**기계학습 알고리즘의 실험적 분석과 결과**

한양대학교

김기환

luceinaltis2020@naver.com

**초록**

실생활에서 이미지 처리를 위해 사용되는 알고리즘은 과거에 비해 다양해졌다. 다양한 알고리즘의 출현은 처한 상황에 적합한 알고리즘을 선택하는 문제로 귀결된다. 본 논문에서는 각 알고리즘들의 분류 속도, 학습 속도, 정확도를 비교해보고 상황에 따른 알고리즘이 무엇인지 분석하고자 한다.

1. **문제 정의**

이미지 분류에서 기계학습 알고리즘을 선택하는 방법은 다양하다. 더 좋은 알고리즘은 더 많은 자원 활용을 요구하기도 하며, 결과적으로 불필요하게 많은 자원을 이용해 낭비하는 경우가 생긴다. 본 논문에서는 이미지 분류 알고리즘에 대해 모델링하고 수치적으로 분석하고자 한다.

* 1. **실험 조건**

실험으로 사용하는 데이터는 MNIST의 손글씨 8 x 8 크기의 이미지를 사용한다. 1797개의 이미지 갯수를 사용하고, 이미지들은 0부터 9까지 해당되는 숫자로 라벨링 되어있다. 데이터는 훈련 데이터와 테스트 데이터를 7대 3의 비율로 나눈다. 이미지는 회색조로 표현된다.

자연이(가) 표시된 사진

자동 생성된 설명

그림 1: MNIST 손글씨

* 1. **분석 지표**

알고리즘은 분류 속도, 학습 속도, 그리고, 정확도를 이용하여 분석한다. 분류 속도는 학습이 완료된 모델에서 테스트 데이터의 답을 추론하는데까지 걸리는 시간을 의미한다. 분류 속도가 느리면 이미지 분류를 위한 입력이 주어졌을때 정답을 추론하는 시간이 길어진다. 반대로 분류 속도가 빨라지면 이미지 분류를 위해 입력이 주어질때 정답을 추론하는 시간이 짧아진다. 분류 속도의 정확한 측정을 위해 필요한 클록 수를 기준으로 평가한다.

학습 속도는 분류를 위해 정답을 추론하는 모델을 학습시키는데 걸리는 시간을 의미한다. 학습 속도가 짧으면 모델이 모든 데이터를 탐색하면서 모델을 학습시키는 시간이 작다는 것을 말한다. 반대로 학습 속도가 길면 모델이 모든 데이터를 탐색하면서 모델을 학습시키는 시간이 길어지는 것을 말한다. 학습 속도의 정확한 측정을 위해 필요한 클록 수를 기준으로 평가한다.

정확도는 알고리즘이 정답을 추론했을때 실제 정답을 얼마나 맞추었는지를 말한다. 정확도가 높을수록 알고리즘이 정답을 맞출 확률이 증가하는 것을 의미한다.

* 1. **알고리즘**

실험에 사용되는 알고리즘은 로지스틱 회귀, 서포트 벡터 머신, 선형 판별 분석, K-최근접 이웃 알고리즘, 심층신경망, 합성곱신경망이다.

로지스틱 회귀 알고리즘은 분류를 위한 선형 모델이다. 최대 엔트로피 분류 혹은 로그 선형 분류라고도 부른다. 로지스틱 회귀 알고리즘은 이진 분류로도 사용할 수 있다. 본 논문에서는 싸이킷런의 이진 분류 방법을 이용하여 실험한다. 여기서 사용되는 공식은 그림 2와 같다.

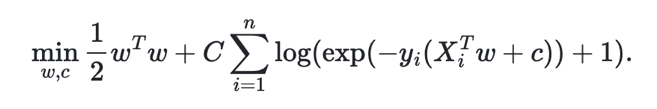


그림 2: 로지스틱 회귀 목표 함수

서포트 벡터 머신은 패턴 인식, 자료 분석을 위한 모델이며 주로 분류와 회귀 분석을 위해 사용한다. 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단한다.

선형 판별 분석은 데이터 분포를 학습해 결정경계를 만들어 데이터를 분류하는 모델이다. 데이터를 결정경계에 해당하는 벡터에 사영시킨 결과를 이용한다. 베이즈 룰을 이용한 방법을 통해서 선형 판별 분석을 이해할 수 있다. 그림 3에서 분모식은 각 데이터에 해당하는 분산을 의미한다. 분자식은 모든 데이터에 해당하는 분산을 의미한다. 선형 판별 분석에서 의도하는 바는 각 데이터에 해당하는 분산은 작게 만들어 같은 라벨에 속하는 데이터끼리는 근접하게하고 모든 데이터에 해당하는 분산은 크게 만드는 것이다. 그러므로 그림 3의 손실함수의 가장 큰 값을 찾는 것이 선형 판별 분석의 목표 함수가 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 3: 선형 판별 분석 손실 함수

K-최근접 이웃 알고리즘은 분류나 회귀에 사용되는 비모수 방식이다. K-최근접 이웃 알고리즘은 가장 가까운 훈련 데이터 K개를 분류에 필요한 데이터로 활용한다. K-최근접 이웃 알고리즘에서 훈련 데이터는 다차원 공간에서의 벡터이다. 분류 단계에서 벡터 간의 사용하는 거리 척도는 실험에서는 유클리드 거리를 이용한다. MNIST 데이터는 8 x 8 크기의 이미지의 64차원이다. K개의 근접 데이터를 분류에 이용한다. 여기서, K개의 갯수가 달라질 경우 변하는 수치를 측정한다. 자세한 사항은 실험의 K-최근접 이웃 알고리즘에 서술한다.

심층신경망은 실험에서 완전 연결된 신경망을 의미한다. 심층신경망에는 입력층, 은닉층, 출력층이 있다. 여기서, 은닉층에 여러개의 레이어로 이루어진 인공신경망이 존재한다. 심층신경망은 다음과 같은 방식으로 학습이 이루어진다.

1. 학습에 필요한 심층신경망 모델을 구성한다.
2. 훈련 데이터를 통해 심층신경망에 데이터를 흘려보낸다.
3. 손실 함수의 기울기를 계산한 후에 역전파 알고리즘을 통해 가중치를 갱신한다.
4. ii와 iii을 손실 함수의 값이 충분히 작아질 때까지 반복한다.

심층신경망은 경사하강법을 사용해 가중치를 갱신한다. 현재 데이터에 해당하는 특징 벡터를 입력으로 했을 때 손실함수의 기울기를 구하고 학습률을 도입하여 손실함수가 최소가 되는 점을 반복하여 추론한다. 심층신경망에서 레이어의 크기는 다양하게 설정할 수 있다. 본 실험에서 사용하는 심층신경망 모델 설명은 실험 심층신경망에서 서술한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 4: 경사하강법

합성곱신경망은 심층신경망에서 나온 새로운 기법이다. 이미지 분류에 적절하며 심층신경망처럼 입력층, 은닉층, 그리고, 출력층으로 구성된다. 또한, 심층신경망의 학습방법과 동일하게 학습이 진행된다. 다만, 합성공신경망은 이미지의 공간적 특성을 반영하여 각 픽셀에 해당하는 데이터와 인접한 픽셀의 데이터를 가중치로 고려한다는 점이 다르다. 이것을 위해 은닉층에 합성곱 연산과 풀링 연산이 추가된다. 합성곱 연산은 해당 레이어의 인접 데이터집합만 다음 레이어와 가중치로 연결된다. 풀링 연산은 인접 데이터집합을 확인하고 가장 높거나 낮은 값을 가지는 부분만 다음 레이어로 흘려보내는 연산이다. 필터는 이미지의 특징을 추출하기위해 사용된다. 필터의 개수를 설정하는 것은 이미지의 분류에 사용되어질 특징의 개수를 설정한다.

알고리즘의 구체적인 구현 방법은 실험에 후술한다.

1. **실험**

분류 속도, 학습 속도, 정확도를 측정하기위해 동일한 조건에서 실험한다. 먼저, 분류 속도를 측정하기위해서 10의 단위로 10부터 100까지의 데이터를 분류할 때 각 알고리즘마다 소요되는 CPU의 클록 수를 측정한다. 학습 속도를 측정하기위해서 MNIST의 데이터 1797개를 모두 학습시키는데 소요되는 CPU의 클록 수를 측정한다. 정확도를 측정하기 위해서 훈련 데이터와 검사 데이터의 비율을 7대 3으로 고정하여 각 알고리즘의 정확도를 비교한다.

* 1. **로지스틱 회귀**

로지스틱 회귀의 사용 공식은 그림 2와 같이 최대우도법을 이용한다.

* 1. **서포트 벡터 머신**

서포트 벡터 머신의 알고리즘은 싸이킷런에서 제공하는 함수를 이용한다.

* 1. **선형판별분석**

선형판별분석의 알고리즘은 싸이킷런에서 제공하는 함수를 이용한다.

* 1. **K-최근접 이웃 알고리즘**

K의 개수는 3개, 5개를 각각 가정한다.

* 1. **심층신경망**

심층 신경망의 레이어는 입력층 64개의 레이어 1개, 은닉층 32개의 레이어 3개, 출력층 10개의 레이어 1개로 이루어진다. 출력층은 소프트맥스로 구현한다.

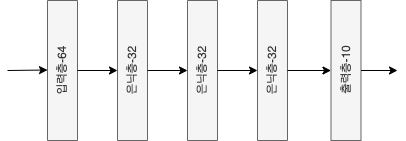


그림 5: 완전연결 심층신경망

* 1. **합성곱신경망**

합성곱 신경망은 입력층 64개의 레이어 1개, 합성곱연산과 풀링연산 레이어 1개, 출력층 10개의 레이어 1개로 이루어진다. 합성곱 연산에서 사용되는 필터의 크기는 3X3이다. 스트라이드의 크기는 1로 가정하고, 패딩은 적용하지 않는다. 합성곱 후에는 활성화 함수로 RELU를 이용한다. 합성곱 연산과 풀링 연산이 끝난 후에는 레이어를 평탄화 시키고 그 뒤의 레이어와 완전연결한다. 출력층은 소프트맥스를 이용한다.

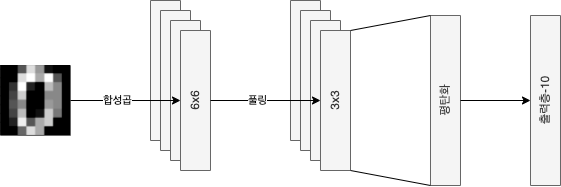


그림 6: 합성곱신경망

1. **결과**

분류 속도, 학습 속도, 정확도를 측정했다. 각 측정치는 나노초 단위로 측정했다.

* 1. **분류 속도**

싸이킷런에서 제공하는 MNIST 1797개 데이터를 7대3의 비율로 나누어 학습한 모델들을 이용했다. 세로축은 백만 나노초 단위로 자연로그 10이 씌어진 형태다. 가로축은 10개부터 100개까지 데이터를 분류했다는 것을 표시했다. 분류 속도에서 측정한 값은 로지스틱 회귀 알고리즘은 분류의 총 개수가 늘어나도 별다른 방향성을 보이지 않았다. 이는 K-최근접 이웃 알고리즘, 선형판별분석, 심층신경망, 그리고, 합성곱신경망에서도 비슷한 경향을 보였다. 다만, 서포트벡터머신의 분류 속도의 경우 분류 개수가 늘어날수록 제곱의 형태로 커졌다.

차트 1: 기계학습 알고리즘의 분류 속도

* 1. **학습 속도**

싸이킷런에서 제공하는 MNIST 1797개 데이터를 7대3의 비율로 나누어 각 알고리즘의 모델 학습 시간을 측정했다. 세로축은 천 나노초 단위로 자연로그 10이 씌어진 형태다. 가장 빠른 학습속도는 1016000ns로 KNN-3이다. 반대로, 가장 느린 학습속도는 21899768000ns로 합성곱신경망이다. 빠른 순서대로 K-최근접 이웃 알고리즘, 선형판별분석, 서포트벡터머신, 로지스틱 회귀 알고리즘, 심층신경망, 합성곱신경망이 도출됐다.

차트 2: 기계학습 알고리즘의 학습 속도

* 1. **정확도**

싸이킷런에서 제공하는 MNIST 1797개 데이터를 7대3의 비율로 나누어 각 알고리즘의 모델을 학습시켰다. 만들어진 모델의 정확도는 표 1과 같이 도출됐다. 가장 높은 정확도는 K-최근접 이웃 알고리즘이다. 가장 낮은 정확도는 로지스틱 회귀 알고리즘이다.

|  |  |
| --- | --- |
| **알고리즘 종류** | **정확도** |
| 로지스틱 회귀 | 0.9574 |
| 서포트 벡터 머신 | 0.9700 |
| 선형판별분석 | 0.9600 |
| K-최근접 이웃 알고리즘-3 | 0.9907 |
| K-최근접 이웃 알고리즘-5 | 0.9907 |
| 심층신경망 | 0.9629 |
| 합성곱신경망 | 0.9722 |

표 1: 기계학습 알고리즘의 정확도

1. **결론**

본 실험의 기계학습 알고리즘들은 모델 설계를 자유롭게 할 수 있고, 그 결과로 계산에 필요한 메모리 사용량, 처리량이 매우 다양해질수 있다. 그러므로, 각 알고리즘간의 분명한 성능 차이에 의의를 두지는 않는다. 그러나, 본 실험 결과로 각 알고리즘의 특징에 대해 유추할 수 있다. 분류 속도에서 서포트벡터머신은 분류 개수가 늘어날수록 필요한 클록수가 급격히 늘어난다. 이를 통해 다른 알고리즘에 비해 분류에 필요한 시간복잡도가 더 높다는 결과를 도출할 수 있다. 그에 비해 로지스틱 회귀, 심층신경망, 합성곱신경망, K-최근접 이웃 알고리즘, 선형판별분석은 서포트벡터머신보다 더 좋은 시간복잡도를 가지고 있는 것을 유추할 수 있다. 그러므로, 서포트벡터머신을 제외한 알고리즘들은 비슷한 시간복잡도를 가지는 것을 의미한다.

학습 속도에서 심층신경망, 합성곱신경망 제외하고 생각해보면 학습에 가장 많은 시간이 드는 알고리즘은 로지스틱 알고리즘이다. 로지스틱 알고리즘은 계산이 다른 알고리즘에 비해 비교적 복잡하므로 많은 학습시간을 필요로 한것으로 결론냈다. 반대로 KNN이 가장 적은 학습시간을 가졌다. 이는 1797개라는 비교적 적은 수의 데이터를 이용했기 때문에 전체적인 데이터를 다 확인해야하는 KNN의 단점이 숨겨진 효과를 가져온 것으로 파악했다. 전체 알고리즘을 봤을때 심층신경망과 합성곱 신경망이 많은 학습시간이 필요했다. 이는, 본 실험에서 복잡한 모델을 설정했기에 나온 결과로 이해했고, 이에 대해서는 큰 의의를 두지 않는다.

정확도에서 K-최근접 이웃 알고리즘이 가장 높은 정확도를 가졌다. 이는 같은 조건일 경우 K-최근접 이웃 알고리즘이 가장 좋은 성능을 가질 수 있다는 것을 의미한다. 이론상으로 K-최근접 알고리즘의 K에 해당하는 개수가 무한히 늘어나면 베이즈 에러에 도달할 수 있기 때문에 좀 더 좋은 성능을 냈다고 파악했다. 그러나, K-최근접 이웃 알고리즘을 제외한 다른 알고리즘들은 평균 0.9655의 정확도를 보였으며 정확도의 수치상으로 큰 차이점을 보이진 않았다. 이는 8 x 8 이미지를 분류할 경우 본 실험에서 사용한 알고리즘으로도 충분히 좋은 정확도를 가질 수 있다는 것을 의미한다. 8 x 8과 같이 복잡하지 않은 이미지의 경우 정확도를 알고리즘 선택의 근거로 판단하기에는 무의미하다고 결론냈다.