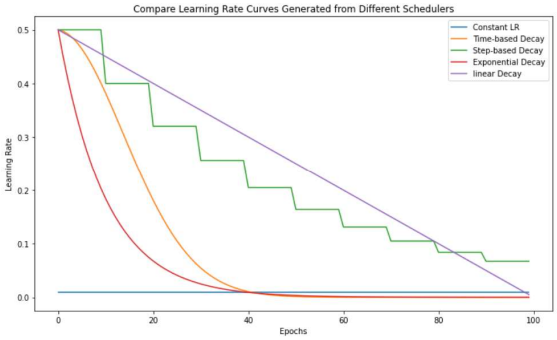
Tobig's 20기 정규세션 6주차 과제

NN Advanced 이론 과제

20기 황태연

1. Learning Rate Scheduler

: **학습률 스케줄러(Learning Rate Scheduler)**는 학습을 진행하면서 학습률을 변화시키는 기법이다. 학습률이 크면 학습 속도는 빠르지만 최적점에 수렴하기 어렵고 학습률이 작으면 학습 속도는 느리지만 최적점에 수렴하기 쉬워진다.



(그림 출처: https://neptune.ai/blog/how-to-choose-a-learning-rate-scheduler)

- 학습률 스케줄러에는 다음과 같은 종류가 있다.

(1) Linear Decay LR(Learning Rate) Scheduler

: 선형적으로 학습률이 감소하는 학습률 스케줄러.

$$\eta_t = \eta_0(1 - \frac{t}{T})$$

(η_t: LR at t epoch, η_0: initial LR, T: total epoch)

(2) Exponential Decay LR Scheduler

: 지수적으로 학습률이 감소하는 학습률 스케줄러. 자주 사용되는 스케줄러이다.

$$\eta_t = \eta_0 e^{-kt} \quad (k\text{: constant})$$

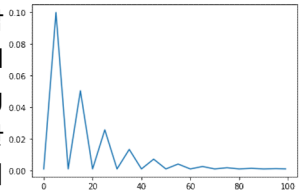
(3) Piecewise constant LR Scheduler

: 일정한 구간마다 학습률을 다르게 설정하여 학습률을 감소시키는 학습률 스케줄러. 다음 식은 Piecewise constant LR Scheduler의 예이다.

$$\eta_t = \eta_0 e^{-[t/10]}$$

(4) Cyclic LR Scheduler

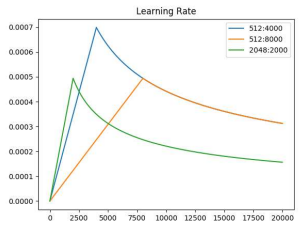
: 주기적으로 학습률을 증가시키고 감소시키는 학습률 스케줄러. Cyclical Learning Rates for Training Neural Networks 논문에서 제시한 방법으로, 학습률을 단조롭게 감소시키는 것보다 더 좋은 성능을 낸다고 한다.



(그림 출처: https://www.kaggle.com/code/isbhargav/guide-to-pytorch-learning-rate-scheduling/notebook)

(5) Noam LR Scheduler

: Transformer를 처음 제안한 Attention is All You Need 논문에서 사용했던 학습률 스케줄러. 초기에는 학습률이 점차 높아지면서 학습 초기에 빠르게 수렴하다가 일정 시점(약 4000 epochs)에서 학습률이 감소하도록 조정한다.



(그림 출처: https://nn.labml.ai/optimizers/noam.html)

2. Training Error와 Generalization Error 사이 간극을 줄이는 방안

: **훈련 오차(Training Error)**는 훈련 데이터로 모델을 학습시킬 때의 오차를 의미하고, **일반화 오차(Generalization Error)**는 훈련 데이터가 아닌 새로운 샘플 데이터에 대한 오차를 의미한다. 일반화 오차를 정확히 도출하는 것은 불가능하므로, 검증 오차(Validation Error)와 시험 오차(Test Error)를 이용하여 일반화 오차에 대한 추정값을 얻는다. 일반적으로 검증 오차와 시험 오차를 줄이는 것이 일반화 오차를 최소화하는 데에 도움이 된다.

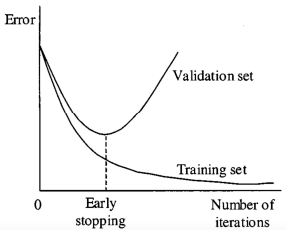
훈련 오차와 일반화 오차 사이의 간극이 크면 훈련 데이터에 과적합(오버피팅, Overfitting)되었다고 볼 수 있다. 간극을 줄이는 것이 머신러닝의 주된 목표 중 하나이다.

- 훈련 오차와 일반화 오차 사이의 간극을 줄이는 방법

(0) 정규세션 강의에서는 다음과 같은 방법을 사용할 수 있다고 소개하고 있다: **가중치 규제(Regularization)**, **드롭아웃(Dropout)**, **데이터 증강(Data Augmentation)**. 이를 제외하고 새로운 방법을 알아보자.

(1) 조기 종료(Early Stopping)

: 조기 종료는 일반화 오차의 추정값인 검증 오차(Validation Error)가 증가하기 시작하면 적절한 때에 학습을 중단하는 기법이다. 일반적으로 k epoch 동안 검증 오차가 감소하지 않으면 학습을 멈춘다.



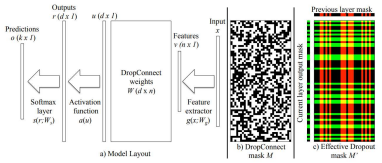
(그림 출처: https://www.researchgate.net/figure/Early-stopping-based-on-cross-validation_fig1_3302948)

(2) 모델의 복잡도 줄이기 (파라미터 수 줄이기)

: 모델의 복잡도가 높아지면 훈련 데이터의 잡음(Noise)까지 학습할 가능성이 있어서 새로운 데이터에 대한 오차가 커져 과적합이 될 수 있다. 따라서 과적합을 막기 위해 모델의 복잡도를 줄이거나 파라미터 수를 줄이는 시도를 할 수 있다.

(3) DropConnect

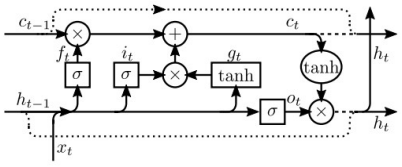
: Regularizarion of Neural Networks using DropConnect 논문에서 제시한 기법으로, Dropout은 일부 노드를 제외하고 학습하는 반면 DropConnect는 일부 가중치를 0으로 차단하고 학습한다. Dropout의 일반화라고 볼 수 있다.



(그림 출처: Regularization of Neural Networks using DropConnect)

(4) Zoneout

: Zoneout: Regularizing RNNs by Randomly Preserving Hidden Activations 논문에서 제시한 기법으로, LSTM에서 사용되는 드롭아웃 기반의 정규화 기술이다. Dropout과 유사한 효과를 LSTM에서도 적용할 수 있다.



(그림 출처: Zoneout: Regularizing RNNs by Randomly Preserving Hidden Activations)