TAVE 서기

|  |  |  |  |
| --- | --- | --- | --- |
| **서기 내용** | | | |
| **서기 일자** | 22.1.5 | **서기** | 서가을 |
| **주제** | 오토인코더 후반부 | | |
| **시간** | 20:30 – 22:30 | **장소** | Zoom 미팅 |
| **스터디**  **인원** | 고성호, 권기호, 이아현, 서가을 : 시작  텍스트, 실내, 사람, 벽이(가) 표시된 사진  자동 생성된 설명  고성호, 권기호, 이아현, 서가을 : 종료  텍스트, 실내, 모니터, 전자기기이(가) 표시된 사진  자동 생성된 설명 | | |
| **내용** | | | |
| **배운 내용** |  | | |
| **Chapter17 오토인코더와 GAN을 사용한 표현학습과 생성적 학습**  **17.7 희소 오코인코더**     * 희소: 좋은 특성을 추출하도록 만드는 제약 방식 * 비용 함수에 적절한 항 추가 -> 오코인코더가 코딩 층에서 활성화되는 뉴런 수 감소하도록 만듦. * 오코인코더가 적은 수의 활성화된 뉴런을 조합하여 입력 표현해야 함. 텍스트이(가) 표시된 사진    자동 생성된 설명      * 결과 이미지 * 전체 뉴런의 활성화 평균이 0 근처   KL Divergrence   * 정보량 * 엔트로피 : 정보량의 평균, 정보량의 기대값 * 즉 엔트로피가 높다는 것은 사건의 분포가 균등한 확률분포를 가져서 불확실하다는 것 * 엔트로피는 예측하기 쉬운 일보다 어려운 일에서 더 높음 * 확률변수가 연속일 경우, 분포가 가우시안일 때 엔트로피 가장 높음 * 크로스 엔트로피 * 실제 분포인 q를 알고 싶을 때 모델링한 분포 p를 통해 예측하려고자 할 때텍스트이(가) 표시된 사진    자동 생성된 설명 * Kullback-Leibler Divergence (KL-Divergence) * 서로 다른 두 분포의 차이를 측정하는 데에 사용텍스트이(가) 표시된 사진    자동 생성된 설명 * => 이것들을 사용해서 훈련할 때 인코딩 층의 실제 sparse함을 측정하고 목표 sparse함과 비교, 차이를 penalty로 줌   텍스트이(가) 표시된 사진  자동 생성된 설명   * 패션 MNIST에서 희소 오토인코더를 훈련한 후 코딩 층에 있는 뉴런 활성화가 거의 0에 가깝고, 전체 뉴런의 평균 활성화가 0.1 근처   **17.8 변이형 오토인코더**   * 변이형 오코인코더(VAE) 의 목적: AE의 인코더 파트를 사용하여 분포(평균과 분산)을 뽑아내고, 분포로부터 랜덤 샘플링된 값 z를 가지고 디코더로 복원하여 새로운 이미지가 생성되도록 학습시키는 것. * 변이형 오토인코더의 속성  1. 확률적 오토인코더: 훈련이 끝난 후에도 출력이 부분적으로 무작위성 2. 생성 오토인코더: 훈련 세트에서 샘플링된 것처럼 새로운 샘플을 생성 가능  * => 변이형 오토인코더와 RBM 유사함. * RBM과 비교한 변이형 오토인코더의 장점  1. 훈련이 더 쉽다. 2. 샘플링 과정이 훨씬 빠르다.  * 잠재 공간: 훈련하는 동안 비용 함수가 코딩을 가우시안 샘플들의 군집처럼 보이도록 점진적으로 이동시키는 공간 * 잠재 공간 값 표현 (AE vs VAE)   AE: 고정된 값(표현 벡터)로 나타냄.  VAE: 가우시안 확률 분포(정규 분포)의 값으로 나타냄.  즉, 잠재 공간의 값은 평균과 분산으로 매핑됨.   * VAE는 인코더를 학습시키는 AE와 달리 디코더를 학습시키는 것이 목표 * 변이형 오토인코더(Variational AutoEncoder) * 랜덤 샘플링된 값 z를 얻기 위해 평균에 분산을 더하는데 분산은 정규분포에서 랜덤하게 뽑아낸 값 곱하여 사용 * 변이형 오토인코더는 복잡해보여도 간단한 가우시안 분포에서 샘플링된 것처럼 보이는 코딩을 만드는 경향이 있음. * 훈련하는 동안 비용 함수가 코딩을 가우시안 샘플들의 군집처럼 보이도록 코딩 공간(잠재 공간)으로 점진적으로 이동시킴. * -> 훈련이 끝난 후 새로운 샘플을 쉽게 생성할 수 있음. * => 가우시안 분포에서 램덤한 코딩을 샘플링해 디코딩하면 됨. * 비용함수   1) 일반적인 재구성 손실: 오토인코더가 입력을 재생산하도록 만듦.  2) 잠재 손실: 단순한 가우시안 분포에서 샘플된 것 같은 코딩을 가지도록 오토인코더를 강제함.   * 잠재 손실의 특징  1. 가우시안 분포와 실제 분포 사이의 KL 발산 사용 2. 가우시안 잡음 때문에 수식이 복잡하지만 식을 통해 간단히 계산 가능        * 패션 MNIST로 변이형 오토인코더 구현 * 텍스트이(가) 표시된 사진    자동 생성된 설명 * 코딩을 샘플링하는 사용자 정의 층 * 텍스트이(가) 표시된 사진    자동 생성된 설명 * 함수형 API를 통해 인코더 생성 * 실제로 사용하는 것은 outputs중 마지막 출력인 codings. * 텍스트이(가) 표시된 사진    자동 생성된 설명 * 디코더 생성 * 함수형 API 대신 시퀀셜 API 사용. * 변이형 오토인코더 모델 생성 * 인코더의 처음 두 개 출력을 무시(코딩만 디코더에 주입)텍스트이(가) 표시된 사진    자동 생성된 설명 * 잠재 손실과 재구성 손실을 추가 * 배치에 있는 모든 샘플의 평균 손실을 계산하고 재구성 손실에 비례해 적절한 크기가 되도록 784로 나눔   - 케라스가binary\_crossentropy를 계산할 때 합이 아닌 784개 전체 픽셀의 평균 계산  - 따라서 필요한 것보다 재구성 손실이 784배 적음  - 간단하게 하기 위해 784로 나눠줌.  - 최종 손실이 784배 작아지므로 더 큰 학습률 사용해야 함.  **17.8.1 패션 MNIST 이미지 생성하기**   * 이미지 생성주방기기이(가) 표시된 사진    자동 생성된 설명 * 가우시안 분포에서 랜덤한 코딩을 샘플링해서 디코딩 * 시맨틱 보간 * 변이형 오코인코더는 시맨틱 보간이 가능 (픽셀 수준 X, 코딩 수준O)  1. 두 이미지를 인코더에 통과시켜 얻은 두 코딩을 보간 2. 보간된 코딩을 디코딩하여 최종 이미지 얻음텍스트이(가) 표시된 사진     자동 생성된 설명      * 결과 이미지 (테두리 – 원본) * 지난 몇 년 간 변이형 오토인코더가 널리 쓰였지만 GAN이 훨씬 실제같이 또렷한 이미지를 만들어 인기가 더 높아짐.   **17.9생산적 적대 신경망**   * GAN은 신경망 두 개로 이루어짐: 생성자 + 판별자 * 생성자: 랜덤한 분포를 입력으로 받고 이미자와 같은 데이터를 출력   판별자 속이기   * 판별자: 생성자에서 얻은 가짜 이미지나 훈련 세트에서 추출한 진짜 이미지를 입력으로 받아 입력된 이미지가 가짜인지 진짜인지 분류 * GAN은 다른 목표를 가진 두 네트워크로 구성 -> 각 훈련 반복은 두 단계로  1. 판별자 훈련  * 훈련 세트에서 실제 이미지 배치 샘플링 * 생성자에서 생성한 가짜 이미지 합침 * 가짜 레이블은0, 진짜 레이블은 1로 설정 * 이진 크로스 엔트로피 사용해 한 스텝 동안 이렇게 레이블된 배치로 훈련 * 역전파는 판별자의 가중치만 최적화  1. 생성자 훈련  * 생성자를 이용해 댜른 가짜 이미지 배치 만듦 * 생성자가 만든 가짜 데이터를 판별자가 실제 데이터라고 추정할 확률을 최대화하도록 학습 * 생성자는 진짜 이미지는 보지 않고 판별자 반응만 보고 학습 * 배치에 이미지를 추가하지 않고 레이블을 1로 세팅하는 과정에서는 판별자의 가중치를 동결해야 함 * 따라서 역전파는 생성자의 가중치에만 영향을 미침 * 패션 MNIST 데이터셋으로 GAN 학습하기텍스트이(가) 표시된 사진    자동 생성된 설명 * 생성자와 판별자 생성 * GAN 모델 컴파일 * 생성자는 GAN 모델 통해서만 컴파일 -> 따로 컴파일 필요 X * GAN 모델도 이진 분류기 -> 이진 크로스 엔트로피 손실 사용텍스트이(가) 표시된 사진    자동 생성된 설명 * 이미지를 순회하는 Dataset 생성텍스트이(가) 표시된 사진    자동 생성된 설명 * 앞서 언급한 두 단계 걸쳐 훈련 * [1단계] * 가우시안 잡음 생성자에 주입하여 가짜 이미지 생성 * 생성한 이미지와 동일한 개수의 진짜 이미지를 합쳐 배치 구성 * 타깃 y1은 가짜일 경우 0, 진짜일 경우1로 설정 * 이 배치에서 판별자를 훈련 * [2단계] * GAN에 가우시안 잡음 주입 * 생성자가 먼저 가짜 이미지 생성, 판별자 추측 * 판별자가 가짜를 진짜로 믿게 하도록 하고 싶으므로 y2를 1로 설정 * 생성된 이미지 출력 * 이보다 좋은 이미지 생성 안 됨   **17.9.1 GAN 훈련의 어려움**   * 내시균형: (게임 예시) 다른 플레이어가 전략을 수정하지 않을 것이므로 어떤 플레이어도 자신의 전략을 수정하지 않는 상태 * GAN은 하나의 내시균형에만 도달 가능  1. 생성자가 완벽하게 실제와 같은 이미지를 생성했을 때:   판별자가 50% 확률로 추측만 할 수 있음  어려움1.: 모드 감소   * 생성자의 출력 다양성 감소 * GAN이 몇 개의 클래스를 오가다가 어떤 클래스에서도 좋은 결과 못 만듦   어려움 2: 파라미터의 감소   * 생성자와 판별자가 서로에게 지속적으로 영향 줌. * -> 파라미터 변동이 크고 불안정해질 수 있음 * 훈련이 안정적으로 시작돼도 갑자기 발산 할 수 있음 * GAN의 파라미터는 매우 민감하므로 튜닝에 많은 노력 필요   훈련방법1: 경험 재생   * 매 반복에서 생성자가 만든 이미지를 재생 버퍼에 저장하고 실제 이미지와 이 버퍼에서 뽑은 가짜 이미지를 더해서 판별자를 훈련 * 판별자가 생성자의 가장 최근 출력에 과대적합 될 가능성 줄임   훈련방법2: 미니배치 판별   * 배치 간 얼마나 비슷한 이미지가 있는지 측정해서 판별자에게 제공 * 생성자가 다양한 이미지를 생성하도록 유도-> 모드 붕괴의 위험 줄임   **17.9.2 심층 합성곱 GAN**   * 큰 이미지를 위해 깊은 합성곱 층을 기반으로 한 GAN 위해 노력 * -> 심층 합성곱 GAN (DCGAN) 제안 * 안정적인 합성곱 GAN 구축을 위한 가이드라인  1. 판별자의 풀링층을 스트라이드 합성곱으로 변경 2. 생성자의 풀링층은 전치 합성곱으로 변경 3. 생성자와 판별자에 배치 정규화 사용 4. 깊은 층 위해 완전 연결 은닉층 제거 5. 생성자의 모든 층은 ReLU 활성화 함수 사용 6. 판별자의 모든 층은 LeakyReLU 활성하 함수 사용텍스트이(가) 표시된 사진     자동 생성된 설명  * 패션 MNIST에서 잘 작동하는 작은 DCGAN 모델 * [생성자] * 크기는 100의 코딩을 받아 6272(7\*7\*128)으로 투영 -> 이 결과를 7\*7\*128 크기의 텐서로 바꿈 * 이 텐서는 배치 정규화 층을 지나 스트라이드가 2인 다른 전치 합성곱에 주입 * Tanh 활성화함수를 사용하므로 출력 범위가 -1 ~ 1이기 때문에 훈련 세트를 동일한 범위로 스케일 조정 + 크기 바꾸고 채널 차원 추가해야 함. * [판별자]. * 이진 분류를 위한 일반 CNN과 비슷 * BUT, 이미지를 다운 샘플링할 때 maxpoooling이 아니라 스트라이드 합성곱 사용물품, 다른, 정렬된, 주방기기이(가) 표시된 사진    자동 생성된 설명 * 훈련 결과텍스트이(가) 표시된 사진    자동 생성된 설명 * DCGAN은 완벽하지 않기 때문에 매우 큰 이미지를 생성하면 일관성 없는 이미지를 얻을 가능성이 높아짐.   **17.9.3 ProGAN**   * 훈련 초기에 작은 이미지를 생성하고 점진적으로 샌성자와 판별자에 합성곱 층을 추가해 갈수록 큰 이미지를 만드는 방법 * ProGAN의 구조 * 한번에 전체 크기 이미지 학습시키기보다 4\*4 저해상도로 large-scale structure를 찾아내고 점자 finer-scale-detail 찾아낼 수 있도록 고해상도로 높임   미니배치 표준편차 층   * 판별자의 마지막 층 근처에 추가 * 입력에 있는 모든 위치에 대해 모든 채널과 배치의 모든 샘플에 걸쳐 표준편차 계산텍스트이(가) 표시된 사진    자동 생성된 설명 * 추가적인 특성 맵이 배치의 모든 샘플에 추가되고 계산된 이 값으로 채워짐      * 생성자가 만든 이미지 다양성 부족하면 판별자는 이 통계를 쉽게 얻을 수 있고 다양성이 아주 적은 이미지를 만드는 생성자에게 속을 가능성 감소 * -> 생성자가 더 다양한 출력을 만들도록 유도(모드 붕괴 위험 감소)   동일한 학습 속도   * He 초기화 대신 가우시안 분포를 사용해 초기화 * BUT, 런타임에 He 초기화 통해 가중치 스케일 낮춤 * RMSPropm Adam 같은 그래디언트 옵티마이저 사용 -> GAN 성능 향상 * 가중치 초기화에서 스케일을 맞추지 않고, 모델 한 부분으로 가중치를 조절 -> 훈련 내내 모든 파라미터의 다이나믹 레인지 동일하게 * -> 모든 가중치가 동일한 속도로 학습되어 훈련 속도와 안정성 높임   픽셀별 정규화 층   * 생성자 합성곱 층 뒤에 추가 * 픽셀 단위별로 초기화해주는 것으로, 동일한 이미지의 동일 위치에 있는 모든 활성화를 채널에 대해 정규화함. * 생성자와 판별자 사이의 과도한 경쟁으로 활성화 값이 폭주되는 것을 막음. * 일반적으로 GAN은 생성자와 판별자의 불필요한 경쟁으로 활성화 값이 폭주되는 것을 억제하기 위해 배치 정규화 사용 BUT, DCGAN은 이 방법이 효과 없음 -> 픽셀 정규화 방식 사용   **17.9.4 StyleGAN**   * 생성자에 style transfer 기법 사용해 생성된 이미지가 훈련된 이미지와 같은 다양한 크기의 국부적인 구조를 갖도록 만듦 -> 생성된 이미지의 품질을 크게 높여줌 * 두 가지 네트워크로 구성됨(매핑 네트워크 + 합성 네트워크)   매핑 네트워크   * StyleGAN은 잠재표현 z(코딩)으로부터 직접 이미지를 생성하지 않고 매핑 네트워크 거침 * 8개의 MLP가 z를 벡터 w로 매핑 * 매핑 네트워크는 코딩을 여러 스타일 벡터로 매핑하는 것   합성 네트워크   * 이미지의 생성을 책임짐 * 일정하게 학습된 입력을 받음 -> 입력을 합성곱 여러 개와 업샘플링 층에 통과시킴 * 차이점  1. 입력과 모든 합성곱 층의 출력에 잡음이 섞임 2. 잡음이 섞인 다음에 적응적 인스턴트 정규화 층이 뒤 따름   믹싱 규제   * 일정 비율의 이미지를 두 개의 다른 코딩으로 생성 * 텍스트이(가) 표시된 사진    자동 생성된 설명   **17.10 연습 문제**   * 1. 오토인코더를 활용할 수 있는 주요 작업?   특성추출, 비지도 사전훈련, 차원 축소, 생성 모델, 이상치 탐지   * 2. 적은 수만 레이블링 되어있는 데이터가 있을 때 오토 인코더가 어떤 도움이 될까요? 어떻게 작업하면 될까요?   비지도 사전훈련  전체 데이터를 통해 오토인코더 훈련  인코더 부분의 가중치 복사 -> 새로운 모델의 하위 층의 가중치로 사용  Output layer에는 softmax 사용해 클래스 구별하도록  레이블된 데이터를 통해 클래스 구별 훈련  레이블 되지 않은 데이터를 집어넣어 레이블 생성  레이블 있는 데이터가 많지 않을 때는 하위 층의 가중치 동결   * 3-1. 오토인코더가 완벽하게 입력을 재구성했다면, 반드시 좋은 오토인코더인가요?   아니다.  오토인코더가 overcomplete 오토인코더일 경우 입력을 코딩 층, output으로 복사하는 것을 학습 할 수도.  Undercomplete여도, 훈련 샘플을 특정 숫자에 매핑 -> 입력 그대로 재구성  하지만 입력을 재구성하지 못하는 오토인코더가 좋은 오토인코더는 아님   * 3-2. 오토 인코더의 성능을 평가할 수 있는 방법은?   reconstruction loss를 사용  물론 낮다고 좋은 오토인코더는 아닐 수 있음.  분류기의 비지도 사전훈련을 위해 사용한다면 분류기의 성능도 평가 지표 중 하나   * 4-1. 과소완전, 과대완전 오토인코더가 무엇인가요?   과소완전 오토인코더:코딩이 입력보다 작은 오토인코더  과대완전 오토인코더:코딩이 입력보다 큰 오토인코더   * 4-2. 지나치게 과소완전인 오토인코더의 주요한 위험은 무엇인가요?   입력을 재구성하는 데에 실패할 확률이 큼  너무 작은 차원의 v벡터로 input data를 줄이다 보면 중요한 특성들을 담을 수 없게 됨   * 4-3. 과대완전 오토인코더의 주요한 위험은 무엇인가요?   유용한 특성을 더 작은 차원의 벡터에 담을 필요가 없기 때문에 입력을 그대로 복사하여 출력하는 오토인코더가 만들어질 가능성   * 5-1. 적층 오토인코더의 가중치를 어떻게 묶나요?   인코더의 가중치 행렬을 전치한 전치 행렬을 디코더의 가중치로 사용   * 5-2. 이렇게 하는 이유는 무엇인가요?   훈련해야 될 파라미터 개수가 반으로 준다  적은 훈련데이터 -> 수렴이 빨라짐 (종종)  훈련 세트에 과대적합될 위험성도 감소   * 6. 생성모델이 무엇이며 생성오토인코더의 종류?   생성모델: 훈련 샘플과 닮은 출력을 랜덤하게 생성하는 모델  대표적으로 variational autoencoder   * 7-1. GAN이 무엇인가요?   서로 반대 목적을 가진 generator, discriminator로 이뤄져 있는 네트워크  Generator: 최대한 진짜같은 가짜 이미지를 만들어 discriminator가 판별하지 못하도록  Discriminator: 가짜 데이터와 진짜 데이터를 판별하는 것이 목적   * 7-2. GAN이 유용한 몇 가지 작업을 나열할 수 있나요?   영상 처리: 초해상도 , 컬러 바꾸기, 이미지 편집(실제같은 배경으로 바꾸기)  데이터 증식, 텍스트 오디오 시계열 같은 여러 다른 종류의 데이터 생성, 다른 모델의 취약점을 식별하고 개선하기 등   * GAN을 훈련할 때 주요 어려움은 무엇인가요?   모드 붕괴: 한 이미지만 생성, 판별하다가 다른 클래스로 옮겨 가며 이전의 학습 내용 망각  파라미터의 변동성: 인코더와 디코더가 서로 영향을 주므로 파라미터의 변동성이 커서 하이퍼파라미터 튜닝이 어려움. | | |
| **과제할당** | 18.1 하정현  18.2 허주희  18.3 고성호  18.4 안세윤  18.5 권기호  18.6 이문기  18.7이아현 | | |
| **특이사항** | 없음 | | |
| **비고** | 없음 | | |