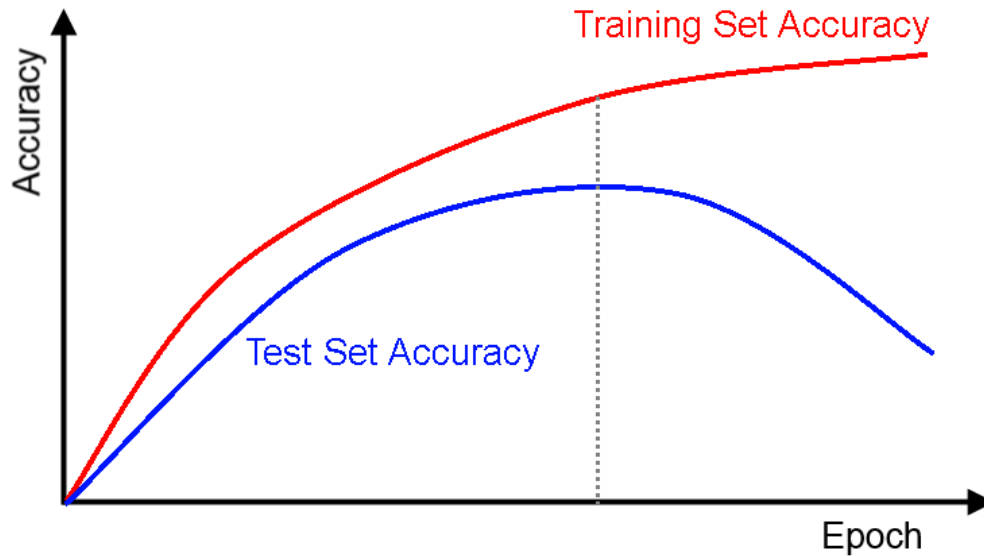


시스템&솔루션개발

[2022년 #차]

문항1) 이미지를 분류하는 인공신경망 모델을 구성하였다. 학습이 진행됨에 따라 분류 정확도가 다음과 같이 나타날 때, 문제점을 올바르게 진단하고 해결책을 제시하지 않은 것을 고르시오. [3점]



- ① 학습데이터에 대한 과대 적합이 발생하고 있으니, 모델의 capacity를 높여서 더 정밀하게 데이터를 분류할 수 있도록 해야 한다
- ② 일반화 성능이 떨어지므로 학습 과정에서 일정 비율의 인공 뉴런을 무작위로 비활성화시켜 일반화 성능을 높여야 한다
- ③ 데이터의 다양성이 떨어지는 것으로 판단되므로, 이미지 변조를 통해 데이터를 증강시켜 다양성을 높여야 한다
- ④ 학습을 하면 할수록 오히려 성능이 나빠지고 있기 때문에 적절한 시점에 학습을 중단시켜야 한다

(정답) 1

(해설) Overfitting 이 발생하고 있음. 모델의 Capacity 를 낮춰야 함

문항2) 적용할 AI 모델을 최종적으로 결정하기 위해서는 가용한 데이터를 목적에 따라 일정 비율로 나눈 후 학습, 검증 그리고 테스트를 진행한다. 각 데이터 셋에 대한 설명 중 잘못된 것을 고르시오. [3점]

- ① Training set 으로 후보 모델들 각각에 대한 최적의 파라미터를 찾는다
- ② 학습한 모델들 중에서 가장 성능이 좋은 최종 모델을 결정하기 위해 Validation set 을 사용한다.
- ③ Training set 으로 학습한 모델을 Validation set 으로 돌려보면서 튜닝하기를 반복한다
- ④ Test set을 돌려서 최종 모델이 실환경에서 어떤 성능을 보일지 예측한다
- ⑤ 최종 모델의 실환경에서의 성능까지 예측한 후에는 더 이상 학습을 시키지 않는다.

(정답) 5

(해설) 최종 모델의 실환경에서의 성능까지 예측한 후에는 Training set, Validation set, Test set 를 합쳐서 선택된 모델을 재학습시켜 추가 튜닝을 진행할 수 있다.

문항3) 시계열 데이터 딥러닝에 대한 설명 중 잘못된 것을 고르시오. [3점]

- ① LSTM 모델은 RNN 모델과 달리 오래된 데이터를 잘 기억할 수 있으므로 길이가 긴 시계열 데이터를 다루기에 적합한 모델이다
- ② Attention 만을 이용하여 시계열 데이터를 학습하도록 모델을 구성하면 병렬 처리 학습이 가능하여 학습 속도가 높아진다
- ③ RNN 모델은 CNN 모델과 달리 시계열 데이터를 학습할 수 있다
- ④ RNN, LSTM 모델은 timestamp 가 길어지면 학습 시간이 크게 증가하는 단점이 있다
- ⑤ LSTM 모델은 RNN 모델과 달리 과거 정보를 얼마나 기억할지, 현재 정보를 얼마나 반영할지 그리고 현재 시점에 계산된 최종 값을 다음으로 얼마나 넘길 지를 조절할 수 있다

(정답) 3

(해설) CNN 도 RNN 모델처럼 시계열 데이터를 학습할 수 있는 모델이다.

문항4) 딥 러닝을 활용하여 솔루션을 개발할 때 데이터 라벨링에 들어가는 비용을 최소화하면서 모델의 성능을 극대화하고자 한다. 학습할 데이터를 선별하는 방식으로 옳은 것을 모두 고르시오. (2개) [3점]

- ① 여러 개의 딥 러닝 모델로 추론한 결과 불일치가 많은 데이터를 선별한다
- ② 딥 러닝 모델이 추론한 결과 판정점수(confidence score)가 높은 데이터를 선별한다
- ③ 학습데이터로 추가했을 때 가중치 값이 가장 많이 변화하는 데이터를 선별한다
- ④ 추론한 결과와 라벨 값이 틀린 데이터를 선별한다
- ⑤ 데이터 전문가의 판단으로 중요도가 높은 데이터를 선별한다.

(정답) 1, 3

(해설)

- 1) Query by Committee - 여러 모델의 추론 결과 불일치 많은 데이터
- 2) Uncertainty Sampling - 판정점수가 낮은 불확실한 데이터
- 3) Expected Impact - 임팩트가 큰 데이터로 성능 높이기
- 4) 라벨링할 데이터를 고르는 과정이므로 라벨 값 없음
- 5) 라벨링에 들어갈 비용을 최소화하기 위해 데이터를 고르는 과정인데, 데이터 전문가가 데이터를 고르는 것은 비용이 올라감

문항 5) 구글 Cloud Functions 는 클라우드 인프라와 서비스에서 발생하는 이벤트에 연결되는 단일 목적의 서비스를 제공할 수 있는 완전 관리형 서버리스 실행 환경이다. Cloud Functions 를 설계하거나 구현하는데 있어 고려해야할 사항으로 잘못된 것을 모두 고르시오.(2 개) [3 점]

- ① 함수에서 사용할 수 있는 임시 파일을 유지할 경우 메모리 누수와 콜드스타트를 유발할 수 있으므로 명시적으로 삭제해야 한다.
- ② Event 기반 함수의 경우 함수 호출 후 Event가 삭제되므로, 안전한 실행 보장을 위해서는 재시도 정책을 설정하고 함수의 멍등성을 구현해야 한다.
- ③ 함수 호출 간에 State(상태)가 유지된다는 보장이 없으므로, 전역변수 등을 사용하여 선/후 호출간의 정보를 공유하지 않도록 구현해야 한다.
- ④ 함수내 백그라운드 활동은 후속 함수 호출을 방해할 수 있으므로, 함수 종료 전에 모든 비동기식 작업이 완료되도록 처리해야 한다.
- ⑤ Network 연결을 이용하는 Cloud Functions는 독립적인 결과를 보장받기 위해 지속적으로 연결된 Network의 재사용을 피해야 한다.

(정답) 3, 5

(해설)

3) 다음 호출에 대비해 Cloud Functions의 상태가 유지된다는 보장은 없지만 Cloud Functions는 종종 이전 호출의 실행 환경을 재활용한다. 전역 범위에서 변수를 선언하면 다시 연산할 필요 없이 후속 호출에서 변수 값을 재사용한다. 전역 범위에서 네트워크 연결, 라이브러리 참조, API 클라이언트 객체를 캐싱하는 것이 특히 중요하다.

5) Network의 지속적인 연결은 권고 사항이며, 지속적인 연결 사용시 각 함수 호출 시 새로 연결하는 데 사용하는 CPU 시간을 절감하고, 연결 또는 DNS 할당량이 부족해질 가능성이 줄어든다.

문항6) Amazon API Gateway 서비스에서 제공하는 HTTP, REST, WebSocket API 중 비즈니스에 가장 적합한 API를 선택하려고 한다. API 선택시 고려해야할 사항 중 잘못된 것을 고르시오. [2점]

- ① HTTP API는 REST API보다 지연시간이 짧고 비용이 저렴하다.
- ② REST API는 API에 대한 플랫폼별 또는 언어별 SDK를 생성하는 기능을 제공한다.
- ③ WebSocket API는 송수신한 메시지 수와 전송한 데이터 양에 대해서 과금된다.
- ④ RESTful API는 HTTP API와 REST API로 구축할 수 있다.

(정답) 3

(해설) HTTP API와 REST API는 수신한 API 호출과 전송한 데이터 양에 대해서 과금되고, WebSocket API는 보내고 받은 메시지 수와 총 연결 시간(분)에 대해서만 과금된다.

문항7) 사용자가 업로드한 이미지를 Amazon EBS 볼륨에 저장하는 웹 애플리케이션을 하나의 Amazon EC2 인스턴스에서 호스팅하고 있다. 확장성과 가용성 향상을 위해 다른 가용 영역에 두 번째 EC2 인스턴스와 EBS 볼륨을 생성하여 Application Load Balancer 뒤에 배치했다. 이러한 변경 이후 웹 사이트를 새로 고칠 때마다 일부 이미지만 조회되는 현상이 발생했다. 모든 이미지를 조회하기 위한 가장 적절한 설정을 고르시오. [3점]

- ① 두 EBS 볼륨에 모든 이미지가 존재하도록 데이터를 복사한다.
- ② 이미지가 존재하는 서버로 요청을 라우팅하도록 Application Load Balancer를 설정한다.
- ③ 두 EBS 볼륨의 데이터를 Amazon EFS로 복사하고, 새로운 이미지를 EFS에 저장하도록 애플리케이션을 수정한다.
- ④ 두 서버 모두에 요청을 보내도록 Application Load Balancer를 설정하고 이미지가 존재하는 서버에서 응답하도록 애플리케이션을 수정한다.

(정답) 3

(해설) EBS는 단일 EC2 인스턴스에서 가장 짧은 지연 시간으로 데이터에 액세스해야 하는 워크로드에 필요한 성능을 제공하기 위해 사용됩니다. 반면 EFS는 최대 수천 개의 EC2 인스턴스를 위한 파일 시스템 인터페이스, 파일 시스템 액세스 시맨틱스(강력한 일관성 및 파일 잠금 등) 및 동시에 액세스 가능한 스토리지를 제공합니다.

