

딥러닝 신경망 이용한 신용카드 부도율 예측의 효용성 분석

윤 중 문 *

여신금융협회 여신금융연구소

〈Abstract〉

본 연구에서는 딥러닝 신경망(deep learning neural network)을 이용해 신용카드 부도율 예측의 정확도를 얼마나 향상시킬 수 있는지를 점검한다. 이를 위해서 기존에 많이 사용되는 머신러닝 알고리즘(Logistic, SVM, RandomForest, Lasso 등)을 딥러닝 신경망 분석의 성능테스트를 위한 비교 지표로 활용한다. 우선, 본고의 딥러닝 신경망 분석은 기본적으로 2개의 은닉층(hidden layers)과 5개의 뉴런(neuron)로 구축하고 활성화함수와 초기값 설정방법에 따른 예측의 정확도의 차이를 도출하고 기존 머신러닝 알고리즘과 비교하였다. 그 결과 딥러닝 신경망 분석이 기존 머신러닝 알고리즘 보다 높은 성능을 나타냈고 그 중 ReLU(활성함수)와 Xavier(초기값)가 가장 높은 예측의 정확도를 보였다. ReLU와 Xavier를 기준으로 은닉층과 뉴런의 수를 15개에서 최대 25개까지 늘려 분석한 결과에서도 유사한 결과를 나타냈다. 다만, 기존 연구에서와 같이 은닉층과 뉴런의 수의 증가에 따른 성능의 향상은 나타나지 않았다. 또한, 이미지 분야에서 높은 성능을 보였던 Dropout과 CNN(convolution neural network) 모델도 뚜렷한 성능의 향상을 보이지 못했다. 이는 신용카드 부도율 데이터의 설명변수와 관측치에 한계가 있기 때문으로 판단된다. 본 연구에서 사용된 개인의 신용카드 부도율 데이터는 횡단면 자료로 패널 및 시계열 데이터에서 높은 성능을 나타내는 딥러닝 신경망 알고리즘인 RNN(recurrent neural network)을 사용하지 못했다. 따라서 추후 신용평가사 및 금융기관의 빅데이터를 활용해 딥러닝 신경망 분석을 실시한다면 보다 향상된 결과값을 도출할 수 있을 것으로 전망한다.

핵심 단어: 딥러닝, 신경망, 머신러닝, 신용카드, 부도율, 금융시장

* 여신금융협회 여신금융연구소 연구위원, 서울시 중구 다동길 43 한외빌딩 12층, Tel: 02)2011-0719, E-mail: idisyun@gmail.com

I. 서론

최근 국내에서는 제4차 산업혁명에 대응한 AI 산업육성을 적극적으로 추진하고 있다. 이는 2016년 알파고와 이세돌의 바둑 대결 이후 AI에 대한 관심이 폭발적으로 늘어난 이후 정부차원에서의 육성전략이 시작됐다고 할 수 있다. 2016년 알파고가 사용한 방법론은 딥러닝 신경망과 트리 검색(deep neural networks and tree search)¹⁾이었다. 여기에서 사용된 딥러닝 신경망 분석은 인간의 신경망을 모방하여 통계학적으로 구현한 것이다. 최근 이미지 식별, 음성인식, 다양한 추천시스템에 이러한 딥러닝 신경망 분석이 도입되었고 이미 인간의 인식 능력을 뛰어넘는 놀라운 성과를 보이고 있다. 특히 이미지 분야에서는 이미지 식별 오류율이 2010년도 26.2%에서 2015년 3%대로 낮아져 인간의 식별 능력을 넘어서게 되었다. 이는 2000년대 이후 컴퓨팅 파워의 증가와 다양한 최적화 알고리즘²⁾의 개발로 인해 오랜 기간 침체기를 겪었던 인공지능망 분석이 중요한 알고리즘으로 떠오르는 계기였다.

하지만 이러한 딥러닝 신경망 분석은 컴퓨터 공학 분야에서 활발히 연구되고 있어서 사회과학 분야, 특히 금융시장에 대한 연구가 부족한 실정이다. 따라서 본 연구는 신용카드 연체 관련 데이터를 이용해서 딥러닝 신경망의 효용성과 이를 통해 신용정보 판단에 있어서 딥러닝 신경망 분석의 확장가능성에 대해서 논의하고자 한다. 현재 국내외 논문에서 금융시장의 데이터를 활용한 딥러닝 신경망 분석은 그리 많지 않은 상황이다. 이는 딥러닝 신경망의 연구가 주로 이미지, 음성인식, 번역 등의 분야에서 활발히 이루어지고 있지만 아직 금융시장의 데이터까지 적용되기는 알고리즘의 숙련도와 적용의 접합성이 낮기 때문으로 풀이된다. 일부 국내 금융회사에서 최근 머신러닝 분야에 관심을 갖고 채용을 늘리고 있지만 아직까지는 딥러닝 신경망과 관련해서 가능성 정도에서 검증하는 단계로 판단된다.

따라서 본 연구에서는 딥러닝 신경망을 이용해 신용카드 부도율을 예측의 정확도를 얼마나 향상시킬 수 있는지를 점검한다. 이를 위해서 과거부터 많이 사용되고 있는 머신러닝 알고리즘(Logistic, SVM, RandomForest, Lasso 등)을 딥러닝 신경망 분석의 성능테스트를 위한 비교 지표로 활용한다. 우선, 본고의 딥러닝 신경망 분석은 기본적으로 2개의

1) Hassabis et al. Mastering the game of Go with deep neural networks and tree search, Nature 529, 484 - 489.

2) Hinton et al.(2006)와 Bengio et al.(2007)는 파라미터의 초기값을 적절히 선택해서 최적화를 빠르게 할 수 있는 방법을 개발하였고 과거 쇠퇴한 신경망 방법론을 Deep이라는 단어를 사용하면서 활성화시키는 계기로 만들었다.

은닉층(hidden layers)과 5개의 뉴런(neuron)로 구축하고 활성화 함수와 초기값 설정 방법에 따른 예측의 정확도를 도출한다. 그 결과 딥러닝 신경망 분석이 기존 머신러닝 알고리즘보다 높은 성능을 나타냈고 그 중 ReLU(활성화 함수)와 Xavier(초기값)가 가장 높은 예측의 정확도를 보였다. ReLU와 Xavier를 기준으로 은닉층과 뉴런의 수를 최대 10개와 25개까지 늘려 분석한 결과에서도 유사한 결과를 나타냈다. 다만, 기존 연구에서와 같이 은닉층과 뉴런의 수의 증가에 따른 성능의 향상은 나타나지 않았다. 또한, 이미지 분야에서 높은 성능을 보였던 Dropout과 CNN(convolution neural network) 모델도 뚜렷한 성능의 향상을 보이지 못했다. 이는 신용카드 부도율 데이터의 설명변수 및 관측치에 한계가 있기 때문으로 풀이된다. 본 연구에서 사용된 개인의 신용카드 부도율 데이터는 횡단면 자료로 패널 및 시계열 데이터에서 높은 성능을 나타내는 딥러닝 신경망 알고리즘인 RNN(recurrent neural network)을 사용하지 못하는 것이 아쉽다. 따라서 추후 신용평가사 및 금융기관의 빅데이터를 활용해 딥러닝 신경망 분석을 실시한다면 과거 보도 향상된 결과를 도출할 수 있을 것으로 기대된다.

본 연구의 구성은 다음과 같다. 먼저 II장에서는 선행연구, III장에서는 머신러닝과 딥러닝 신경망 분석방법에 대해서 알아 볼 것이다. 그리고 IV장에서는 딥러닝 신경망 모형을 이용한 신용카드 연체율 분석과 해당 결과에 대해서 검토할 것이다. 마지막으로 V장에서는 결론과 시사점을 도출할 것이다.

II. 선행연구

딥러닝 신경망 분석은 앞에서 언급한 것처럼 이미지, 소리, 언어분석 등 과거 인간의 영역 이외에서 분야에서 활발하게 이루어지고 있다. 특히 이미지 분야에서는 이미지 식별 오류율이 인간의 능력을 넘어서게 되면서 해당 분야에 있어서 다양하게 적용되고 있다. 하지만 사회과학 분야 특히 금융시장에서의 신경망 분석은 아직 초보적 단계에 머무르고 있다. 이는 금융 데이터가 가지고 있는 속성 자체와 분석결과에 대한 해석의 어려움 등에 기인하는 것으로 판단된다. 또한, 딥러닝 신경망 분석이 주로 컴퓨터 공학을 중심으로 연구가 진행되고 있어서 금융시장 데이터를 이용한 연구가 관심 분야인 이미지 등에 비해 소외되고 있고 금융전문가들의 딥러닝 신경망 분석에 대한 이해도가 기존 계량방법론에 비해 낮다고 볼 수 있다.

우선, 딥러닝 신경망 분석은 인공신경망(Neural Network)로부터 시작되었는데, 이는 인간의 신경망 모델을 알고리즘을 본 딴 1958년 Rosenblatt가 제시한 퍼셉트론(perceptron)이 시작이라고 할 수 있다(Rosenblatt(1958)). 하지만 이는 현재의 logistic regression과 유사한 형태로 간단한 XOR 문제³⁾도 학습하지 못하는 등의 문제점을 나타냈다. 이로 인해

서 한동안 인공신경망 분석이 활성화되지 못하고 표류하게 되었다. 하지만 다중 퍼셉트론(multilayer perceptron)이 출현하면서 XOR 문제를 간단히 해결할 수 있었고 이는 은닉층(hidden layer)을 늘리면 분석능력이 향상되는 결과를 가져왔다. 딥러닝 신경망의 의미는 이러한 은닉층을 늘려 분석능력을 향상시킨다는 의미로 해석할 수 있다. 하지만 은닉층이 늘어나면서 추정해야 되는 계수(parameter)가 기하급수적으로 늘어나 학습이 어려운 문제 있었다. 이를 Rumelhart et al.(1985)이 역전파 알고리즘(backpropagation Algorithm)을 개발하여 다중 퍼셉트론의 학습을 가능하도록 하였다. 하지만 역전파 알고리즘도 은닉층이 많아지면서 학습 능력이 떨어진다는 것(vanishing gradient problem)과 과적합(overfitting) 및 극소(local minimum) 문제점들이 노출하였다. 특히, 은닉층이 많아지면서 학습능력이 떨어지는 것은 그동안 이용하던 활성화함수(activation function)인 sigmoid(logistic) 함수와 같은 경우 학습이 진행될수록 기울기가 0에 가까워져 경사하강법(Gradient descent)이 거의 일어나지 않아 학습이 되지 않는 단점 때문이다(Bengio et al.(1994)). 위와 같은 단점으로 인공신경망 분석이 잘 활용되지 못하였는데, 2006년 볼츠만 머신(Boltzman Machine)을 이용한 학습방법이 주목받으면서 활성화되는 계기가 되었다. 볼츠만 머신은 비지도학습(unsupervised learning)으로 데이터를 충분히 학습시키고 이후 역전파 알고리즘과 같은 지도학습(supervised learning)을 실행해 기존의 다중 퍼셉트론의 단점인 vanishing gradient와 과적합 문제를 해결하고 극소 문제도 일정 부분 해결할 수 있는 가능성을 보였다(Smolensky(1986); Hinton and Salakhutdinov(2006); Bengio(2009)). DBN(Deep Belief Network)는 이러한 볼츠만 머신을 이용한 다중 퍼셉트론 학습 방법을 얘기한다. 이후 볼츠만 머신과 같은 비지도학습 방법이 아닌 지도학습 방법론을 이용한 다중퍼셉트론의 문제를 개선하려는 노력들이 나오기 시작하였다. 이중 대표적인 아이디어로는 Rectified linear unit(ReLU)와 Dropout이 있다. vanishing gradient의 문제는 weight(계수)의 기울기가 0에 급속도로 가까워진다는 것이다. 이를 극복하기 위해서 sigmoid 함수가 1 이상의 값을 가지지 못하는 문제를 바꿔 입력값이 0 이상인 경우 기울기를 1로 하는 ReLU함수를 이용하였다. 이를 통해서 기존의 방법 보다 학습성능이 높고 볼츠만 머신과 같이 pre-training(fine tuning) 없이 학습 가능하도록 만들었다(Nair and Hinton 2010; Glorot, Bordes, and Bengio 2011).

다중 퍼셉트론 학습의 문제점 중 하나는 관측치(Labeled data)의 부족으로 인해 과적합(overfitting)이 발생하는 것이다. 이에 Hinton(2012)은 Dropout 방법론을 제시해 앙상블(ensemble) 모형과 유사한 분석방법론을 도입하였다. 이는 학습과정에서 은닉층의 값을 모두 사용하지 않고 일정비율(50%~70%)만을 이용해 학습하고 실제 테스트 데이터를 통

-
- 3) 간단하게 보면 XOR 논리 회로를 만드는 것으로 두 개의 더미변수가 있을 때 배타적 논리합으로 두 변수 중 하나만 1일 때 1이 된다는 논리로 선형으로 이를 추정할 수 없는 문제를 의미한다.

해 분석할 때는 모든 학습된 가중치(weight)를 사용해 분석의 과적합 문제를 해결하고 분석의 정확도를 높일 수 있다. 또한, 이미지 인식과 자연어 처리 등의 분야에서 혁신적인 성과를 얻고 있는 Convolutional Neural Network(CNN)이 있다. CNN은 기존의 신경망 분석에 Convolution 작업을 추가한 것으로, 설명변수의 데이터를 일정한 필터를 통해서 새롭게 의미 있는 값으로 뽑고 또한 통합 계층(pooling layer) 과정을 통해 전체 학습 능력을 높이는 방법론이다. CNN는 처음 이미지 인식 분야에서 LeCun et al.(1998)에 의해서 시도된 LetNet5가 있고 이후 Krizhevsky et al.(2012)에 사용한 AlexNet이 있다. 그리고 Szegedy et al.(2014)은 GoogLeNet을 개발해 ILSVRC 2014에서 이미지 식별분야에서 우승하였고 He et al.(2015)은 ResNet을 개발해 ILSVRC 2015에서 우승하였고 이미지 식별 오차를 3.6%까지 줄여 인간의 인식오류(5%)를 능가하는 결과를 보여주었다. 이외에도 시계열 데이터를 이용한 RNN(Recurrent Neural Network), GAN(Generative Adversarial Network) 등 새로운 방법론이 출현하면서 딥러닝 신경망 분석이 지속적으로 발전하고 있다.

그렇다면 금융분야에서는 이러한 딥러닝 신경망 분석이 어떻게 연구되고 있는지를 알아보자. 우선, 신용평점 관련해서 딥러닝 신경망 분석을 이용한 논문으로는 Kavamme et al.(2016), Tran(2016), Luo(2017) 등이 존재한다. Li(2006)은 초기 신경망 모델인 MLP(multilayer pereptron)과 SVM 알고리즘을 이용하여 소비자신용의 연체여부에 대한 연구를 실시하였다. 분석결과로는 SVM이 신경망 모델에 비해 예측력이 더 좋은 것으로 나타났다. Luo(2017)은 CDS 데이터를 이용해 신용평가 모델의 성능을 검증하였다. 성능 평가에 사용된 알고리즘으로는 볼츠만 머신을 이용한 DBN(deep belief networks)와 같은 딥러닝 신경망과 로지스틱 회귀분석, MLP(multilayer pereptron), SVM(support vector machine) 등을 이용하였다. 결과를 보면, DBN이 정확도와 AUC 값이 각각 100%와 1를 가장 높은 성능을 보였다. Kvamme et al.(2016)는 시계열 데이터를 이용하여 신용평점에 딥러닝 신경망 분석을 적용하였다. 이 논문은 최신 딥러닝 신경망 분석 방법을 가장 많이 적용한 논문으로 보인다. 모기지론, 자동차 담보대출, 신용카드 사용 내역 등을 이용하여 1년 내 부도가 날 확률을 계산하였다. 분석방법론으로는 MLP, CNN, RNN 등을 이용하였고 현재 위험관리모델인 DNB의 AUC(0.866)에 비해 CNN의 AUC가 0.9186으로 높게 나타나 향상된 성능을 보였다. 하지만 RNN의 결과는 제시되지 않았다. Sirignano et al.(2016)는 모기지론의 연체율에 대해서 딥러닝을 도입하였고 기존의 로지스틱 회귀분석보다 좋은 성능을 보였다.

한편, 신용카드 데이터를 이용해 딥러닝 신경망 분석을 한 논문으로 신용카드 부정거래 판별이 주를 이루고 있다. 최근 논문들을 보면, Niimi(2015)는 딥러닝 신경망 알고리즘을 이용해 신용카드의 부정거래 여부를 판별하고 이것의 예측정확도를 분석하였다. 분석데이터는 UC Irvine에서 제공하는 신용카드 관련 UCI 머신러닝 데이터셋을 이용하였

다. 데이터는 총 690개의 관측치와 16개의 변수(설명변수 15개, 부정거래 더미변수)로 구성되어 있다. 이때 부정거래 더미변수는 307개가 정상거래이고 383개는 부정사용으로 나누어진다. 총 690개의 관측치 중 590개는 분석모형의 파라미터를 추정하는 훈련(training) 데이터로 사용하고 나머지는 해당 분석모형이 실제 어느 정도 부정사용을 탐색할 수 있는지를 검증하는 데이터로 사용하였다. Niimi(2015)는 딥러닝 신경망 모델의 신용카드 부정거래탐지시스템의 정확도를 판별하기 위해서 많이 사용되는 머신러닝 알고리즘과 비교분석하였다. 비교모형으로는 Logistic Regression⁴⁾과 SVM(support Vector Machine)⁵⁾, Random Forest⁶⁾, XGBoost⁷⁾을 이용하였다. 딥러닝 신경망 모델은 정보의 산출방식인 활성화(activation) 방법론에 따라서 결과가 상당히 달라지기 때문에 ReLU, Tanh, Maxout으로 구분해서 분석하였고 이와 함께 Dropout⁸⁾ 방법론을 추가 사용하였다. 추정방법론별 오차율을 보면 SVM 모델(Gaussian Kernel 이용)이 11.0%로 가장 높은 예측율을 나타낸다. 딥러닝 신경망 분석은 활성화 방법(ReLU, Tahn, Maxout 등)과 은닉층(hidden layer) 개수에 의해서 예측정확도가 상당히 달라질 수 있는데, 분석 결과를 보면 활성화 방법으로 Maxout과 Dropout를 추가한 방식이 12.4%의 오차율로 딥러닝 신경망 방법론 중 가장 높은 정확도를 보였다.⁹⁾ 하지만 딥러닝 신경망 분석이 SVM 모델 보다 낮은 예측력을 보여줘서 기존 방법론에 비해 추가적인 효용성을 보여주고 있지는 못하다. Ramanathan(2014)은 PayPal의 데이터를 이용하여 부정거래를 탐지하는 분석모델을 테스트하였다. Niimi(2015)가 딥러닝 신경망을 포함한 다양한 분석모델을 이용해 정확도를 검증하였다고 하면 Ranmanathan(2014)는 딥러닝 신경망 방법론만 집중해서 분석하였다.

-
- 4) 부정거래여부(0/1)와 같은 종속변수와 다수의 설명변수로 이루어진 경우 실행하는 통계방법론인데, logistic 함수를 이용해서 계수를 추정하고 예측하는 특징을 가지고 있다.
 - 5) 종속변수를 설명변수의 특성으로 분류하는 방법론의 일종으로 설명변수의 특성을 초평면(hyperplane)으로 구분하고 이를 가장 크게 나눌 수 있는 방법을 찾는 방식으로 신경망 모델보다 쉽고 예측력이 좋은 것이 특징이다. 선형으로 결정경계(decision boundary)를 나누지 못할 때, 다차원으로 설명변수를 변형하는데, 이를 보다 쉽게 할 수 있는 계산방법인 커널트릭(kernel trick)을 이용할 수 있다. 이때 커널함수의 따라서 모형은 Linear, Gaussian, Sigmoid, Polynomial 등으로 구분가능하다.
 - 6) 회귀분석, 분류 등에서 사용하는 앙상블(ensemble) 학습으로 랜덤(중복허용)으로 훈련(training) 데이터셋을 구성하고 10회 정도 분석을 실시해 해당 계수 값의 평균을 이용하는 등의 방법론을 말하고 오버피팅(overfitting) 문제를 일부 해결할 수 있다.
 - 7) Gradient Boosing의 일종으로 지도학습 방법론 중 트리모델 중 하나이다.
 - 8) 훈련데이터를 통해서 은닉층의 계수를 수치최적화를 통해서 추정할 때 모든 은닉층의 계수를 추정하지 않고 랜덤하게 일부만 추정하는 방식으로 추정하는 방식으로 계수값을 찾는 방식이다. 다만, 실제 데이터를 통해서 계수를 이용할 때는 모든 계수를 이용해서 신용카드 부정사용을 예측하는 방법론이다.
 - 9) 10회 상호검증(10-fold cross-validation) 방법론과 은닉층(hidden layer)은 3개, 뉴런 수는 200개를 이용하셔서 검증하였다.

Ranmanathan(2014)는 은닉층(hidden layer)과 뉴런(neuron)의 수, 활성화 방법에 따른 결과를 비교분석하였다. Ramanathan(2014)는 Niimi(2015) 보다 훨씬 많은 1천만 개의 관측치와 1,500개의 설명변수(범주형 150개)를 이용하였다. 이때 Ramanathan(2014)는 Niimi(2015)와 달리 모델평가를 AUC(Area under the Curve)를 이용한다. Ramanathan(2014)에 따르면, 분석모델에서 은닉층(hidden layer) 수가 많아지면 부정거래탐지를 보다 정확히 수행하는 것으로 나타났다. 은닉층이 2개에서 8개로 늘면 AUC의 값이 0.762에서 0.839까지 증가하였다. 또한, 은닉층의 뉴런 수가 증가하면 추정의 소요시간이 증가하고 과적합의 우려가 존재하지만 추정의 정확도가 일정 순간까지 증가하는 것을 알 수 있다. 뉴런의 수가 300개에서 600개로 증가할 때 AUC는 0.809에서 0.859까지 증가하였다. 한편 활성화 방법에 따른 AUC의 변화를 보면 Niimi(2015)의 결과와 유사하게 Rectifier (ReLU) with Dropout이 가장 높은 정확도를 나타냈다. 마지막으로 전체 1,500개 변수 중에서 일부만을 이용해 추정한 결과는 subset3이 AUC가 0.849로 가장 높은 정확도를 나타냈는데, 이는 일부 변수만을 이용해도 정확도 손실은 크지 않을 수 있다는 것을 나타낸다. 다만, 이는 추정결과가 간단해지고 추정속도가 빨라진다는 장점이 있지만 충분한 컴퓨팅 파워가 있을 때는 모든 변수를 이용하는 것이 더 나은 방법으로 판단된다.

I. 모델

2.1 로지스틱 회귀모형

이항 로지스틱 회귀모형은 신용점수(credit scoring)와 부도확률 예측에서 가장 많이 사용되는 모형 중 하나이다. 종속변수가 연체 혹은 부도여부에 따라서 0 혹은 1로 구성되어 있는 이산형 변수에서 사용할 수 있다.

변수:

- Y 는 이산형 변수, $Y_1, Y_2, \dots, Y_n \sim iid. Bin(n_i, \pi_i)$

$Y_i = 1$ 는 신용카드 대출 연체 발생시

$Y_i = 0$ 는 신용카드 대출 연체가 발생하지 않음

- $X = (X_1, X_2, \dots, X_k)$ 는 신용카드 사용규모, 성별, 교육수준, 나이 등

추정방법:

- Maximum likelihood estimator (MLE)를 통해 추정

$$L(\beta_0, \beta_1) = \prod_{i=1}^N \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i} = \prod_{i=1}^N \frac{\exp\{y_i(\beta_0 + \beta_1 x_i)\}}{1 + \exp(\beta_0 + \beta_1 x_i)}$$

2.2 서포트벡터머신(SVM: Support Vector Machine)

Support Vector Machine (SVM)은 1992년에 소개되었다(Boser et al., 1992). 이는 범주형 종속변수 데이터에 머신러닝 기술을 이용하여 과적합(overfitting) 문제를 피하고 추정의 정확성을 높여주는 알고리즘 중 하나이다. SVM은 단순하거나 복잡한 범주형 모델을 학습할 수 있고 고도의 수학적 원리를 이용해 과적합 문제를 피할 수 있다. 또한, 커널(Kernel) 함수를 이용해서 계산량을 줄이면서 다차원 공간으로 설명변수를 확장시켜 모형의 설명도를 높일 수 있는 장점이 있다. SVM은 현재까지 나온 머신러닝 알고리즘 중 추정의 정확도가 높은 것으로 알려져 있다.

- SVM 문제

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|_2^2 \\ \text{s.t.} \quad & y_i (\langle w, x_i \rangle + b) \geq 1 \end{aligned}$$

- 라그랑지 문제

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i y_i ((\langle w, x_i \rangle + b) - 1)$$

- 커널함수

- Linear kernel: $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$

- Polynomial kernel of degree d :

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d$$

- Sigmoid kernel: $K(\vec{x}_i, \vec{x}_j) = \tanh(\kappa \vec{x}_i \cdot \vec{x}_j - \delta)$

- Gaussian RBF (Radial Basis Function) kernel:

$$K(\vec{x}_i, \vec{x}_j) = \exp \left[-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2} \right]$$

2.3 Lasso 회귀분석

머신러닝의 일종인 Lasso(least absolute shrinkage and selection operator) 회귀분석은 최신 계량방법론으로 추정의 정확도를 높이기 위해서 변수 선택과 제약(regularization)을 이용한 분석방법이다. 이는 Leo Breiman의 nonnegative garrote를 기초로 Robert Tibshirani(1996)의 의해 소개되었는데, Ridge와 Lasso는 제약의 비용(cost)에만 차이가 존재할 뿐 큰 차이는 없다. 기본 OLS 추정식에서 제약을 추가해 추정계수가 너무 많아져 과적합 되는 문제를 일부 개선하고 추정의 정확도를 높이는 효과가 존재한다.

- Lasso 문제

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 \right\} \text{ subject to } \|\beta\|_1 \leq t.$$

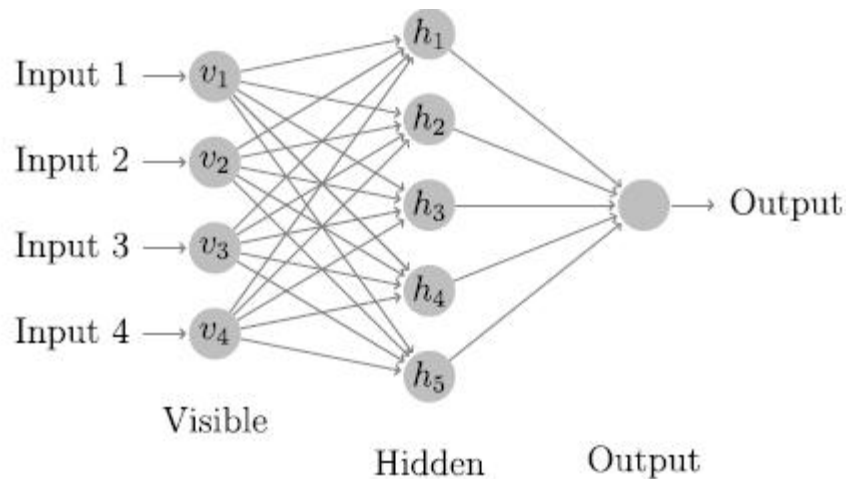
- 라그랑지 문제

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$

2.4 딥러닝 신경망 분석

2.4.1 다층신경망분석(MLP: Multilayer perceptron)

다층신경망 분석은 투입층, 은닉층, 출력층으로 구성되어 있는 단순한 feed-forward 신경망 분석이다. MLP 네트워크는 패턴 분류(pattern classification), 인식(recognition), 예측에 있어서 널리 사용되는 분석 방법론이다. 아래 그림은 기본적인 하나의 투입층과 은닉층으로 구성되어 있는 모형이다.



MLP에 의한 분석 결과는 은닉층의 개수(h)와 은닉층의 뉴런의 수(v)에 따라서 결과가 상당히 많이 바뀔 수 있다. 또한, 뉴런의 출력값을 결정하는데 사용되는 활성화함수(activation function)에 따라서 결과치가 달라질 수 있다. 활성화함수로는 Sigmoid, ReLu,

Tahn, Maxout 등이 존재한다. 이때 MLP는 Back Propagation (Paul(1874,1982), Hinton(1986)) 학습 알고리즘을 사용한다.

- 활성화함수 (Sigmoid)

$$f(x) = \frac{1}{1 + e^{-x}}$$

- 은닉층 신호 전달

$$n_j = \sum_{i=1}^n v_i w_{ij} + b_j, \quad j = 1, 2, \dots, m.$$

- 뉴런의 활성화

$$a_j = f(n_j) = f\left(\sum_{i=1}^n v_i w_{ij} + b_j\right)$$

단, v_i 는 입력층 혹은 전단계 은닉층 데이터

b_j 는 bias 값.

- 출력값

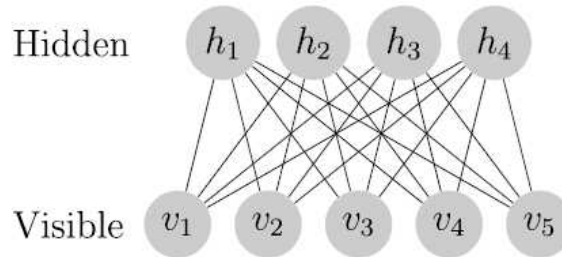
$$Y_k = \sum_{j=1}^m f\left(\sum_{i=1}^n v_i w_{ij} + b_j\right) w_{jk} + b_k$$

단, w_{ij} 와 w_{jk} 는 각각 은닉층과 출력층의 가중치를 의미

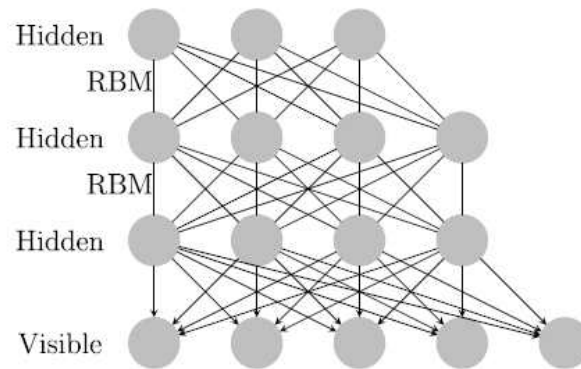
2.4.2 DBN(Deep belief Network)

Hinton et al.(2006)에 의해 제시된 Deep Belief Network는 다층신경망 분석으로 계수 추정을 위해 초기 값을 로컬수렴 방식을 이용해서 두 개의 층간의 계수를 훈련시켜서 최종 은닉층과 출력층까지 확대하기는 Restricted Boltzmann Machines (RBM) 방식 (Salakhutdinov et al. 2007)이다. 이를 통해서 초기 계수값을 설정해서 다층신경망의 모든 계수를 수렴하도록 만드는 분석방법이다. 신경망 분석은 1960~70년대 출현했지만 다수의 은닉층의 계수값을 추정하는데 어려움이 존재해 관련 방법론이 침체되었지만 초기 값 설정에 대한 연구가 급속도로 발전하면서 대중화되는 계기를 마련되었다.

Restricted Boltzmann Machine (로컬계수 추정)



Deep Belief Network (DBN)



RBM은 로컬 계수를 추정하는데, 두 개의 입력층(은닉층)과 은닉층(출력층)만을 대상으로 양방향으로 추정해 계수값이 수렴할 때까지 지속하는 방식이다. 아래의 energy 함수를 극소화하는 w 을 추정하는 방식으로 볼 수 있다. RBM 학습은 Contrastive Divergence algorithm (Hinton et al., 2006)을 이용해 효과적으로 달성할 수 있다. DBN은 다층신경망 분석으로 RBM을 각 입력층, 은닉층, 출력층으로 확장해서 추정할 수 있다.

Restricted Boltzmann Machine (energy function)

$$E(v, h) = - \sum_{i=1}^n a_i v_i - \sum_{j=1}^m b_j h_j - \sum_{i=1}^n \sum_{j=1}^m w_{ij} v_i h_j$$

단, a_i, b_j 는 설명변수(v_i)와 은닉층 노드(h_j)의 bias 임

w_{ij} 는 가중치(계수) 값을 의미

Boltzmann Distribution

$$P(v, h) = \frac{1}{Z} e^{-E(v, h)}$$

단, Z 는 distribution의 합을 1로 만들어주는 partition function 임

RBM은 초기값을 설정해주는 방식의 하나로 이후 Xavier 초기값(Glorot and Bengio, 2010)와 He's 초기값(He et al.(2015)) 설정방법 등 출현해 보다 간단하고 성능이 좋은 초기 계수 값을 설정할 수 있다. Xavier 초기값은 입력값과 출력값의 수를 입력값의 수의 제곱근으로 나눈 값을 최소와 최대값으로 해서 랜덤하게 뽑은 값을 초기값으로 주면 된다. He's 초기값 설정방법은 제곱근의 값에서 입력값으로 fan_in/2로 넣어주면 된다.

Xavier 초기값 방식

Random(fan_in, fan_out)/sqrt(fan_in)

단, fan_in은 입력변수의 수, fan_out은 출력값의 수

He's 초기값 방식

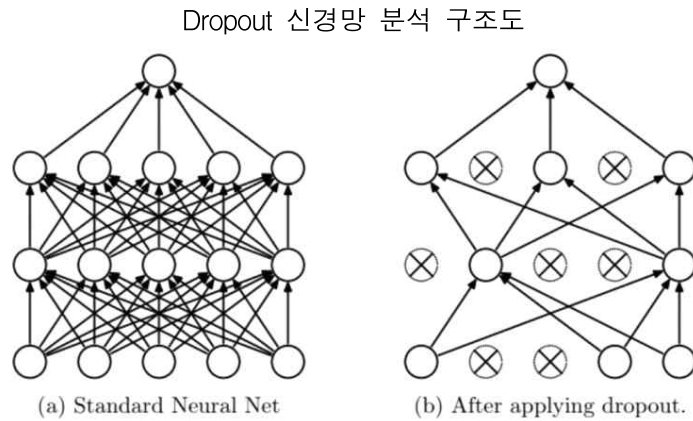
Random(fan_in, fan_out)/sqrt(fan_in/2)

단, fan_in은 입력변수의 수, fan_out은 출력값의 수

2.4.3 Dropout 신경망 분석

기존 MLP 분석에서는 각 은닉층의 뉴런(노드)을 모두 이용해서 각 가중치(weight)를 추정하였지만 Dropout에서는 각 은닉층의 뉴런을 랜덤하게 선택해서 해당 뉴런만을 학습시키는 방법론이다. 이를 반복적으로 추정하고 최종적으로 테스트 데이터를 이용할 때에는 모든 뉴런을 이용하면서 학습능력과 정확도를 높이는 방식이라고 할 수 있다. 이는

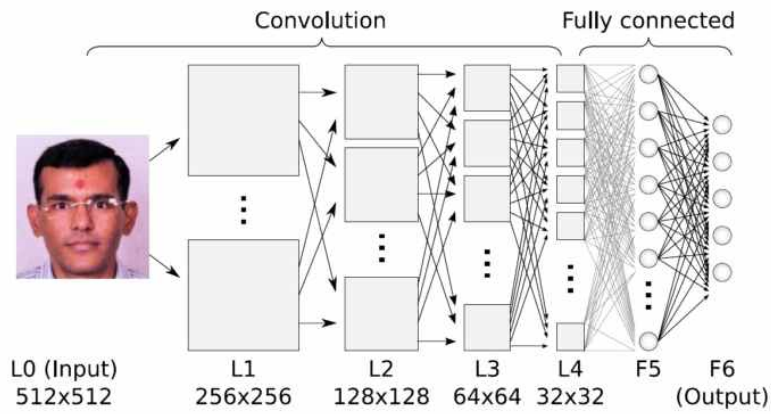
여러 분석방법을 이용하여 추정하는 방식인 앙상블(ensemble)과 유사한 개념으로도 생각할 수 있다.



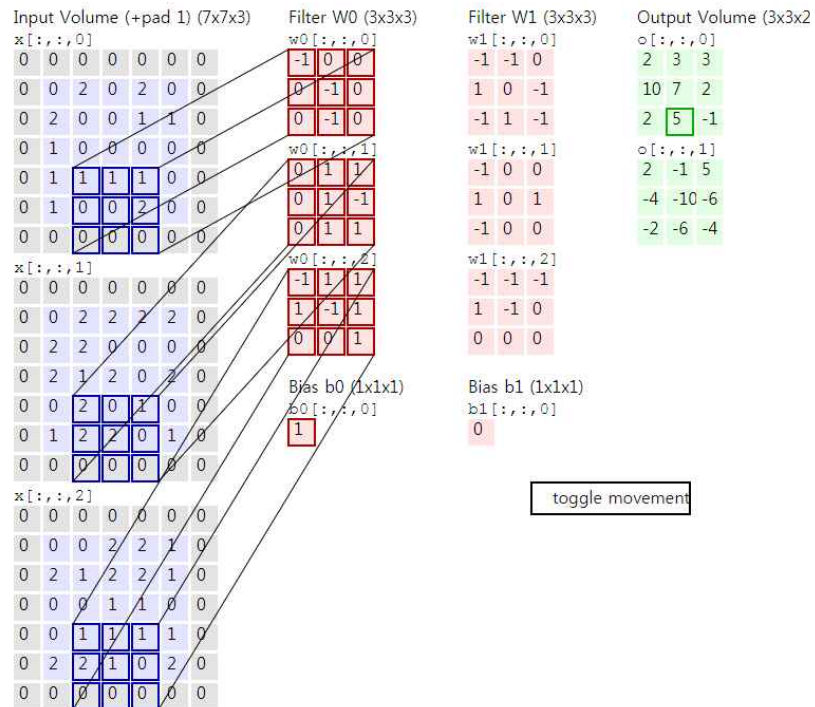
2.4.4 CNN(Convolution Neural Network)

CNN은 이미지 식별 분야에서 괄목할 만한 성능을 보여주고 있다. 이는 이미지의 일부분만을 인식하는 값을 필터(연구자에 의해 크기 조정가능)를 통해서 새로운 값을 만들어주고 이 값들을 지속적으로 연결시켜줘서 이미지의 특징을 보다 잘 이해할 수 있도록 하는 방법론이다. 현재까지 CNN은 이미지 데이터 특징에 맞게 사용되어 왔지만 본 연구에서는 이들 방법론을 금융시장 데이터에 적용해서 성능을 테스트 하려고 한다. 하지만 CNN 방법론도 역시 필터의 수, 크기, 특성 등에 영향을 크게 받을 수 있고 금융데이터의 특성과 설명변수의 배열순서에 따라서 결과가 크게 달라질 수 있다는 위험도 존재한다.

Convolution Neural Network



자료: Makwana(2016)



자료: CS231n Convolutional Neural Networks for Visual Recognition

III. 추정결과

3.1 데이터

분석에 사용된 데이터는 2010년 설립된 예측모델 및 분석대회 플랫폼인 Kaggle 데이터셋을 이용한다. Kaggle은 기업 및 단체에서 데이터와 해결과제를 등록하면 데이터 과학자들이 이를 해결하는 모델을 개발하고 경쟁하는 플랫폼이고 2017년 3월 구글에 인수되었다. 본 연구의 데이터는 2005년 4월~ 2006년 9월까지 대만의 신용카드 지급결제 부도(default payments)와 관련된 자료로 인구통계학 특징(demographic factors), 신용 데이터(credit data), 지급결제 내역, 신용카드 고객의 청구명세서(billing statement)의 정보를 가지고 있다. 총 24개의 설명변수와 1개의 종속변수로 구성되어 있고 있다. 이 중 성별, 결혼 유무, 부도여부는 범주형 데이터이고 나머지 교육수준, 나이, 지급결제 및 청구서 정보는 수치형 데이터로 구분해 분석하였다. PAY 데이터는 각 기준년월의 연체 기간(월단위)을 의미하고 있다. 그리고 BILL_AMT는 각 기준년월의 카드 청구금액을 의미한다. PAY_AMT는 각 기준년월의 이전 지불금액을 나타낸다. 각 금액단위는 NT Dollar(New Taiwan Dollar)인 신대만 달러로 나타나 있다.

앞에서의 범주형 데이터는 더미(dummy) 변수로 변환하였고 수치형 데이터는 머신러닝 알고리즘과 딥러닝 신경망 분석의 특성으로 0~1로 표준화시켜 줬다. Logistic, SVM, Lasso 등 머신러닝 알고리즘은 데이터 수치의 크기가 너무 차이가 나거나 크면 수렴(converge)이 어려운 특성이 있다. 그리고 이들 데이터는 금액단위 데이터로 크기 보다는 금액의 정도가 중요하다. 또한, 딥러닝 신경망 분석은 앞에서의 머신러닝 알고리즘 보다 추정해야 되는 계수(parameter)가 훨씬 많아서 데이터의 수준이 커지면 수렴이 더욱 어려워지는 문제가 발생한다. 따라서 본 연구에서는 수치형 데이터는 모두 0~1로 표준화 과정을 걸쳐서 분석한다.

UCI credit card 데이터는 총 3만개의 개인별 관측치가 존재한다. 이 데이터를 통해 다양한 알고리즘의 성능을 테스트하기 위해서는 데이터를 훈련데이터(train)와 테스트 데이터로 나뉘어야 한다. 훈련데이터를 이용해서 머신러닝 및 딥러닝 신경망 분석 방법론을 훈련시키고 이를 통해서 추정된 가중치(weight)와 테스트 데이터의 설명변수를 이용하여 테스트 데이터의 부도여부를 추정하고 실제 값과 차이를 통해서 정확도와 AUC를 계산한다. 이를 통해 알고리즘의 성능을 검증할 수 있다. 본 연구에서는 총 3만개의 데이터 중 70%는 훈련데이터 사용하고 나머지 30%는 테스트 데이터로 사용한다.

데이터 설명

변수명	내역
ID	ID of each client
LIMIT_BAL	Amount of given credit in NT dollars (includes individual and family/supplementary credit
SEX	Gender (1=male, 2=female)
EDUCATION	(1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
MARRIAGE	Marital status (1=married, 2=single, 3=others)
AGE	Age in years
PAY_0	Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, ... 8=payment delay for eight months, 9=payment delay for nine months and above)
PAY_2	Repayment status in August, 2005 (scale same as above)
PAY_3	Repayment status in July, 2005 (scale same as above)
PAY_4	Repayment status in June, 2005 (scale same as above)
PAY_5	Repayment status in May, 2005 (scale same as above)
PAY_6	Repayment status in April, 2005 (scale same as above)
BILL_AMT1	Amount of bill statement in September, 2005 (NT dollar)
BILL_AMT2	Amount of bill statement in August, 2005 (NT dollar)
BILL_AMT3	Amount of bill statement in July, 2005 (NT dollar)
BILL_AMT4	Amount of bill statement in June, 2005 (NT dollar)
BILL_AMT5	Amount of bill statement in May, 2005 (NT dollar)
BILL_AMT6	Amount of bill statement in April, 2005 (NT dollar)
PAY_AMT1	Amount of previous payment in September, 2005 (NT dollar)
PAY_AMT2	Amount of previous payment in August, 2005 (NT dollar)
PAY_AMT3	Amount of previous payment in July, 2005 (NT dollar)
PAY_AMT4	Amount of previous payment in June, 2005 (NT dollar)
PAY_AMT5	Amount of previous payment in May, 2005 (NT dollar)
PAY_AMT6	Amount of previous payment in April, 2005 (NT dollar)
default.payment.next.month	Default payment (1=yes, 0=no)

데이터 기초통계

변수명	최소값	최대값	중간값	평균	표준편차
ID	1	30,000	15,001	15,001	8,660
LIMIT_BAL	10,000	1,000,000	140,000	167,484	129,748
SEX	1	2	2	2	0
EDUCATION	0	6	2	2	1
MARRIAGE	0	3	2	2	1
AGE	21	79	34	35	9
PAY_0	-2	8	0	0	1
PAY_2	-2	8	0	0	1
PAY_3	-2	8	0	0	1
PAY_4	-2	8	0	0	1
PAY_5	-2	8	0	0	1
PAY_6	-2	8	0	0	1
BILL_AMT1	-165,580	964,511	22,382	51,223	73,636
BILL_AMT2	-69,777	983,931	21,200	49,179	71,174
BILL_AMT3	-157,264	1,664,089	20,089	47,013	69,349
BILL_AMT4	-170,000	891,586	19,052	43,263	64,333
BILL_AMT5	-81,334	927,171	18,105	40,311	60,797
BILL_AMT6	-339,603	961,664	17,071	38,872	59,554
PAY_AMT1	0	873,552	2,100	5,664	16,563
PAY_AMT2	0	1,684,259	2,009	5,921	23,041
PAY_AMT3	0	896,040	1,800	5,226	17,607
PAY_AMT4	0	621,000	1,500	4,826	15,666
PAY_AMT5	0	426,529	1,500	4,799	15,278
PAY_AMT6	0	528,666	1,500	5,216	17,777
default.payment .next.month	0.00	1.00	0.00	0.22	0.42

훈련데이터 기초통계

구분	훈련데이터			테스트데이터		
	중간값	평균	표준편차	중간값	평균	표준편차
ID	0.13	0.16	0.13	0.13	0.16	0.13
LIMIT_BAL	0.00	0.39	0.49	0.00	0.40	0.49
SEX	0.33	0.31	0.13	0.33	0.31	0.13
EDUCATION	0.22	0.25	0.16	0.22	0.25	0.16
MARRIAGE	0.20	0.20	0.11	0.20	0.20	0.11
AGE	0.20	0.19	0.12	0.20	0.18	0.12
PAY_0	0.20	0.18	0.12	0.20	0.18	0.12
PAY_2	0.20	0.18	0.12	0.20	0.18	0.12
PAY_3	0.20	0.17	0.11	0.20	0.17	0.11
PAY_4	0.20	0.17	0.12	0.20	0.17	0.11
PAY_5	0.17	0.19	0.07	0.17	0.19	0.06
PAY_6	0.09	0.11	0.07	0.09	0.11	0.07
BILL_AMT1	0.10	0.11	0.04	0.10	0.11	0.04
BILL_AMT2	0.18	0.20	0.06	0.18	0.20	0.06
BILL_AMT3	0.10	0.12	0.06	0.10	0.12	0.06
BILL_AMT4	0.27	0.29	0.05	0.27	0.29	0.04
BILL_AMT5	0.00	0.01	0.02	0.00	0.01	0.02
BILL_AMT6	0.00	0.00	0.01	0.00	0.00	0.01
PAY_AMT1	0.00	0.01	0.02	0.00	0.01	0.02
PAY_AMT2	0.00	0.01	0.02	0.00	0.01	0.03
PAY_AMT3	0.00	0.01	0.04	0.00	0.01	0.03
PAY_AMT4	0.00	0.01	0.03	0.00	0.01	0.03
PAY_AMT5	0.00	0.22	0.41	0.00	0.23	0.42
PAY_AMT6	0.00	0.00	0.04	0.00	0.00	0.05
default.payme nt.next.month	0.00	0.46	0.50	0.00	0.45	0.50

3.2 분석결과

우선 분석결과를 보기 전에 비교 알고리즘으로의 머신러닝 분석방법과 딥러닝 신경망 분석에 대해 설명하고자 한다. 앞에서 다양한 머신러닝 알고리즘을 설명한데로 기본적으로 Logistic 회귀분석, SVM, Random Forest, Lasso를 비교 모델로 이용한다. SVM 모델은 커널함수에 따라서 Gaussian과 Linear로 나뉘서 분석결과를 나타낸다. 그리고 딥러닝 신경망 분석은 모형, 활성화함수, 초기값 설정방식에 따라서 다양한 결과가 나올 수 있고 특히, 은닉층과 뉴런(혹은 노드)의 수에 따라서 추정의 정확도 상당히 달라질 수 있다. 다만, 딥러닝 신경망 분석이 각 조건에 따라서 너무 많은 결과를 도출해야 하기 때문에 활성화 함수로는 최근 지도학습 방법으로 가장 많이 사용하고 있는 ReLU를 사용한다. Tanh과 Sigmoid는 Vanishing gradient의 문제점이 있기 때문에 본 연구에서는 따로 분석하지 않는다. 그리고 초기값 설정에 있어서도 RBM은 비지도학습 방법으로 MLP에 나오는 문제점을 일부 해결할 수 있지만 현재 Xavier(2010)과 He's(2015)의 초기값 설정이 보다 향상된 결과를 가져오고 있기 때문에 본 연구에서는 지도학습 방법을 중심으로 결과를 도출하려고 한다. 그리고 분석결과 비교를 위해서 정확도(Accuracy)와 AUC(Area under the curve)를 사용한다. AUC는 모델의 예측정확도를 측정하는 값으로 AUC가 높으면(1에 가까울수록) 예측정확도가 높은 모델이라는 의미이고 0.5 이하일 경우는 분석모델의 가치가 없다는 것을 의미한다.

머신러닝 및 딥러닝 신경망 알고리즘 현황

모형	내용		
Deep Learning	모형	활성함수	초기값 설정방식
	MLP	• Tanh	• Random Number
	DBN	• Rectifier Linear Unit(ReLU)	• RBM(2006)
	CNN	• Maxout	• Xavier(2010)
Logistic Regression			
Support Vector Machine	Gaussian Kernel	$f(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{2\sigma}\right)$	
	Linear Kernel	$f(x_i, x_j) = x_i \cdot x_j$	
Random Forest			
Lasso	$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \ y - X\beta\ _2^2 \right\} \text{ subject to } \ \beta\ _1 \leq t.$		

우선, 머신러닝 분석을 보면, Logistic, SVM, Random Forest, Ridge 및 Lasso 분석의 정확도를 보면 최저 SVM(Gaussian Kernel)의 0.7495에서 최고 Random Forest의 0.8099까지 나타났다. 정확도(accuracy)와 AUC의 순서가 알고리즘 모두 일치해서 여기에서 정확도만을 보고한다. 결국, 딥러닝 신경망을 제외한 머신러닝 알고리즘에서는 Random Forest가 가장 성능이 좋다고 할 수 있다. 한편, 딥러닝 신경망에서는 기초적으로 은닉층 2개와 뉴런(노드) 5개를 이용해 학습한다. 딥러닝 신경망 분석별로 정확도를 측정하면, 활성화 함수는 ReLU이고 초기값 설정 방법은 Xavier(2010)을 사용한 것이 가장 높은 정확도를 달성하고 있다. ReLU with Dropout은 기존 논문과 달리 ReLU만을 이용한 것보다 낮은 성과를 보이고 있다. 한편, CNN을 이용한 알고리즘의 정확도는 0.8072로 ReLU를 이용한 딥러닝 신경망 분석보다는 낮은 결과를 나타내고 있다. 이는 이미지 데이터와 달리 filtering 과정에서의 정보 추출 능력이 낮고 설명변수의 개수가 적기 때문에 CNN을 통한 분석의 장점이 발휘되지 않았기 때문으로 보인다. 이미지 분야에서는 CNN 모델이 다른 딥러닝 신경망 분석보다도 높은 성과를 보이고 있다. 결과적으로 보면 ReLU를 이용한 신경망 분석이 다른 머신러닝 알고리즘 보다 높은 성과를 보여서 딥러닝 신경망 분석의 효용성이 있다고 판단할 수 있다.

머신러닝 및 딥러닝 신경망 알고리즘 정확도 및 AUC 결과

알고리즘	구분		정확도(Accuracy)	AUC
	활성함수	초기값		
Deep Learning with hidden layer 2, node 5	ReLU	Random	0.8137	0.7630
		Xavier(2010)	0.8156	0.7629
		He's(2015)	0.8047	0.7236
	ReLU with Dropout	RBM(2006)	0.7701	0.7303
		Xavier(2010)	0.8013	0.7488
		He's(2015)	0.8027	0.7503
	CNN with filter(2,2), kernel(1,1), Maxpool(same), Xavier		0.8072	0.7314
Logistic Regression			0.8037	0.7287
Support Vector Machine	Gaussian Kernel		0.7495	0.6377
	Linear Kernel		0.7701	0.6475
Random Forest			0.8099	0.7569
Ridge Regression			0.7807	0.7274
Lasso Regression			0.7821	0.7284

하지만 이와 같은 분석결과는 일반적인 딥러닝 신경망 분석이 은닉층을 4개 이상 갖

고 뉴런도 설명변수 이상을 갖는다는 측면에서는 초기 분석에 불구하다. 따라서 우리는 다양한 은닉층과 뉴런 수의 변화를 통해서 딥러닝 신경망 분석의 결과가 어떻게 달라지는지를 봐야한다. 은닉층 수와 뉴런수를 조정하여 정확도와 AUC를 본 결과는 은닉층 2개를 사용했을 때보다 의미있게 향상되지는 못했다. 하지만 은닉층 10개와 뉴런수 15개 일 때 가장 높은 AUC를 보이는 것으로 나타났다. 이러한 결과는 데이터셋의 설명변수와 관측치가

은닉층과 뉴런의 변화에 따른 딥러닝 신경망 정확도 및 AUC 결과

은닉층수	뉴런수	정확도(Accuracy)	AUC
4	15	0.8135	0.7614
	25	0.8123	0.7663
6	15	0.8149	0.7277
	25	0.8085	0.7650
8	15	0.8127	0.7545
	25	0.8109	0.7608
10	15	0.8161	0.7726
	25	0.8113	0.7663

주: 활성화함수는 ReLU를 사용하였고 초기값 설정은 Xavier를 이용

VI. 결론

현재 국내외 논문에서 금융시장의 데이터를 활용한 딥러닝 신경망 분석은 거의 전무한 상황이다. 이는 딥러닝 신경망의 연구가 주로 이미지, 음성인식, 번역 등의 분야에서 활발히 이루어지고 있지만 아직 금융시장의 데이터까지 적용되기는 알고리즘의 숙련도와 적용의 접합성이 낮기 때문으로 풀이된다. 일부 국내 금융회사에서 최근 머신러닝 분야에 관심을 갖고 채용을 늘리고 있지만 아직까지는 딥러닝 신경망과 관련해서 가능성 정도에서 테스트하는 수준으로 판단된다.

따라서 본 연구에서는 딥러닝 신경망을 이용해 신용카드 부도율 예측의 정확도를 얼마나 향상시킬 수 있는지를 점검한다. 이를 위해서 과거에서부터 많이 사용되고 있는 머신러닝 알고리즘(Logistic, SVM, RandomForest, Lasso 등)을 딥러닝 신경망 분석의 성능테스트를 위한 비교 지표로 활용한다. 우선, 본고의 딥러닝 신경망 분석은 기본적으로 2개의 은닉층(hidden layers)과 5개의 뉴런(neuron)로 구축하고 활성화함수와 초기값 설정방법

에 따른 예측의 정확도를 도출한다. 그 결과 딥러닝 신경망 분석이 기존 머신러닝 알고리즘 보다 높은 성능을 나타냈고 그 중 ReLU(활성함수)와 Xavier(초기값)가 가장 높은 예측의 정확도를 보였다. ReLU와 Xavier를 기준으로 은닉층과 뉴런의 수를 최대 10개와 25개까지 늘려 분석한 결과에서도 유사한 결과를 나타냈다. 다만, 기존 연구에서와 같이 은닉층과 뉴런의 수의 증가에 따른 성능의 향상은 나타나지 않았다. 또한, 이미지 분야에서 높은 성능을 보였던 Dropout과 CNN(convolution neural network) 모델도 뚜렷한 성능의 향상을 보이지 못했다. 이는 신용카드 부도율 데이터의 설명변수 및 관측치에 한계가 있기 때문으로 풀이된다. 본 연구에서 사용된 개인의 신용카드 부도율 데이터는 횡단면 자료로 패널 및 시계열 데이터에서 높은 성능을 나타내는 딥러닝 신경망 알고리즘인 RNN(recurrent neural network)을 사용하지 못하는 것이 아쉽다. 따라서 추후 신용평가사 및 금융기관의 빅데이터를 활용해 딥러닝 신경망 분석을 실시한다면 과거 보다 향상된 결과를 도출할 수 있을 것으로 기대된다.

본 연구는 실무적인 관점에서 금융데이터를 활용해 얼마나 부도율을 낮출 수 있는지에 대한 문제로 전환할 수 있다. 그리고 신용평점을 책정하는 데에 있어서도 도움이 될 수 있을 것이다. 이를 통해서 보다 정교한 신용평가 가능해진다면 금융회사는 대손비용을 줄일 수 있고 금융소비자들은 보다 낮은 금융비용으로 서비스를 이용할 수 있게 될 것이다. 따라서 이에 대한 금융당국 및 금융기업들의 딥러닝 신경망 분석 연구에 대한 지원과 장려가 필요한 시점이라고 생각한다.

참고문헌

- Bengio, Y.(2009), Learning Deep Architectures for AI. Foundations & Trends in Machine Learning, 2(2009), 1-127.
- Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. 1994. "Learning Long-Term Dependencies with Gradient Descent Is Difficult." Neural Networks, IEEE Transactions on 5 (2). IEEE: 157 - 66.
- Bengio, Yoshua. 2009. "Learning Deep Architectures for Ai." Foundations and Trends in Machine Learning 2 (1). Now Publishers Inc.: 1 - 127.
- By Dr R M Makwana - December 8, 2016
- Cuicui Luo Desheng Wu Dexiang Wu, 2016, A deep learning approach for credit scoring using credit default swaps Engineering Applications of Artificial Intelligence, Volume 65, October 2017, Pages 465-470
- Deep Face Recognition Using Deep Convolutional Neural Network
- Fischer, Asja, and Christian Igel. 2012. "An Introduction to Restricted Boltzmann Machines." In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 14 - 36. Springer.
- Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. 2011. "Deep Sparse Rectifier Networks." In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. Jmlr W&CP Volume, 15:315 - 23.
- Gupta, S.(2016), Deep Learning vs. traditional Machine Learning algorithms used in Credit Card Fraud Detection, Masters thesis, Dublin, National College of Ireland.
- Han-Hsing, Tu. n.d. "[ML, Python] Gradient Descent Algorithm (Revision 2)." <http://hhtucode.blogspot.kr/2013/04/ml-gradient-descent-algorithm.html>.
- Hassabis et al.(2016), Mastering the game of Go with deep neural networks and tree search, Nature 529, pp.484~489.
- Hinton, G.E. Osindero, S., The, Y.W.(2006), A fast learning algorithm for deep belief nets, Neural Computation, 18(7), 1527-1554.
- Hinton, Geoffrey E, and Ruslan R Salakhutdinov. 2006. "Reducing the Dimensionality of Data with Neural Networks." Science 313 (5786). American Association for the Advancement of Science: 504 - 7.
- Hinton, Geoffrey E, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. "Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors." arXiv Preprint arXiv:1207.0580.
- Hinton, Geoffrey E. 2009. "Deep Belief Networks." Scholarpedia 4 (5): 5947.
- Hinton, Geoffrey, Simon Osindero, and Yee-Whye Teh. 2006. "A Fast Learning Algorithm for Deep Belief Nets." Neural Computation 18 (7). MIT Press: 1527 - 54.
- Hinton, Geoffrey. n.d. "Coursera: Neural Networks for Machine Learning."

- <https://class.coursera.org/neuralnets-2012-001>.
- Honkela, Antti. n.d. "Multilayer Perceptrons." <https://www.hiit.fi/u/ahonkela/dippa/node41.html>.
- http://www.dmi.usherb.ca/~larocheh/projects_deep_learning.html.
- Larochelle, Hugo. n.d. "Deep Learning."
- LeCun, Yann, and F Huang. 2005. "Loss Functions for Discriminative Training of Energybased Models." In. AISTats.
- LeCun, Yann, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. "A Tutorial on Energy-Based Learning." Predicting Structured Data.
- Lee, Honglak, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. 2009. "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations." In Proceedings of the 26th Annual International Conference on Machine Learning, 609 - 16. ACM.
- Li, F., Jonson, J., Yeung, S.(2017), Convolutional Neural Networks for Visual Recognition Lecture Note, 2017.4.13.
- Minsky, M., Papert, S.(1969), Perceptrons: an introduction to computational geometry, MIT Press.
- Muehlhauser, Luke. n.d. "A Crash Course in the Neuroscience of Human Motivation." http://lesswrong.com/lw/71x/a_crash_course_in_the_neuroscience_of_human/.
- Nair, Vinod, and Geoffrey E Hinton. 2010. "Rectified Linear Units Improve Restricted Boltzmann Machines." In Proceedings of the 27th International Conference on Machine Learning (Icml-10), 807 - 14.
- Niimi, A., 2015, Deep Learning for Credit Card Data Analysis, World Congress on Internet Security (WorldCIS-2015).
- Paul, W.(1982), Applications of advances in nonlinear sensitivity analysis, System modeling and optimization, Springer, 762 - 770.
- Ramanathan, V.(2014), Fraud Prevention Using Deep Learning, H2O World 2014, 2014.11.19.
- Ranzato, Marc'Aurelio. n.d. "Deep Learning for Vision: Tricks of the Trade." www.cs.toronto.edu/~ranzato.
- Rosenblatt, Frank. 1958. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." Psychological Review 65 (6). American Psychological Association: 386.
- Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams. 1985. "Learning Internal Representations by Error Propagation." DTIC Document.
- Salakhutdinov, Ruslan, and Geoffrey E Hinton. 2009. "Deep Boltzmann Machines." In International Conference on Artificial Intelligence and Statistics, 448 - 55.
- Sheng-Tun Li, Weissor Shiue, Meng-Huah Huang, 2006, The evaluation of consumer loans using support vector machines, Expert Systems with Applications Volume 30, Issue 4, May 2006, Pages 772-782.

- Smolensky, Paul. 1986. "Information Processing in Dynamical Systems: Foundations of Harmony Theory." Department of Computer Science, University of Colorado, Boulder.
- Wan, Li, Matthew Zeiler, Sixin Zhang, Yann L. Cun, and Rob Fergus. 2013. "Regularization of Neural Networks Using Dropconnect." In Proceedings of the 30th International Conference on Machine Learning (ICML-13), 1058 - 66.
- Wan, Li. n.d. "Regularization of Neural Networks Using Dropconnect." <http://cs.nyu.edu/~wanli/dropc/>.
- Wikipedia. n.d. "Wikipedia." http://en.wikipedia.org/wiki/Restricted_Boltzmann_machine.
- Woodie, A(2016), How Credit Card Companies Are Evolving with Big Data, Datanami, 2016.3.3.
- Zeiler, Matthew D, M. Ranzato, Rajat Monga, M. Mao, K. Yang, Quoc Viet Le, Patrick Nguyen, et al. 2013. "On Rectified Linear Units for Speech Processing." In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 3517 - 21. IEEE.