기계학습 모델을 이용한 MNIST 데이터셋 분석

한양대학교 공과대학 컴퓨터소프트웨어학부 최가온

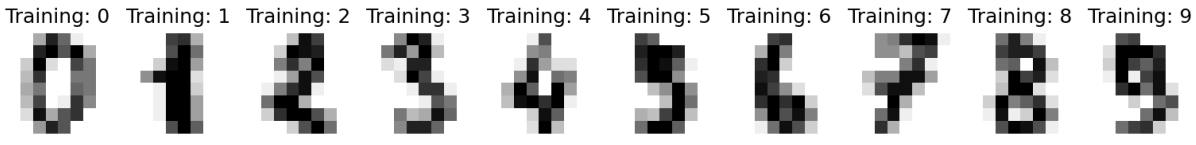
[LINK] https://github.com/Gaon-Choi/CSE4007/tree/main/research_mnist

Abstract

기계학습 알고리즘에는 Linear Regression, Logistic Regression, SVM, K-Nearest Neighbor, Fisher Discriminant Analysis(FDA), Multi-Layer Perceptron(MLP) 등 다양한 모델이 존재한다. Scikit-learn 라이브러리에서는 이러한 모델들을 손쉽게 사용할 수 있는 API를 제공하고 있다. 인공지능과 관련하여 풀 수 있는 문제는 크게 회귀(Regression)와 분류(Classification) 두 가지로 나눠서 생각해볼 수 있다. 이번 프로젝트에서는 손글씨 숫자 데이터에 해당하는 MNIST 데이터셋을 이용한 분류 문제를 풀어보고자 한다. 그리고, 각각의 모델별로 같은 문제를 반복하여 적용해보고 각각의 특성을 파악하는 것이 이번 연구의 목적이다.

Problem Definition

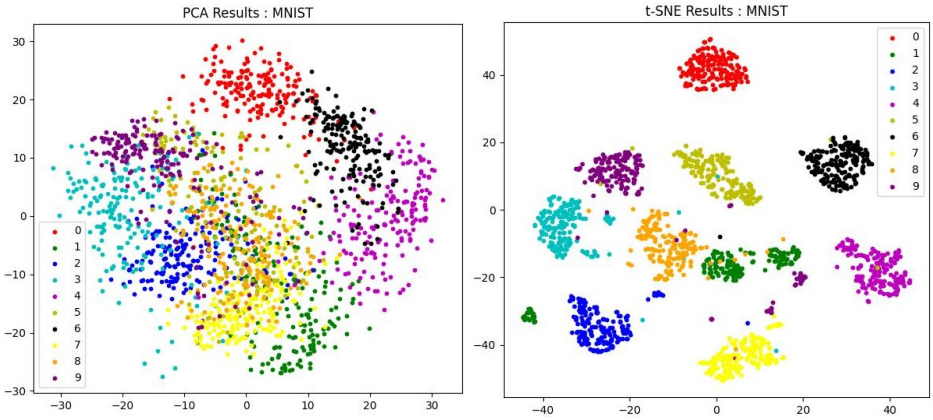
MNIST란 미국의 고등학생과 인구조사국 직원들이 손으로 직접 쓴 약 7만여개의 작은 숫자 이미지 데이터셋이다. sklearn.load_images( )을 통해 MNIST 데이터셋을 받으면 일부의 데이터를 내려받아 사용할 수 있다. 총 1797개의 숫자 데이터를 포함하고 있다. 기본적으로 사전(dict) 형태로 구성되어 있는데, 이미지 픽셀에 해당하는 mnist['images']에는 8 x 8 형태로 데이터가 있고, 각각의 값들은 그레이 스케일 기반으로 0~255의 값 중 하나를 가지고 있다. 데이터셋 학습의 과정에서 8x8 형태의 배열을 그대로 다루기보다는 64x1 형태의 벡터로 바꾸어 진행하였다. 아래의 그림은 MNIST 데이터셋의 일부를 시각적으로 표시한 것이다.



따라서, 이 학습에서의 입력은 64차원 벡터이고, 출력은 0부터 9까지의 범위에 해당하는 정수(스칼라) 하나이다. 즉, $R^{64} \rightarrow R$로 표시할 수 있다.

대부분의 문제에서도 그렇지만, 고차원 데이터 샘플이 있다고 할 때 그것이 모든 차원에 걸쳐 균일하게 펴져 있는 경우는 매우 이상적이다. 따라서, 전체 특성이 아닌 일부의

특성이 서로 강하게 연관되어 있는 경우가 많기 때문에 고차원의 데이터를 저차원으로 투영(projection)시키는 것은 데이터셋의 특성을 파악하는 한 가지 방법이 될 수 있다. 아래의 그림은 각각 PCA와 t-SNE를 이용하여 MNIST 데이터셋을 2차원 상으로 투영한 결과이다. 상대적으로 t-SNE를 적용할 때의 결과에서 데이터셋의 클러스터링 특성이 뚜렷하게 나타남을 확인할 수 있다.



Computation

가. 다양한 모델 분석하기

첫번째 실험에서는 다양한 기계학습 알고리즘을 이용하여 MNIST 데이터셋을 학습시킨 후 정확도, 소요 시간이라는 두 가지 실험값을 중점적으로 기록하였다. 이 실험에서 사용한 모델의 목록은 아래와 같다.

i. Logistic Regression
ii. Linear Discriminant Analysis
iii. Linear SVC
iv. SVC
v. K-Nearest Neighbor
vi. MLP

이때, MLP에서는 활성화 함수로 'relu'를 사용하였으며, K-nearest neighbor 알고리즘에서는 n=5 기준으로 실험을 진행하였다.

나. KNN 알고리즘에서 N값에 따른 정확도 분석하기

두 번째 실험에서는 KNN 알고리즘에서 N의 값이 올라갈 때 정확도가 어떤 변화 양상을 보일 것인지를 측정하였다. 이번 실험에서는 N의 범위를 최대치 200으로 설정하고 진행하였다.

다. Decision Boundary 분석을 통한 모델의 특성 파악

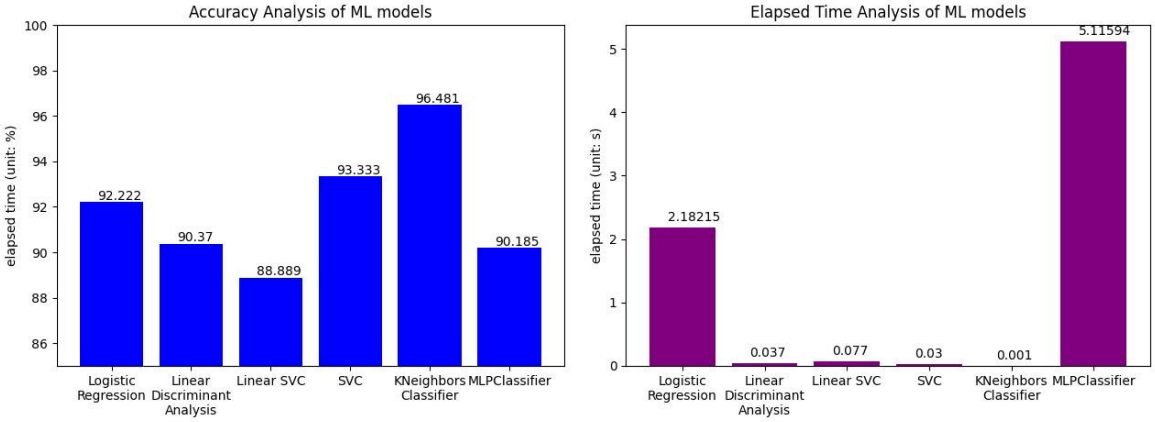
이전에 서술하였듯 데이터의 분포를 시각적으로 그려보는 것은 그것의 총체적인 특성을 파악하는데 유용한 방법이다. 이 실험에서는 t-SNE로 변환한 2차원 상의 데이터셋을 기반으로 첫 번째 실험에서 사용한 모델들로 학습의 과정을 다시 거친 후 그 그래프 위에 모델의 Decision Boundary를 표시하여 각각의 기계학습 모델의 특성을 파악하고자 한다.

Results and Discussions

가. 다양한 모델 분석하기

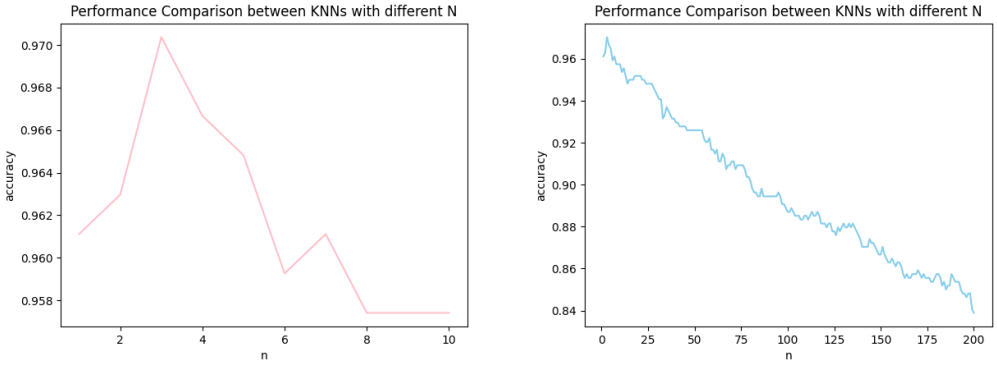
아래의 그림은 실험에서 사용한 기계학습 모델별 정확도(accuracy)와 학습에 소요된 시간을 나타낸 그래프이다. 다수의 실험을 진행한 결과, 정확도 측면에서는 KNN > SVC > Logistic Regression > FDA >= MLP > Linear SVC로 성능의 순위가 나타났다. 또한, 정확도 측면에서의 성능 평가는 confusion matrix(혼동행렬)을 통해서도 시각적으로 확인할 수 있다.

소요 시간의 측면에서는 Logistic Regression과 MLPClassifier에서 상대적으로 많은 시간이 소요됨을 관찰할 수 있었다. 먼저 Logistic Regression의 경우 직접적으로 non-linear한 문제를 풀 수 없기에, non-linear feature에 대해 변환 과정이 필요하다. 이러한 과정에서 소요시간이 늘어났을 것이라 추정된다. 또한 MLPClassifier에서는 한 epoch에서 모든 노드의 weight와 bias가 업데이트되며, 다층 구조 내에서 propagation이 이뤄진다. 이러한 이유로 hidden layer가 복잡한 구조를 가질수록 학습시간이 늘어나게 된다. 반면, KNeighbors Classifier의 경우 learning의 과정이 필요가 없다. 이러한 특성 때문에 소요시간이 거의 0에 근접하는 수준으로 나타났다. k의 값이 커질수록 계산량이 커지지만 일반적인 k의 범위에 대해서는 그래프 상으로 무시할 수 있을 수준으로 관찰되었다.



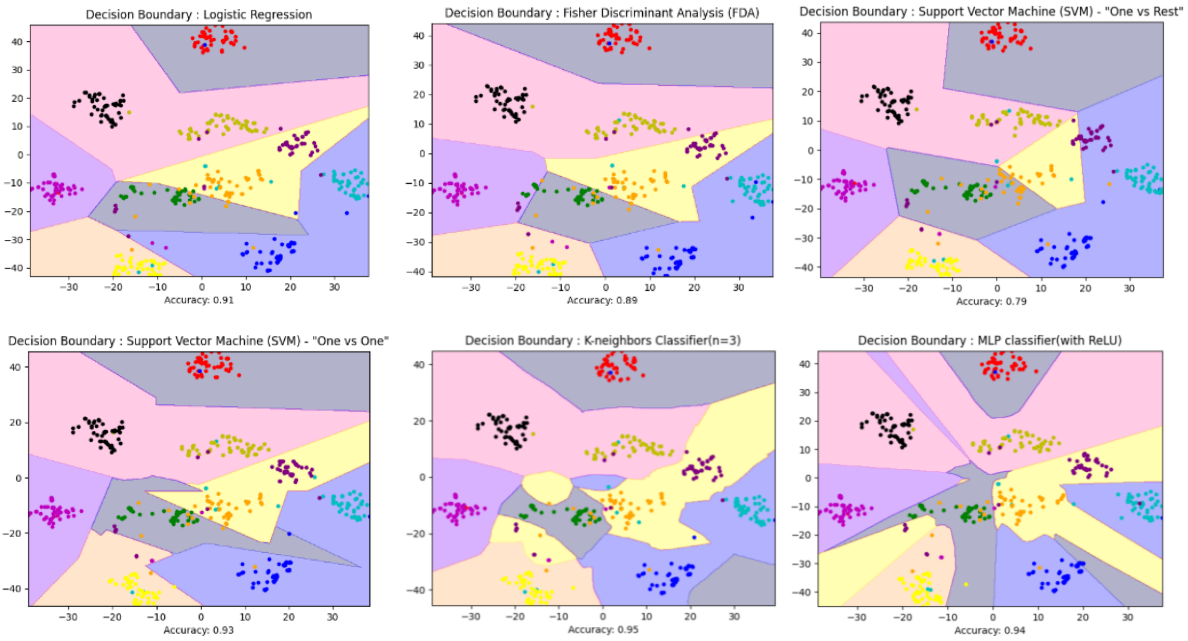
나. KNN 알고리즘에서 N값에 따른 정확도 분석하기

N의 값을 변화시킴에 따라 정확도가 어떠한 양상을 보이는지를 나타낸 그래프 결과는 아래와 같다. N값이 작을 때에는 데이터의 지역적(local) 분포에 영향을 많이 받게 되고 (노이즈에 강건하지 않음), 반대로 N값이 크면 데이터의 군집 분포를 불필요할 정도로 넓게 느슨한 학습이 이루어져 underfitting이 일어나 모델의 성능이 악화된다. 이 실험에서 풀고자 하는 문제에 대해서는 N=3일 때의 KNN이 최적의 모델임을 알 수 있다. 일반적으로 feature의 개수의 제곱근으로 설정하거나 홀수 k를 지정하는 것이 흔하나, 실루엣 점수 등의 개념을 도입하여 객관적인 최적의 k값을 구할수도 있다.



다. Decision Boundary 분석을 통한 모델의 특성 파악

아래의 그림은 이번 실험에서 사용한 6가지 기계학습 모델에 대해 각각 MNIST 데이터셋을 학습시킨 후, validation data와 decision boundary를 시각적으로 나타낸 것이다.



MNIST 데이터셋을 학습시킴에 있어서, train set과 test set의 비율은 7:3의 비율이 되도록 데이터를 split하였다. 이때 test set은 통제변인에 해당하기 때문에 shuffle은 비활성화하여 매번 test set의 분포는 같도록 실험을 구성하였다. test set의 2차원 임베딩 결과를 플롯팅하고 그 위에 각각의 모델의 decision boundary를 표시하였다. Decision Boundary의 시각적 표현은 해당 모델의 특성을 그대로 반영한다고 볼 수 있다. 일례로 위의 그래프를 보면, decision boundary가 일직선의 형태로 주로 나타나는 모델과 곡선의 형태로 나타나는 모델로 뚜렷하게 구분됨을 알 수 있다. 앞의 4개의 모델은 Linear Model이라는 것을 확인할 수 있다.

k = 3인 경우 KNN 모델에서 나타난 decision boundary를 보면 노랑색 영역에 하나의 주황색 포인트가 찍혀있는 것을 확인할 수 있는데, 해당 점은 전체적인 데이터셋의 분포를 고려하였을 때 노이즈에 해당한다고 볼 수 있다. 이로부터 k의 값이 작을 경우 각각의 데이터의 local distribution의 영향을 많이 받게 된다는 것을 추론할 수 있다.

또한 첫번째 실험에서 Linear SVC의 accuracy는 가장 낮게 측정되었는데, Decision Boundary를 분석해보면 주황색으로 표시된 데이터들이 회색 영역과 노랑색 영역에 나누어 표시된 것을 확인할 수 있다. 즉, fitting된 결과가 test set의 분포를 제대로 예측하지 못한다는 점을 시사한다.

Conclusions

이번 연구에서는 다양한 방식을 사용하여 MNIST 데이터셋을 기반으로 여러 기계학습 모델의 성능을 측정하였다. 정확도 측면에서는 KNN에서 가장 높은 정확도를 보였으며 동시에 이 모델은 learning의 과정이 수반되지 않기에 소요시간이 0에 가까웠다. 반면, MLP와 Logistic Regression은 소요시간이 길게 측정되었다.

KNN의 k값이 작을 때에는 local distribution의 영향을 많이 받아 불리했으나, k값이 일정 부분 커지면 underfitting 문제가 발생하여 성능이 악화된다는 것이 관찰되었다. 주로 최적의 k값을 정하는 몇가지의 규칙이 있음을 서술하였다. 마지막으로, 각 모델의 Decision Boundary를 시각적으로 분석하며 linear model과 non-linear model의 차이점을 명확하게 인지할 수 있었다. 추가적으로, 어떠한 데이터에서 제대로 예측이 되지 않는지 등의 성능적인 부분들도 세밀하게 검토할 수 있다.

References

[1] Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow

[2] https://gaussian37.github.io/ml-concept-t_sne/

[3] https://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html#sphx-glr-auto-examples-classification-plot-digitsclassification-py

[4] Lecture Note, CSE4007(HYU, Artificial Intelligence)