학습계획서

팀	커피시키신분	구성원	이기문, 유승아,이재현

일정	발제자	주제	주요내용
1일차 (5 / 27)	이기문	오리엔테이션	향후 학습방향 조정 및 교육컨텐츠 선정
2일차 (5 / 28)	유승아	Basic Machine Learning(1)	• Lec 01: 기본적인 Machine Learning 의 용어와 개념 설명 • Lec 02: Simple Linear Regression • Lab 02: Simple Linear Regression 를 TensorFlow 로 구현하기 • Lec 03: Linear Regression and How to minimize cost • Lab 03: Linear Regression and How to minimize cost 를 TensorFlow 로 구현하기
3일차 (5 / 29)	이재현	Basic Machine Learning(2)	• Lec 04: Multi-variable Linear Regression • Lab 04: Multi-variable Linear Regression 를 TensorFlow 로 구현하기 • Lec 05-1: Logistic Regression/Classification 의 소개 • Lec 05-2: Logistic Regression/Classification 의 cost 함수, 최소화 • Lab 05-3: Logistic Regression/Classification 를 TensorFlow 로 구현하기 • Lec 06-1: Softmax Regression: 기본 개념소개 • Lec 06-2: Softmax Classifier의 cost함수 • Lab 06-1: Softmax classifier 를 TensorFlow 로 구현하기 • Lab 06-2: Fancy Softmax classifier 를 TensorFlow 로 구현하기
4일차 (5 / 30)	유승아	Basic Machine Learning(3)	• Lab 07-1: Application & Tips: 학습률(Learning Rate)과 데이터 전처리(Data Preprocessing) • Lab 07-2-1: Application & Tips: 오버피팅(Overfitting) & Solutions • Lab 07-2-2: Application & Tips: 학습률, 전처리, 오버피팅을 TensorFlow 로 실습 • Lab 07-3-1: Application & Tips: Data & Learning • Lab 07-3-2: Application & Tips: 다양한 Dataset 으로 실습
5일차 (5 / 31)	이기문	DNN(1)	• Lec 08-1: 딥러닝의 기본 개념: 시작과 XOR 문제

			● Lec 08-2: 딥러닝의 기본 개념2: Back-propagation 과 2006/2007 '딥'의 출현 ● Lec 09-1: XOR 문제 딥러닝으로 풀기 ● Lec 09-2: 딥넷트웍 학습 시키기 (backpropagation) ● Lab 09-1: Neural Net for XOR ● Lab 09-2: Tensorboard (Neural Net for XOR)
6일차 (6 / 3)	이재현	DNN(2)	● Lab 10-1: Sigmoid 보다 ReLU가 더 좋아 ● Lab 10-2: Weight 초기화 잘해보자 ● Lab 10-3: Dropout ● Lab 10-4: Batch Normalization
7일차 (6 / 4)	이기문	RNN(1)	● Lec 12: NN의 꽃 RNN 이야기 ● Lab 12-0: rnn basics ● Lab 12-1: many to one (word sentiment classification) ● Lab 12-2: many to one stacked (sentence classification, stacked)
8일차 (6 / 5)	유승아	RNN(2)	● Lab 12-3: many to many (simple pos-tagger training) ● Lab 12-4: many to many bidirectional (simpled pos-tagger training, bidirectional) ● Lab 12-5: seq to seq (simple neural machine translation) ● Lab 12-6: seq to seq with attention (simple neural machine translation, attention)
9일차 (6 / 7)	이재현	프로젝트 기획	머신러닝을 이용한 프로젝트 기획
10일차 (6 / 10)	유승아	기능 설계	프로젝트의 구체적인 기능을 정의하고 개발 스케줄 작성

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
5/28	유승아	• Basic Machine Learning(1)

주요 내용 요약

Lec 00

도커란?

- 컨테이너 기반의 가상 환경 시스템
- 독립된 운영체제를 굳이 여러개 띄울 필요가 없다
- 리눅스 기반이다
    - 윈도우 및 맥에서는 GPU 사용 불가
    - 리눅스에서만큼의 성능이 나오지 않을 수 있다

컨테이너

- 운영체제 자체는 동일한 상태로 나머지 필요한 부분만 묶어 가볍게 가상화한 것

Lec 01

Machine Learning

- 등장 배경: 명시적 프로그래밍의 한계로 인해
- 종류
    - Supervised Learning
    - Unsupervised Learning

Supervised Learning

- 정해진 데이터(training set)로 학습하는 방식
- 예: 이미지 분류(labeling), 이메일 스팸 필터링, 시험 점수 예측 등
- training set으로 학습한 내용을 배경으로 test data를 입력 받았을때 결과를 예측하는 방식
- 종류별 예
    - regression: 시험공부한 시간을 바탕으로 시험 점수 예측
    - binary classification: 시험공부한 시간을 바탕으로 pass/fail 예측
    - multi-label classification: 시험공부한 시간을 바탕으로 A, B, C, D, F 등 등급

예측

Lec 02

Regression

- 어떠한 크기의 데이터가 나와도 이 데이터들은 전체 평균으로 회귀하려는 속성이 있다.
- 데이터간 관계를 해석하는것

Linear Regression

- 목표: 데이터를 가장 잘 나타낼 수 있는 직선형 방정식을 찾는것
- Hypothesis(가설): 데이터를 가장 잘 나타낼 것으로 예측되는 직선 방정식 - H(x) = Wx + b
- Cost: 가설과 실제 데이터간의 차이(loss, error)

$$W := W - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(W(x_i) - y_i \right)$$

  m: 전체 데이터 수
  cost function: 전체 cost 제곱의 평균

- learning: Cost를 최소화하는 W, b를 찾는것

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
5/29	이재현	• Basic Machine Learning(2)

주요 내용 요약

Lec 04

Multi Variable Regression

- 변수 갯수가 늘어나면 가중치 갯수도 그만큼 늘어난다
- 행렬 곱(Dot product)을 통해 간결하게 표현: H(X) = XW
    - 행, 열 갯수 중요: X의 열 갯수와 W의 행 갯수가 일치해야 행렬 곱 가능
    - W의 행 갯수 = 입력 데이터의 열 갯수, W의 열 갯수 = 결과 데이터의 열 갯수
    - 변수의 갯수가 많을 경우 matrix를 사용하면 표현, 성능면에서 유용하다.

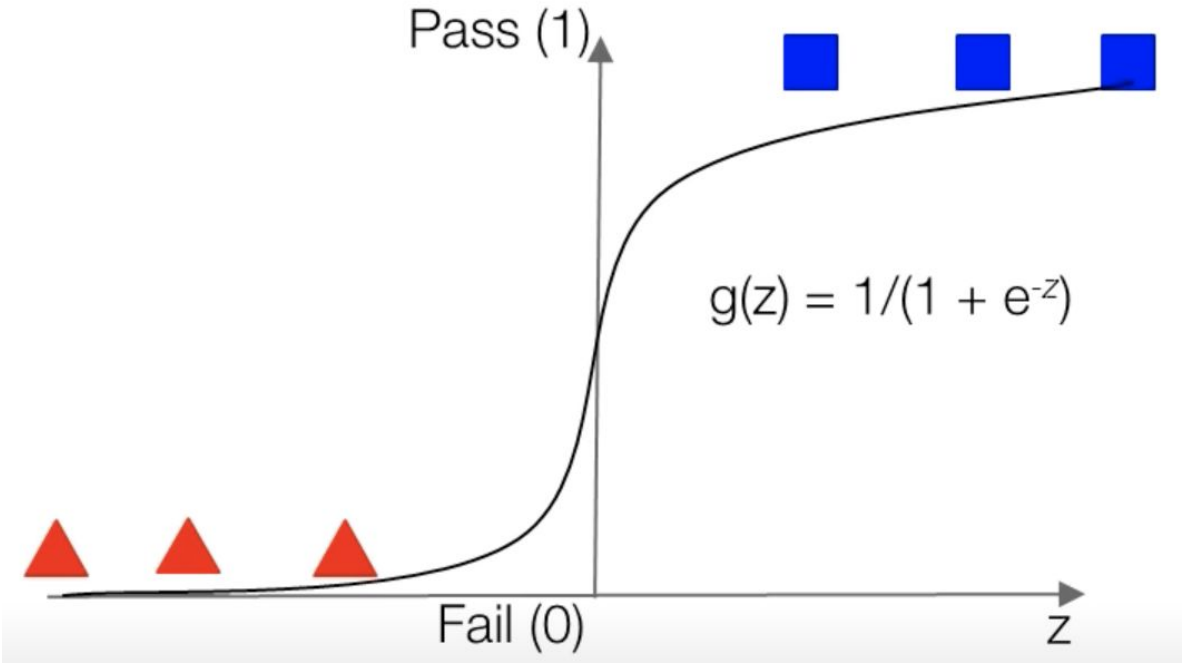
Lec 05

Logistic Regression

- 데이터를 구분하기 위해 사용(classification)
- 적용 가능한 데이터들의 특징: binary classification이 가능한 데이터
- Linear Regression과의 차이
    - Linear Regression: 데이터들이 연속적이다(Numeric)
    - Logistic Regression: 데이터가 흩어져있고 구분이 가능하다(One hot)
- 데이터를 Logistic Function으로 나타낸 후 decision boundary를 적용해 classification 실행
- Neural Network의 하나의 component

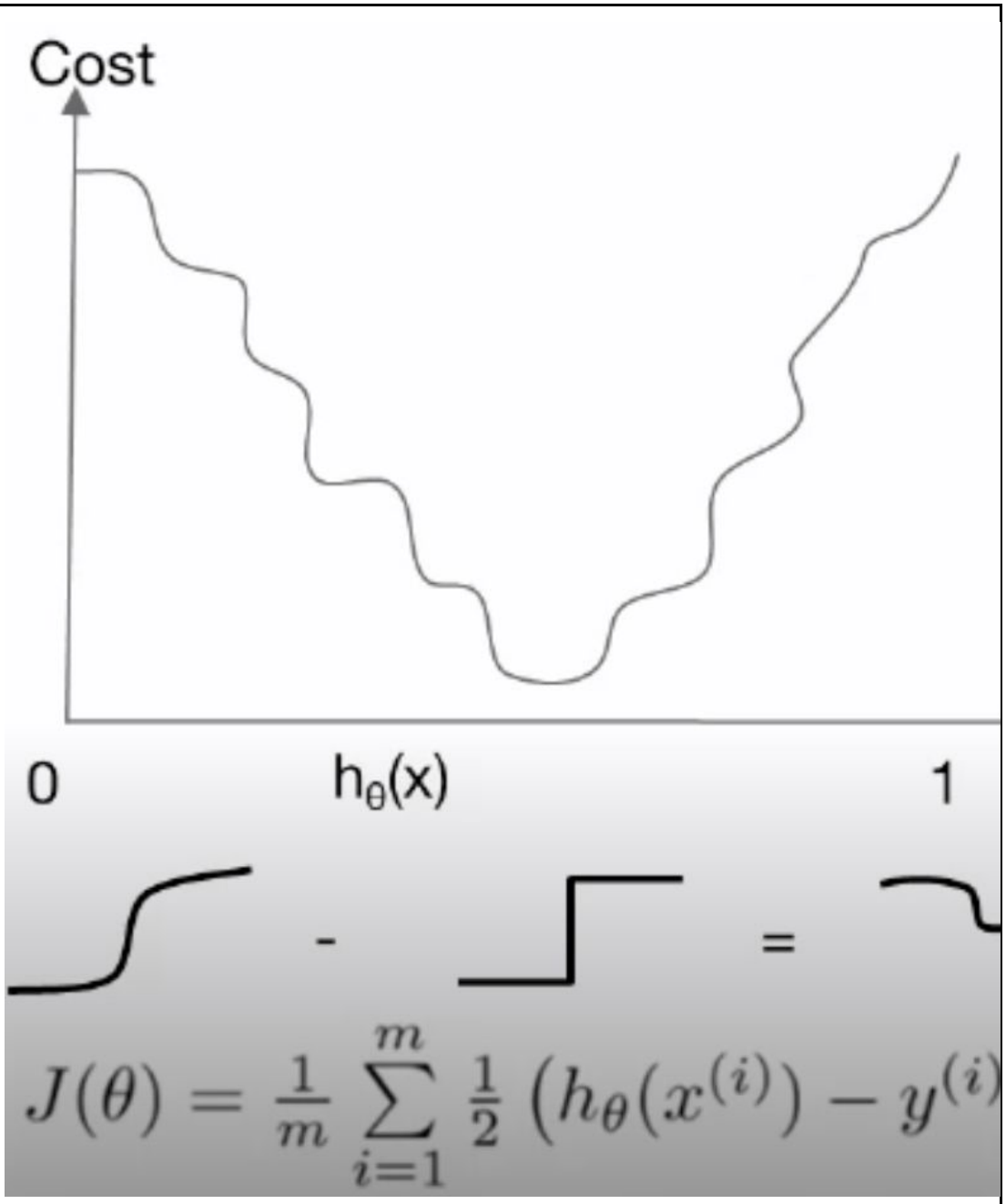
Sigmoid Function

- g function
- linear regression을 통해 얻은 값을 sigmoid function을 통해 0, 1로 변환한다
- decision boundary: 데이터 구분을 위한 경계
    - linear / non-linear function 관계없이 설정 가능



Cost Function

- 초기의 random한 W값을 fitting하기 위함
- Sigmoid Function의 값 - binary classification한 값이므로 그래프의 형태가 구부러진 형태가 된다



- 형태
  - 예측값이 1이면: cost(h(x), y) = -log(h(x))
  - 예측값이 0이면: cost(h(x), y) = -log(1 - h(x))
- Optimization: cost function 값을 최소화하는 작업
  - Gradient Descent를 통해 최적화

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
5/30	유승아	• Basic Machine Learning(3)

주요 내용 요약

Lec 06

Multinomial Classification

- binary classification을 여러번 수행하는 방식
- 각각의 classifier를 만드는 대신 softmax를 활용해 한번에 처리 가능

Softmax

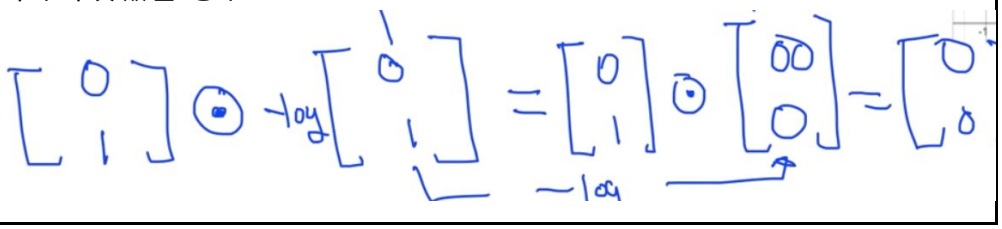
- 예측 결과가 여러 label에 대해 각각의 label에 속할 확률로 나타남(결과값 합 = 1)
  - one hot encoding 적용하면 가장 큰 확률만 1로 나타나고 나머지는 0으로 나타남

Cost function

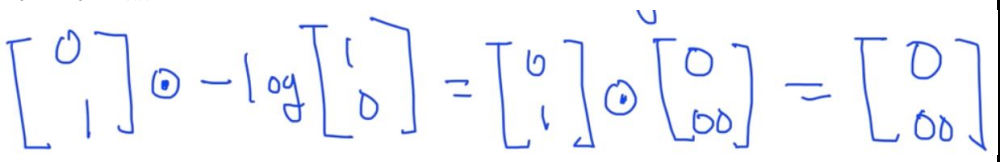
- cross entropy: softmax 결과 예측한 값과 실제 label 값의 차이를 구하기 위해 사용

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

  - y: 실제 label 값
  - p: softmax를 통해 예측한 값
  - y와 p의 각각의 element를 곱한 후 모든 element의 값을 합하는 방식
  - 예측이 맞았을 경우



    - 예측이 틀렸을 경우



    - 예측이 맞으면 0, 틀리면 매우 큰 값이 된다
- gradient descent를 통해 cost를 최소화한다

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
5/31	이기문	• DNN(1)

주요 내용 요약

Lec 08

Activation Function

1. 특정 신호가 들어온다
2. 그 신호에 weight을 곱한 후 전부 더한다
3. 총합에 bias를 더한다
4. 계산 결과가 특정 값 이상일 경우 1, 아니면 0의 신호를 준다

XOR Problem

- AND, OR의 결과값은 선형 함수로 separate할 수 있다
- XOR의 결과값은 선형 함수로 separate할 수 없다
    - 여러개의 선형 함수로는 가능: MLP(Multi Layer Perceptron) 필요
    - MLP를 어떻게 학습시킬것인가

Backpropagation

- Multi Layer의 앞에서 구한 error를 토대로 뒤쪽 layer를 training해나가는 방법
- 문제점: layer 수가 늘어나면 제대로 동작하지 않는다 - layer 수가 늘어날수록 성능

감소

Convolutional Neural Network

- 그림을 한번에 입력시키지 않고 잘라서 각각의 layer에 보낸 후 나중에 합치는 방법
- 문자, 숫자 인식에 주로 사용
- 사진 분류, 사진 설명도 가능

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

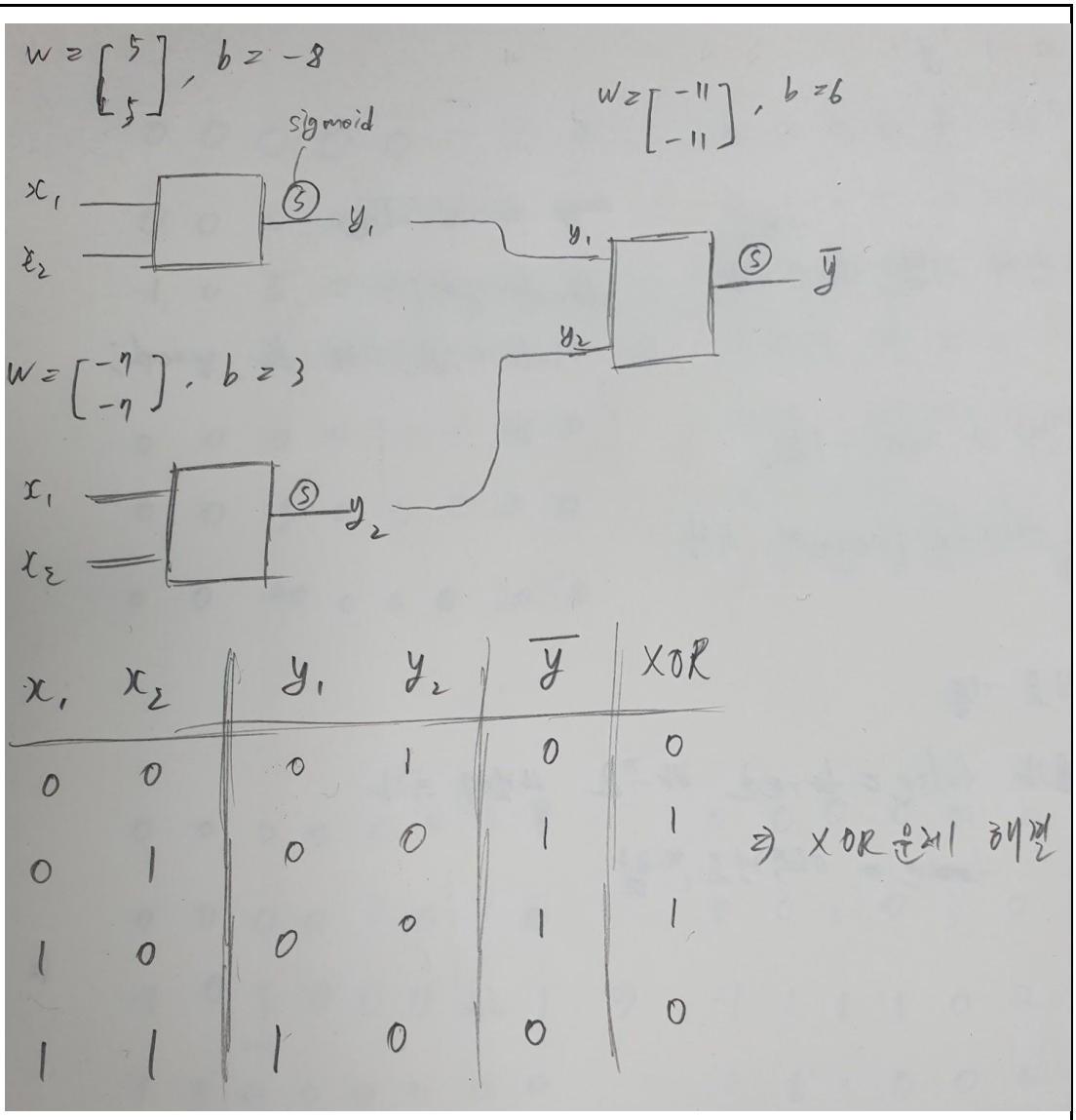
일정	발제자	주제
6/3	이재현	• DNN(2)

주요 내용 요약

Lec 09

Solving XOR Problem with Neural Network

- Neural Network: layer별로 각각의 weight, bias를 가진다
- Forward Propagation으로 해결 가능

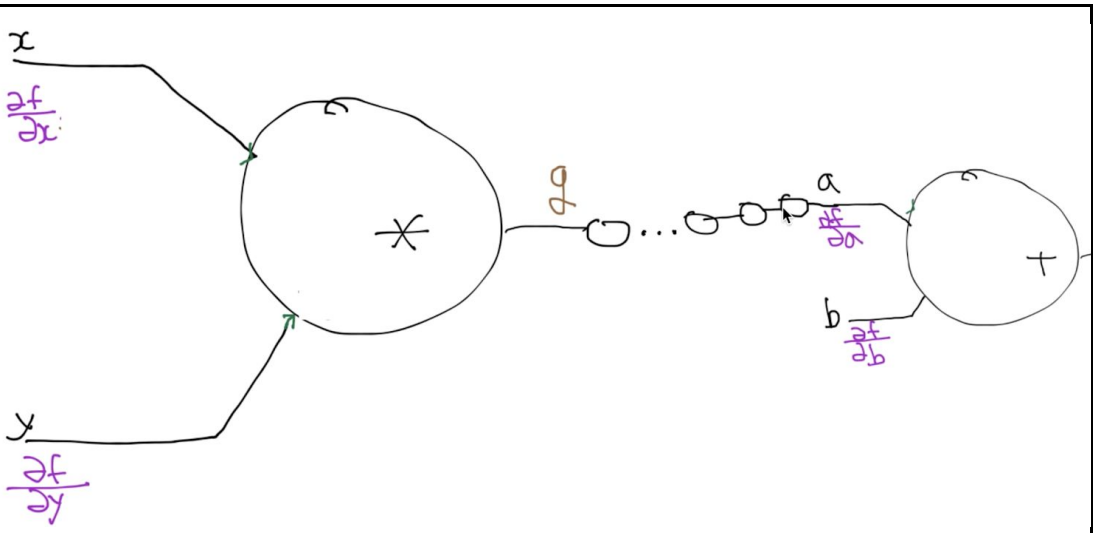


- layer가 늘어나며 W, b의 차원이 증가 => Multinomial Classification처럼 행렬 곱으로 처리
- 구현 형태

```
# NN
K = tf.sigmoid(tf.matmul(X, W1) + b1)
hypothesis = tf.sigmoid(tf.matmul(K, W2) +
```

Backpropagation

- NN의 각각의 layer들을 어떻게 학습시킬 것인가: layer가 늘어나며 gradient descent 과정이 복잡해진다
- 예측값과 출력값을 비교한 후 계산된 cost 값을 뒤에서부터 앞으로 돌려서 미분값과 실제로 조절할 값 계산
- 해결: backpropagation을 활용하여 복잡한 미분 계산
- f = wx + b, g = wx => f = g + b로 정의



- chain rule을 활용하여 f에 가장 가까운 layer부터 미분을 진행하며 다음 layer 미분 진행
- 결과: 최종적으로 x, y가 f에 미치는 영향을 알 수 있다

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
6/4	이기문	• RNN(1)
주요 내용 요약		
Activation Functions • Sigmoid • tanh • relu • elu		

- selu

Sigmoid 학습 과정

- input 입력
- network에서 input에 대한 output 도출
- 실제값과 비교하여 loss 구함
- loss function을 미분하는 backpropagation을 통해 학습
- gradient: loss의 미분값
- 문제점: backporpagation 과정에서 network 내 각 layer들의 gradient가 곱해지는 중 매우 작은 값들이 곱해지며 gradient가 소실되는 현상 발생(Vanishing Gradient)

Relu 함수

- f(x) = max(0, x) 형태
- 0보다 클때: gradient = 1
- network가 deep해도 gradient 소실되지 않음
- 0보다 작을때: gradient = 0
- gradient가 전달되지 않는 문제
- 0보다 작을때 전달되지 않는 문제가 있어도 많이 쓰인다
- leaky relu: 0보다 작은 경우 gradient = 특정 값과 x를 곱한 값

Weight initialization

- Xavier Initialization (Glorot Initialization)
  - 임의의 출발점에서 global minima를 찾는 것이 목적
  - 문제점: 출발점에 따라 local minima에 빠지거나 saddle point에 도달하는 경우
  - 출발점을 잘 고르는데 도움을 준다
  - 작동 원리
    - 평균 = 0
    - 분산 = 2 / (Channel_in + Channel_out)
      - Channel_in: input으로 들어가는 channel 갯수
      - Channel_out: output으로 나오는 channel 갯수
- He Initialization
  - Relu 함수에 특화
  - 작동 원리
    - 평균 = 0
    - 분산 = 4 / (Channel_in + Channel_out)

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
6/5	유승아	• RNN(2)

주요 내용 요약

Underfitting

- training dataset에 대한 정확도가 매우 낮음
- test dataset 정확도 낮음

Overfitting

- training dataset에 대한 정확도가 매우 높음
- test dataset 정확도 낮음: training dataset에만 맞춰져 있어 새로운 데이터에 대한 정확도가 낮아짐

Dropout

- underfitting, overfitting을 방지하는 regularization 기법
- 학습 과정에서 전체 neuron(node) 중 일부만 사용하는 방식
- 어떤 node를 사용할지는 random
- test시에는 전체 node 사용
- 학습 과정에서 데이터의 전체를 보는 것이 아닌 부분을 보고 예측하는 방식

Batch Normalization

- Internal Covariate Shift 방지
  - Internal Covariate Shift: layer들을 거치며 input으로 들어온 distribution이 변형되는 현상
- 각 layer의 input으로 들어오는 distribution을 계속 normalize하여 distribution을 항상 일정하게 유지

학습 정리

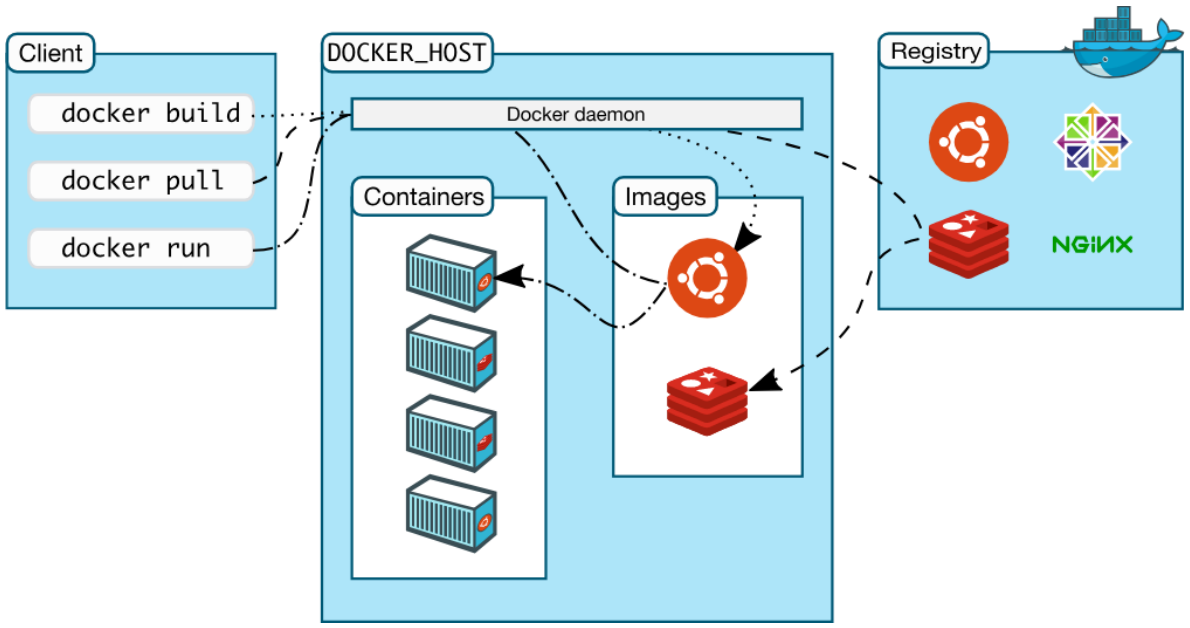
팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
6/7	이재현	• 프로젝트 기획
주요 내용 요약		
Cobbee 커피 시키신 분 : 사무실 막내 커피 봇 - 협업툴인 Slack에 직접 구현한 Deeplearning Model을 이용하여, 커피 주문내용 자동 추합 봇을 개발 1. RNN을 이용하여 자연어를 가공 2. IDE : Pycharm 3. Remote Interpreter : Docker 4. Language : Python		

학습 정리

팀	커피시키신분	구성원	이기문, 유승아, 이재현

일정	발제자	주제
6/10	유승아	• 기능설계
주요 내용 요약		



Cobbee 커피 시키신 분 : 사무실 막내 커피 봇

- 메신저에 사용자들이 자신이 원하는 커피의 종류와 옵션을 말한다.
- 코비봇 프로세스를 통해 커피의 종류, 각각의 옵션, 갯수가 정리되어 추출된다.
- 사무실 막내는 추출된 결과를 통해 간편하게 커피 주문을 한다 !