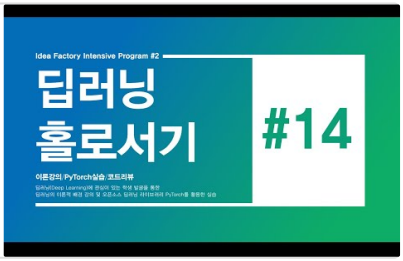
딥러닝 홀로서기#14

주제 : Hyperparameter Tuning Guide

링크 :

[#14.Lec] Hyperparameter Tuning Guide - 딥러닝 홀로서기
발표자료 링크 : https://github.com/heartcored98/Standalone-DeepLearning/blob/master/Lec3/Lec3-E.pdf자료 저장소 링크 : https://github.com/heartcored98/Standalone-DeepLearn...
https://youtu.be/-i8b-srMhGM



1. Purpose of Hyperparameter Tuning?

- Hyperparameter Tuning의 목적
  - Model의 Performance를 향상시키기 위해서
    - Model의 True Risk(Generalization Error)를 줄이기 위해서
      - Validation Set에 대한 True Risk를 줄이기 위해서

- Two Approach of Hyperparameter Tuning
  - Model Related
    - Model Capacity - Error 관점에서 Validation Loss가 증가하는 Optimal 한 시점, Capa. 값
    - Option
      - No. of hidden layer
      - No. of hidden unit
      - Activation Function : Non-Linear Act. Function 중 하나 선택
  - Optimization Related

    - Epoch - Error 관점에서 Validation Loss가 증가하는 Optimal한 시점, Epoch값
    - Option
        - Type of Optimizer : SGD, ADAM 등 어떤 Optimizer를 선택
        - Learning Rate
        - L2 Coef : L2 Norm의 Weight 비중 조절
        - Dropout Rate : 얼마나 높은 확률로 Neuron을 죽일 것인가
        - Batch Size : Batch Size
        - Epoch : 언제 끝낼 것인가

- Tunning Order
    - Model Related Hyperparameter Tuning 후 Optimization Related Tuning 수행

- 하이퍼파라메터 튜닝의 정답은 없기에 실무적 관점에서 주관적인 경험에 의해 정함
    - Model Related
        - MLP의 경우 하이퍼파라메터는 2~3개밖에 없음 CNN에서는 4~5개가 더 추가됨
        - Number of hidden layer
            - MLP의 경우 2~3개 넘어갈 경우 효율이 좋지 않다 (1~10 Layers까지 가능)
            - CNN의 경우 ~152, ~1000 Layers까지 가능
        - Number of hidden unit
            - 일반적으로 Layer당 1000~2048까지 Try가능
        - Activation Function
            - Sigmoid (X)
            - Tanh (X)
            - ReLU (O)

        - Leaky ReLU (O)
        - GeLU (O)
        - ELU ...
- Optimization Related
    - Type of Optimizer
        - 다음의 Optimizer들 중 좋은 성능을 나타내는 것
        - GD (X) : 데이터를 통째로 받고 매 Iteration마다 모든 데이터셋을 다 보고 업데이트하는 방식
        - SGD (Stocastic (O) : 데이터셋을 청킹을 하여 특정 덩어리를 랜덤으로 추출하여 업데이트하는 방식
        - RMSProp (O)
        - ADAM (O) : 가장 많이 사용하며, 복잡한 문제도 성능이 좋음
        - AdaDelta (O)
    - Learning Rate
        - Log Scale 단위 변경 : 10^{-5}~10^{-1}
        - Log Scale단위에서 괜찮은 Range가 발견되면 그때부터 Linear로 변경
            - 0.00001, 0.001, 0.001, 0.01, 0.1 → 0.001, 0.003, 0.006
    - L2 coef
        - Log Scale 단위 변경 : 10^{-5}~10^{5}
        - 문제 상황에 따라 Term의 Magnitude가 달라질 수 있으니 보정하기 위해 적절한 값을 입력
    - Dropout Rate
        - 0~1사이의 값인데, 일반적으로 0.1 ~ 0.5를 주로 쓴다
        - 만약, 모델이 너무 Complex한데 데이터가 부족할 경우, 0.7까지도 사용
    - Batch Size
        - Batch Size를 일단 키움 Out of Memory 발생하는 언저리까지

        - 이유 : GPU 가 같은 Batch에 대해서는 병렬로 연산이 되기 때문에, Batch Size와 상관없이 동일한 시간이 걸림 따라서, 최대한 키울 수록 좋음
    - Overfitting이 발생될 경우, Batch size를 줄여본다
    - 일반적으로 128, 256 ... 512사이의 값을 많이 씀 (2^n승으로 올리는 이유는 하나의 Batch가 메모리에 로딩될 때 2^n개씩 로딩되기 때문. 남으면 못씀)
- Epoch
    - Train Loss & Validation Loss를 정기적으로 모니터링
    - Validation Loss가 N Epoch을 기다려줬는 데도 감소되지 않으면 학습 종료

2. Four Way to Tune Experiment

- 하이퍼 파라미터 튜닝 4가지 방식
    - Grid Search : 찾고자 하는 하이퍼파라미터의 범주를 설정해서 적당히 Grid로 쭉 돌려보는 방식
    - Random Layout : 랜덤으로 하이퍼파라미터 스페이스에서 샘플링하여 Combination을 만든 후 돌려보는 방식
    - Hand Tuning : 메뉴얼로 하나하나 돌려보는 방식
    - Bayesian Optimization : 하이퍼파라미터 스페이스에서 그 다음 Combination을 추출할 때 Accuracy를 기록하여 Accuracy의 Tendency를 체크한 후, 통계적으로 Accuracy가 가장 높게 나올 수 있는 Range에서 샘플링하여 계속하여 Accuracy가 높은 하이퍼파라미터 Combination을 찾아내는 방식
    - Auto ML : Machine Learning을 Machine Learning으로 학습시키는 방식
- 권장 순서

  ① Grid Search : 하이퍼파라미터에 따른 전체적인 경향성 파악

  ② Random Layout : Random Search로 미처 알지 못한 괜찮은 조합 탐색

  ③ Hand Tuning : 논문 읽는 척하면서 계속 하이퍼 파라미터 튜닝

  ④ Bayesian Optimization : 어느 정도 파악이 되면 탐색하고 싶은 구간을 설정하고 오토 튜닝

3. Caution : Human Bias During Tuning Process

- 일반적으로 수행하는 과정
  1. Train, Validation, Test Set을 나누라고 시키길래 나누긴 나눔
  2. 학습을 한번 돌려보고 Test Acc를 확인한다
  3. 그럭저럭 시작치곤 나쁘지 않다
  4. 하이퍼파라미터를 조금 바꿔보고 Test Acc와 Train, Val Loss 그래프를 확인해본다
  5. 오버피팅이 감지되면 고칠 만한 부분을 생각하고 해당 하이퍼파라미터를 바꿔본다
  6. 다시 Test Acc와 Train, Val Loss
  7. 시행착오 끝에 Test Acc. 99% 달성~!

- 붉은 부분이 문제인데, 사람이 Test Set을 확인하며 학습하는 과정은 발생되어서는 안된다
  - 학습 과정중에는 Test Set이 아닌 Validation Set의 Accuracy를 가지고 하이퍼파라미터 튜닝을 해야한다
  - Validation Accuracy가 괜찮아 졌을때, 객관적인 평가를 하기 위해서 Test Set의 Accuracy를 확인한다