



(인공지능과빅데이터) 연습문제_09


1. 다음 중 AutoEncoder에 대한 설명으로 가장 적절한 것은 무엇인가?
 1. 지도 학습(Supervised Learning)의 한 형태로, 입력 데이터와 정확히 일치하는 레이블이 필요하다.
 2. 입력 데이터를 저차원 잠재 공간으로 압축한 후, 이를 다시 원본 데이터로 복원하는 신경망 모델이다. ☒
 3. 주로 분류(Classification)나 회귀(Regression) 문제 해결에 사용되는 생성 모델(Generative Model)이다.
 4. 오직 인코더 부분만으로 구성되어 있어, 데이터의 차원을 확장하는 데 사용된다.
 5. 데이터의 노이즈를 증가시켜 모델의 일반화 성능을 저하시키는 역할을 한다.
2. AutoEncoder의 주요 구성 요소인 인코더(Encoder)와 디코더(Decoder)의 역할에 대한 설명으로 가장 적절하지 않은 것은 무엇인가?
 1. **인코더**: 입력 데이터를 받아 저차원의 잠재 공간 표현(Latent Space Representation)으로 변환한다.
 2. **인코더**: 데이터의 특징을 압축하고 불필요한 정보를 제거하는 역할을 수행한다.
 3. **디코더**: 잠재 공간 표현을 다시 원본 데이터 형태의 출력으로 재구성(reconstruct)한다.
 4. **디코더**: 인코더가 학습한 압축된 정보를 사용하여 데이터의 차원을 감소시킨다. ☒
 5. **인코더와 디코더**: 모두 가중치와 편향을 학습하여 입력과 출력 간의 관계를 최적화한다.
3. AutoEncoder의 주요 활용 분야나 특징에 대한 설명으로 가장 거리가 먼 것은 무엇인가?
 1. **차원 축소(Dimensionality Reduction)**: 고차원 데이터를 저차원 잠재 표현으로 압축하여 데이터 시각화나 효율적인 저장을 가능하게 한다.
 2. **노이즈 제거(Denoising)**: Denoising AutoEncoder와 같이 손상된 입력으로부터 깨끗한 원본 데이터를 재구성하도록 학습하여 노이즈 제거에 활용될 수 있다.
 3. **이상 탐지(Anomaly Detection)**: 정상 데이터에 대한 재구성 오차는 작지만, 이상 데이터에 대한 재구성 오차가 큰 경향을 이용하여 이상치를 탐지할 수 있다.

4. **데이터 증강(Data Augmentation):** Variational AutoEncoder(VAE)와 같은 변형 오토인코더는 잠재 공간에서 새로운 샘플을 생성하여 데이터 증강에 활용될 수 있다.
 5. **강화 학습(Reinforcement Learning)의 보상 함수 설계:** 오토인코더는 직접적으로 강화 학습의 보상 함수를 설계하는 데 사용되는 주요 기법이다. ☒
4. Variational AutoEncoder (VAE)가 일반적인 AutoEncoder와 구별되는 주요 특징으로 가장 적절한 것은 무엇인가?
1. VAE는 디코더 없이 인코더만으로 구성되어 있어 데이터의 차원 축소에만 사용된다.
 2. VAE는 잠재 공간이 특정 확률 분포(예: 가우시안 분포)를 따르도록 제약 조건을 부여한다. ☒
 3. VAE는 주로 지도 학습 환경에서 분류 문제 해결에 활용된다.
 4. VAE는 재구성 오차 외에 어떠한 추가적인 손실 함수도 사용하지 않는다.
 5. VAE는 노이즈가 없는 깨끗한 데이터만을 입력으로 받아 학습해야 한다.
5. VAE의 학습 과정에서 사용되는 손실 함수(Loss Function)는 크게 두 가지 항목으로 구성됩니다. 다음 중 이 두 가지 손실 함수에 대한 설명으로 가장 적절한 것은 무엇인가?
1. **재구성 손실(Reconstruction Loss):** 잠재 공간의 정규성을 유지하기 위한 손실이며, **KL Divergence Loss:** 원본 데이터를 얼마나 정확하게 복원하는지 측정하는 손실이다.
 2. **재구성 손실:** 원본 데이터를 얼마나 정확하게 복원하는지 측정하는 손실이며, **KL Divergence Loss:** 잠재 공간이 표준 정규 분포와 얼마나 다른지 측정하는 손실이다. ☒ MSE, Binary Cross-Entropy
 3. **재구성 손실:** 인코더의 가중치를 조절하는 손실이며, **KL Divergence Loss:** 디코더의 가중치를 조절하는 손실이다.
 4. **재구성 손실:** 생성된 데이터의 다양성을 증가시키는 손실이며, **KL Divergence Loss:** 생성된 데이터의 품질을 향상시키는 손실이다.
 5. VAE는 단일 손실 함수만을 사용하여 학습하며, 별도의 두 가지 손실 함수를 사용하지 않는다.
6. VAE의 잠재 공간(Latent Space)에서 새로운 샘플을 생성하는 과정에 대한 설명으로 가장 적절한 것은 무엇인가?
1. 훈련 데이터셋에 포함된 기존 샘플 중에서 무작위로 하나를 선택하여 디코더에 입력한다.
 2. 잠재 공간의 평균(μ)과 분산(σ)을 사용하여 표준 정규 분포에서 무작위로 잠재 벡터를 샘플링한 후 디코더에 입력한다. ☒
잠재 공간이 따르도록 학습된 확률 분포(예: 표준 정규 분포)에서 무작위로 잠재 벡터(latent vector)를 샘플링
 3. 인코더에 새로운 입력 데이터를 주어 잠재 벡터를 얻은 후, 이를 다시 인코더에 재입력하여 새로운 데이터를 생성한다.

4. 디코더의 모든 가중치를 무작위로 초기화한 후, 순방향 전파를 통해 새로운 데이터를 생성한다.
 5. VAE는 데이터를 압축하고 복원할 뿐, 새로운 데이터를 생성하는 능력은 없다.
7. Variational AutoEncoder (VAE) 학습에서 ELBO(Evidence Lower Bound)를 최대화하는 것이 가지는 의미로 가장 적절한 것은 무엇인가?
1. 인코더와 디코더의 신경망 구조를 복잡하게 만들어 모델의 표현력을 높이는 것이다.
 2. 잠재 변수가 따르는 사후 확률 분포 $p(z|x)$ 를 정확히 추정하는 것과, 입력 데이터를 효과적으로 재구성하는 두 가지 목표를 동시에 달성하는 것이다. 
 3. 모델이 무조건적으로 더 많은 데이터를 생성하도록 유도하는 것이다.
 4. 잠재 공간의 차원 수를 늘려 정보 손실을 최소화하는 것이다.
 5. 재구성 오차만을 최소화하여 모델의 일반화 성능을 향상시키는 것이다.
8. PyTorch 프레임워크를 사용하여 TensorBoard로 신경망 모델 학습을 모니터링할 때, 각 Epoch 또는 Step마다 손실(Loss) 값의 변화를 시각적으로 추적하기 위한 코드적인 접근 방법으로 가장 적절한 것은 무엇인가?
1. `torch.save(model.state_dict(), 'model.pth')` 함수를 사용하여 매 학습 단계마다 모델 가중치를 저장한다.
 2. `print(f"Epoch {epoch}, Loss: {loss.item()}")` 명령어를 사용하여 콘솔에 손실 값을 출력한다.
 3. SummaryWriter 객체를 생성하고, `writer.add_scalar()` 메서드를 사용하여 스칼라 값을 TensorBoard 이벤트 파일에 기록한다. 
`torch.utils.tensorboard.SummaryWriter` : 객체 생성,
`writer.add_scalar('Loss/train', loss.item(), global_step=epoch)` : 기록
 4. `matplotlib.pyplot.plot()` 함수를 사용하여 학습이 완료된 후 손실 값을 그래프로 그린다.
 5. `torch.optim.Adam` 옵티마이저의 `step()` 메서드를 호출하여 손실 값을 자동으로 기록한다.
9. MLflow를 사용하여 머신러닝 모델의 학습 과정을 모니터링하고, 특정 실험(Experiment) 내에서 각 에포크(Epoch) 또는 스텝(Step)마다 손실(Loss)과 정확도(Accuracy) 같은 메트릭(Metric)의 변화를 기록하고자 할 때, 가장 적절한 MLflow 코드 사용법은 무엇인가?
1. `mlflow.save_model(model, "my_model")` 함수를 사용하여 매 학습 단계마다 모델을 저장한다.
 학습이 완료된 모델을 저장
 2. `mlflow.set_tag("loss", current_loss)` 함수를 사용하여 손실 값을 태그로 기록한다.
 특정 속성(예: Git 커밋 해시, 모델 버전 등)을 기록하는 데 사용되며, 시간 흐름에 따른 값의 변화를 추적하는 메트릭 기록에는 적합하지 않음

3. `mlflow.start_run()` 컨텍스트 내에서
`mlflow.log_param("learning_rate", 0.01)` 함수를 사용하여 학습률을 기록한다.
실험 시작 시 설정된 고정된 하이퍼파라미터(예: 학습률, 배치 크기)를 기록할 때 사용
4. `mlflow.start_run()` 컨텍스트 내에서
`mlflow.log_metric("train_loss", loss_value, step=epoch_num)`
함수를 사용하여 메트릭을 기록한다. 
5. `mlflow.log_artifact("data.csv")` 함수를 사용하여 학습 데이터를 아티팩트로 저장한다.
학습 데이터, 그래프, 모델 요약 파일 등 모델 학습과 관련된 파일을 저장하는 데 사용

10. PyTorch에서 자신만의 데이터를 신경망 모델에 효율적으로 공급하기 위해 커스텀 Dataset 클래스를 정의하고자 합니다. 다음 중 PyTorch의 `torch.utils.data.Dataset` 을 상속받아 커스텀 데이터셋을 구현할 때, 반드시 재정의(override)해야 하는 메서드로 올바르게 짝지어진 것은 무엇인가?

1. `__init__` 과 `__call__`
2. `__len__` 과 `__iter__`
3. `__getitem__` 과 `__setitem__`
4. `__len__` 과 `__getitem__` 
5. `__init__` 과 `__getitem__`