

2강 TensorFlow 소개 및 기본 사용법

TensorFlow 소개

- **정의:** Google 에서 개발한 오픈 소스 머신러닝 라이브러리입니다.
- **주요 특징:**
 - **텐서 (Tensor):** 데이터를 표현하는 기본 단위로, 행렬을 일반화한 다차원 배열입니다.
 - **그래프 기반 계산:** 계산을 노드(연산)와 엣지(데이터 흐름)로 표현하는 그래프 기반 계산을 사용합니다.
 - **자동 미분:** 모델의 매개변수를 자동으로 미분하여 최적화할 수 있는 기능을 제공합니다.
 - **Eager Execution:** 코드를 실행하는 즉시 결과를 확인할 수 있어 디버깅이 편리합니다. (TF 2.0 이상 버전).
 - **Keras 통합:** 직관적인 고수준 API 를 제공하여 모델 개발이 용이합니다.
 - **하드웨어 가속:** GPU 및 TPU 하드웨어 가속화 지원으로 모델 학습 시간을 단축합니다.

텐서 조작 및 데이터 전처리

- **텐서 생성:** `tf.constant()` 상수, `tf.Variable()` 가중치, `tf.zeros()`, `tf.ones()` 등으로 텐서를 생성합니다.
- **데이터 정규화 기법:**
 - **표준화 (Standardization):** 데이터를 평균 0, 표준편차 1로 변환합니다.
 - **최소-최대 정규화 (Min-Max Scaling):** 데이터를 0-1 범위로 변환합니다.
- **tf.data API:** 데이터 입력 파이프라인 구축을 위한 고수준 API 로, `shuffle()`, `batch()`, `prefetch()`, `cache()` 등의 메서드를 활용하여 데이터 로딩 병목 현상을 줄이고 성능을 최적화합니다.

모델 훈련 및 평가

- 케라스 Sequential API: 다층 신경망 모델을 간결하게 구축할 수 있습니다.
- 모델 훈련(model.fit): epochs(반복 횟수), batch_size(배치 크기), validation_split(검증 세트 비율), callback(조기 종료, 모델 저장 등)를 설정하여 학습을 진행합니다.
- 옵티마이저(Optimizer): 경사하강법 알고리즘(예: SGD, Adam)을 사용하여 손실 함수를 최소화하는 방향으로 가중치를 업데이트합니다.
- 모델 평가 지표: 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1 점수 등을 사용하여 모델 성능을 측정합니다.

3강 회귀 분석

회귀분석 모델 종류

- **선형 회귀 (Linear Regression):** 데이터 간의 직선적 관계를 모델링합니다.
- **다항식 회귀 (Polynomial Regression):** 비선형 관계를 곡선 형태로 모델링합니다. 과적합 위험이 존재합니다.
- **비선형 회귀(Nonlinear Regression):** 다층 신경망(Multi-layer Perceptron) 구조를 활용하여 비선형 활성화 함수(ReLU, Sigmoid 등)를 통해 복잡한 비선형 관계를 학습합니다.
- **정규화 회귀 (Regularized Regression):** 릿지(Ridge, L2), 라쏘(Lasso, L1), 엘라스틱 넷(Elastic Net) 등을 통해 과적합을 방지합니다.

정규화

- **과적합 방지 기법**
 - **정규화:** L1, L2 정규화로 모델 복잡도에 페널티를 부과합니다.
 - **드롭아웃 (Dropout):** 학습 시 무작위로 뉴런의 일부를 비활성화하여 과적합을 방지합니다.
 - **조기 종료 (Early Stopping):** 검증 손실(val_loss)이 특정 횟수(patience) 동안 개선되지 않으면 학습을 중단하고 최적의 가중치로 복구합니다.
- **K-Fold 교차 검증 (Cross-Validation):** 데이터를 K 개의 폴드(부분집합)로 나누어 각 폴드를 번갈아 가며 검증 세트로 사용하여 모델의 일반화 성능을 더 정확하고 안정적으로 평가합니다.

4 강 경사하강법

기본 개념

요소	역할
비용 함수 (Loss Function)	모델의 예측값과 실제값 사이의 오차를 측정하는 함수. 경사 하강법의 목표는 이 값을 최소화하는 것입니다.
경사 (Gradient)	비용 함수를 파라미터(가중치)로 미분한 값으로, 현재 위치에서 비용 함수가 가장 가파르게 증가하는 방향과 크기를 나타냅니다.
학습률 (Learning Rate)	파라미터를 업데이트할 때 이동할 보폭을 결정하는 하이퍼파라미터입니다.

- 경사 하강법 (Gradient Descent): 손실 함수의 값이 최소가 되는 지점을 찾아가는 최적화 알고리즘입니다.
 - 가중치 업데이트

$$W_{new} = W_{old} - \eta \times \left(\frac{\partial L}{\partial W} \right)$$

학습률(η): 가중치 업데이트 보폭을 결정하는 중요한 하이퍼파라미터입니다.

학습률의 중요성

- 학습률이 너무 클 때: 최솟값을 건너뛰어 발산하거나 불안정한 학습이 발생할 수 있습니다.
- 학습률이 너무 작을 때: 최솟값에 도달하는 데 너무 오랜 시간이 걸리거나 지역 최솟값(Local Minimum)에 갇힐 수 있습니다.

경사하강법의 종류

구분	전체 데이터 사용량	특징
배치 경사 하강법 (Batch GD)	전체 데이터셋을 한 번에 사용	안정적이지만, 데이터가 클 때 학습 속도가 매우 느립니다.
확률적 경사 하강법 (SGD)	랜덤하게 선택된 1개의 데이터만 사용	학습 속도가 빠르지만, 비용 함수의 움직임이 불안정하고 불규칙합니다.
미니 배치 경사 하강법 (Mini-Batch GD)	Batch Size 크기만큼의 데이터셋을 사용	가장 널리 사용되는 방식으로, 안정성과 속도를 모두 확보할 수 있습니다.

응용 알고리즘 (Optimizer)

- **Adam (Adaptive Moment Estimation):** 현재 가장 널리 사용되는 옵티마이저 중 하나로, 과거 그래디언트의 지수 이동 평균을 활용하여 각 파라미터에 대한 학습률을 독립적으로 조정합니다.
- **RMSProp:** 그래디언트의 제곱에 대한 지수 감쇠 평균을 사용하여 학습률을 조정합니다.
- **Adagrad:** 학습 초기에 큰 업데이트를 하다가 점차 업데이트 크기를 줄이는 방식입니다.

5 강 합성곱 신경망(CNN) 기초

CNN 개요 및 특징

- **정의:** 합성곱 신경망(CNN, Convolutional Neural Network)은 딥러닝에서 주로 이미지나 비디오 데이터를 처리하는 데 사용되는 인공 신경망의 한 종류입니다.
- **특징:** 인간의 시신경이 사물을 인식하는 방식을 모방하여 설계되었으며, 이미지의 공간적 계층 구조를 학습하여 특징을 효과적으로 추출합니다.
- **DNN 과의 차이점:** 완전 연결 신경망(DNN)과 달리, CNN 은 합성곱 연산을 통해 이미지의 지역적 특징을 추출하고 유지하여 공간적 정보 손실을 방지합니다.

CNN 핵심 구성 요소

구성 요소	역할	주요 특징
합성곱 계층 (Convolutional Layer)	필터(커널)를 사용하여 입력 데이터의 지역적 특징을 추출합니다.	파라미터 공유를 통해 효율적인 학습이 가능합니다.
활성화 함수 (Activation Function)	신경망에 비선형성을 도입하여 모델의 표현력을 향상 시킵니다.	주로 ReLU 함수($f(x)=\max(0, x)$)가 사용되며, 계산 효율성과 경사 소실 문제 완화에 장점이 있습니다.
풀링 계층 (Pooling Layer)	특성 맵의 크기를 줄여 계산 효율성을 향상시키고, 위치 변화에 대한 내성을 제공합니다.	최대 풀링 (Max Pooling)과 평균 풀링(Average Pooling) 방식이 있습니다.
완전 연결 계층 (Fully-Connected Layer)	추출된 특징을 기반으로 최종 분류/예측을 수행합니다.	이전 계층의 특성 맵을 1차원 벡터로 평탄화(Flatten)하여 처리합니다.

CNN 동작 원리

- **합성곱 연산:** 필터(커널)를 입력 데이터 위에서 이동시키면서 필터와 입력 데이터 해당 부분 간의 곱셈-누적 연산을 수행합니다.
- **스트라이드 (Stride):** 필터가 입력 데이터 위를 한 번에 이동하는 간격으로, 값이 클수록 특성 맵의 크기가 작아집니다.
- **패딩 (Padding):** 입력 데이터의 가장자리에 특정 값(주로 0)을 추가하여 출력 특성 맵의 크기를 조절하는 기법입니다. Same Padding 을 통해 입력과 출력의 크기를 동일하게 유지할 수 있습니다.

6 강 의료영상 분석에서의 CNN 모델

의료영상 분석의 필요성: 세그멘테이션

- 세그멘테이션 (Segmentation): 이미지를 의미 있는 영역으로 분할하는 작업으로, 각 픽셀에 레이블을 할당합니다.
- 중요성: 종양 영역을 정확히 표시하여 크기 측정, 장기 경계 추출을 통한 수술 계획 수립, 방사선 치료 계획 등 정확한 진단과 치료를 위한 필수 기술입니다.
- 분류 vs 세그멘테이션: 분류는 이미지 전체에 하나의 레이블을 부여하지만, 세그멘테이션을 픽셀별 레이블을 부여하여 정확한 위치 정보를 제공합니다.

U-Net 아키텍처

- 개발 및 활용: 2015 년 개발되었으며, 현재 의료영상 세그멘테이션의 표준 모델로 사용됩니다.
- 특징:
 - U 자 형태의 대칭 구조 (Encoder-Decoder)를 갖습니다.
 - Skip Connection 이 있어 다운샘플링 시 손실된 공간 정보를 복원하고 정확한 경계 검출을 가능하게합니다.
 - 적은 데이터로도 높은 성능을 낼 수 있어 의료 데이터 레이블링 비용 문제를 해결하는 데 유리합니다.
- 구성:
 - Encoder (Contracting Path): Double Conv + Max Pooling 으로 공간 해상도를 낮추고 채널 수를 늘리며 맥락(Context) 특징을 포착합니다.
 - Decoder (Expansive Path): Conv2DTranspose(Upsampling) + Skip Connection(Concatenation) + Double Conv 로 특징을 복원하고 공간 해상도를 높입니다.

U-Net 학습 및 평가

- 손실 함수: Combined Loss (Binary Cross-Entropy + Dice Loss)를 주로 사용하여 경계선과 전체 형태 유사도를 모두 최적화합니다.
- 평가 지표: Dice Coefficient (Dice Score) 및 IoU (Intersection over Union)가 사용되며, 1 에 가까울수록 정답과 예측이 유사합니다.
- 학습 팁: Adam 옵티마이저($lr = 1e - 4$), Data Augmentation, ModelCheckpoint, ReduceLROnPlateau 콜백 등을 활용하여 안정적인 학습을 수행합니다.