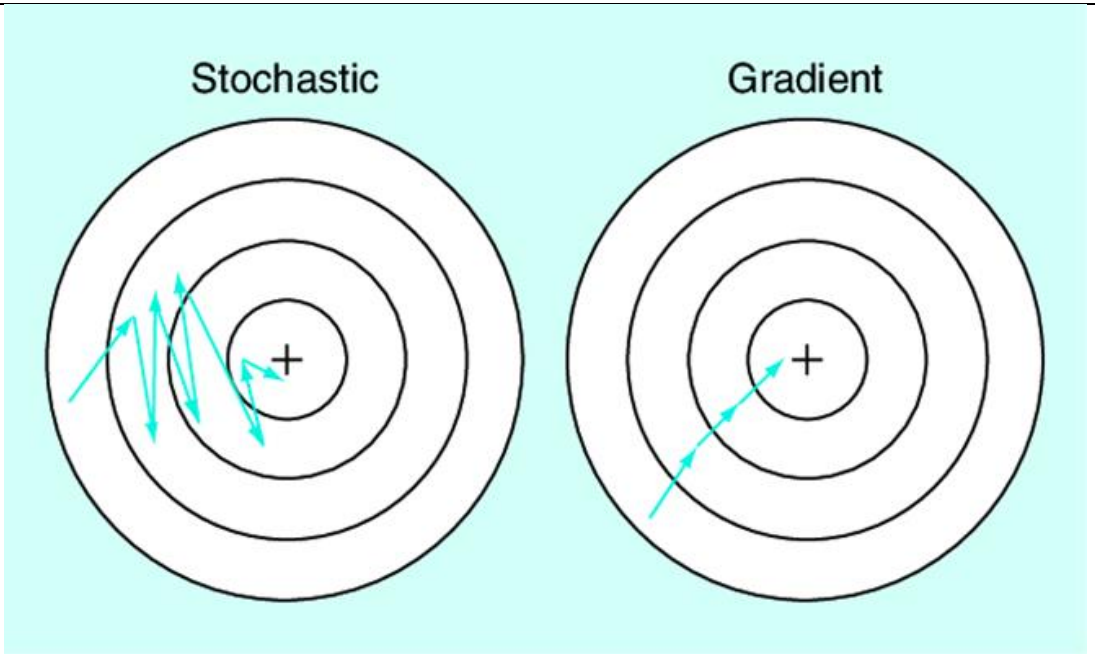
3차 일석이조 조별보고서	
작성일 : 2023년 10월 4일	작성자 : 이학빈
조 모임 일시 : 10월 3일	모임 장소 : 구글미트
참석자 : 이준용, 유정훈, 김동규, 이학빈, 탁성재	조원 : 이준용, 유정훈, 김동규, 이학빈, 탁성재

구분	내용
학습 범위와 내용	5주차 온라인 강의 내용
논의 내용 (조별 모임 전에 조장이 지시)	2.3. 최적화 2.3.1. 매개변수 공간의 탐색 2.3.2. 미분 2.3.3. 경사 하강 알고리즘
질문 내용 (모임 전 공지된 개별 학습 범위에서 이해된 것과 못한 것)	Q(1): SGD와 BGD의 개념이 확실하게 잡히지 않아 팀원들과 함께 더 깊숙히 논의해보았다. A(1) 확률적 경사 하강법(SGD) 확률적 경사 하강법(SGD)은 가장 기본적으로 사용되는 최적화 알고리즘 중 하나로, 손실 함수를 최소화하는 방향으로 가중치를 조정하는 방법이다. 각 학습 단계마다 무장위로 선택한 미니 배치(mini-batch)의 데이터를 사용하여 가중치를 업데이트한다. SGD의 작동원리는 1. 랜덤하게 초기화된 모델 가중치를 설정 2. 학습 데이터에서 미니 배치를 무장위로 선택 3. 선택한 미니 배치를 사용하여 모델의 출력과 정답 사이의 손실을 계산 4. 손실을 사용하여 가중치 업데이트 5. 위 단계를 반복하여 모든 학습 데이터에 대해 가중치를 업데이트함

	SGD의 하이퍼파라미터 - 학습률(learning rate): 가중치를 업데이트 할 때 적용되는 스케일 조정 계수. 학습률이 너무 작으면 수렴이 느려지고, 너무 크면 발산할 수 있다. 보통 0.1과 0.001 사이의 값을 사용한다. - 모멘텀(momentum): 가중치 업데이트에 이전 업데이트의 영향을 추가하는 계수입니다. 모멘텀을 사용하여 SGD의 수렴 속도가 향상될 수 있다. - 가중치 감소(weight decay): 오버피팅을 방지하기 위해 가중치 값이 작아지도록 하는 계수. 보통 0.001~0.0001사이의 값 사용한다. SGD 장점: - 메모리 사용이 적다 - 대규모 데이터 셋 학습이 가능 SGD 한계 - 수렴 속도가 느리고, 지역 최소값에 빠지기 쉬움 - 하이퍼파라미터를 적절히 조정해야 최적의 값이 나옴 Q(2) 야코비안, 헤시안 행렬이란? A(2). 결과를 추출하기 위한 퍼셉트론의 함수 f(x)를 이용해 만들어지는 행렬. f(x)가 가지는 각 벡터들에 관한 식에 대하여 편미분, 거기에 한번 더 편미분을 시행해 구할 수 있다. 각 벡터들은 여러 조건들을 가지고 만들어져 있으며, 어떠한 변수x에 의해 발생하는 변화율을 확인할 수도 있다. f(x)는 x에 갯수에 따른 차원으로 표현될 수 있는데, 해당 차원은 일반적인 선형 공간이 아닌 비선형 공간을 가질 수도 있다. 하지만 야코비안 행렬을 이용하면 f(x)의 차원을 유사한 선형 공간으로 변형시켜 계산

	을 조금 더 쉽게 만들 수 있다. Q(3) 경사 하강법의 또 다른 종류에 대해 공부해보았습니다. A(3) 1. 배치 경사 하강법: 배치 경사 하강법(Batch Gradient Descent)은 가장 기본적인 경사 하강법으로 Vanilla Gradient Descent 라고 부르기도 합니다. 배치 경사 하강법은 데이터셋 전체를 고려하여 손실함수를 계산합니다. 배치 경사 하강법은 한 번의 Epoch 에 모든 파라미터 업데이트를 단 한 번만 수행합니다. 즉, Batch 의 개수와 Iteration 은 1 이고 Batch size 는 전체 데이터의 개수입니다. 파라미터 업데이트할 때 한 번에 전체 데이터셋을 고려하기 때문에 모델 학습 시 많은 시간과 메모리가 필요하다는 단점이 있습니다. 2. 확률적 경사 하강법: 확률적 경사 하강법(Stochastic Gradient Descent)은 배치 경사 하강법이 모델 학습 시 많은 시간과 메모리가 필요하다는 단점을 개선하기 위해 제안된 기법입니다. 확률적 경사 하강법은 Batch size 를 1 로 설정하여 파라미터를 업데이트하기 때문에 배치 경사 하강법보다 훨씬 빠르고 적은 메모리로 학습이 진행됩니다.



위의 **그림 2** 는 경사 하강법 종류에 따라 최적의 해를 찾아가는 과정을 시각화한 자료입니다. 좌측 은 확률적 경사 하강법을, 우측은 배치 경사 하강법을 활용한 경우입니다. 확률적 경사 하강법은 파라 미터 값의 업데이트 폭이 불안정하기 때문에 배치 경사 하강법보다 정확도가 낮은 경우가 생길 수도 있습니다. 그럼에도 불구하고, 하나의 데이터(Batch size=1)에 대해서만 손실함수를 계산하고 파라미터를 업데이트하면 되기 때문에, 적은 시간과 메모리로도 모델을 학습시킬 수 있다는 장점이 있습니다.

1. 미니 배치 경사 하강법: 미니 배치 경사 하강법(Mini-Batch Gradient Descent)은 Batch size 가 1 도 전체 데이터 개수도 아닌 경우를 말합니다. 미니 배치 경사 하강법은 배치 경사 하강법보다 모델 학습 속도가 빠르고, 확률적 경사 하강법보다 안정적인 장점이 있습니다. 덕분에, 딥러닝 분야에서

	가장 많이 활용하는 경사 하강법입니다. Batch size 는 일반적으로 32, 64, 128 과 같이 2 의 n 제곱에 해당하는 값으로 사용하는 게 보편적입니다. Q(4) 수업에서 소개된 경사하강 알고리즘의 핵심 원리에 대한 보충 설명이 필요합니다. A(4) 경사 하강 알고리즘은 머신 러닝과 최적화에서 중요한 역할을 하는 알고리즘 중 하나로, 함수의 최솟값(또는 최대값)을 찾기 위해 사용됩니다. 주로 비용 함수(cost function)를 최소화하기 위해 모델의 파라미터를 조정하는 데 사용됩니다. 이 알고리즘의 핵심 원리는 다음과 같습니다: 1. 초기화: 경사 하강 알고리즘을 시작하기 전에 모델의 파라미터를 초기화합니다. 이 초기 파라미터 값은 주로 무작위로 선택하거나, 다른 방법을 사용하여 설정합니다. 2. 비용 함수 계산: 초기 파라미터 값을 사용하여 비용 함수를 계산합니다. 비용 함수는 모델의 예측값과 실제 값 사이의 오차를 측정하는 함수입니다. 목표는 이 비용 함수를 최소화하는 것입니다. 3. 기울기(Gradient) 계산: 비용 함수의 기울기(Gradient)를 계산합니다. 기울기는 비용 함수가 어떻게 변화해야 하는지를 나타내는 벡터로, 각 파라미터에 대한 편미분값을 포함합니다. 4. 파라미터 업데이트: 기울기를 사용하여 파라미터를 업데이트합니다. 경사 하강은 현재 파라미터 위치에서 기울기의 반대 방향으로 조금씩 움직여 비용 함수를 줄이는 방향으로 진행합니다. 업데이트할 때 사용되는 학습률(learning rate)은 각 업데이트 단계에서 얼마나 크게 이동할지를 결정합니다.

	5. 반복: 위의 단계를 여러 번 반복합니다. 주어진 에포크(epoch) 또는 일정한 조건이 충족될 때까지 반복하여 파라미터를 계속 업데이트합니다. 이 과정을 통해 파라미터가 최적의 값을 찾게 됩니다. 6. 종료 조건: 종료 조건을 설정하여 알고리즘이 멈추는 시점을 결정합니다. 종료 조건은 일정한 에포크 수, 기울기의 크기, 또는 비용 함수 값의 변화량 등을 고려할 수 있습니다. 경사 하강 알고리즘은 최적화 문제를 해결하는 데 사용되며, 모델 학습, 신경망 훈련 등 다양한 머신 러닝 작업에서 중요한 역할을 합니다. 학습률과 초기 파라미터 설정 등 하이퍼파라미터의 조정이 경사 하강의 성능에 영향을 미치므로 조심스럽게 설정해야 합니다. Q(5) 매게변수 공간의 탐색 부분에서 낱낱탐색, 무작위 탐색 이외의 최적화 문제 해결에 대해 알아보았습니다. A(5) 어떤 알고리즘을 선택할지는 문제의 특성과 매개변수 공간의 형태에 따라 다를 수 있습니다. 1. 그리드 탐색(Grid Search): 그리드 탐색은 매개변수 공간을 일정한 간격으로 나누고, 각 조합에 대해 모델을 학습 및 평가하여 최적 조합을 찾는 방법입니다. 간단하고 직관적이지만, 탐색 공간이 크면 계산 비용이 급증할 수 있습니다. 2. 에볼루션 기반 최적화: 에볼루션 기반 최적화 알고리즘은 생물학적 진화 원리를 모방하여 최적화를 수행합니다. 대표적인 알고리즘으로는 유전 알고리즘(Genetic Algorithm)이 있습니다. 다양한 매개변수 조합을 시도하고, 가장 적합한 조합을 선택하는 데 사용됩니다. 3. 베이지안 최적화: 베이지안 최적화는 베이지안 통계를 기반으로 하여 매개변수 공간을 조사하고 가장 가능성 있는 후보들을 선택하는 방법입니다. 비선형 및 복잡한 최적화 문제에서 효과적입니다.

	4. 자동화된 하이퍼파라미터 튜닝: 자동화된 하이퍼파라미터 튜닝 라이브러리(예: Hyperopt, Optuna, BayesianOptimization 등)를 사용하여 최적화를 자동화할 수 있습니다. 이러한 라이브러리는 다양한 최적화 알고리즘을 지원하며, 계산 비용을 줄일 수 있습니다. 매개변수 공간의 탐색에서 가장 적절한 방법을 선택하려면 문제의 특성, 계산 자원, 시간 제약 등을 고려해야 합니다. 또한 다양한 알고리즘을 조합하여 혼합 최적화 방법을 사용하기도 합니다.
질문내용	Q(1) SGD의 중요한 하이퍼파라미터를 적절하게 조정하면 학습의 수렴 속도와 결과에 영향을 미칠 수 있다고 알고 있습니다. 모델을 훈련 중인 경우, 각 데이터 세트와 모델에는 다양한 하이퍼파라미터 세트가 필요합니다. 하이퍼파라미터를 구할때 우리는 수동으로 직접 값을 설정하면서 하이퍼파라미터를 구하는 것을 하이퍼파라미터라고 생각하는데, 자동으로 값을 변경해서 구해주는 함수, 알고리즘, 라이브러리를 사용해서 구하는 것도 하이퍼파라미터라고 봐야할까요? Q(2) g= dΘ=∂J/∂Θ이고 p= 학습률일 때, Θ=Θ-pg에서 pg를 적절히 조절해야 한다고 하셨는데 정확한 이해가 되지 않습니다.