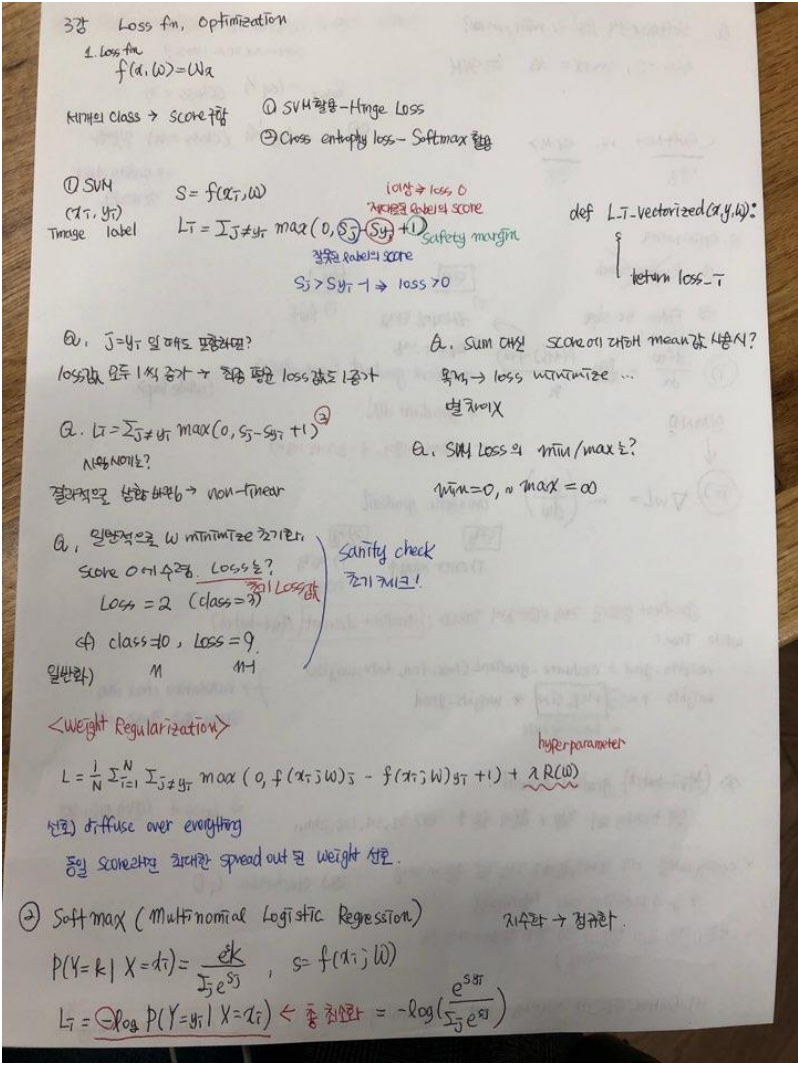


팀 미팅 회의록

팀명	3조_달달한 컴퓨터 달콤	차수	6 차
일 시	2018 년 3 월 28 일		
장 소	7호관		
참석자	전원		
불참자	無		
회의내용	<p>- 이번 주까지의 주요 기능 완성 및 오류 해결</p> <ol style="list-style-type: none"> 1. 앱이 서버에서 mp3 받아오는 기능 2. 알람 작동 스위치 오류 해결 3. 딥러닝 강의 공부 		
			

Q. Softmax loss의 min, max?
 $\min = 0, \max = \infty \Rightarrow SVM$

Q. weight = 0. (9=0)
 Softmax의 loss?

$loss = -\log \frac{1}{3}$ (class = 3).

cf) $-\log \frac{1}{n}$ (class = n) 일반화

→ sanity check
 확인하기

<SoftMax vs. SLM>
 SoftMax vs. SLM

Q. Optimization

① Random search

② Follow the slope
 gradient check

(1) $\frac{df(w)}{dw} = \lim_{h \rightarrow 0} \frac{f(w+h) - f(w)}{h}$

numerical gradient (vector 형태)

→ gradient dW

(- : 업데이트 기호, + : 불확실 기호)

실제 사용

(2) $\nabla WL = \dots \left(\frac{dL}{dw} \right)$

analytic gradient

단점

1) error 가능성 ↑

장점

1) 정확

2) 빠름

gradient 방향으로 계속 이동해서 최적화! gradient descent (full-batch)

while True:

weights_grad = evaluate_gradient(loss_fun, data, weights)

weights += \ominus step_size * weights_grad
 = learning rate

→ validation check 해서
 최적의 값을 찾아줘야 함.

cf) (Mini-batch) gradient descent

일부 training set 활용 → 정확도 ↑ ex) 32, 64, 128, 256, ...

→ Loss가 시간이 따라 감소

* learning rate를 너무 낮게, 높게도 하는 것은 좋지 않음

→ good learning rate 실험해봐야 함

ex) Stochastic GD.

보통 1) 조금 높게 설정 2) 점점 낮춰가는 식

(decay)

cf) feature 추출이 일반적이지만 deep learning에선 이미지 자체를 classifying 단계를