|  |  |  |
| --- | --- | --- |
| 7주차 일석이조 조별보고서 | | |
| 작성일 : 2023년 10월 17일 | | 작성자 : 이준용 |
| 조 모임 일시 : 10월 17일 | | 모임 장소 : 구글meet |
| 참석자 : 이준용, 유정훈, 김동규, 이학빈, 탁성재 | | 조원 : 이준용, 유정훈, 김동규, 이학빈, 탁성재 |
| 구분 | 내용 | |
| 학습 범위와 내용 | 1. 7주차 온라인 강의 내용  2. 교재 3장 내용 | |
| 논의 내용  (조별 모임 전에 조장이 지시) | 3.4.1 목적함수의 정의  3.4.2 오류 역전파 알고리즘 설계  3.4.3 오류 역전파를 이용한 학습 알고리즘  3.5 미니배치 스토캐스틱 경사 하강법  3.6 다층 퍼셉트론에 의한 인식  3.7 다층 퍼셉트론의 특성 | |
| 질문 내용  (모임 전 공지된 개별 학습 범위에서 이해된 것과 못한 것) | **Q1**  미니배치 스토캐스틱 경사하강법(minibatch SGD)에서는 미니 배치를 무작위로 선별해서 훈련시킨다고 알고 있다. 무작위로 선별할때 선택되지 않은 훈련 집합도 존재할 것인데 이 때의 문제는 무엇인가?  **A1**  미니배치는 무작위로 뽑기 때문에 선택되지 않은 훈련 집합이 존재하므로, 데이터 전체의 경향을 반영하기 힘들다.  그래서 업데이트를 꼭 좋은 방향으로만 하지 않아 현재 학습을 진행하는 데이터 한 개에 대해서는 cost function의 값이 줄어들더라도, 이로 인해 다른 데이터에 대해서는 비용(cost)가 증가할 수 있다. 갈팡질팡 헤매는 경우가 생길 수 있다.  배치와 스토캐스틱 방식에서는 모든 샘플을 사용하지만 너무 느리고, 데이터 한 개를 사용하면 빠르지만 너무 헤맨다.  이러한 문제는 수십 개부터 많게는 수백 개의 데이터를 한 그룹으로 하여 처리함으로써, 적당한 속도로 전체 데이터를 최대한 반영함으로써 해결한다.  **Q2**  팀원들과 https://playground.tensorflow.org/에서 여러 파라미터와 데이터를 조정해보면서 딥러닝의 학습에 대해서 공부해보자!  **A2**  **data1 → 학습률: 0.0001 batch size: 10 활성함수: ReLU 문제: 회귀 Regression feture: 2개 은닉층: 0**    **학습률: 0.0001 batch size: 10 활성함수: ReLU 문제: 분류 classifiction feture: 2개 은닉층: 2 layer**  **은닉층과 학습 뉴런을 늘릴 경우 잘 맞춤. 학습이 잘됨.**      **학습률: 0.0001 batch size: 10 활성함수: ReLU 문제: 분류 classifiction feture: 2개 은닉층: 2 layer**    **학습률: 0.0001 batch size: 10 활성함수: ReLU 문제: 분류 classifiction feture: 2개 은닉층: 2 layer**  **data가 구분이 잘 되어있는 경우 학습이 잘되는 것을 확인함.**    **학습률: 0.0001 batch size: 10 활성함수: ReLU 문제: 분류 classifiction feture: 2개 은닉층: 2 layer**  **다음과 같은 나선형 data에서는 은닉층과 feature수가 적어서 분류를 잘 못함.**    **이후 여러가지 파라미터를 조정해보면서 더 공부해보았습니다.**  **Q3**  역전파 오류 알고리즘이란?  **A3**  퍼셉트론은 해당 방향을 구하기 위해 가중치를 계속 새로 갱신합니다. 역전파 오류 알고리즘은 해당 계산에 사용되는 대표적인 알고리즘입니다.  우선 기존에 주어진 가중치를 각 은닉층에 적용시키고, 받은 입력값들을 출력값으로 계산합니다. 그에 따른 결과값들은 가중치와 변수의 곱, 그 결과들의 합으로 계산됩니다. 이후 결과값들은 실제값과의 에러의 크기를 구하게 됩니다.    이때 각 가중치에 대한 오차는 해당 가중치에 대한 편미분으로 분리시킬 수 있으며, 기존 가중치에서 해당 값들을 빼는 형태로 갱신이 이루어집니다.  해당 내용을 학습 횟수만큼 반복해서 최적해를 찾아나갈 수 있습니다.  **Q4**  미니배치 스토캐스틱 경사하강도가 무엇인가요?  **A4**  배치란 네트워크에서 학습을 시킬 때 사용되는 데이터의 그룹을 의미한다. 만약 데이터가 1000개가 있다고 했을 때 한 번 학습을 시행 할 때 1000개의 데이터를 모두 넣어 학습을 시킨다면 그럴 경우엔 Full batch , 한 번에 10개의 그룹씩의 데이터를 10번씩 넣어 학습을 시키겠다면 그럴 경우엔 mini batch 라고 이야기 한다. 이 때 mini batch의 경우에는 한 mini batch 데이터셋이 학습될 때 마다 가중치가 업데이트 된다. 네트워크를 학습시킬 때 우리가 도착하고자 하는 Global minimum 은 결국 전체 데이터셋에 대한 Global minimum이다. 하지만 mini batch를 이용한 경사하강법을 실생활에서는 더욱 많이 쓰고 선호한다.    그것은 데이터의 크기가 커질 수록 전체 데이터를 한 번 학습할 때 시간이 매우 오래 소요될 것이다. 좀 더 자세하게 가정을 해본다면  1. 데이터 1억개가 있으며 1억개를 모두 학습 할 때까지 걸리는 시간은 10시간 소요  2. 전체 데이터가 global minimum 에 도착하는데까지 필요한 경사 하강 수는 10회  가정이 위와 같을 경우 Full batch를 이용하여 데이터를 학습하게 된다면 1 epoch (10시간) \* 10회이니 100시간이 걸려야만 Global minimum 에 도달할 수 있다. Full batch 를 사용하게 된다면 한 번 경사하강을 통해 내려가는 정도는 클 지 언정, 학습까지의 시간이 너무나도 오래 걸리는 것이다. 이번엔 동일한 데이터셋으로 mini batch를 이용해보자  1. 1억개의 데이터를 1000개의 batch size를 가진 10개의 batch set으로 나눠 학습  2. 하나의 batch set을 학습할 때까지 걸리는 시간은 1시간 소요  3. 전체 데이터가 Global minimum에 도착하는데까지 필요한 경사 하강 수는 30회  mini batch set을 가지고 학습을 하게 된다면 전체 데이터가 아닌 소량의 데이터 셋 그룹인 batch set 을 가지고 global minimum을 찾으려고 하니 필요한 경사 하강수는 증가하였지만, 개별 batch set 을 학습하는데 시간이 매우 줄어들어 전체 걸리는 시간은 1 epochs (1시간) \* 30회 = 30시간으로 매우 줄어들게 되는 것이다.  이 때 mini batch size 가 1일 경우, 즉 하나의 데이터를 조금씩 넣을 경우에는 stochastic Gradeint Descent (SGD, 확률적 경사하강법) 이라고 하고 mini batch size 가 2개 이상일 경우엔 mini batch Gradient Descent, 전부를 사용할 경우엔 Batch Gradient Descent 라고 한다.  확률적 경사하강법에서의 확률적이란, 선택되는 데이터 하나를 확률적으로 랜덤하게 뽑기 때문에 확률적 경사하강법이라고 한다.  mini batch gradient descent의 경우에는 움직임이 Batch Gradient Descent에 비해서 불안정한 모습을 볼 수 있는데 그 이유는 mini batch 들의 Global minimum이 전체 데이터의 Global minimum 과 차이가 있을 수 있기 때문이다.  **Q5**  휴리스틱이란?  **A5**  휴리스틱(Heuristic)은 문제 해결이나 결정을 내릴 때 사용되는 규칙 또는 방법론을 의미합니다. 휴리스틱은 일반적으로 빠른 결정을 내리기 위해 경험에 기초한 규칙 또는 간략화된 절차를 사용하는 것을 의미하며, 항상 최적의 해결책을 보장하지는 않습니다. 복잡한 문제나 정보 부족 상황에서 사용되며, 복잡한 문제를 단순화하거나 분해하여 해결책을 빠르게 찾기 위해 사용되고 정보가 제한된 상황에서 최선의 결정을 내릴 때 사용됩니다. 그리고 기계 학습 및 인공 지능에서 데이터를 기반으로 패턴을 발견하고 결정을 내릴 때 사용됩니다.  휴리스틱의 특징  1. 복잡한 문제를 단순화하고 빠르게 해결할 수 있는 방법을 제공합니다. 이것은 실시간 의사 결정, 긴급 상황에서의 빠른 대응, 생산 프로세스의 효율화, 고객 서비스 개선 등 다양한 상황에서 중요합니다.  2. 필요한 정보를 수집하거나 처리할 시간과 자원이 없을 수 있습니다. 제한된 정보나 자원으로도 합리적인 결정을 내릴 수 있도록 도와줍니다.  3. 일부 문제는 최적의 해결책을 계산하기 어렵거나 시간이 너무 오래 걸릴 수 있습니다. 문제를 간단화하고 관리 가능한 부분 문제로 분해함으로써 이러한 상황에서 도움을 줍니다.  4. 기계 학습 알고리즘은 휴리스틱을 활용하여 패턴 인식, 분류, 예측 등을 수행합니다. 휴리스틱은 이러한 기술을 지원하며 빅데이터 분석 및 자동 의사 결정 시스템의 핵심 구성 요소입니다.  휴리스틱은 다양한 도메인에서 사용되며, 예를 들어 경영, 공학, 의학, 컴퓨터 과학 등 다양한 분야에서 적용됩니다. 그러나 항상 최적의 해결책을 찾아내지는 못할 수 있으므로 상황에 따라서 다른 방법이나 최적화 알고리즘과 함께 사용되기도 합니다.  그러나 휴리스틱을 사용할 때 주의가 필요합니다. 완벽한 최적화나 정확한 결정을 필요로 하는 상황에서는 휴리스틱만 사용하는 것이 적절하지 않을 수 있으며, 추가적인 검토 및 분석이 필요할 수 있습니다. 또한 휴리스틱은 경우에 따라 오류나 편향을 가질 수 있으므로 신중한 선택과 상황에 맞는 적절한 조절이 필요합니다.  **Q6**  휴리스틱과 딥러닝의 관계  **A6**  휴리스틱과 딥러닝은 모두 문제 해결과 패턴 인식 분야에서 사용되는 방법론이지만 서로 다른 방식으로 작동하며 서로 다른 용도에 적합합니다. 이들 간의 관계는 다음과 같이 설명할 수 있습니다:  1. 작동 원리:  • 휴리스틱: 휴리스틱은 주로 경험적이고 근사적인 규칙 또는 전략을 사용하여 문제를 해결합니다. 휴리스틱은 문제 도메인에 대한 지식과 경험에 기반하며, 최적 솔루션을 찾는 대신 적절한 근사치에 가까운 결과를 빠르게 얻는 데 중점을 둡니다.  • 딥러닝: 딥러닝은 인공 신경망과 같은 기계 학습 기술의 하위 집합입니다. 딥러닝은 데이터 기반 학습을 통해 모델을 훈련하고 복잡한 패턴을 인식하는 데 사용됩니다. 딥러닝 모델은 보다 많은 데이터를 필요로 하며, 기본적으로 최적 솔루션을 찾으려고 시도합니다.  2. 문제 복잡성:  • 휴리스틱: 휴리스틱은 상대적으로 간단하거나 중간 정도의 복잡성을 가진 문제에 적합합니다. 주로 실행 시간을 단축하고 실용적인 솔루션을 찾는 데 사용됩니다.  • 딥러닝: 딥러닝은 고도로 복잡한 문제와 대규모 데이터셋을 다루는 데 더 적합하며, 이미지 인식, 음성 인식, 자연어 처리 및 복잡한 패턴 인식과 관련된 작업에 특히 유용합니다.  3. 데이터 요구:  • 휴리스틱: 휴리스틱은 대부분 데이터에 의존하지 않거나 작은 양의 데이터를 사용합니다. 대신 도메인 전문 지식에 의존합니다.  • 딥러닝: 딥러닝은 대량의 데이터가 필요하며, 효과적인 결과를 얻기 위해 많은 양의 훈련 데이터가 필요합니다.  4. 적용 분야:  • 휴리스틱: 휴리스틱은 주로 제한된 데이터와 지식을 기반으로 문제를 해결하는 경우에 사용됩니다. 예를 들어, 조립 라인 스케줄링, 전자 게임의 게임 플레이, 경로 최적화 등에 적용됩니다.  • 딥러닝: 딥러닝은 대규모 데이터셋을 처리하고 복잡한 패턴을 인식하는 경우에 주로 사용됩니다. 이를테면, 이미지 분류, 음성 인식, 자연어 처리, 자율 주행 자동차 및 의료 진단 등에 사용됩니다.  딥러닝과 휴리스틱은 서로 보완적이며, 특정 문제에 어떤 접근 방식이 더 효과적인지는 문제의 성격과 요구 사항에 따라 다릅니다. 때로는 이 두 가지를 결합하여 최상의 결과를 얻는 것도 가능합니다. | |
| 질문내용 | Q: playground.tensorflow.org 사이트를 이용해서 학습해보면서 배치 사이즈를 줄일수록, test 오차가 커지는 현상을 발견할 수 있었는데, 학습률과 배치사이즈와의 상관관계에 대해 잘 이해가 되지 않습니다.   * 작은 batch size와 낮은 learning rate * 큰 batch size와 큰 learning rate   위 두가지 경우일 때 왜 학습이 잘 되는 것인가요? 아니면 이것도 어떤 데이터를 사용하는 것에 따라 달라질 수 있는 문제인가요? | |