

## 2강

지난 강의에서는 이미지 분류에 대한 간략한 소개

이미지 분류를 할 때, 시스템이 이미지를 입력 받으면 시스템은 예상 카테고리나 분류를 인지함

단순한 문제일 수도 있으나 기계에 한해서는 상당히 난이도 있는 작업

컴퓨터가 이미지를 볼 때, 시각적 정보가 아닌 특정 수들의 집합으로 정보를 받아들임

각각의 픽셀은 RGB 3가지 채널을 가진 고유 값을 지님 - 이러한 픽셀 값들이 컴퓨터가 실제로 보게 되는 것, 이미지의 약간의 변형을 주어도 이미지 값의 변동폭이 큼

광원, 자세, 폐색, 배경 등과 같은 요소도 이미지 인식에 반영되어 이미지 인식에 차질을 줄 수 있음 - 알고리즘이 해결해야 할 문제

지난 강의에서 조금 언급했듯이, 물체를 분석할 때 시각적 인식에 관하여 윤곽은 대개 주요한 위치를 차지함 - 각각의 윤곽을 계산하고, 모서리와 경계를 분석하고 정렬하여 규칙성을 찾고 물체를 인식

하지만 이러한 방식은 작동하지 않음

- 오류 발생이 잦으며, 다른 카테고리에 대해 접근 시 전 과정을 다시 시작해야함

모든 것을 작동하게 하기 위한 방식이 데이터 기반 접근 방식에 대한 아이디어

데이터 접근 방식

- 1 이미지와 레이블 데이터셋 수집
- 2 분류기를 훈련시키기 위한 머신러닝
- 3 새로운 이미지로 모델 평가

단순 분류 알고리즘: K-최근접 이웃

- 모든 데이터와 레이블을 학습시킨 후, 가장 비슷한 훈련 이미지를 토대로 예측-CIFAR10

비교 기능에 따른 분류 방식의 차이: L1, L2

L1: 각 이미지의 개별 픽셀 값의 차이를 계산, 이후 전체 픽셀 값의 합산을 통한 분류 / 좌표축을 따르는 경향

최근접 이웃 알고리즘을 적용할 때의 모습: 2차원 평면으로 구성, 일종의 공간을 쪼개고, 인근 지점에 따라 색으로 구분하는 방식 - K개의 가까운 이웃의 비율에 따라 영역 형성, 이에 따른 K값 조정

L2: 각 거리의 제곱합의 합으로 구성/자연스러운 분류

입력 기능이 있고 개별 벡터의 중요성이 부각된다면 L1이 적합할 수 있음/개별 벡터의 정체를 판단할 수 없거나, 중요성이 크지 않은 경우, L2에 좀 더 적합

K값 설정 방법과 같은 선택 사항: 하이퍼파라미터

하이퍼파라미터 설정 관련

1 데이터셋에서 제일 잘 작동하는 하이퍼파라미터 선정

- K값을 1로 설정 시, 다음과 같은 조건에 부합/새로운 데이터에 대한 적응 X

2 전체 데이터 세트를 가져와, 훈련과 테스트 세트로 분할

- 다양한 하이퍼파라미터 적용 후, 테스트 세트로 확인, 새로운 데이터에 대한 성능 보장 X

3 데이터셋을 훈련/검증/테스트로 분류 후, 검증세트에 하이퍼파라미터 적용 후 테스트 세트로 평가

- 검증과 테스트 세트의 데이터 분류 必

4 교차검증

- 한 데이터를 여러 개의 세트로 분할, 작은 데이터셋을 다룰 땐 유용하나, 딥러닝 분야에선 자주 이용되지 않음/하이퍼파라미터에 대한 높은 정확성 - 많은 재원을 필요로 함

훈련세트와 검증세트의 차이점

- 알고리즘이 훈련 세트의 레이블을 볼 수 있는 지 없는 지/검증세트의 알고리즘은 직접적인 레이블에 접근 불가

테스트 세트가 모든 데이터를 나타내는 것은 아님-기본 확률 분포에서 추출되어야 하는 것을 전제

K-최근접 이웃은 이미지에선 거의 사용되지 않음

- 테스트에 매우 큰 시간 할애 + 픽셀에 거리에 따른 분류방식이 적합하지 않음
- 차원의 저주

선형 분류

- 전체 신경 네트워크와 CNN을 구축하는 데 도움을 줌
- 여러 개의 선형 분류가 합쳐서 신경망 구성

CIFAR10에서의 선형 분류

- 파라메트릭 모델/이미지를 받아들인 후, 데이터  $x$ 와 매개변수  $w$ 를 받은 함수 계산 - 10

개의 범주 각각에 해당하는 점수 출력 – 점수를 바탕으로 확률이 제일 높은 개체 선정

- 테스트에 실제 훈련 데이터가 필요 X, 쾌적한 환경에서 구동 가능

### 3강

손실 함수의 최적화

인식의 어려움, 데이터 기반 접근 방식에 대한 이해

컴퓨터가 보는 수치적 숫자와 실제로 보는 이미지의 차이

- 전체 데이터 기반 사고방식에 대한 간단한 소개로 k-최근접 이웃 분류기 등이 존재

훈련 데이터를 기반으로 데이터의 요점을 클래스로 분기하기 위한 경계를 학습되는 데 사용됨

- 이는 곧 교차 검증 및 하이퍼파라미터 변수 설정에 대한 논의로 이어짐

선형 분류기 – 이미지를 가져온 후 긴 벡터로 확장, 이후 이미지 픽셀을 나타내는 벡터를 변환, 이후 매개변수를 거쳐 CIFAR-10의 경우 10개 클래스 각각에 대한 확률을 계산

- 10개의 클래스에 대한 매개변수가 해당 픽셀이 해당 클래스에 어느 정도의 영향을 미치는지 알려주는 항목  $W$  – 가중치

알맞은 매개변수를 설정

손실 함수 – 매개변수에 대한 성능을 지표로 나타내는 함수

최적화 절차 수반

손실 함수에 대한 논의를 할 때, 이미지의 실제 픽셀 값과 예측을 원하는 항목(레이블)로 분류(1-10)

실제 대상  $Y$ 와 함께 함수  $f$ 에서 나오는 예측 점수를 가져와 해당 훈련 예제에 대한 예측의 성능을 계산하는 손실함수 출력

- 최종손실  $L$ 은 각각에 대한 전체 데이터 세트에 대해 합산된 손실의 평균

다중 클래스 SVM 손실

- 각 클래스 쌍의 손실 합계
- 전체 데이터 세트에 대한 손실 – 각 클래스 쌍의 손실 합계 /  $N$

최소 0과 다른 값을 취하는 이러한 형태의 손실 함수 – 힌지 손실/특정 값을 넘어가는 순간 0으로 고정(예제에 대한 분류를 마침)

20:45

손실이 0인 함수를 찾았을 때, 손실이 0인 다른 함수도 존재할 수 있음

머신러닝의 요점은 훈련 데이터를 사용하여 일부 분류기를 찾은 후 테스트세트에 적용

- 훈련 데이터의 성능 < 테스트 데이터의 성능에 더 높은 관심

정규화

- 과적합을 낮추기 위하여, 손실 함수에 항을 추가 – 간단한 매개변수를 선정하는 데 도움
- 데이터 손실, 정규화 손실

L2정규화, 가중치 감소 등을 일반적으로 사용

- 여러 방식으로 변형 가능

훈련 데이터에 너무 맞추기 보단 모델의 복잡성을 줄이는 것에 초점

다항 로지스틱 회귀, 소프트맥스

- 딥러닝 분야에선 좀 더 일반적인 선택지
- 손실에 몇 가지 추가 의미를 부여
- 모든 점수를 양수화 시킨 후 해당 지수의 합으로 다시 정규화
- 소프트맥스를 활용한 확률 분포로 표현 – 합계 1

손실이 실제 클래스 확률의  $-\log$ 의 형태(손실을 측정하는 손실함수의 원리)

최소 손실은 0, 최대 손실은 무한대( $-\log 0$ )

소프트맥스 / SVM

선형 분류 측면

SVM: 각 클래스 간의 차이값을 계산, 이후의 작업 X

소프트맥스(교차 엔트로피 손실): 확률을 계산 / 항상 확률 합을 1로 조정, 이에 따른 가중치 부여

구조적으로 매개변수의 값이 주어지면 알고리즘이 얼마나 잘 작동하는지 결정하는 손실 함수를 지정하고, 모델 복잡성에 패널티를 부여하는 방법에 대한 정규화 진행 후 결합

- 손실을 최소로 하는 W 값을 찾아야함

- 최적화 과정 수반

최적화

외부에서 가져온 손실 함수 사용 지양

기울기 / 도함수 계산(미분)을 통한 경사 측정

경사가 클수록 함수의 증가폭이 커짐

- 많은 딥러닝은 함수의 기울기를 계산하고, 이를 사용하여 매개변수 벡터를 반복적으로 업데이트

모든 구간을 미분하는 것은 매우 비효율적인 계산 방식

- 손실에 대한 표현식을 활용하여 시간 단축
- 기울기에 대한 분석 표현식을 알아낸 후, 적용

-실제 기울기를 계산할 땐 분석 표현식을 사용

경사하강법

매개변수를 일정 값으로 초기화한 후, 손실과 경사를 계산한 다음 경사 방향의 반대 방향으로 가중치를 업데이트

- 함수가 가장 크게 증가하는 방향-마이너스 기울기는 가장 크게 감소하는 방향(학습률)

전체 훈련 세트에 대한 손실과 기울기를 계산할 시, 표본이 많을수록 분석 속도가 매우 더디게 됨

- SGD경사 하강법: 매 반복마다 미니배치를 샘플링하여 전체 합계의 추정치와 실제 기울기의 추정치를 계산

모든 심층 신경망에 사용하는 기본 알고리즘

색상 히스토그램

- 해당 색상 버킷 중 하나로 매핑한 다음 각 픽셀에 속하는 픽셀 수를 계산
- 해당 이미지의 색상 분포를 알 수 있음 / 간단한 특징의 벡터