데이터 과학 연습문제 과제

이름: 강민성

학번:2226082

1. 텐서플로를 한마디로 어떻게 정의할 수 있나요? 텐서플로의 주요 특징은 무엇인가요? 인기 있는 딥러닝 라이브러리는 어떤 것이 있나요?

정의: 텐서플로(TensorFlow)는 데이터 흐름 그래프를 활용하여 수치 연산을 수행하고, 주로 딥러닝 모델을 개발 및 학습하는 데 사용되는 오픈소스 라이브러리입니다.

특징:

주요 특징:

1.다차원 배열(텐서) 기반 연산 지원

2.자동 미분(Gradient Tape)을 통한 효율적인 최적화

3.다양한 플랫폼(CPU, GPU, TPU)에서 실행 가능

4.모델 구축, 학습, 배포를 위한 도구 제공 (e.g., Keras API)

인기 있는 딥러닝 라이브러리:

1.PyTorch

2.TensorFlow

3.Keras (TensorFlow에 포함됨)

4. MXNet, JAX 등

2. 텐서플로는 넘파이를 그대로 대체할 수 있나요? 두 라이브러리의 주요 차이점은 무엇인가요?

넘파이와의 차이:

1.TensorFlow는 GPU 및 TPU 가속을 지원하지만, NumPy는 CPU에서만 작동합니다.

2.TensorFlow는 자동 미분(Autograd)을 지원하며, 딥러닝 모델의 학습에 최적화되어 있습니다.

3.NumPy는 주로 수치 연산과 배열 연산에 특화되어 있어 일반적인 데이터 과학 작업에 적합합니다.

3. tf.range(10)과 tf.constant(np.arange(10))의 결과는 같은가요?

결과 비교:

1.tf.range(10): TensorFlow 텐서를 반환하며, [0, 1, 2, ..., 9]의 정수 배열입니다.

2.tf.constant(np.arange(10)): NumPy 배열을 TensorFlow 텐서로 변환한 것으로, 같은 값인 [0, 1, 2, ..., 9]를 반환합니다.

차이: 자료 생성 방식이 다르며, tf.range()는 TensorFlow 네이티브 방식으로 생성됩니다.

4. 일반 텐서 외에 텐서플로에서 사용할 수 있는 6가지 데이터 구조는 무엇인가요?

TensorFlow 데이터 구조:

1.tf.Tensor: 기본적인 다차원 배열

2.tf.Variable: 학습 가능한 변수

3.tf.SparseTensor: 희소 행렬 표현

4.tf.RaggedTensor: 불규칙적인 텐서

5.tf.data.Dataset: 데이터 파이프라인 구성

6.tf.TensorArray: 텐서를 동적으로 저장

5. tf.keras.losses.Loss클래스를 상속하거나 일반 함수를 작성하여 사용자 정의 손실 함수를 정의할 수 있습니다. 언제 사용해야 하나요?

사용 시점:

1.기본 손실 함수(e.g., MSE, CategoricalCrossentropy)로 원하는 동작을 구현하기 어려울 때.

2.문제 특화된 손실 함수가 필요할 때 (e.g., Custom Regularization).

6. tf.keras.metrics.Metric클래스를 상속하거나 함수를 정의하여 지표를 정의할 수 있습니다. 언제 사용해야 하나요?

사용 시점:

1.기존 지표(e.g., Accuracy, Precision, Recall)가 요구 사항에 맞지 않을 때.

2.학습 과정에서 추가적인 통계를 추적할 필요가 있을 때.

7. 언제 사용자 정의 층 또는 사용자가 정의한 모델을 만들어야 하나요?

사용 시점:

1.기본 레이어나 모델이 문제를 해결하기에 적합하지 않을 때.

2.예를 들어, 복잡한 attention mechanism, custom RNN cell, 또는 domain-specific operation을 구현해야 할 때.

8.사용자 정의 훈련 반복을 만들어야 하는 경우는 언제인가요?

사용 시점:

1.매우 복잡한 학습 루프(e.g., reinforcement learning)나 맞춤 최적화 알고리즘이 필요한 경우.

9. 케라스의 사용자 정의 구성 요소 입력 요소가 입력의 파이썬 코드로 담을 수 있나요?

구성 요소:

1.사용자 정의 build()및 call()메서드를 통해 입력 데이터와 레이어를 설정할 수 있습니다.

10. 텐서플로로 특정 분류를 맞추는 함수를 만든다면 따라야 할 주요 규칙은 무엇인가요?

규칙:

1.입력 데이터 형태를 정확히 설정 (input\_shape).

2.손실 함수 및 최적화 알고리즘을 명확히 정의.

3.데이터 정규화 및 배치를 적절히 수행.

4.검증 데이터셋으로 성능 검증.

11. 동적인 케라스 모델을 만들어야 할 때는 언제인가요? 어떻게 만들 수 있나요? 왜 전체 모델을 동적으로 만들지 않나요?

동적인 모델을 만들어야 할 때:

1.네트워크 구조가 입력 데이터에 따라 달라져야 하는 경우

2.순환 구조(예: RNN에서 시퀀스 길이가 동적인 경우)

3.복잡한 계산 흐름(조건문, 반복문 등)이 필요한 경우

동적 모델 구현 방법:

1.tf.keras.Model클래스를 상속받아 사용자 정의 모델을 작성

2.Python의 제어 흐름(조건문, 반복문 등)을 활용

전체 모델을 동적으로 만들지 않는 이유:

1.동적인 모델은 디버깅 및 최적화가 어렵고 속도가 느려질 수 있음.

2.일반적인 딥러닝 작업에는 정적인 구조(시퀀셜 모델, 함수형 API)가 더 효율적임.

12번 층 정규화(Layer Normalization)를 수행하는 사용자 정의 층 구현

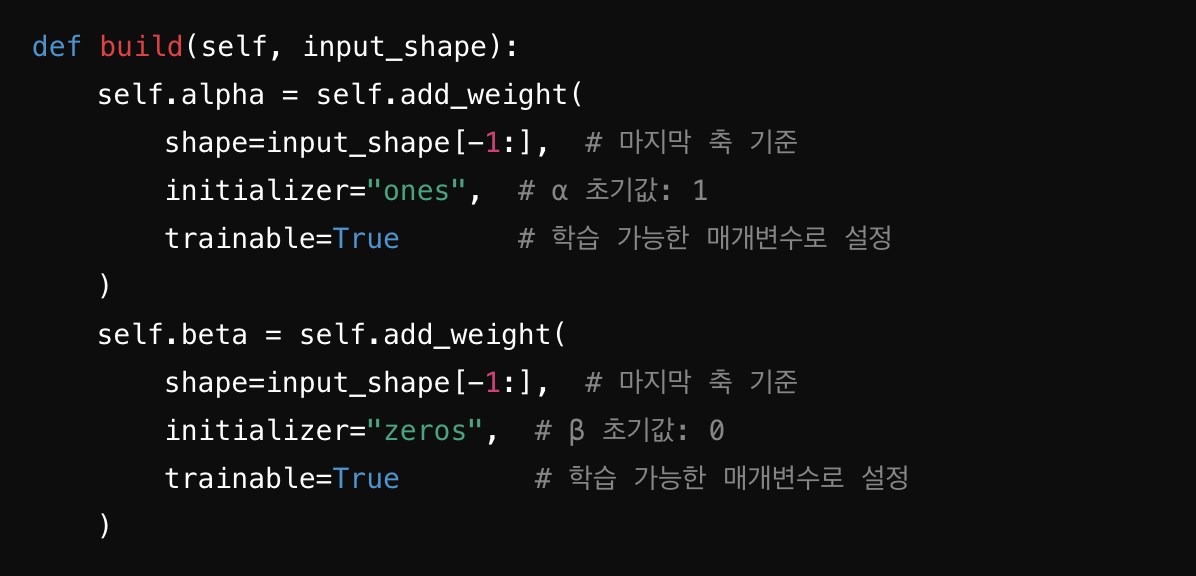
a. build() 메서드에서 두 개의 학습 가능한 가중치 α와 β를 정의합니다.

설명:  
build()메서드는 레이어를 초기화하는 역할을 합니다. 여기에서 학습 가능한 매개변수(α와 β)를 정의하며, 이들은 정규화된 값에 적용됩니다.

1.α는 정규화된 값의 스케일을 조정하며, 초기값은 1이다.

2.β는 정규화된 값에 바이어스를 추가하며, 초기값은 0이다.

a 문제의 코드

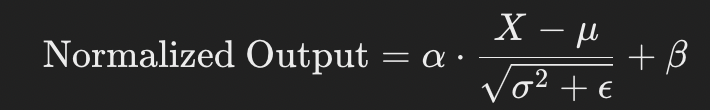


b. call() 메서드는 샘플의 특성마다 평균 μ와 표준 편차 σ를 계산해야 합니다.

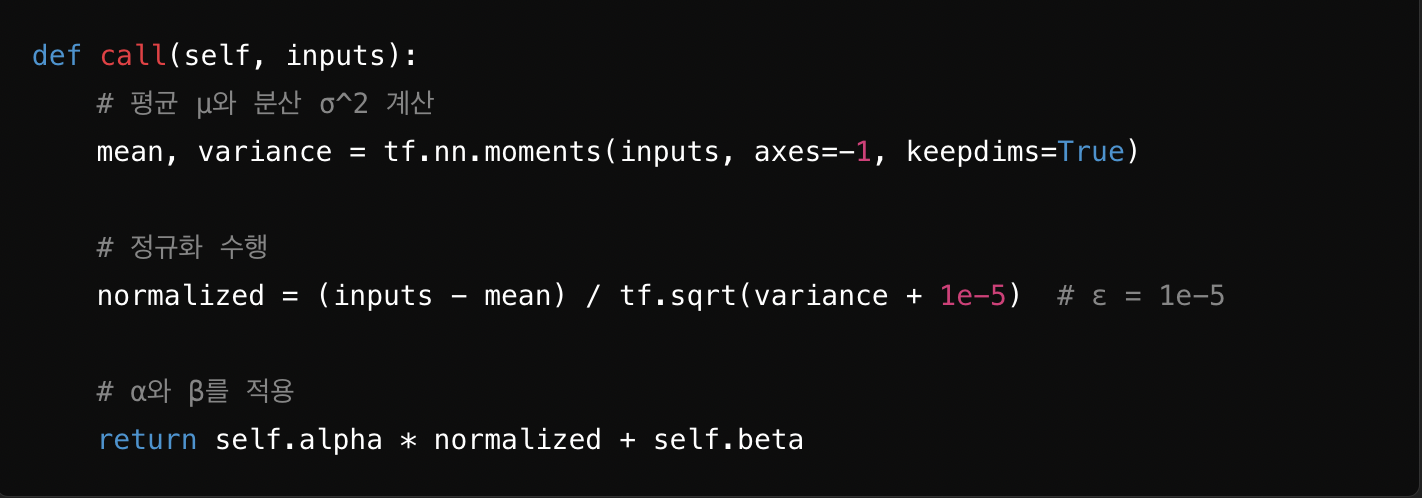
설명:

1.입력 텐서에서 평균 μ와 분산을 계산하여 정규화 수행.

2.안정성을 위해 ε(아주 작은 값)를 추가하여 0으로 나누는 상황 방지.

공식을사용: 

tf.nn.moments를 사용하여 평균과 분산 계산.

b문제코드

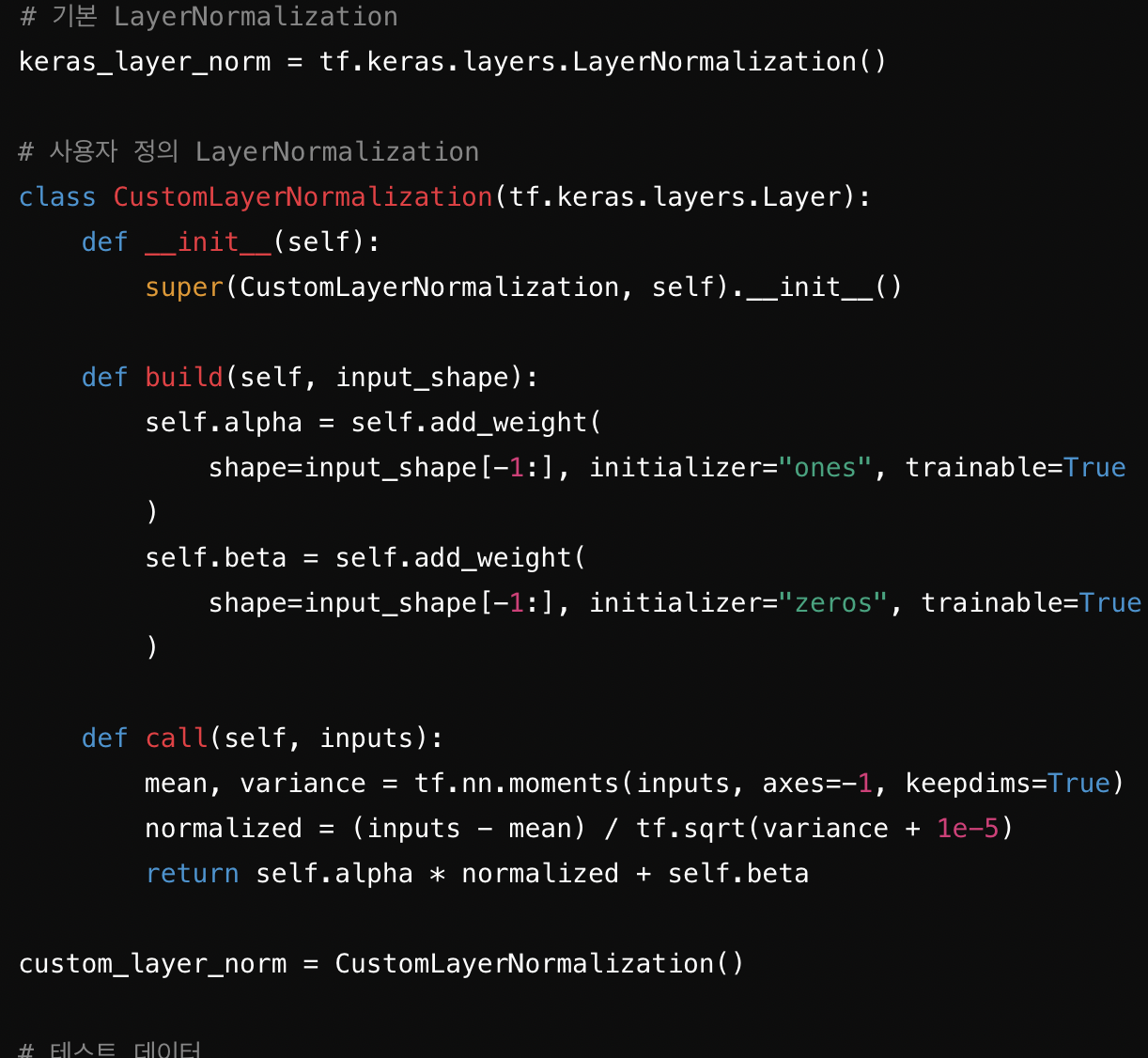
c. 사용자 정의 층이 tf.keras.layers.LayerNormalization층과 동일한 출력을 만드는지 확인하세요.

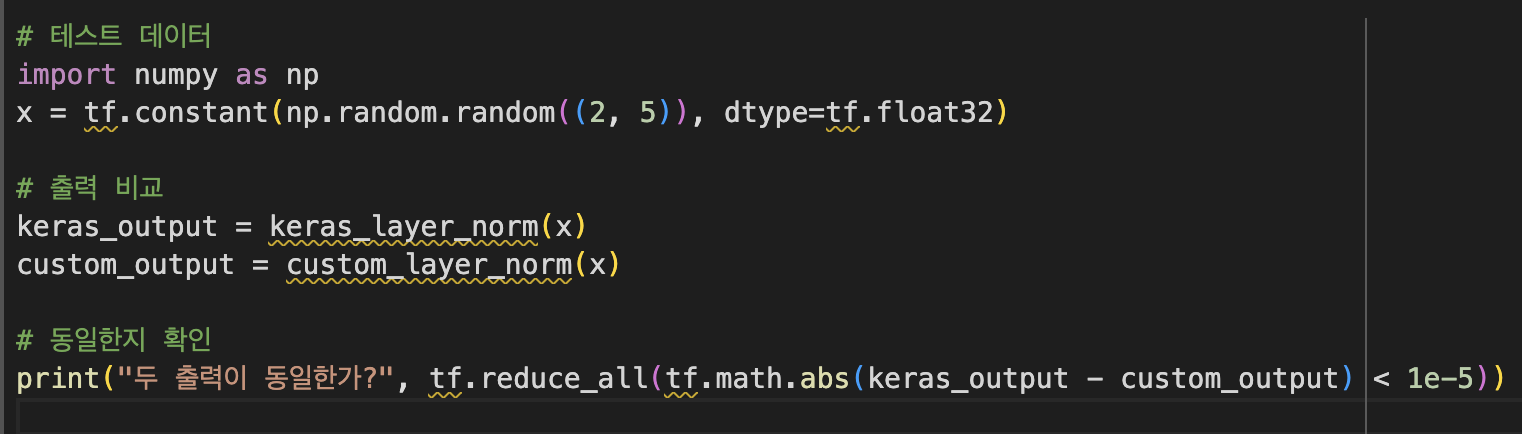
설명:

1.tf.keras.layers.LayerNormalization은 기본 제공 레이어로, 위에서 구현한 사용자 정의 층과 같은 동작을 합니다.

2.두 층의 출력 비교를 통해 동작이 동일한지 확인합니다.

c문제코드





13.사용자 정의 훈련 반복을 사용해 패션MNIST데이터로 모델을 훈련 해보세요

a. 사용자 정의 훈련 반복 및 에포크 출력

1.각 에포크와 반복에서 손실, 정확도를 출력합니다.

2.최종적으로 검증 손실과 정확도를 출력합니다.

b. 상위 층과 하위 층의 학습률이 다른 옵티마이저 사용

1.옵티마이저를 구성하여 특정 층마다 다른 학습률을 적용합니다.

13번 코드 줄이 많은 관계로 파일로 제출하겠습니다.