8/10(화) 회의록

작성일시	@2021년 8월 6일 오후 5:57
작성자	하람 이하람
참석자	
최종 편집일시	@2021년 8월 17일 오후 2:00
회의 유형	일일 회의

☝학습 내용 공유

1. 강의 내용 중 질문하기🙋‍♀️

Q1. 토치 익숙하신분, 여기서 self.modules()로 가져온 iterable객체는 위에서 self.net.add_moddule로 만들어진건가요? nn.Module의 메소드 같기도 하군요

```
        prev_hdim = hdim
    # Final layer (without activation)
    self.layers.append(nn.Linear(prev_hdim,self.ydim,bias=True))

    # Concatenate all layers
    self.net = nn.Sequential()
    for l_idx,layer in enumerate(self.layers):
        layer_name = "%s_%02d"%(type(layer).__name__.lower(),l_idx)
        self.net.add_module(layer_name,layer)

    self.init_param() # initialize parameters

def init_param(self):
    for m in self.modules(): # 토치 익숙하신분께 질문, 여기서 self.modules()로 가져온 iterable 객체는 위에서 self.net.add_module로 생긴건가요?
        if isinstance(m,nn.Conv2d): # init conv
            nn.init.kaiming_normal_(m.weight)
            nn.init.zeros_(m.bias)
        elif isinstance(m,nn.Linear): # init dense
            nn.init.kaiming_normal_(m.weight)
            nn.init.zeros_(m.bias)

def forward(self,x):
    return self.net(x)
```

nn.Module의 메소드가 맞습니다

https://pytorch.org/docs/stable/generated/torch.nn.Module.html

Model 클래스가 nn.Module을 상속하고 있기 때문에, self.modules() 를 호출하면 nn.Module의 내장 메소드가 호출되는 것 같습니다. 근데 add_module 때문만은 아니고, self.modules()

를 호출하면 __init__에서 선언된 nn.Linear 들을 전부 불러올 수 있는 것 같습니다. (nn.Linear나 nn.Conv2d 등은 nn.Module을 상속하고 있습니다.)

Q2. 이번 강의 cross-validation 관련해서 질문이 있습니다. train data를 k개의 fold로 나누는 과정이라고 강의 중에 나왔습니다. 또한, 최적의 hyperparameter를 찾는데 사용된다고 말씀하셨는데, hyperparameter를 찾을 때 학습 데이터는 1개의 fold가 되는건가요?? 아님 1개의 validation set을 제외한 나머지 전부가 되는건가요??

- 후자가 맞는 것 같습니다!
- 그럼 hyperparameter가 정해진 후에 train set을 전체로 학습하라고 하셨는데 그건 validation set까지 포함시켜서 학습을 시키라는 얘기인거죠??
- A)hyperparameter가 정해진 후에 train set을 전체로 학습할 수도 있긴 한데, 그러면 보성님이 올려주신대로 학습 중에 과적합이 일어나도 모르고 계속 학습 시킬 수가 있어요.
- 그래서 제 생각에는 hyperparameter가 정해지면 validation set의 비율을 줄이고 학습 시킬 수 있다는 것 아닐까요.

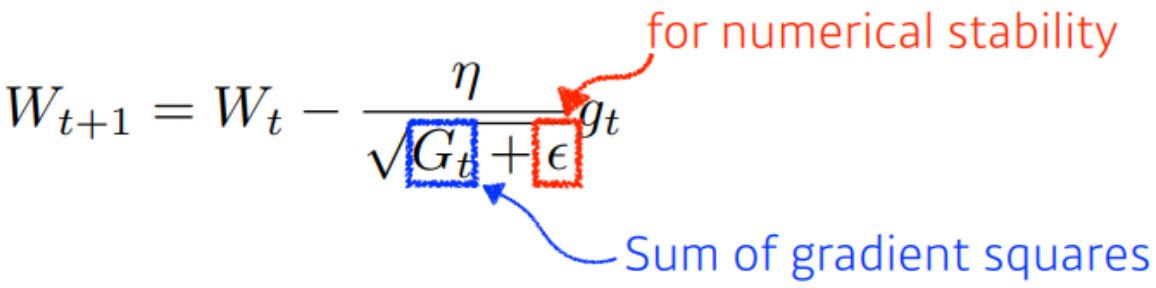
	Training dataset	Validation dataset	Test dataset
학습 과정에서 참조할 수 있는가?	O	O	X
모델의 인자값 (가중치) 설정에 이용되는가?	O	X	X
모델의 성능 평가에 이용되는가?	X	O	O

-요건 트레이닝데이터셋/벨리데이션/테스트데이터셋의 차이점 정리입니다.

출처)https://untitledtblog.tistory.com/158

-요건 제 사족인데 보통 0.7을 트레이닝셋으로, 0.2를 밸리데이션셋으로, 0.1을 테스트 셋으로 쓰는걸 자주 봤던것 같지만 이건 크게 의미가 없겠네요 ㅋㅋ 밸리데이션은 어쨌든 트레이닝 셋에 안넣는게 정론입니다...! 매 에폭마다 셔플을 걸어도 트레이닝 셋 안에만 돌리는게 맞지 싶어요...!

Q3. Adagrad의 아이디어가 ‘지금까지 많이 변화하지 않은 변수들은 step size를 크게 하고, 지금까지 많이 변화했던 변수들은 step size를 작게 하자’라고 말씀하셨는데 이게 수식에서 어디서 나타나는건지 궁금합니다.



여기 적당한 설명이 있읍니다

Q4. 24:37 강의 설명중에서 (momentum 과 NAG 비교)

설명에서 두 개의 차이를 잘 모르겠습니다.

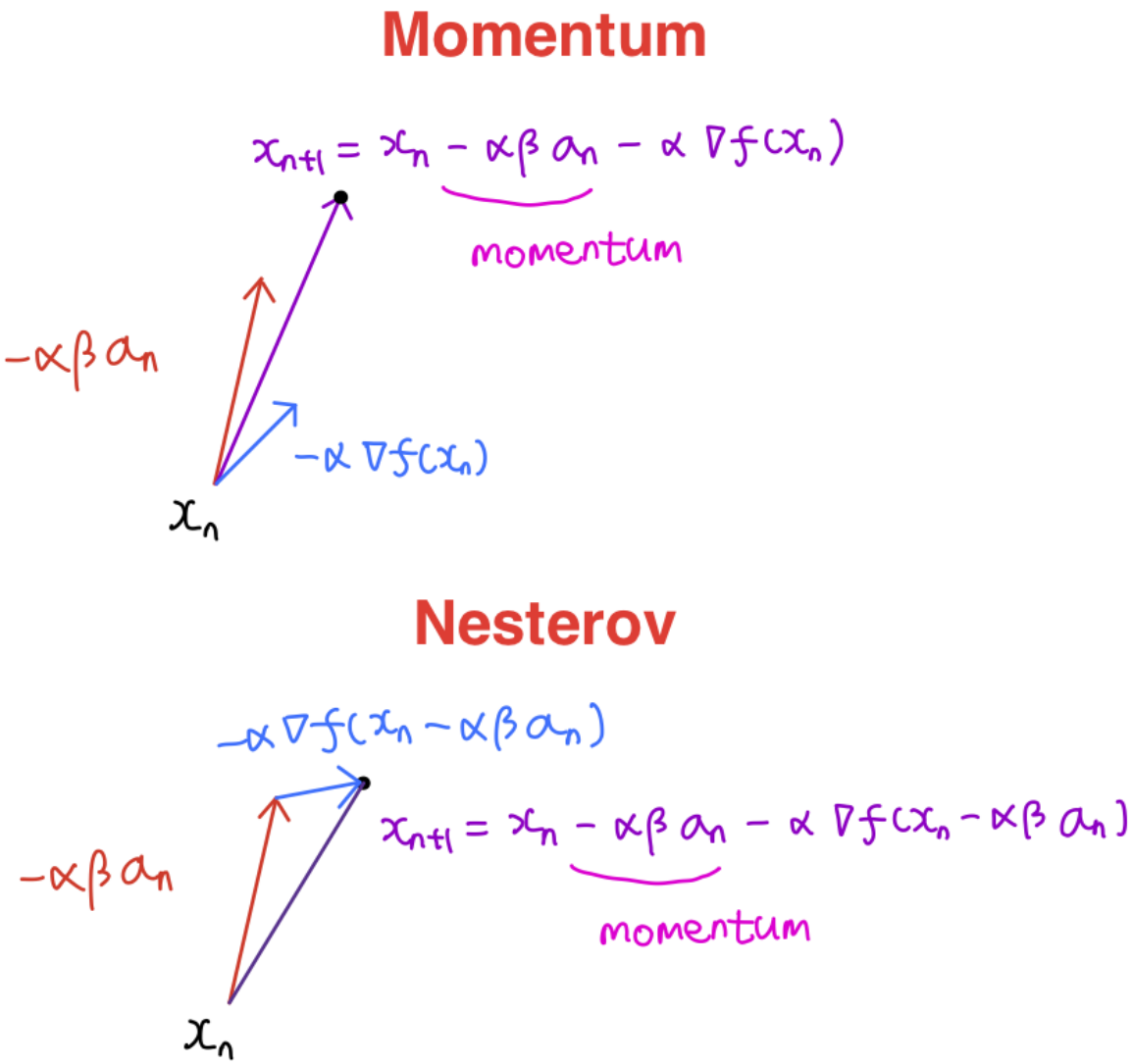
momentum은 momentum는 slope 반대편에 도달해서 다시 gradient를 계산해서 방향이 반대로 바뀌어도 이전 방향 그대로 이동한다고 말씀하신 후에 slope를 왔다 갔다 한다고 말하셨는데, 계속 올라가는데 어떻게 다시 내려올 수 있나요?

그리고 converging하게 되어서 local minimum으로 converging 하게 된다고(??)하셨는데 말씀을 잘못하신건지 제가 잘못 들었던건지 이해가 잘 되지 않습니다ㅠㅠ

NAG는 local minimum을 지나서 gradient를 계산하는 것이 아니라 한번 지난 점에서 계산하기 때문에 local minimum이 한쪽 아래로 흘러가는 효과가 생긴다고 말씀하셨는데, 설명을 듣다 보니 momentum과 같은 말씀을 하신 걸로 이해를 했습니다. 이것도 설명해주실 수 있나요??

an+1:=β·an+∇f(xn−α·β·an)an+1:=β·an+∇f(xn−α·β·an)

xn+1:=xn−α·an+1



Q5. Bagging 과 Ensemble의 차이

Bagging이 ensemble기법 중 하나라고 알고 있습니다.

2. CNN 논문 리뷰 🧐

(1) VGG

발표자료 : https://github.com/Barleysack/BoostCampPaperStudy/blob/main/VGG.pdf

(2) Batch Normalization

발표자료 :
https://github.com/Barleysack/BoostCampPaperStudy/blob/main/BatchNormalization.pdf

1d-mlp

2d-conv2d.

dropout 몇개 빼도 된다고 하네요 이걸 쓰면.

3. 도메인 특강 🔎

NLP 엔지니어의 역량:

데이터 크롤링/전처리/관리/버저닝 능력 요구
뉴럴넷 아키텍쳐 이해도 필요! (cnn/rnn/transformer)
논문 수치 재현 능력도 필요하죠!
DevOps/MLOps
서빙을 위한 API 제작, 로그 관리/분석 능력

인상 깊었던 논문 On the Measure of Intelligence
Link : https://arxiv.org/pdf/1911.01547.pdf
Youtube : https://www.youtube.com/watch?v=3_qGrmD6iQY

NLP 논문 Attention is all you need
Link : https://arxiv.org/abs/1706.03762
Annotation : http://nlp.seas.harvard.edu/2018/04/03/attention.html
https://github.com/KimDaeUng/PLM-Implementation/tree/main/01_Vanilla-Transformer