그림으로 배우는 딥러닝 정리

1. 머신러닝 개요

1. 머신러닝 : 데이터에서 의미있는 정보를 추출하는 기술
2. 지도학습 : 레이블이 되어있다는 말, 데이터 세트와 할당된 레이블 쌍을 제공, 호랑이 사진과 '호랑이' 라는 레이블을 제공해 줌.
3. 비지도학습 : 데이터와 연결된 레이블이 없는 상태에서 클러스터링(분류)를 하거나 그룹화, 클러스터링 과정이라고 함.
4. 강화학습 : 컴퓨터가 '좋을 것이다' 라거나 '더 낫다' 라는 식의 일반적 방법으로 순위를 매기면 되는 가능한 접근.
5. 딥러닝 : 일련의 단계와 레이어들의 연산을 사용하는 머신러닝 알고리즘
6. 네트워크를 훈련하거나 가르치는 건 모든 입력이 원하는 출력을 만들어내도록 가중치에 대한 값을 찾는 것에 지나지 않음.
7. 학습은 보통 원하는 답을 얻을 때까지 가중치를 점차적으로 바꾸는 과정을 의미함.

2. 필수 통계

1. 난수 : 난수는 머신러닝 알고리즘에 중요하다. 사용해서 시스템을 초기화 하고 학습과정에서 단계를 제어하며 출력값에 영향을 주기도 한다.
2. 평균 : 모든 항목의 합계를 목록의 항목 수로 나눈 값을 의미한다.
3. 최빈값 : 목록에 가장 자주 발생하는 값
4. 중앙값 : 가장 작은 값에서 가장 큰 값 순으로 정렬된 목록에서 중간에 있는 값. 짝수 배열의 경우는 양 옆 값의 평균을 취한다.
5. 확률분포 : 가능합 옵션에서 사건이 발생할 총 확률은 100%이기 때문, 정규화된 버전이라고도 말하며, 모든 값의 합이 1이 되는 것을 의미함.
6. 확률 질량함수 : 이산확률분포라고도 하며 반환값이 유한한 경우에 대한 확률 함수를 의미함. 연속되지 않은 범위에 대한 확률(5개의 차량 중 1개를 뽑을 확률)
7. 확률 밀도함수 : 연속 확률분포라고도 하며 이는 0과 1사이에서 모든 값을 반환할 수 있는 여지가 있다. (차에 있는 오일의 양)
8. 균등분포 : 기본 균등분포는 1의 값을 가지는 0과 1사이를 제외한 부분에서 모두 0이다.
9. 정규분포 : 가우스 분포라고도 하며 분포값이 높은 곳에서 밀도가 높고, 분포가 낮은 곳에서 밀도가 더 낮다. 측면으로 갈수록 0에 가까워지나 도달하지 않는다.
10. 정규분포에서 평균은 분포의 중심이며, 표준편차는 벌어진 정도를 나타낸다. 표준편차의 제곱은 분산이며 뜻하는 바는 같다.
11. 정규분포에서는 3시그마 규칙을 적용하며, 68-95-99.7의 확률을 가진다.
12. 베르누이분포 : 0과 1의 두가지 값만 반환함.
13. 기대값 : 확률분표에서 값을 선택 후 다른 값을 또 선택하면 시간이 지남에 따라 긴 목록이 생기는데 숫자인 경우 이를 기댓값이라고 한다.

14. 서로 연관성을 가지는 변수를 종속변수라고 하며, 서로 상관이 없는 변수를 독립변수라고 한다.
15. 비복원 추출 : 뽑은 걸 또 뽑기 가능, 독립적이다.
16. 복원 추출 : 뽑은 건 못 뽑음, 종속적이다.
17. 부트스트래핑 : 모집단에 대해 표본 평균을 바탕으로 파악하는 것. 표본 평균의 집단은 정규분포를 따른다. 이를 부트스트래핑이라고 함.
18. 공분산 : 한 값이 증가하는데 다른 게 증가하면 이를 양의 공분산을 나타낸다고 한다. 반대는 음의 공분산이라고 한다. y가 x의 변화를 일관되게 따라갈수록 공분산이 강해진다고 의미한다. 두 변수에 일관된 움직임이 없으면 공분산은 0이다. 공분산 아이디어는 변화값이 서로의 배수인 경우에 한해 관계를 가진다.
19. 상관관계는 공분산의 확장 버전으로 -1과 1사이에 값을 제공한다. +1은 완벽한 양의, -1는 완벽한 음의 상관관계를 의미한다. 값이 0에 가까울수록 상관관계가 약한 것이다.
20. 조건부 확률 : B가 참일 때, A가 참일 확률. B가 참인 상황에서만 적용 가능하므로 이를 조건부 확률이라고 한다.
21. 정확도 : 모든 데이터 셋에서, 제대로 분류한 참양성과 참 거짓의 비율 = 제대로 분류한 비율, 올바르게 예측한 샘플의 비율
22. 정밀도 : 양성으로 분류한 것들 중 제대로 참양성을 분류한 비율 = 양성 분류 중 참양성의 비율(거짓 양성 제외), 거짓양성의 비율
23. 재현율 : 모든 양성 데이터 중에서 참 양성의 비율 = 양성 데이터 중 참양성 데이터 비율, 거짓음성의 비율
24. 정밀도와 재현율은 트레이드 오프 관계이므로, 상호 반비례 관계이다.
25. 빈도주의자 : 측정과 관찰 불신, 많은 수의 측정을 수행했을 때 가장 자주 나온 값이 가장 가능성이 높다.
26. 베이지안 접근 : 프로세스 끝에서 발견되기를 기다리는 값이 없다.
27. 베이즈 규칙 : 조건부 확률 계산에 대한 것. 사후확률 = (우도 * 사전확률) / 증거, 그리고 이 사후 확률이 다시 사전확률이 된다. 이러한 루프를 베이즈 루프라고 한다.
28. 베이지안 추론 : 하나의 정답에 초점을 두지 않음, 작은 범위에 점차 큰 확률을 할당한다. 범위의 어떠한 값도 답은 될 수 있다는 개념
29. 함수에서 가장 작은 값 : 전역 최소값, 함수에서 가장 큰 값 : 전역 최대값
30. 일부 구간에서 가장 작은 값 : 지역 최소값, 큰 값 : 지역 최대값
31. 미분 : 임의 점에서 가지는 기울기, 미분값이 0에 가까울수록 최대나 최소값에 가까워진다. 미분값이 음수일 때 오른쪽으로 이동하면 최소값에 가까워진다. 양수일때는 왼쪽으로 이동해서 최소값에 가까워진다.
32. 그래디언트 : 3차원 이상의 공간에서의 미분을 일컫는 단어.
33. 물은 가장 가파른 길을 따라서 흐른다. 이를 최대하강이라고 하고, 반대경우는 최대상승이라고 한다. 하강하려면 그래디언트의 음의 경사도만 따른다.
34. 안장점 : 한쪽 방향으로는 최소값, 한쪽 방향에서는 최대값을 가져서 움직이지 못하는 지점

3. 분류

1. 훈련 데이터 세트 : 분류하려는 샘플이나 데이터 조각의 집합
2. 레이블 : 훈련 데이터 세트에 할당된 클래스(호랑이 사진에 호랑이라는 이름을 붙여주는 것)
3. 이진분류 : 모든 입력에 대해서 가능한 출력이 두개다(예스 or 노)
4. 2D 다중클래스 분류 : 하나대 하나, 하나대 나머지
5. 클러스터링 : 데이터 세트 자체를 유사한 것들끼리 묶는 것, 비지도 학습, 사전에 몇개로 묶을지에 대해 설정하는 K-평균 클러스터링
6. 차원의 저주 : 피처가 많을 수록 분류에 유리하지만, 일정 시점을 넘으면 역효과가 난다.
7. 차원의 저주는 시스템 훈련시 엄청난 양의 데이터가 필요한 이유이다.
8. 테스팅 : 시스템이 이전에 본적이 없는 데이터를 얼마나 잘 평가하는지 확인하는 것
9. 훈련 데이터 세트 : 레이블이 있는 학습할 모든 샘플
10. 훈련 프로세스가 과정을 통해 한 샘플씩 실행됨에 따라서 분류기 내부의 변수들은 레이블 예측을 점차 잘하는 값으로 변화하게 되는데 전체 훈련 데이터 세트를 실행할때마다 한 에폭동안 훈련했다고 말함
11. 테스트 데이터 세트 : 처음 보는 데이터 세트
12. 훈련 및 테스트의 필수 규칙은 절대 테스트 데이터로 학습하지 않는 것이다. 만약 이를 발생할 경우 데이터 누출 또는 오염된 데이터라고 함
13. 일반적으로 60%를 훈련데이터 20%를 검증데이터, 나머지 20%를 테스트 데이터에 할당한다.
14. 과적합 : 시스템이 훈련데이터로 너무 잘 학습되어 새로운 데이터에 대해 잘 작동하지 않는 것
15. 과소적합 : 시스템이 잘 학습되지 않아서 새로운 데이터에 대해 잘 작동하지 않는 것
16. 과적합을 제어하기 위해서 첫째 규칙이 너무 구체적인 때를 포착해 프로세스를 중지한다 얼리스토핑이라고도 한다. 둘째는 일반화를 사용해서 일반적 규칙을 학습하도록 보장하고, 과적합 시작을 지연한다.
17. 훈련 오차가 개선되는 동안 검증오차의 증가세가 꺾이거나 악화될 때 과적합이라고 한다.
18. 일반화 수행이나 얼리스토핑의 핵심은 분류기에서 사용하는 파라미터 값을 제한하는 것이다. 파라미터 값을 낮추면 과적합이 발생하기 전까지 더 오래 일반 특성에 대한 학습이 가능하다.
19. 일반화 : 드롭아웃, 배치 정규화, 레이어 정규화, 가중치 일반화 등이 있다.
20. 편향 : 시스템이 잘못된 것을 지속적으로 학습하는 경향
21. 분산 : 관련없는 세부사항을 학습하는 경향
22. 각 샘플은 피처라는 값의 목록이다.
23. 각 피처는 크게 수치형 또는 범주형의 유형을 가진다.
24. 수치형 : 부동소수나 정수로 된 단순한 숫자, 정량적 데이터
25. 범주형 : 레이블을 설명하는 문자열의 형태

26. 원핫 인코딩 : 더미변수라고도 한다. 레이블을 숫자로 된 목록으로 처리하는 것
27. 정규화 : 모든 수의 합이 1이 되게 하는 것.
28. 표준화 : 데이터의 평균이 0, 표준편차가 1이 되도록 조정하는 것.
29. 데이터 세트 축소 : 피처 선택(반드시 필요한 데이터 피처만 선택 및 갈무리), 차원 축소(몸무게와 키를 합쳐서 하나의 BMI로 만든다), 주성분 분석(공통 분모가 되는 주 컴포넌트를 식별해서 가중치에 따라 구분한다)
30. 대표적 분류 알고리즘 : K-최근접 이웃, 의사결정 트리, 서포트 벡터머신, 나이브 베이즈
31. 분류기의 유형 : 모수적, 비모수적 알고리즘으로 나뉜다.
32. K-최근접 이웃 : K-평균 클러스터링은 비지도 학습이며, 이는 지도 학습이다. 새로운 샘플의 클래스 결정을 위해 k개의 가장 가까운 샘플들의 클래스를 계산한다.k = 9로 설정하면 주변에 가장 가까운 9개의 샘플들에 대한 클래스를 살핀다. 간단하여 훈련속도가 빠르지만, 모든 데이터를 메모리에 올려야 하므로 느려질 수 있다. 또한 근처에 많은 이웃이 있어야 효과적이다.
33. 의사결정 트리 : 스무고개 게임과 비슷하다. 트리 분할점은 노드, 각 선은 링크 또는 가지라고 한다. 마지막 노드는 리프 또는 터미널 노드라고 한다. 중간에 있는 노드들을 내부 노드라고 한다. 이의 흥미로운 특성은 이진이라는 것이다. 모든 노드는 예/아니오의 값을 가진다. 이는 입력 샘플에 매우 민감하게 반응하므로 과적합 경향이 있다. 과적합 제어를 위해 트리밍이라는 가지치기를 수행한다.
34. 서포트 벡터머신 : 클러스터 사이의 가장 완벽한 경계선을 찾고 싶다. 이는 모든 샘플에서 가장 거리가 먼 선을 찾는다. 서포트 벡트는 가장 거리가 먼 점선 상에 위치한 데이터 들이다. 그리고 실선에서 서포트 벡터를 통과하는 점선까지의 간격을 마진이라고 한다.마진 영역에 데이터 노이즈가 있다면? C라는 파라미터를 제어하는데 이는 마진에 점을 얼마나 허용하는지 통제한다.
35. 나이브 베이즈 : 이는 데이터에 대한 가정에서 시작하므로 빠르게 작동한다. 예를 들면 샘플의 모든 피처가 가우스 분포, 정규분포를 따른다고 가정한다. 데이터를 나이브 베이즈 분류기에 넣으면 각 피처 집합이 정규분포에서 비롯된다고 가정한다. 실제로 잘 처리된다. 나이브 베이즈는 매우 빠르기 때문에 어떤 데이터인지 알고 싶을 때 적용하는게 일반적이다.
36. 정리하면 ? 분류기는 모수적, 비모수적으로 나뉘는데 데이터에 대해 선입견이나 가정이 들어가면 모수적이라고 한다. 비모스는 K-최근접 이웃, 의사결정 트리가 있고, 나머지는 서포트 벡터머신과 나이브베이즈가 있다. 예를 들어 서포트 벡터 머신은 훈련 데이터를 클래스별 구분하는 선이나 면을 찾는다. 나이브 베이즈는 데이터에 고정된 데이터 분포가 있다고 가정하고 이를 각 피처에 적용하려는 노력을 한다.
37. 앙상블 : 유사한 학습기들의 그룹을 앙상블이라고 한다. 일반적 방법은 다수 투표이다. 학습기는 예측값에 대해서 한 표를 던지고, 가장 많은 표를 받는 예측값이 승자가 된다.
38. 의사결정 트리의 앙상블 예시 : 배깅, 랜덤 포레스트, 부스팅

4. 딥러닝 기본

1. 딥러닝 : 연산 요소간의 네트워크 구축을 기반으로 하는 알고리즘, 기본 단위를 인공 뉴런이라고 한다.
2. 퍼셉트론 : 뉴런의 단순화된 수학적 모델로, 1962년에 처음 제안되었다. 모든 입력은 부동소수로 표현된다. 각 입력에는 가중치라고 하는 부동소수가 곱해지고, 이 곱셈의 결과는 모두 함께 더해진다. 그리고 이 결과를 임계값과 비교한다/ 합산의 결과가 0보다 크면 +1의 출력을, 작으면 -1을 출력한다.
3. 현대의 '인공 뉴런' : 퍼셉트론에서 약간만 일반화 되었다. 첫째, 각 뉴런에 하나를 입력으로 추가 제공한다. 이를 편향이라고 한다. 이는 이전 뉴런의 출력에서 나오지 않으며, 모든 가중입력의 합에 더해지는 숫자다. 두 번쨰는 출력이다. 인공 뉴런은 활성화 함수를 통해서 다양한 출력값을 반환한다.
4. 피드포워드 네트워크 : 정보가 한 방향으로 흐르게 뉴런을 배열하는 방법.
5. 출력 레이어 : 심층 네트워크 : 뉴런을 포함하는 최상위 레이어
6. 레이어의 깊이 : 계산하는 레이어의 수
7. 히든 레이어 : 입력과 출력 사이의 레이어들
8. 완전연결 레이어 : 모든 뉴런이 다음 레이어의 모든 뉴런에게 가중치를 전달하는 구성 3 * 3에서는 총 9개의 연결이 발생한다. 레이어가 다음처럼 완전연결로만 구성된 것을 다중레이어 퍼셉트론, 완전 연결 네트워크라고 한다.
9. 텐서 : 주어진 차원수와 각 차원의 크기를 가진 숫자 블록일 뿐, 넘파이는 cpu, 텐서는 gpu 연산이 가능하다.
10. 활성화함수 : 전달함수 또는 비선형성이라고 한다. 입력을 받아 함수 값으로 리턴하여 새로운 부동소수 값을 돌려준다. ReLU, 누수 ReLU, 소프트 플러스, ELU, 시그모이드, 하이퍼볼릭 탄젠트 등이 있다. ReLU나 누수ReLU가 대표적이고 그 다음으로 시그모이드와 하이퍼볼릭 탄젠트가 많이 사용된다.
11. 소프트맥스 : 일반적으로 출력 뉴런대상에만 적용하는 연산이다. 목적은 출력되는 원시숫자를 클래스의 확률로 변환해주는 것이다. 출력값의 총 합은 1이며, 모두 0과 1사이의 부동소수다.
12. 오차 또는 손실 또는 페널티 : 레이블과 예측값 사이의 불일치 정도를 설명, 네트워크가 오차를 0에 가깝게 낮추는 유일한 방법은 가중치를 변경하는 것이다.
13. 경사하강법 : 각 가중치의 양이나 음의 방향 중 어느쪽으로 조금씩 움직일지를 미리 알면 훈련 속도를 늘릴 수 있다. 가중치에 대한 오차의 그래디언트에서 이 정보를 얻을 수 있다. 오차를 바탕으로 오차곡선을 그릴 수 있는데, 각 가중치 위의 오차 고선 기울기를 찾아서 특정 가중치 값에서 오차의 미분값을 찾을 수 있다. 가중치 값에서 미분값이 음수이면 우측으로 이동시 오차가 줄어든다. 역으로 양수이면 우측 이동시 오차가 늘어난다. 미분 값을 사용해 각 가중치를 이동시키고 오차 곡선에서 더 낮은 값을 찾아 가므로 이를 경사하강법 또는 gradient descent라고 부른다.

14. 역전파 : 결국 각 뉴런의 델타값을 찾는 것이 전부다. 각 뉴런의 변화량 또는 미분값을 찾는 것이 전부인 것이다. 각 뉴런에서의 델타값이 결국 오차의 양에 영향을 주는 것이다 예를 들어 A 뉴런에서 1이 변하면 오차는 4가 변하는 식의 룰이다. 역전파에서는 체인룰이 적용된다. 1뉴런에서의 1의 변화가 2뉴런에서 0.5의 영향을 발휘하고 3에서는 2의 영향을 발휘한다. 모든 것은 비례하여 엮인다.
15. 편미분을 통해서 우리는 모든 뉴런의 델타를 계산할 수 있다. 계산은 다음 뉴런의 델타값에 의존하며, 출력뉴런에는 델타 값이 없다.
16. 학습률 : 가중치를 너무 많이 변경하면 모델이 과적합되어 효과적이지 않다. 실전에서는 학습률이라는 하이퍼 파라미터를 사용해서 가중치 변화량을 제어한다. 쉽게 말하면 걸음을 크게 걸을 것인가, 작게 걸을 것인가. 일반적으로 최적의 학습률을 찾는 것이 과제 중 하나다.
17. 옵티마이저 : 목표는 경사하강을 더 빠르게 하고 멈추게 하는 일부 문제를 피하는 것이다. 최적의 학습률을 찾는 작업을 자동화한다. 학습의 속도를 향상시키기 위한 알고리즘을 옵티마이저라고 한다. 여러 옵티마이저가 공유하는 중요 개념은 학습률을 변화시켜 학습을 개선한다는 것이다.
18. 옵티마이저 - 시간에 따라 학습률 변화 : 큰 학습률로 시작해서 점차적으로 감소시킨다. 쉬운 말로 학습률에 거의 1에 가까운 숫자를 계속해서 곱해주는 것이다. 0.99 같은 수. 스텝을 수행하며 학습률이 점차 작아진다. 스텝을 좁히는 것이다.
19. 옵티마이저. - 신경망 오차 모니터링 : 오차가 줄면 현재 학습률을 유지하고, 학습이 멈추면 감쇠를 적용해 더 좁게 스텝을 이동한다.
20. 옵티마이저 - 볼드 드라이버 : 시간이 지나며 학습률을 더 높이는 방법이다.
21. 경사하강법 적용 전략(옵티마이저의 종류) : 배치 경사하강, 확률적 경사하강, 미니배치 경사하강, 모멘텀 경사하강, 네스테로프 모멘텀 경사하강, Adagrad, Adam, RMSProp
22. 배치 경사하강법 : 모든 샘플에 대해 평가한다음 각 에폭당 한번만 가중치를 업데이트 하는 것. 이는 부드러운 학습 진도를 보이지만 자칫 시간이 오래걸릴 수 있다. 데이터가 무거워지는 경우에는 알고리즘이 느려질 수 있다.
23. 확률적 경사하강법(SGD) : 매 샘플마다 가중치를 업데이트 하는 방법. 훈련용 샘플을 임의 순서로 보여주기 때문에 가중치가 한 샘플에서 다음 샘플로 어떻게 바뀔지 예측이 불가하여 붙은 이름이다. 매 샘플마다 업데이트를 하므로 300개의 샘플로 구성된 데이터는 300번 가중치를 업데이트 한다.
24. 미니배치 경사하강법(미니배치 SGD) : 일정 갯수 샘플을 평가하고 가중치를 업데이트 한다. 미니배치 크기는 대부분 32 256사이의 2의 거듭제곱수이며 GPU의 병렬처리 성능을 고려한 선택이다. 만약 300개의 샘플을 가지고 32개 샘플로 구성된 미니배치를 사용했으므로, 매 에폭당 10개의 미니배치가 존재한다. 통상 미니배치 SGD를 그냥 SGD로 부른다.
25. 모멘텀 경사하강법 : 각 스텝에서 가중치를 얼마나 변경해야 하는지 계산하고 스텝에서 변화를 약간씩 더 추가한다. 모멘텀이 너무 과하면 오버슈팅을 하니 주의해야 한다. 이는 그래디언트를 찾고, 현재 학습률로 스케일링한 뒤 이전의 변화에 이를 더해서 새로운 위치를 얻는다. 모멘텀은 안장의 얕은 곳을 극복하도록 도와준다.

26. 네스테로프 모멘텀 : 현재 위치한 곳의 그래디언트만 사용하는 대신 예상 지점의 그래디언트도 사용한다. 이동할 다음 움직임이 지난 움직임과 비슷하다면, 더 큰 스텝을 내딛는다. 하지만 반대라면 좁게 내딛는다.
27. Adagrad : 적응형 그래디언트 학습의 줄임말이다. 각 가중치에 대해서 학습률 감쇠를 수행한다. 각 가중치에 대해 업데이트 스텝에서 사용하는 그래디언트를 취하고 제곱한다음 이를 가중치 실행합계로 나눈다. 학습률을 0.01 정도로 설정 후 시작한다.
28. Adadelta와 RMSprop : adadelta는 각 스텝의 가중치 실행 합계를 사용해서 스텝별 가중치 업데이트 양을 적응적으로 변경한다. RMSprop도 매우 유사하다 다만 그래디언트에 추가할 조정을 결정하기 위해 제곱 평균 제곱근 연산을 사용한다. 시작에 좋은 값은 0.9이다.
29. Adam : 적응적 모멘트 추정이라고 부르며, 두개의 베타 파라미터를 설정한다. Beta1 = 0.9, beta2 = 0.99로 설정하는 것이 일반적이다.
30. 옵티마이저 선택 : 간단한 테스트 예제는 네스테로프 모맨텀을 사용한 미니배치 SGD가 빠르다. 광범위한 데이터 세트와 네트위크는 Adadelta, RMSprop, Adam 세가지가 매우 유사하게 수행된다. 모든 상황에 최적인 옵티마이저는 없다.
31. 일반화 : 과적합을 지연시키는 기술, 드롭아웃, 배치 정규화가 대표적 방법
32. 드롭아웃 : 드롭아웃 레이어의 역할은 이전 레이어에 있는 일부 뉴런 연결을 일시적으로 끊는 것이다. 드롭아웃 레이어는 무작위로 이전 레이어에 있는 뉴런을 비율만큼 선택하고 이들의 입출력을 일시적으로 네트워크에서 분리한다. 이는 순전파 역전파에서도 어떠한 영향을 받지 않는다. 배치가 완료되고 나머지 가중치가 업데이트 되고 나면 연결을 복원한다.
33. 배치 정규화 : 배치놈이라고도 하며, 이는 레이어에서 나오는 값을 수정한다. 배치놈은 이전 레이어에서 나온 모든 값을 크기를 조정하고 이동시켜서 가장 유용한 값을 취한다.