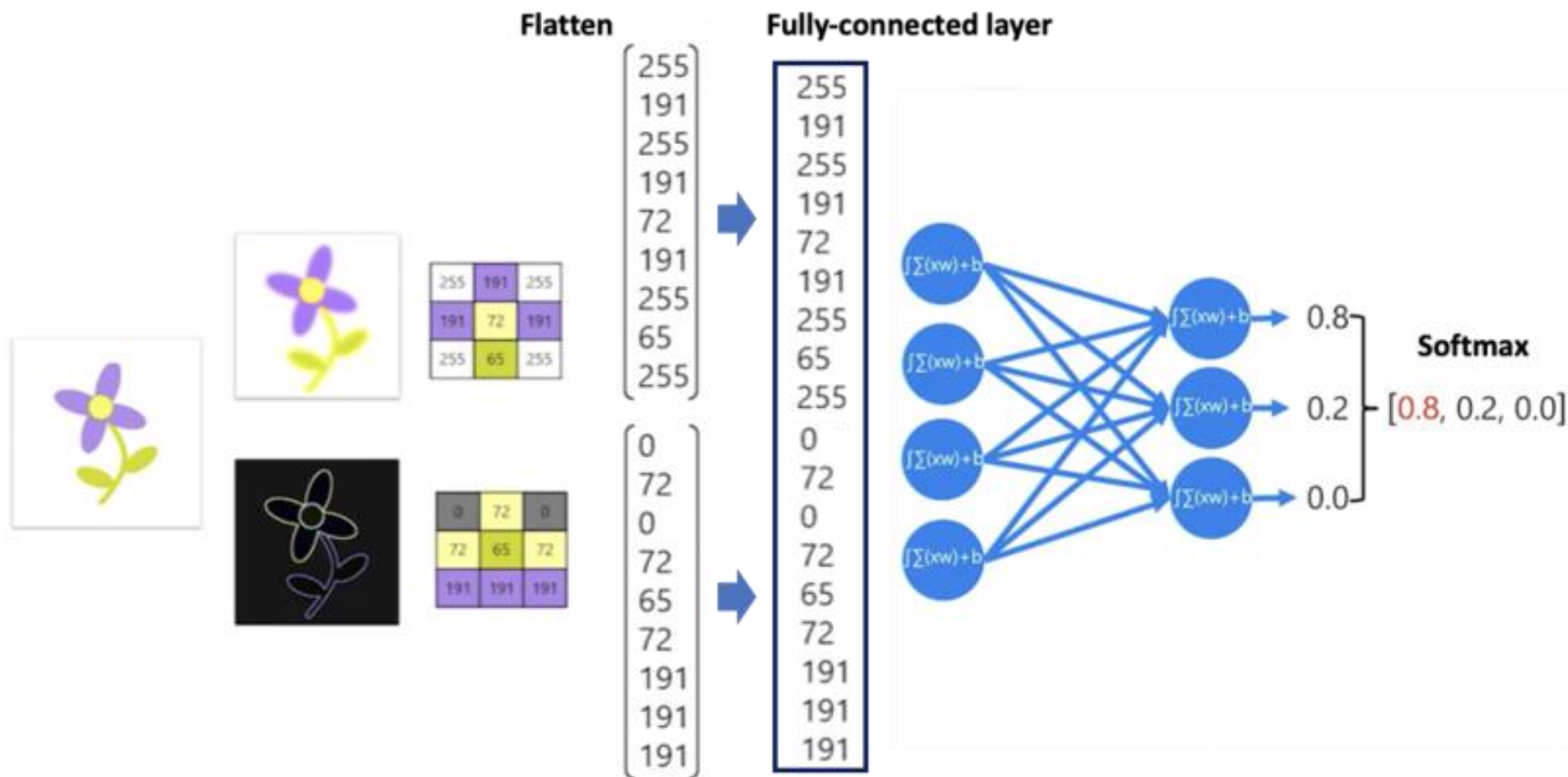
입력 공간을 추상화하는 레이어를 말한다. 예를 들면 영상 데이터의 경우 픽셀의 수가 많으면 서브 샘플링(sub-sampling) 등의 과정을 통해 차원 축소의 효과를 얻는다. 주로 맥스 풀링(max-pooling)을 기반으로 구현된다.



▲ Max-pooling 기반 풀링 계층의 동작

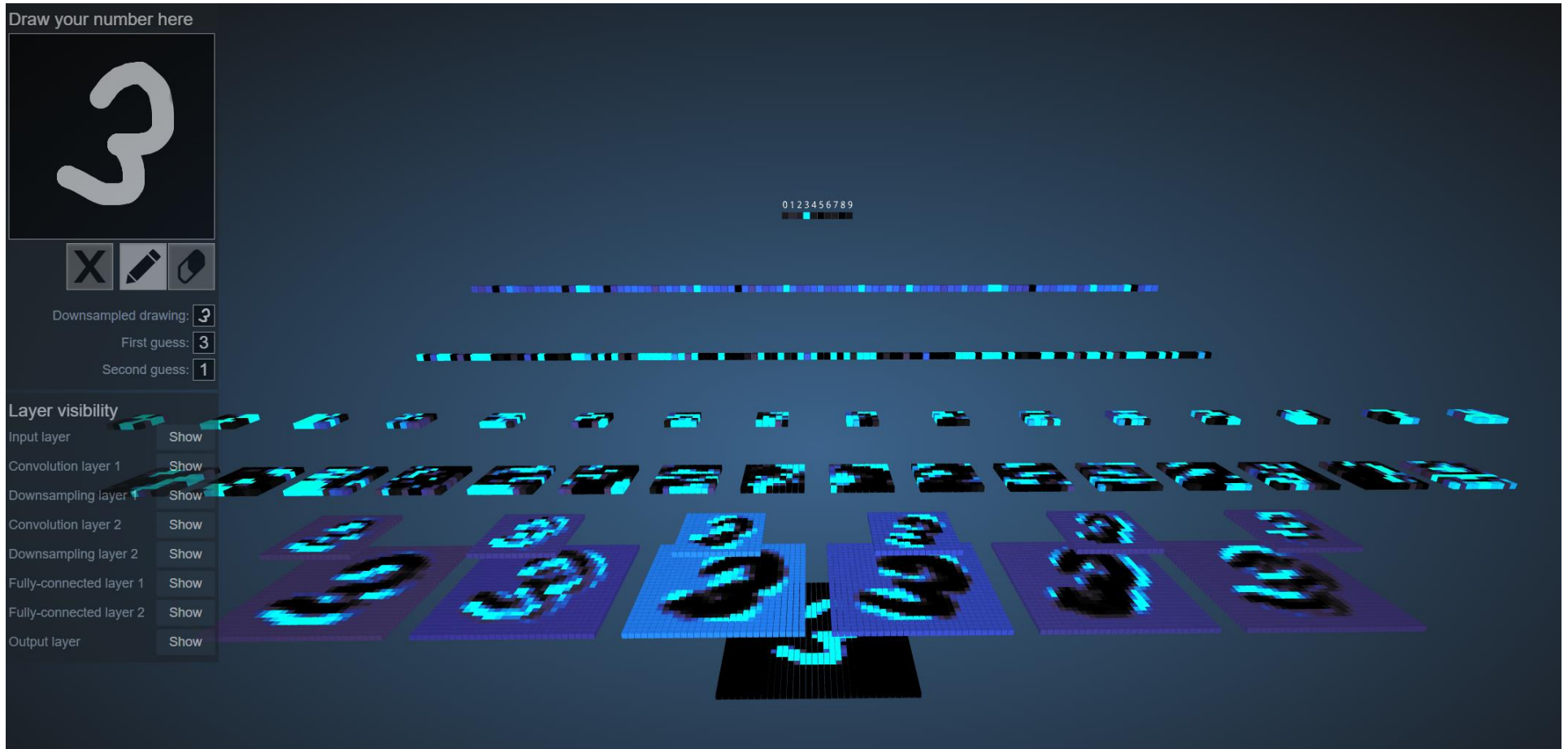
합성곱 신경망 처리 과정

3. 분류 및 회귀 : Fully Connected

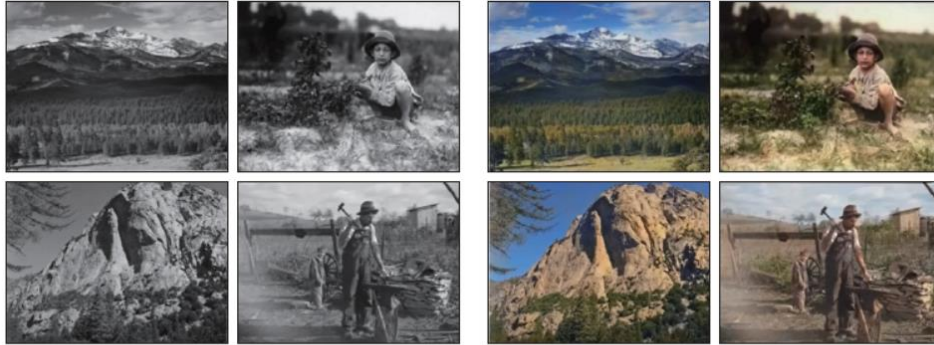


합성곱 신경망 과정 보기

<http://www.cs.cmu.edu/~aharley/vis/conv/>



합성곱 신경망 활용



▲ 출처: https://www.youtube.com/watch?time_continue=3&v=ys5nMO4Q0iY

CNN기술을 적용해 기존 흑백사진과 영화에 입혀 컬러영화로 새롭게 탄생

CCTV를 이용해 사람들의 실시간 행동을 분석



▲ 출처: <https://www.youtube.com/watch?v=xhp47v50BXQ>

인터넷상에서 플래쉬몹과 유사한 애니메이션을 만들기 위해 실제 플래쉬몹에 참여한 사람들의 움직임을 분석



▲ 출처: <https://www.youtube.com/watch?v=pW6nZxeWIGM>

인공지능 실습

인공지능 실습

<https://www.autodraw.com/>



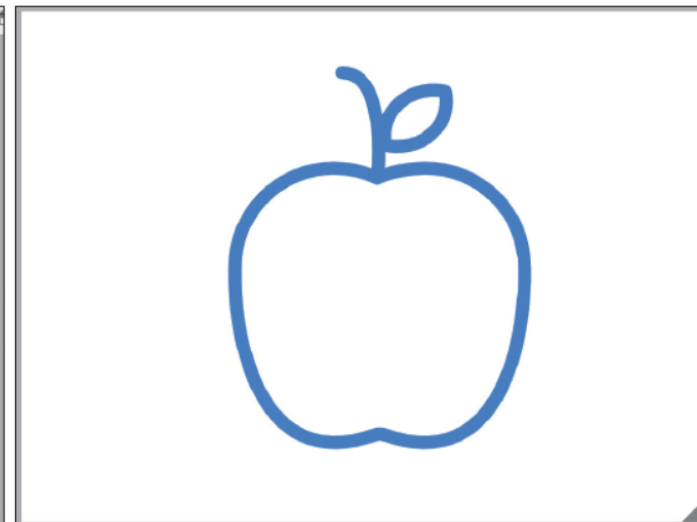
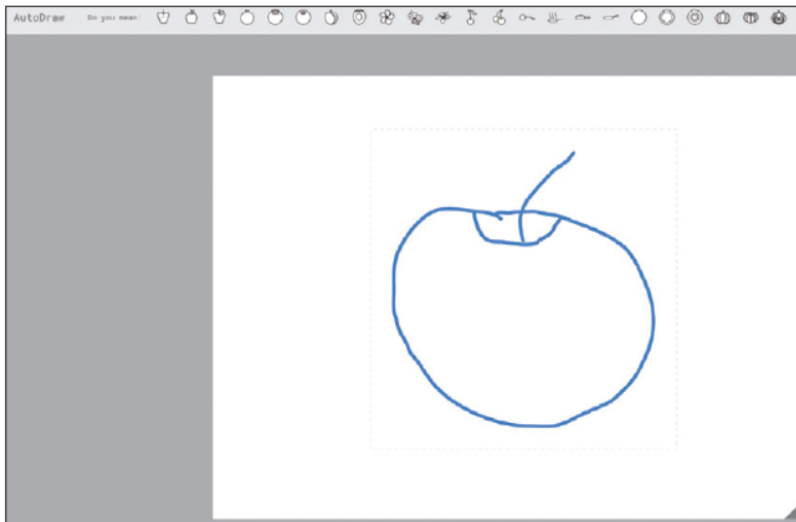
AutoDraw

Fast drawing for everyone.

그리기 시작

빠른 사용법*

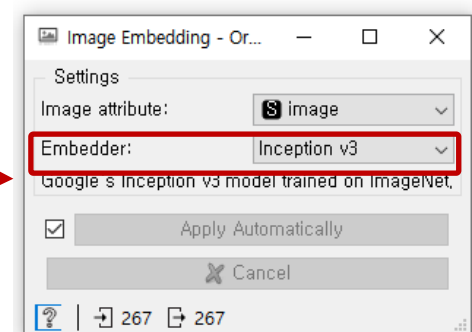
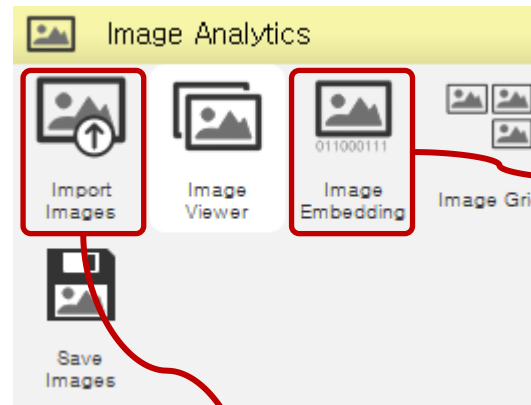
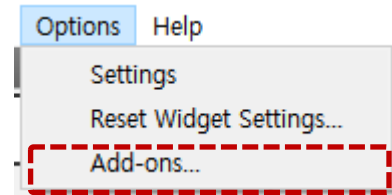
Do you mean:



인공지능 실습

실습 1 : 신경망을 이용한 이미지 분류

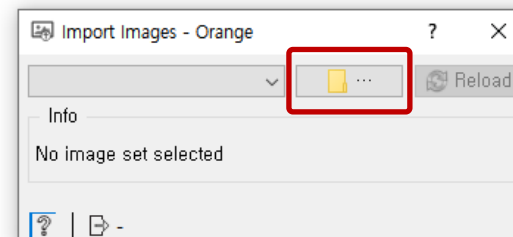
- 데이터 모델 : Image Analytics



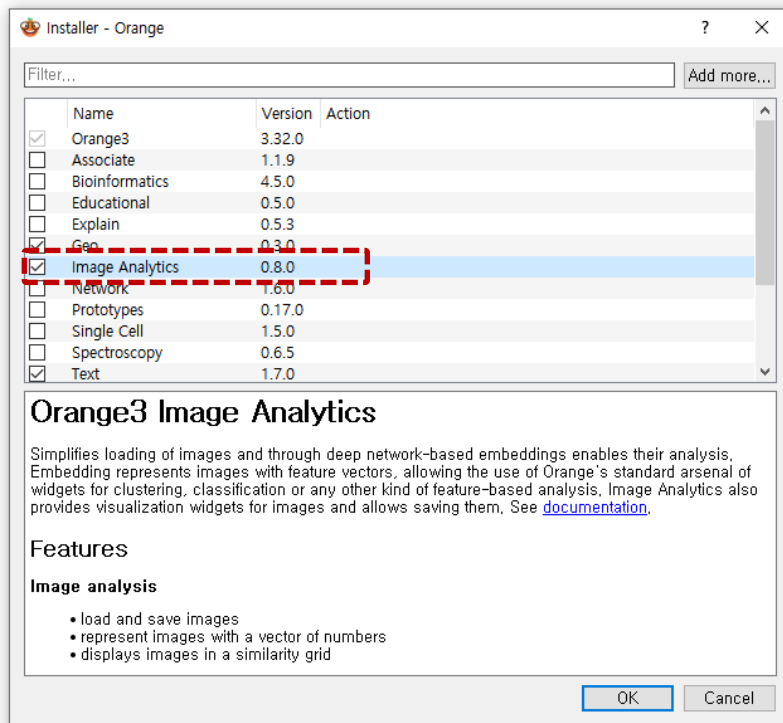
이미지 전처리 작업

: 이미지 데이터를 분석에 적합한
형태로 가공 (벡터화)
: 딥러닝을 이용하여 특징 추출
(인터넷 연결 필요)

- inception v3 선택
- SqueezeNet (인터넷 미연결시)

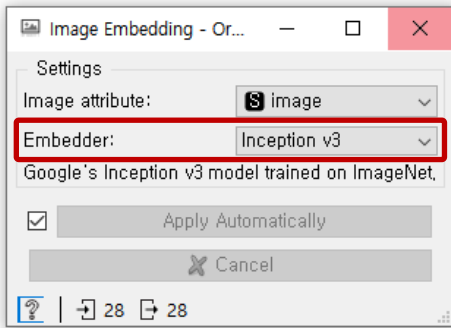
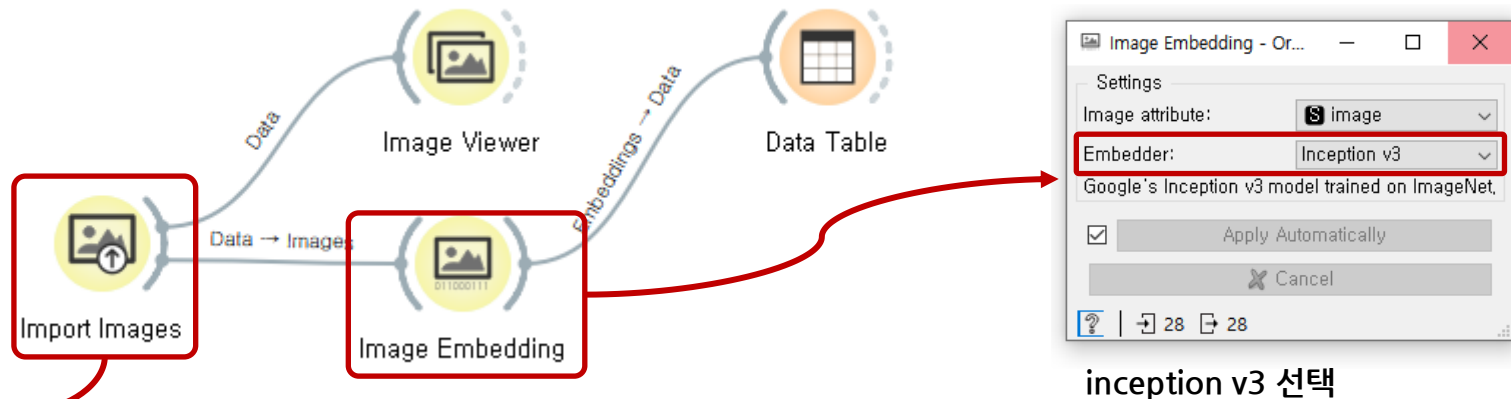


Training 폴더 선택

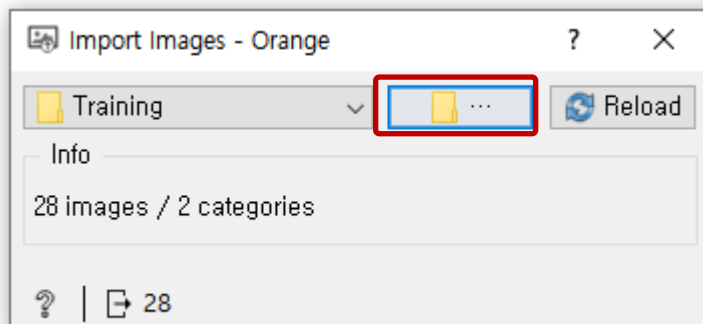


인공지능 실습

1) 이미지 데이터 읽기 : Dog Dataset - 'Training'



inception v3 선택



Training 폴더 선택

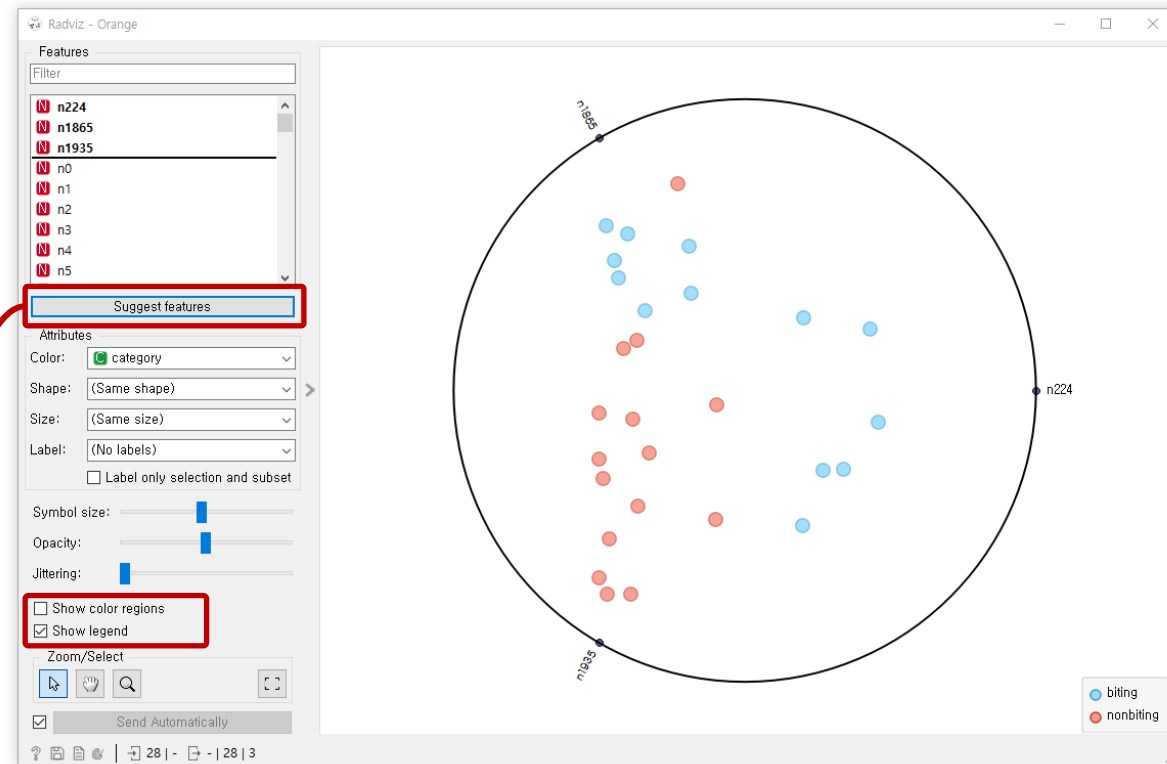
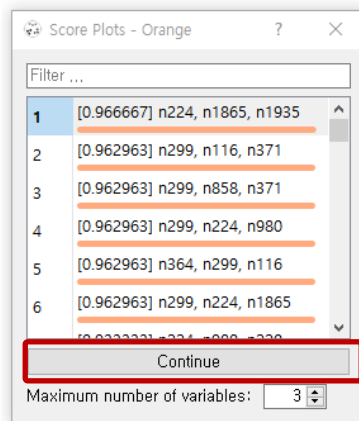
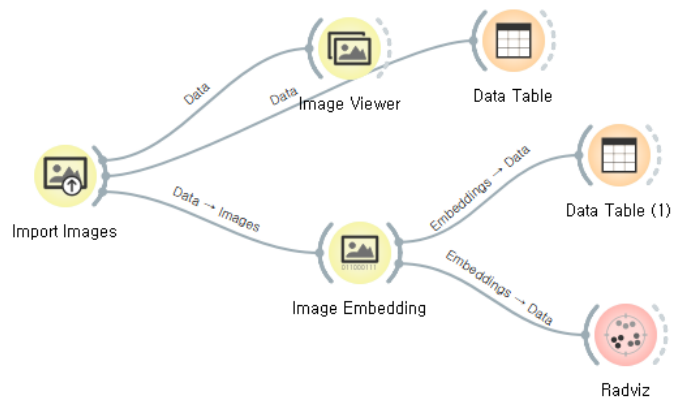
* 2048개 특징 추출

The screenshot shows the 'Data Table (1) - Orange' window. It displays a table with 2048 features extracted from the images. The table has columns for 'hidden origin name' and various feature names (n2038, n2039, n2040, n2041, n2042, n2043, n2044, n2045, n2046, n2047). The first row shows the values for these features for the first instance.

hidden origin name	n2038 True	n2039 True	n2040 True	n2041 True	n2042 True	n2043 True	n2044 True	n2045 True	n2046 True	n2047 True
1	1.45107	1.28922	1.0025	0	0.957372	0.150402	0.319577	0.164622	0.610624	0.181107
2	0.59201	0.783973	1.25194	0.0923097	1.00489	0.452998	0.306797	0.0633911	0.261035	0.350738
3	0.457493	0.710513	1.24759	0.0762569	0.826199	0.259089	0.282513	0.0419464	0.450901	0.476047
4	0.555999	0.595953	1.19195	0.303339	0.815815	0.253301	0.230144	0.424374	0.680997	0.751758
5	0.476643	0.60967	1.30141	0.127445	0.708539	0.468226	0.271128	0.120908	0.732463	0.329509
6	0.717629	1.02385	1.08534	0	1.22171	0.169689	0.236921	0.304574	1.09688	0.252783
7	1.14208	1.14438	1.30183	0	0.85049	0.196166	0.247028	0.0638952	0.416768	0.229296
8	1.51869	1.11793	0.576036	0.219123	0.782458	0.161359	0.46252	0.571362	0.464587	0.211698
9	0.690075	0.446996	0.542317	0.059243	1.04785	0.177531	0.1516	0.290961	0.908031	0.172723
10	0.744465	1.07832	0.200391	0.256496	0.429983	0.0750902	0.227882	0.709565	0.758177	1.09769
11	0.503825	0.683267	1.04935	0.117224	0.697197	0.194308	0.269141	0.195571	0.66449	0.508929
12	1.0135	0.786323	1.16373	0.164245	1.39167	0.368238	0.650926	0.112411	0.27268	0.149133
13	0.79736	0.717456	0.892112	0.0720466	1.2646	0.305075	0.348569	0.142569	0.364018	0.193356
14	0.497312	0.529561	1.35374	0.0954138	0.670197	0.741116	0.364518	0.11885	0.444099	0.295936
15	0.795993	0.778787	1.22809	0.0640778	0.803792	0.233956	0.120514	0.600265	1.1006	0.708568
16	1.05888	0.80648	0.826742	0	0.699987	0.0856347	0.191468	0.403788	0.755427	0.207575
17	1.51289	0.971708	0.996986	0	0.586851	0.122786	0.283775	0.451921	0.641333	0.425136
18	1.53906	1.19727	0.6565	0.000751968	0.496325	0.0612442	0.492492	0.672967	0.828384	0.276728

인공지능 실습

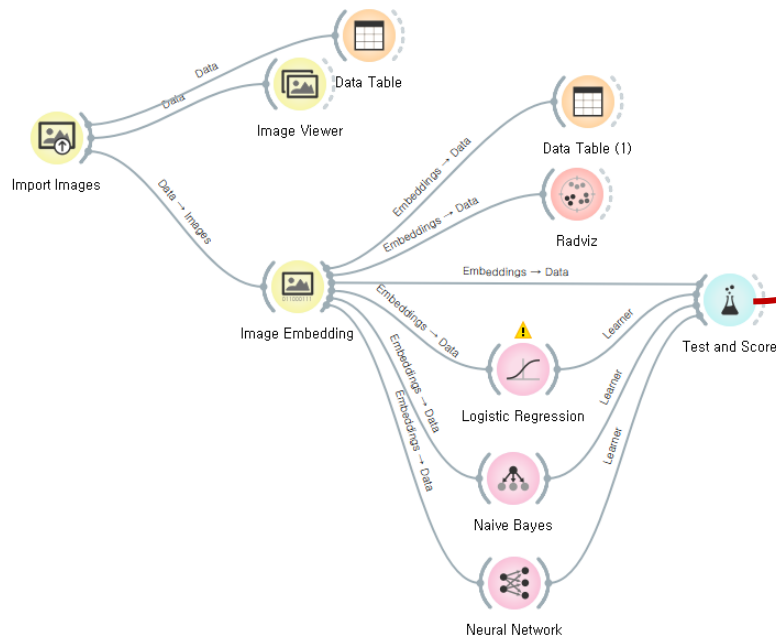
2) 어떤 속성이 분류를 잘 해내는 가 찾기



inception v3 선택

인공지능 실습

3) 문제에 적합한 모델 찾기



Test and Score - Orange

☒ Cross validation
Number of folds: 10
☒ Stratified

☐ Cross validation by feature
☐ Random sampling
Repeat train/test: 10
Training set size: 80 %
☐ Stratified

☐ Leave one out
☐ Test on train data
☐ Test on test data

Evaluation results for target (None, show average over classes)

Model	AUC	CA	F1	Precision	Recall
Neural Network	0.892	0.786	0.786	0.786	0.786
Naive Bayes	0.728	0.679	0.679	0.681	0.679
Logistic Regression	0.641	0.571	0.571	0.571	0.571

Compare models by: Area under F
☐ Negligible diff.: 0.1

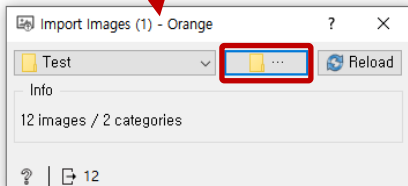
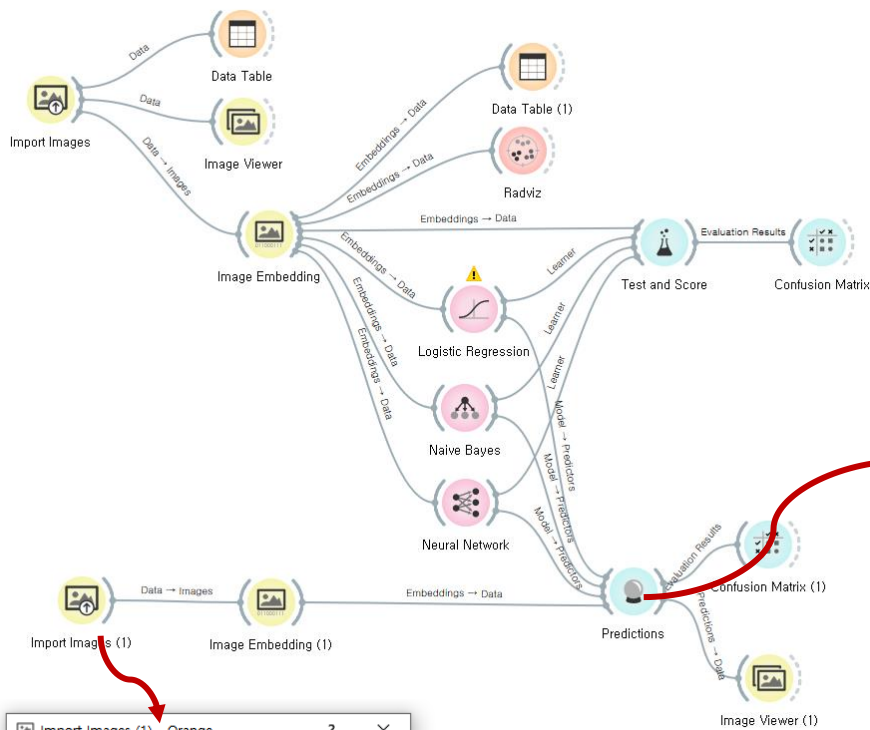
	Neural Netw...	Naive Bayes	Logistic Repr...
Neural Network		0.836	0.500
Naive Bayes	0.164		0.164
Logistic Regression	0.500	0.836	

Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.

? | 28 | - | 28 | 3x28

인공지능 실습

4) 테스트 데이터 적용하기 : Dog Dataset - 'Test'



Test 폴더 선택

Predictions - Orange

Show probabilities for: Classes in data

Restore Original Order

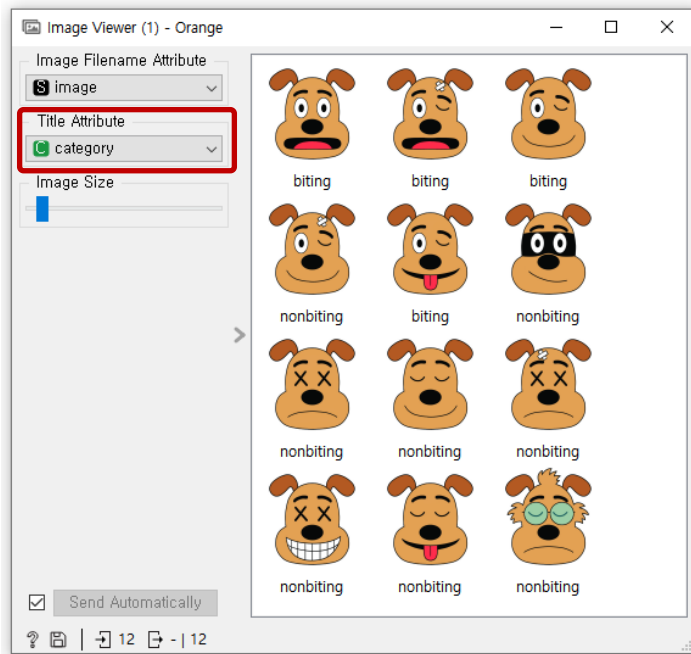
	Logistic Regression	Naive Bayes	Neural Network	category	image name	image
1	0.90 : 0.10 → biting	1.00 : 0.00 → biting	0.97 : 0.03 → biting	biting	d13	biting#d13.jpg
2	0.70 : 0.30 → biting	1.00 : 0.00 → biting	0.79 : 0.21 → biting	biting	d18	biting#d18.jpg
3	0.87 : 0.13 → biting	1.00 : 0.00 → biting	0.96 : 0.04 → biting	biting	d34	biting#d34.jpg
4	0.77 : 0.23 → biting	1.00 : 0.00 → biting	0.87 : 0.13 → biting	biting	d6	biting#d6.jpg
5	0.28 : 0.72 → nonbit...	0.00 : 1.00 → nonbit...	0.17 : 0.83 → nonbit...	nonbiting	d11	nonbiting#d11.jpg
6	0.31 : 0.69 → nonbit...	0.00 : 1.00 → nonbit...	0.08 : 0.92 → nonbit...	nonbiting	d20	nonbiting#d20.jpg
7	0.62 : 0.38 → biting	1.00 : 0.00 → biting	0.33 : 0.67 → nonbit...	nonbiting	d21	nonbiting#d21.jpg
8	0.73 : 0.27 → biting	1.00 : 0.00 → biting	0.79 : 0.21 → biting	nonbiting	d26	nonbiting#d26.jpg
9	0.28 : 0.72 → nonbit...	0.00 : 1.00 → nonbit...	0.10 : 0.90 → nonbit...	nonbiting	d3	nonbiting#d3.jpg
10	0.21 : 0.79 → nonbit...	0.00 : 1.00 → nonbit...	0.12 : 0.88 → nonbit...	nonbiting	d31	nonbiting#d31.jpg
11	0.25 : 0.75 → nonbit...	0.00 : 1.00 → nonbit...	0.05 : 0.95 → nonbit...	nonbiting	d32	nonbiting#d32.jpg
12	0.40 : 0.60 → nonbit...	0.00 : 1.00 → nonbit...	0.13 : 0.87 → nonbit...	nonbiting	d8	nonbiting#d8.jpg

☒ Show performance scores Target class: (Average over classes)

Model	AUC	CA	F1	Precision	Recall
Logistic Regression	0.969	0.833	0.838	0.889	0.833
Naive Bayes	0.969	0.833	0.838	0.889	0.833
Neural Network	0.969	0.917	0.919	0.933	0.917

인공지능 실습

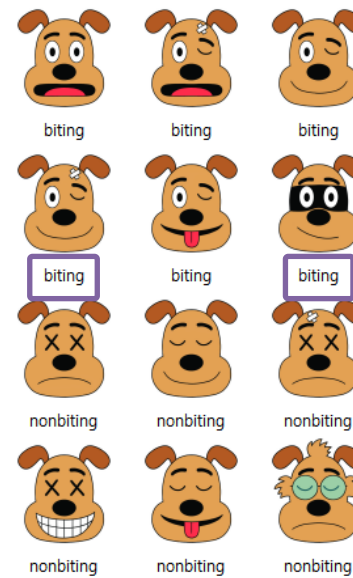
5) 분류 비교



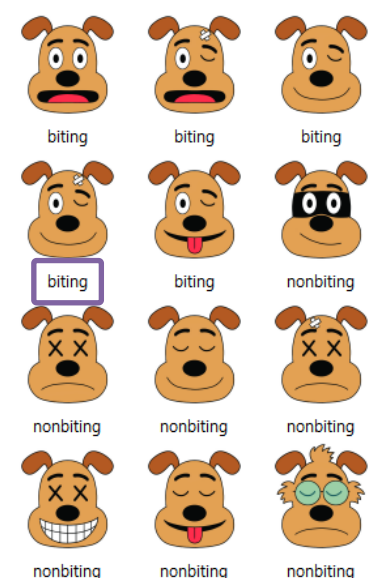
Logistic Regression



Naïve Bayes



Neural Net

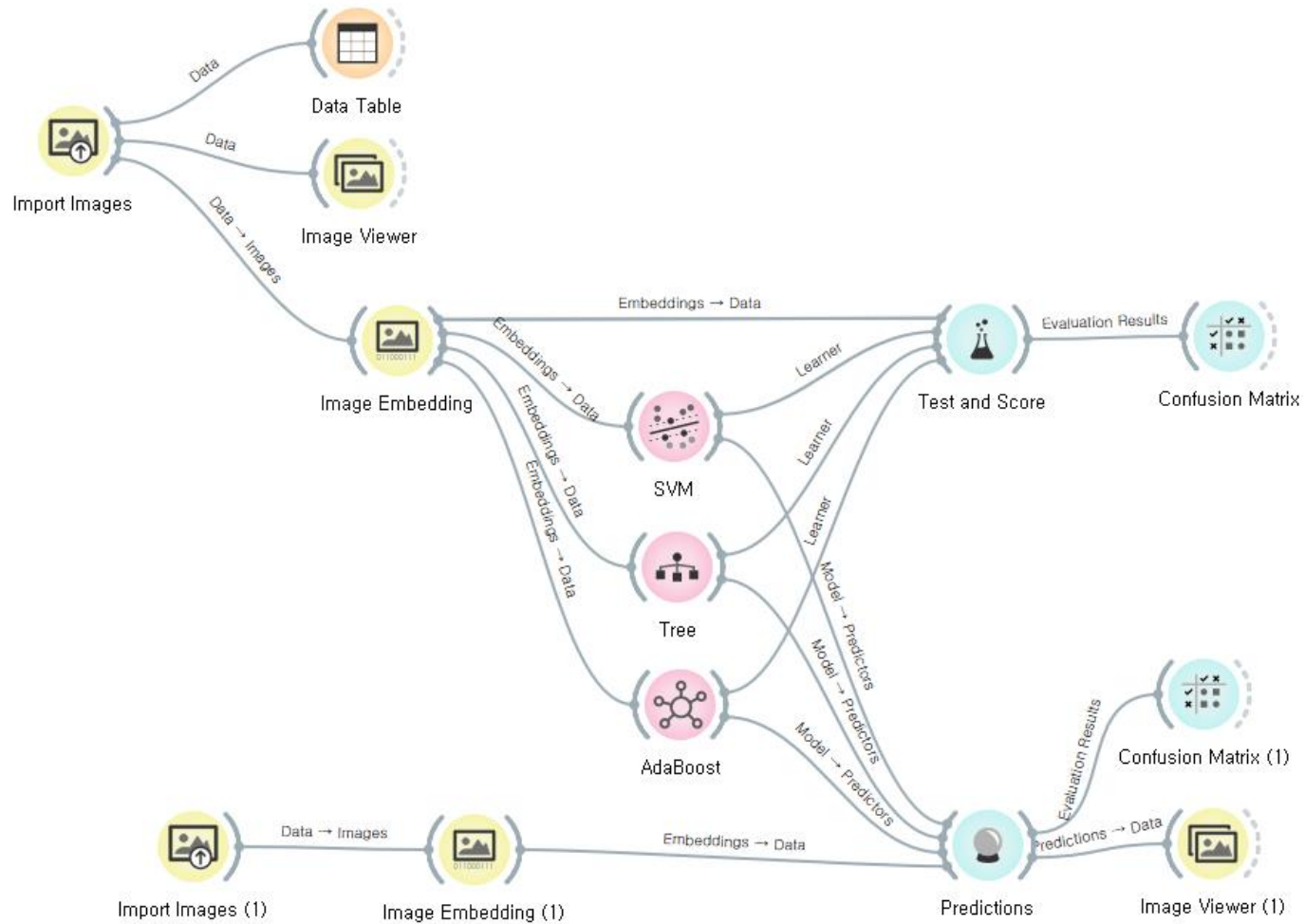


인공지능 실습

실습 1 : 다른 알고리즘을 이용해서 이미지 분류를 잘하는 알고리즘 찾기

- 데이터는 Dog Dataset을 이용
- 적용 알고리즘 : SVM, Tree, AdaBoost
- 훈련용 이미지 데이터는 Dog Dataset Train 데이터를 사용
- 테스트용 이미지 데이터는 Dog Dataset Test 데이터를 사용
- 어느 모델이 무슨 강아지 인식하는 데 더 좋은 가?

인공지능 실습



인공지능 실습

실습 2 : X-ray 이미지를 학습하고 이상여부 판별하기

- 적용 알고리즘 : SVM, Neural-Network
- 훈련용 이미지 데이터는 xray_train 데이터를 사용
- 테스트용 이미지 데이터는 xray_test 데이터를 사용

MBlock 실습

MBlock 인공지능 실습

Open API를 이용한 날씨 안내 앱

기후 데이터(날씨 Open API), TTS 확장 추가

The screenshot shows the MBlock workspace with various extensions. Two extensions are highlighted with red boxes:

- 기후 데이터 (Weather Data):** By mBlock official. 확장은 실시간 기후 데이터에 대한 빠른 액세스를 제공합니다. (The extension provides fast access to real-time climate data.)
- Text to Speech:** 개발자: MIT Media Lab. Make your projects talk. (not available in China yet)

Below the workspace, there are two buttons:

- 기후 데이터** (Weather Data)
- 텍스트 음성 변환** (Text to Speech Conversion)

◎ 기온 불러오기 (최저/최고)



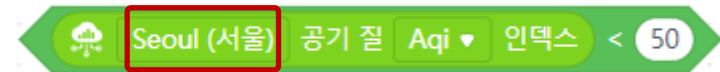
도시 지정

◎ 날씨 불러오기



도시 지정

◎ 공기질 불러오기

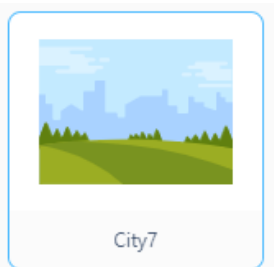


도시 지정

MBlock 인공지능 실습



코스들 수: 1



```
클릭했을 때
  언어를 로 설정 한국어
  만약 ☁️ Seoul, KR 최저 온도 (°C) > 10 이(가) 참이면
    오늘의 날씨는 따뜻하게 시작하겠습니다. 음(를) 2 초 동안 말하기
    말할 오늘의 날씨는 따뜻하게 시작하겠습니다.
  아니면
    오늘도 매서운 추위가 있을 예정입니다. 옷을 따뜻하게 입어야 할 날씨입니다. 음(를) 2 초 동안 말하기
    말할 오늘도 매서운 추위가 있을 예정입니다. 옷을 따뜻하게 입어야 할 날씨입니다.
  만약 ☁️ Seoul, KR 날씨 = Clear 이(가) 참이면
    그래도 오늘은 맑은 날씨를 기록할 예정입니다. 음(를) 2 초 동안 말하기
    말할 그래도 오늘은 맑은 날씨를 기록할 예정입니다.
  아니면
    오늘은 외출하기에 좋은 날씨는 아닙니다. 음(를) 2 초 동안 말하기
    말할 오늘은 외출하기에 좋은 날씨는 아닙니다.
  만약 ☁️ Seoul (서울) 공기 질 Aqi ▼ 인덱스 < 50 이(가) 참이면
    오늘의 공기 질은 괜찮은 수준입니다. 빨래를 하셔도 될 것 같은데요. 음(를) 2 초 동안 말하기
    말할 오늘의 공기 질은 괜찮은 수준입니다. 빨래를 하셔도 될 것 같은데요.
  아니면
    오늘은 힘드시더라도 가급적 환기를 시키지 않는 것이 좋겠습니다. 음(를) 2 초 동안 말하기
    말할 오늘은 힘드시더라도 가급적 환기를 시키지 않는 것이 좋겠습니다.
```


MBlock 인공지능 실습 2

문자인식을 이용한 영어단어 골든벨

인식서비스, TTS 확장 추가

The screenshot shows the MBlock AI Lab interface with a grid of AI services. Three services are highlighted with red boxes:

- 인식 서비스** (Recognition Service): By mBlock official. 인지 서비스 API를 사용하면 비디오, 음성, 언어 및 지식과 같은 다른 기능을 추가할 수 있습니다. [더 보기](#)
- Translate**: 개발자: MIT Media Lab. Translate text into many languages. (not available in China yet)
- Text to Speech**: 개발자: MIT Media Lab. Make your projects talk. (not available in China yet)

Below the screenshot are three buttons:

- 인식 서비스
- 번역하기
- 텍스트 음성 변환

◎ TTS 기능

언어를 로 설정 한국어 ▼ 말할 **정답입니다.**

◎ 번역하기

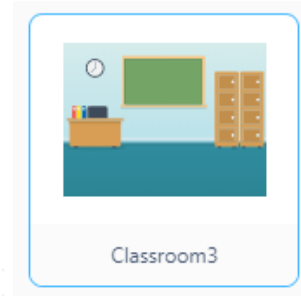
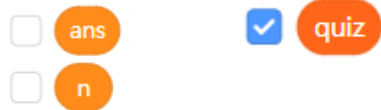
ans ▼ 을(를) 번역하기 quiz ▼ 의 n 번째 항목 받는 사람 영어 ▼ 로(으로) 설정하기

◎ 문자 인식 (영어)

10 ▼ 초 동안 적혀진 영어 인식하기

ans = 문자 인식 결과

MBlock 인공지능 실습 2



팀 활동
