

기계 학습을 활용한 이미지 분류 및 환율 예측 프로그램 설계

김선민(2012122034), 김훈창(2012122079), 김현수(2013122071)

지도교수: 최지훈

요 약

본 논문에서는 기계 학습 알고리즘을 이용하여 모션 인식 및 지정된 동작 수행 프로그램과 환율 예측 프로그램을 설계한다. 첫 번째 모션 인식 및 지정된 동작 수행 프로그램에서는 6가지 모션을 정의하고 다양한 환경에서 촬영한 모션 이미지를 Alexnet의 전이 학습 (Transfer Learning) 기능을 이용해서 학습한다. 그리고 사용자가 특정 모션을 취하면 Alexnet에서 학습된 결과를 이용하여 6가지 모션 중 가장 유사한 모션으로 인식해서 미리 지정된 동작을 수행한다. 두 번째 환율 예측 프로그램에서는 환율 변동 요인을 분석한 후 선형 회귀 (Linear Regression) 기법을 이용한 환율 예측과 신경망(Neural Network)을 이용한 환율 등락 예측을 수행한다. 이를 위해 선형 회귀 모델을 정의한 후 기계 학습을 이용하여 계수를 최적화하고, MATLAB에서 제공되는 기본 기능을 이용하여 환율 등락 예측을 위한 신경망을 구현한다. 환율 예측 정확도 개선을 위해 파라미터 최적화 및 사전 연산 기법을 적용하고, 모의실험을 통해 실제 환율과 예측된 환율의 오차를 계산한다.

Keywords : 기계 학습, 모션 분류, Alexnet, 환율, 선형 회귀, 신경망

I. 서 론

4차 산업혁명의 일환으로 막대한 양의 데이터를 이용한 인공지능 기술이 급속도로 발전하고 있다. 인공지능을 활용하여 인간이 수행하기 힘든 일을 대체하거나 좀 더 편리한 삶을 살 수 있는 기술들이 다양하게 개발되고 있다. 특히 이미지 인식, 빅데이터를 이용한 미래 예측, 보안 및 투자 분야에 널리 활용되고 있다.

인공지능 기술을 이용하는 데에 가장 중심적인 알고리즘은 기계학습이다. 기계학습은 알고리즘은 크게 지도 학습 (Supervised Learning) 과 비지도 학습 (Unsupervised Learning)으로 나눌 수 있다 [1]. 지도 학습은 데이터 값과 그 데이터의 정답 값이 프로그램에 주어진다. 그리고 이것을 연속적인 값 또는 이산적인 값을 결과로 하느냐에 따라 회귀와 분류로 나뉜다. 회귀와 분류의 대표적인 알고리즘으로는 각각 선형 회귀와 신경망이 있다. 선형 회귀는 Hypothesis Function을 정의하고 예측값과 결과값의 차이를 줄이는 방식으로 설계된 알고리즘이다. 신경망의 경우 입력값에 서로 다른 가중치를 두고 가중치에 따른 Sigmoid Function을 적용하여 결과값을 예측한다.

본 논문에서는 수집한 데이터와 다양한 머신러닝 알고리즘을 활용한 두 가지 프로그램을 설계한다. 첫째로 모션 인식 프로그램이다. 이미지 인식 알고리즘으로는 Alexnet을 사용하였으며 [2], 각 모션에 따른 특정 기능을 지정하여 특정 모션이 인식되면 해당 기능을 수행하

도록 프로그램을 설계한다.

두 번로 환율 예측 프로그램이다. 설계에 앞서 환율 변동 요인들에 대해 학습하고 이를 바탕으로 한국은행 경제 통계시스템을 통해 데이터를 수집하였다 [3]. 이를 선형 회귀와 신경망에 각각 적용하여 미래의 환율을 예측하는 프로그램을 설계한다.

본 논문의 II장에서는 Alexnet을 활용한 모션 인식 프로그램의 설계 과정을 설명하고, III장에서는 선형 회귀와 신경망을 활용한 환율 예측 프로그램의 설계 과정을 설명한다. IV장에서는 모의실험을 통해 각 프로그램의 성능을 평가하고, V장에서는 결론을 제시한다.

II. 딥러닝 기반 모션 인식 프로그램

본 주제에서는 마우스나 키보드를 사용하지 않고 웹캠을 통해서 지정한 특정 모션을 인식시키면 녹음, 동영상 촬영, 사진 촬영, 웹 브라우저 실행 등의 기능이 자동으로 실행되는 프로그램을 설계한다.

1. 모션 및 수행 기능 정의

가. bye (종료)

인사 하는 손동작을 표현한 모션이다. 이 모션을 인식시켜 실행중인 기능을 종료하거나 프로그램을 종료할 수 있다.



그림 1. 지정한 각 모션과 기능 설명

나. v (사진 촬영)

사진촬영 시 대표적인 포즈를 표현한 모션이다. 이 모션을 인식시키면 3초 후 사진이 촬영된다. 촬영된 사진은 파일로 저장 한 후 확인용으로 2초 간 화면에 출력해준다.

다. mic (녹음)

마이크를 쥐는 듯한 동작을 표현한 모션이다. 이 모션을 인식시키면 3초 후 음성 녹음이 시작된다. 녹음 시간이 화면 밑에 표시되고 'bye' 모션을 인식시켜 중지시킬 수 있다. 녹음된 음성은 파일로 저장한 후 확인용으로 한 번 재생해 준다.

라. finger snap (동영상 촬영)

손가락을 튕겨 소리를 내는 동작을 표현한 모션이다. 이 모션을 인식시키면 3초 후 동영상 촬영이 시작 된다. 촬영 시간이 화면 밑에 표시되고 'bye' 모션을 인식시켜 중지시킬 수 있다. 촬영된 영상은 파일로 저장 한 후 확인용으로 한 번 재생해 준다.

마. i (웹 브라우저 실행)

검지 손가락을 이용해 알파벳 I를 표현한 모션이다. 이 모션을 인식시키면 MATLAB 내부에서 인터넷 웹 브라우저가 실행된다. 'bye' 모션을 인식시켜 웹 브라우저를 종료시킬 수 있다.

바. waiting (대기)

모션 인식을 하기 위한 대기 상태이다. 지정한 5가지의 모션을 취하지 않는다면 이 상태로 분류된다.

2. 모션 인식을 위한 기계 학습

조원들이 직접 촬영한 이미지 데이터와 대표적인 CNN model인 Alexnet을 이용해 전이 학습시켰다. 5가지 모션과 인식 대기를 포함해서 6개의 class로 구성했고, class당 600장의 이미지를 사용했다. 배경, 조명 등 주위의 환경에 의해 인식 정확도가 크게 영향을 받지 않게끔 장소를 변화시켜가며 학습 이미지를 촬영했다.

한 장소에서 손의 위치나 얼굴의 각도, 표정 등을 변화시켜가며 한 모션에 대해 10장의 이미지를 촬영한다. 같은 방법으로 다른 모션들의 이미지를 촬영한다. 조명을 변화시킬 수 있다면 같은 장소에서 조명에 변화준 후 같은 방법으로 각 모션에 대해 10장씩 이미지를 촬영한다. 조명에 변화를 주기 어렵다면 장소를 옮겨 같은 방법으로 이미지를 촬영한다. 위 과정을 반복하여 수집된 이미지 데이터를 학습에 이용한다.

3. 모션 인식 정확도 개선

학습 이미지 2000장을 사용해 이미지 분류 방식의 모션 인식 초기 모델을 설계한 이후 데모 시현을 통해 실제 동작을 확인했다. 모션 간의 분류는 준수한 정확도를 보였지만, 아무런 모션을 취하지 않았을 때 특정 모션으로 인식되어 제멋대로 기능이 실행되는 경우가 빈번히 발생하였다. 이러한 경우를 '오 분류'로 정의하고, 이를 낮추기 위해 다음과 같은 시도를 하였다.

가. 학습 데이터 (Training Data) 확대

전체적인 모션 인식 정확도를 향상시키기 위해 학습 이미지를 추가로 촬영했다. 기존에 조원 한 명이 촬영한 사진 2000장에 조원 3명 모두의 사진을 추가했고 총 3600장의 사진으로 새롭게 학습시켰다.

나. 임계값 (Threshold) 설계

오 분류 확률을 낮추기 위해 테스트 이미지로 측정한 각 모션별 확률 분포를 이용했다. CCDF 그래프를 이용해 적절히 임계값을 설계했고, 이미지 분류 출력 확률에 임계값을 적용하여 그 값을 넘지 않으면 waiting으로 분류되는 부분을 추가하여 오 분류를 개선했다.

III. 환율 예측 프로그램

과거 환율, 주식 가격, 금리 변동, 무역 수지 등 환율에 영향을 주는 요인을 발굴하고, 기계 학습 기법을 적

용하여 환율을 예측하는 프로그램을 설계한다. 우선 선형 회귀 모델을 사용하여 환율 값을 예측하고, 다음으로 신경망을 이용하여 환율의 증감 여부를 예측한다.

1. 파라미터 정의

데이터로 수집한 환율에 영향을 미치는 n 개의 요인들은 Feature Matrix \mathbf{X} 로 정의하고 환율 예측을 위한 입력으로 사용된다. 이 때, ‘일’ 단위의 환율 예측을 위해 ‘월’ 단위로 수집된 데이터에 대해서 Interpolation을 적용한다. 실제 환율 정보는 Output Vector \mathbf{y} 로 정의하고 예측한 환율과 비교하여 오차를 줄이기 위해 노력한다. 수집한 데이터 중 예측에 사용되는 70%에 해당하는 m 개의 데이터는 Training Set, 성능 비교에 사용되는 30%의 데이터는 Test Set으로 구분하고 각각의 Feature matrix를 \mathbf{X}_{train} , \mathbf{X}_{test} , Output Vector를 \mathbf{y}_{train} , \mathbf{y}_{test} 로 정의한다.

2. 환율 예측을 위한 입력 인자(Input Feature) 발굴

과거 환율 변동에 영향을 주는 요인은 수집 단위에 따라 ‘일’ 단위로와 ‘월’ 단위의 데이터로 구분했다. ‘일’ 단위 데이터로는 과거의 환율정보, 외국인 순매수(코스닥 코스피), 한국은행 기준금리, 정부대출금 금리, CD, 국고채, 나스닥, 미국 10년 채권, 다우지수, S&P를 수집했다. ‘월’ 단위의 데이터로는 한국의 물가지수, 대미 수출입 총액, KOSPI, 경상수지, 자본수지, 금융계정, 대출금리, US LIBOR 등을 수집했다.

3. 선형 회귀를 이용한 환율 예측

미래의 환율 값을 예측하기 위해 선형 회귀 모델을 사용한다.

가. 환율 예측 모델 정의 및 계수 설계

환율 값을 예측하기 위한 모델로 아래와 같은 Hypothesis Function을 정의한다.

$$h_{\theta}(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n \quad (1)$$

식 (1)과 같은 함수가 정의되면 \mathbf{X}_{train} , \mathbf{y}_{train} 을 입력으로 대입하여 최적의 계수 $\theta = [\theta_0, \dots, \theta_n]^T$ 는 Normal Equation을 이용하여 다음과 같이 구한다.

$$\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (2)$$

식 (2)에서 구한 최적 θ 를 이용하여 \mathbf{X}_{test} 를 입력으로 사용할 때 환율 예측값 $h_{\theta}(x)$ 를 구한 뒤 \mathbf{y}_{test} 의 차이를 계산하여 MSE(Mean Square Error)를 구한다. 이

때, 환율 예측 시 며칠 전까지의 정보를 이용할 것인지를 나타내는 변수 no 를 변화시키면서 실험을 진행하여 MSE가 최소가 되도록 하는 no 를 찾는다.

나. 환율 예측 성능 개선

먼저 모델에 반영되는 Training Set과 Test Set을 수정한다. 주말 및 공휴일에는 데이터가 제공되지 않는 일부 Feature들을 고려하여 해당 날짜의 데이터는 삭제하도록 한다. 더불어 정확한 데이터 반영을 위해 ‘월’ 단위 데이터는 일 단위 변환없이 그대로 적용한다. 또한, ‘월’ 단위 데이터는 정보 공개 지연을 고려하여 3달 전 데이터를 사용하도록 수정한다.

이와 같이 예측 모델을 수정한 뒤, 성능을 개선하기 위해 환율에 영향을 미치는 요인들을 추가적으로 발굴한다. 그리고 러닝 과정에서 발생할 수 있는 Overfitting 현상을 막기 위해 정규화(Regularization)를 수행한다. 정규화 계수 λ 를 변화시켜가며 성능을 측정하여 최적의 λ 를 도출한다. 수정이 완료되면 기존 모델과의 성능을 비교한다.

4. 신경망을 이용한 환율 증감 예측

MATLAB에서 제공되는 신경망 Toolbox를 이용하여 환율 증감을 예측한다 [4]. 환율 증감 예측에 사용되는 입력 인자는 2절에서와 동일하다.

가. 신경망 기반 환율 예측 모델 정의

설계에 앞서 과거 환율 정보에 대한 증감 여부를 2가지의 State로 정의한다. 하루 전에 비해 환율이 하락한 경우 ‘0’, 환율이 상승한 경우 ‘1’로 분류하여 \mathbf{y} 를 재설정한다. 이후 Backward Propagation을 통해 최적의 θ 를 구한 뒤, Forward Propagation을 통해 결과값을 도출한다. 이렇게 얻어진 결과 값을 반올림하여 ‘0’ 또는 ‘1’로 분류하고 이를 예측값 $h_{\theta}(x)$ 에 대입한다. $h_{\theta}(x)$ 와 \mathbf{y}_{test} 의 각 요소들을 1대1로 비교하여 0~100% 사이의 정확도로 성능을 측정한다. 앞선 예측 모델에서와 동일하게 no 를 변화시켜가며 같은 과정을 반복하고 최대 정확도를 나타낼 때의 no 를 바탕으로 성능을 판별한다.

나. 환율 증감 예측 성능 개선

성능 개선을 위해 (1)에서 얻어진 θ 를 분석하여 각 입력 인자별로 상대적인 영향력을 검출한다. 이를 이용하여 환율에 영향이 큰 입력 인자에 다양한 전처리 함

수를 적용한다. 반대로 영향력이 작은 입력 인자에 대해서는 이를 최소화할 수 있는 전처리 함수를 적용하거나 해당 입력 인자를 제거한 뒤 러닝을 시도한다.

또한, 신경망의 은닉 계층의 개수를 증가시키면서 성능을 비교하고, 구현 복잡도를 고려하여 적절한 은닉 계층 수를 정한다. 일정 수준의 정확도를 얻었다면 환율의 증감범위를 고려하여 State의 개수를 늘려가며 성능을 측정하고 이전 모델의 결과와 비교한다.

IV. 모의실험

1. 딥러닝 기반 모션 인식 프로그램

테스트 이미지 600장을 이용해 모션 인식 정확도를 측정했다. 기존의 2000장의 이미지로 학습된 모델은 63%의 정확도를 보였고, 학습 데이터를 늘려 3600장의 이미지로 학습된 모델은 83%의 정확도를 보였다. 하지만 여전히 높은 오 분류 확률이 나타났다.

표 1. (a) 한 명의 사진(2000장)으로 학습된 모델의 정확도

정상 분류 확률	63%
오 분류 확률	37%

(b) 3명의 사진(3600장)으로 학습된 모델의 정확도

정상 분류 확률	83%
오 분류 확률	36%

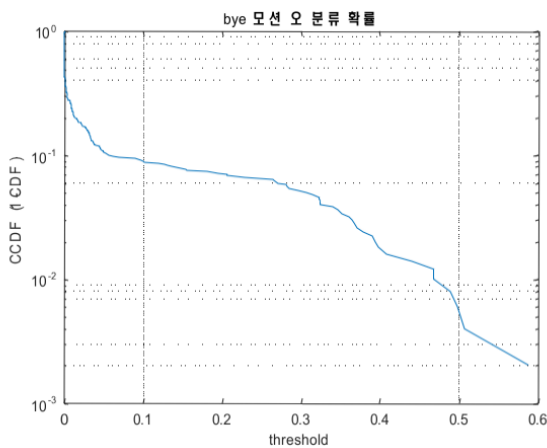


그림 2. bye 모션의 확률 분포 그래프

표 2. 각 모션별 임계값

모션	임계값
bye	0.4682
finger snap	0.5605
i	0.4961
mic	0.9869
v	0.4251

모션당 100장의 테스트 이미지를 이용해 정확도를 측정했고 이 과정에서 얻어진 모션별 확률 분포를 분석해 오 분류 확률이 1% 이하가 되는 지점을 임계값으로 설계했다. 그림 2는 'bye' 모션에 대한 임계값 설계를 위한 CCDF 분석 결과를 보여주고, 표 2는 모션별 설계된 임계값을 나타낸다. 각 모션 별로 임계값을 적용한 후 분류 정확도는 이전과 동일하게 83%로 나타났고, 설계한 대로 1%의 오 분류 확률을 보였다.

2. 환율 예측 프로그램

가. 선형 회귀 기반 환율 예측

선형 회귀 모델을 이용한 환율 예측 프로그램의 성능을 측정한다. 성능 비교는 MSE를 통해 이루어지며 총 3단계로 나누어 진행한다.

처음으로 III-3-가 에서 설계한 초기 모델에 대한 실험을 진행한다. no를 1부터 70까지 변화시켜가며 Training Error와 Test Error를 비교한 결과, no = 16일 때 Training Error가 약 6.7원, Test Error가 약 5.7원으로 최소의 오차를 나타냈다.

다음으로 성능 개선을 위한 III-3-나 에서의 시도 후의 모델의 성능을 측정한다. 먼저 데이터 통합 및 반영 시기 등을 고려해 모델을 수정한 뒤의 성능을 측정한 결과, no = 28일 때 Training Error가 약 6.7원, Test Error가 약 9.1원으로 이전 모델에 비해 오차가 증가하였다. 이러한 성능 하락은 '월' 단위 데이터의 경우 예측 시 3달 전의 데이터를 사용해야하는 불가피한 상황으로 인한 것으로 볼 수 있다.

위의 실험 결과에서 발생한 Overfitting 현상을 막기 위해 정규화를 수행하고 Input Feature를 추가 발굴한 뒤 성능을 측정하였다. 그 결과, 표 3에서와 같이 no = 4이고 $\lambda = 0.55$ 일 때 Training Error와 Test Error가 약 7.8원으로 나타남을 확인했다.

표 3. 진행 사항에 따른 평균 오차

진행 사항	초기 모델	데이터 수정	Feature 추가, Regularization
Training Error	6.7원	6.7원	7.8원
Test Error	5.7원	9.1원	7.8원

결과적으로 초기 모델에 비해 최종 모델의 예측 오차가 증가하였으나 데이터 수정을 통해 현실적 요건을 고려하였으며, 이 과정에서 증가한 오차를 줄였다는 점에서 의미 있는 결과를 얻을 수 있었다.

나. 신경망 기반 환율 증감 예측

신경망 모델을 이용한 환율 예측 프로그램의 성능을 측정한다. 앞선 실험과 동일하게 초기 모델에서 시작해 성능 개선 노력 이후의 실험 결과들을 비교한다.

하루 전에 비해 환율이 하락한 경우 '0', 환율이 상승한 경우 '1'로 분류하여 2가지의 State를 가지고 실험을 진행한다. 10개의 은닉 계층을 사용하여 진행하였으며 그 결과 $no = 5$ 일 때, Training Accuracy는 67.3%, Test Accuracy는 65.1%로 낮은 정확도를 보였다.

정확도 개선을 위해 계층을 분석한 후, 각 Feature에 대한 전처리 함수를 적용하였다. 그 결과 $no = 5$ 일 때, Training Accuracy는 68.1%, Test Accuracy는 65.7%로 눈에 띄는 성능 향상을 확인하지 못했다.

이후 State의 개수를 증가시켜 실험을 진행하였다. 환율이 5원 이상 하락한 경우 '0', 0~5원 하락한 경우 '1', 0~5원 상승한 경우 '2', 5원 이상 상승한 경우 '3'으로 총 4개의 State로 분류했다. 표 4에서는 전처리 적용 여부에 따른 환율 증감 예측 성능을 나타낸다. 2 State와 동일한 조건에서 실험을 진행한 결과 $no = 4$ 일 때, Training Accuracy는 48.3%, Test Accuracy는 42.8%로 낮은 정확도를 보였다. 이후 전처리 함수를 적용하여 성능 향상을 시도했으나 2-State에서의 결과와 마찬가지로 성능 개선 정도는 미미하게 나타났다.

표 4. State 개수별 전처리 적용 전후 성능 비교

진행 사항	2-State (전처리x)	2-State (전처리o)	4-State (전처리x)	4-State (전처리o)
Training Accuracy	67.3%	68.1%	48.3%	48.5%
Test Accuracy	65.1%	65.7%	42.8%	42.9%

마지막으로 은닉 계층의 개수에 따른 성능을 비교한다. 이 때, 정확도뿐만 아니라 실행 시간(Runtime)을 측정하여 복잡도 대비 성능 향상을 분석한다. $no = 5$ 로 고정하고 2-State 모델에서 실험을 진행했을 때, 그림 3과 같은 결과를 보였다. 은닉 계층의 수가 증가할수록 예측 정확도가 상승하고, 특정 개수 이상이 되면 정확도 상승폭이 줄어들면서 수렴하는 현상을 보였다. 반면,

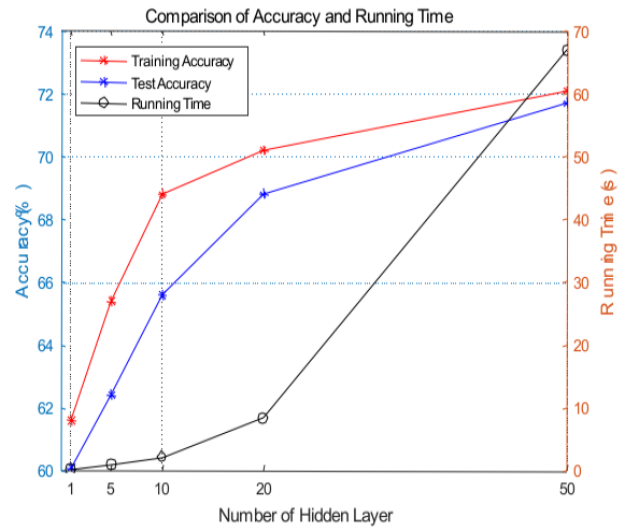


그림 3. 은닉 계층의 개수에 따른 정확도와 복잡도 비교

실행 시간은 은닉 계층의 수가 증가함에 따라 급격하게 증가했다.

전반적으로 성능 개선을 위한 시도에도 불구하고 예측 모델이 낮은 예측 정확도를 나타냄을 확인할 수 있었다. 이는 예측 모델에 반영 불가능한 정치적, 사회적 이슈들과 환율 변동 요인 중 모델에 반영하지 못한 요인의 존재 등으로 원인을 분석할 수 있다.

V. 결 론

본 논문에서는 수집한 데이터를 가지고 대표적인 기계학습 알고리즘인 선형 회귀, 신경망, Alexnet 을 활용하여 이미지 인식을 통해 원하는 기능을 수행하도록 하거나 특정 환율 값을 예측하는 프로그램을 구현하였다. 또한, 측정 결과에 따른 오차율, 정확도를 개선하기 위해 훈련 이미지와 입력 인자를 추가하거나 확률 분포 분석을 통한 임계값 조정, 정규화 수행 등을 시도하였다. 모션 인식 프로그램의 경우 사용자의 모션을 인식해서 미리 지정된 동작을 수행하는 정확도를 분석했고, 환율 예측 프로그램의 경우 일 단위 환율값 예측과 환율 증감 예측 정확도를 제시하였다.

VI. 참고문헌

- [1] https://en.wikipedia.org/wiki/Machine_learning/
- [2] <https://en.wikipedia.org/wiki/AlexNet/>
- [3] <http://ecos.bok.or.kr/>
- [4] <https://kr.mathworks.com/products/statistics.html>