

tensorflow 기본/실무 교육

ref.

- 1일: [텐서플로 기본/실무 교육 TensorFlow\(SK Planet\).pdf](#)
- 2일: [tensorflow 기본/실무 교육.pdf](#)
- 3일: <https://www.slideshare.net/JEEHYUNPAIK/tensorflow-service-machine-learning>
- 4일: <https://docs.google.com/presentation/d/1MCq5u4n0dU7Lqm-fhF3jTpuOpFsEQNv1rYLLxk4rxI/pub?start=false&loop=false&delayms=3000&slide=id.p4>
- 5일: https://docs.google.com/presentation/d/12AZClT2Z0iFiHFu_9JgnK0wEwFrrVfGpNF8wmPYEc4/pub?start=false&loop=false&delayms=3000&slide=id.p4
- 6일: https://docs.google.com/presentation/d/1HRv04pLmfn_R2xRxZrIPA7O4muJlop0D5TroycKIqIs/pub?start=false&loop=false&delayms=3000
- source : git clone https://github.com/TensorMSA/tensormsa_jupyter.git
- docker : docker pull tf_edu_docker_skp
- docker-compose: 슬라이드 참조

1. 실습환경

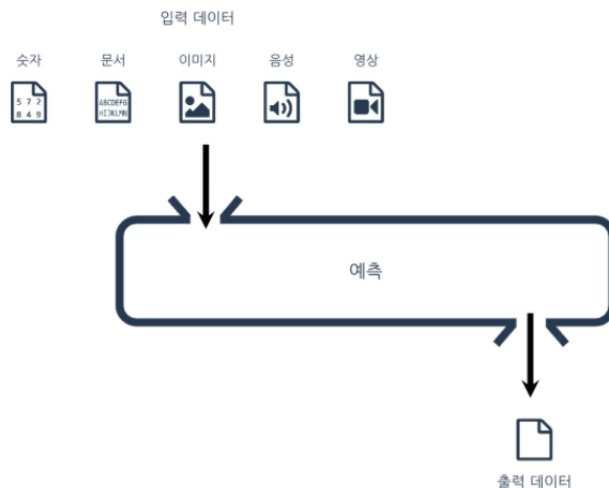
2. 사전지식

a. AI, 머신러닝, 딥러닝 정의

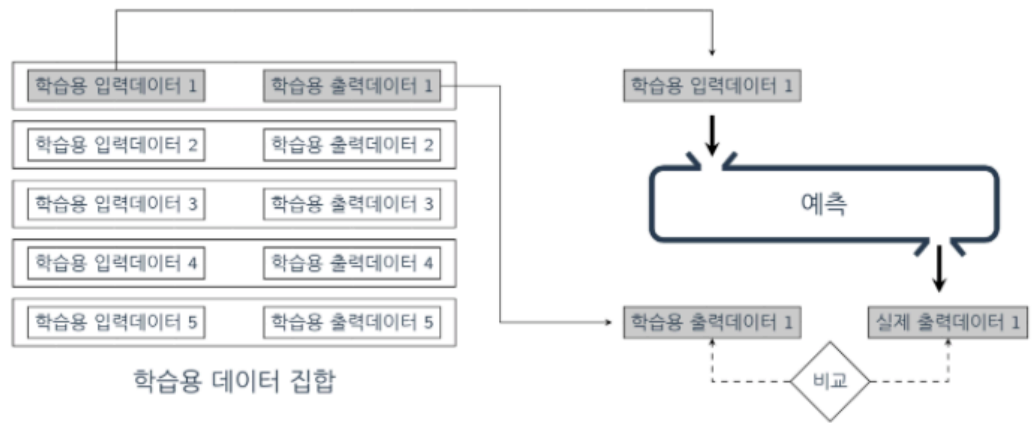
- AI: 인공지능,
인공이란말은 자연적으로 생성된것이 아닌 사람이 만든것 하지만 지능이란 것에 대해서 정의하기가 쉽지 않아
명확한 정의는 없는 상태, 튜링테스트를 통해 우회적으로 정의
예시
1. rule based ex) if 조건
2. search ex) 31
3. inference : $a \rightarrow b, b \rightarrow c \Rightarrow a \rightarrow c$
- 머신러닝: 기계'가' 일일이 코드로 명시하지 않은 동작을 데이터로부터 '학습'하여 실행할 수 있도록 하는
'알고리즘'을 개발하는 연구 분야이다. (1959년 아서 사무엘)
Experience E 를 사용하여 (Learning하여) task T의 performance P가 개선이 되도록 하는 program (Algorithm)
- 딥러닝: 심층학습
여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화(abstractions, 다량의 데이터나 복잡한 자료들 속에서
핵심적인 내용 또는 기능을 요약하는 작업)를 시도하는 기계학습(machine learning) 알고리즘의 집합으로 정의
일반적으로 Neural Network 구조에서 hidden layer가 3개 이상을 말함. hidden layer가 많아지면서 발생하는 여러
문제들을 해결한 학습 알고리즘
조합가능한 basis feature에 까지 학습시키는 효과를 얻을 수 있음

b. 예측(prediction)문제에서 hypothesis, f, model 추정

- $x \rightarrow f \rightarrow y$ 의 구조



- 이전에 발생한 데이터를 이용해 어떠한 동작을 하는 시스템(f, classifier, hypothesis, model)을 추정하는 것



- 그 중 Classification
데이터를 분류하는 미리 정의한 클래스로 분류하는 작업
ex)

예시

상황: 신입으로 입사한 박M에게 선임인 강M가 첫 업무로 배치를 실행하라고 하였다. 그 배치는 X1, X2라는 입력을 받아 Y(-1,1)라는 컬럼의 값을 채우는 배치였다.

그런데 박M이 실수로 배치를 삭제하였다. 비지니스 로직을 모르는 박M은 혼자서 다시 배치를 개발할수 없다. 강M이 새롭게 들어온 데이터를 주며 다시 배치를 돌리라고 한다.

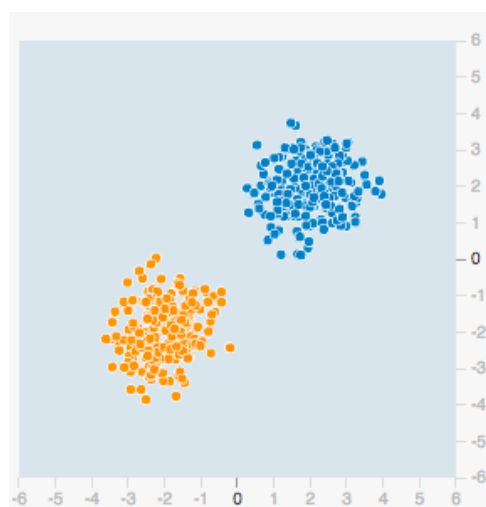
배치를 삭제한 사실을 말하면 혼날것 같아 무서운 박M은 혼자서 해결해보려고 한다. 박M은 이전까지의 X1, X2,Y가 있는 테이블만을 가지고 있고 배치의 로직에 대해선 전혀 모른다.

테이블이 가진 row의 수는 1000개이다.

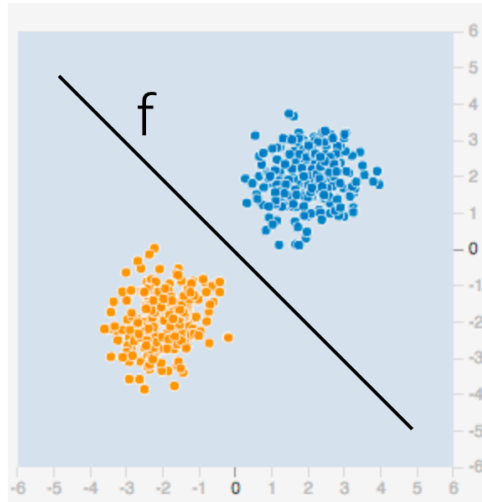
박M은 삭제한 배치를 흉내내는 배치를 만들려고 한다.

index	x1	x2	Y
1	1	1	-1(파랑)
2	-1	-1	1(주황)
3	2	1	-1
4	-3	-4	1
...			

먼저 박M은 각 row를 x1, x2축으로 하는 이차원 평면에 표시해보기로 하였다.



그리고 f라는 선을 그려서 f라는 선보다 위쪽에 있으면 파랑(-1), 아래에 있으면 주황(1)이라고 가정하였다.



f라는 선은 수식으로 표현하면 아래와 같다.

$$f(x_1, x_2) = \begin{cases} w_1 * x_1 + w_2 * x_2 > 0 \Rightarrow 1 \\ w_1 * x_1 + w_2 * x_2 < 0 \Rightarrow -1 \end{cases}$$

그리고 실제로 알고 있는 x1과 x2를 대입하여 w라는 가중치를 결정해보기로 하였다.

1번 row에 대해서 계산해보니 f(1번 row) = w1 + w2 이고 그때 y의 값은 -1 였다.

위 식대로라면은 y가 -1이 되기 위해서 w1 + w2 < 0 이어야만 한다.

2번 row에 대해서도 계산해보니 f(2번 row) = -w1 - w2 이고 그때 y는 1 이었다.

이번에도 미리 가정한 식을 만족하려면 -w1 - w2 > 0 이고 w1 + w2 < 0 이어야 한다.

그래서 위 부등식을 만족하는 w1과 w2를 모두 -1 임의로 정하였다 그때 f는

$$f(x_1, x_2) = \begin{cases} -x_1 - x_2 > 0 \Rightarrow 1 \\ -x_1 - x_2 < 0 \Rightarrow -1 \end{cases}$$

이였다.

3번과 4번에 대해서 이 함수에 대입하여 보니

f(3번 row) = -3 < 0 => -1이라고 판단하였고 실제로 y는 -1 이였다.

f(4번 row) = 7 > 0 => 1이라고 판단하였고 실제로 y는 1 이였다.

이 결과를 보고 박M은 기뻐지만 소심한 박M은 다른 의문들이 생겼다.

그런데 데이터 1,2번을 만족하는 f가 3,4번 데이터를 올바르게 예측했다고 해서 이게 나머지 데이터 996개에 대해 맞을까 ?

또, 1000개의 데이터를 일일이 손으로 풀면 강M가 다시 올때까지 데이터를 채울 수 없다 ?

엄청난 노력으로 1000개에 대해서 만족하는 f를 만들었지만 새롭게 준 데이터를 예측하였을 때 그 값이 정답이라고 확신할 수 있을까 ?

또한 중간에 이전의 f에 맞지 않은 결과가 나오면 어떻게 w들을 수정해야할까 ?

이에 대해 머신러닝에서 사용하는 용어로 바꿔보자면

테이블의 x1,x2컬럼은 데이터가 가지는 feature(속성)

y 는 예측할 class, label

f는 classifier, model, hypothesis

w는 가중치 weight

데이터를 대입한 정답과 예측값을 비교하는 과정 => loss, cost수식을 통해 error를 구하는 과정, feed과정
구해진 error를 통해 weight를 구하는 과정 => 최적화, 피드백과정
최종적으로 w값이 결정된 f함수 => 학습을 통해 만들어진 모델

1,2번을 통해 구해진 값을 3,4번과 비교하는 과정 => validation
강M이 준 데이터를 예측하는 과정 => prediction

w를 어떻게 변화시킬 것인가? => optimize rule
w를 변화시킬때 얼마만큼 변화시킬것인가? => learning rate
몇개의 row를 계산해보고 움직일 것인가? => batch size
모든 데이터에 대해서 몇번이나 반복할 것인가? => epoch

전체 과정을 보면

- 1) feature정의, class정의, f 가정
- 2) 학습데이터를 f에 넣어 y'라는 예측값을 구함
- 3) y' 와 실제값 y를 비교하여 error를 구함
- 4) 이 error에 따라서 f의 파라미터인 w를 바꾸어줌
- 5) 이 과정을 더이상 error가 변하지 않을때까지 수렴때까지 수행
- 6) 하이퍼 파라미터를 바꾸면서 validation 진행

최종적인 f를 이용하여 test data를 예측하고 평가

c. tensorflow 개요
딥러닝을 위한 lib

- d. 간단한 NN 튜닝하기
<http://playground.tensorflow.org/>
- 선형 분류 가능
 - 비선형 분류
 - 원형
 - XOR
 - 나선형