

**Team Project Final Report**

**The Relationship between News and Exchange Rates**

nlp Team project

20143592 최영재

20141261 송제웅

**Indexes**

1. **Main Topic**
2. **Modified Factors**
3. **System Development**
   1. **Data Collection**
   2. **NLP Processing**
   3. **Machine Learning**
4. **Evaluation**
5. **Conclusion**

**1. Main Topic**

- **The Relationship between News and Exchange rates**

저희는 국제뉴스에 보도된 특정 국가에 대한 기사들이 환율과 어떠한 관련성을 보일 것이라는 가설을 가지고 이를 자연어 처리를 통해 검증하려 합니다.

공신력 있는 국제뉴스에서 특정 국가에 대해 긍정적인 평가를 보인 기사가 많았던 날에는 환율이 오르고, 반대이면 낮아질 것이라는 예측입니다.

NLP를 통하여 얻어낸 데이터로 ANN을 이용한 머신 러닝으로 예측 모델을 생성하여, 해당 모델의 판단을 근거로 모의 투자를 실시하여 수익을 내는 것이 목표입니다.

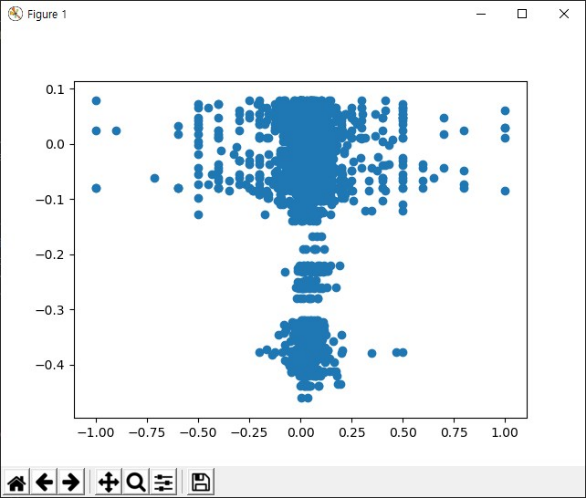
**2. Modified Factors**

프로젝트를 진행하며 최초 Proposal에 작성하였던 사항으로부터 변경점이 발생하여 설명 드리겠습니다.

|  |  |  |
| --- | --- | --- |
|  | Before | After |
| News Dataset | Only New-York Times Data | New-York Times Data&  New-York Daily Data |
| Sentiment Analyzing | Use(TextBlob Library) | Not Use |
| Lemmatization | Only Make term-doc table  (Gensim Library) | Nltk |
| Vectorization | Sklearn |
| LSA | Not Use | Use |
| ML | Training = Sentiment Data  Label = Exchange rate Data  (Numpy) | Training= Topic modeling Data  Label = Exchange rate Data  (Numpy & Cupy) |
| Development Environment | Local, | Jupyter notebook with Colab |

먼저 프로젝트를 진행하며 더 많은 데이터의 필요를 느껴 기존 New-York Times 뿐만 아니라 New-York Daily news 로부터 또한 데이터를 수집하기로 변경하였습니다.

최초 계획은 TextBlob Library의 rule-based Sentiment Analyze를 이용하여, 그 값을 토대로 기계학습을 진행하려고 하였으나

 위와 같이 해당 라이브러리를 이용하여 얻은 감정 수치와 환율 데이터 간 일련의 연관성을 찾아보기 힘들었고, rule-based Sentiment Analizing이 부정확하다는 결론을 내렸습니다.

따라서 저희는 해당 라이브러리의 사용을 철회하고, 원문 텍스트에서 Term-Doc TF-IDF matrix를 추출하여 NLP (Lemmatization, LSA 등)를 이용해 수치화하여 학습을 진행하는 것으로 방향을 바꾸었으며 이에 따라 Library Gensim, TextBlob을 사용하지 않고 Nltk, Sklearn을 사용하게 되었습니다.

또한, 기계학습 과정에서 GPU연산을 이용해 실행시간을 단축하기 위하여 기존 Local Computer에서 진행하는 것으로 계획되었던 프로젝트 중 기계학습 부분을 Google Colab을 이용한 jupyter notebook으로 대체하게 되었습니다.

**3. System Development**

**A. Data Collection**

프로젝트 진행을 위해 두 가지의 데이터가 필요하였습니다.

1. **공신력 있는 외신의 특정 국가에 대해 언급한 기사 텍스트 데이터**

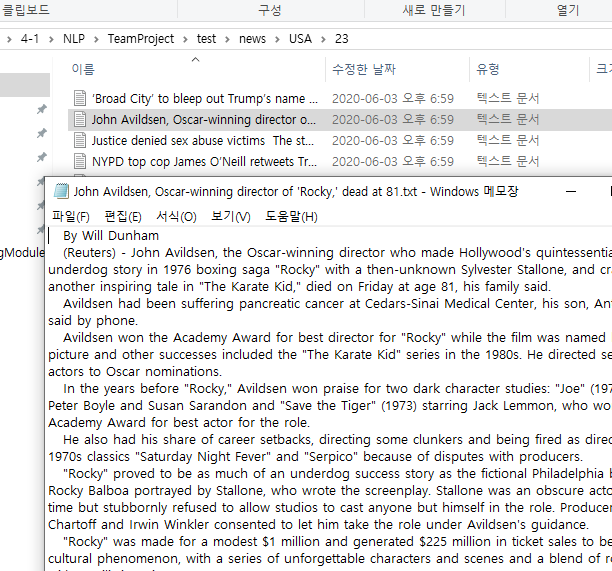
방대한 양의 텍스트 데이터를 크롤링을 통하여 얻기 위해 검색 기능이 존재하고 url을 통해 손쉽게 리퀘스트 요청이 가능한 The New York Times 사이트 및 New York Daily news 사이트를 선정했습니다.

각 나라 이름을 키워드로 검색하여 나온 기사들을 주차별로 분류하여 150주치 뉴스를 수집하였습니다.

동적 웹 크롤링을 위하여 Selenium을 활용하였고, html parsing에 BeautifulSoup이 이용되었습니다.

수집된 기사 데이터는 디렉토리로 구분되어 txt file 형식으로 저장됩니다.

* Chrome version 83에 최적화 되어 있습니다.
* 실행 시 버전이 다른 경우 해당 버전의 크롬드라이버를 다운로드 받아 resource 폴더에 넣으셔야 합니다.

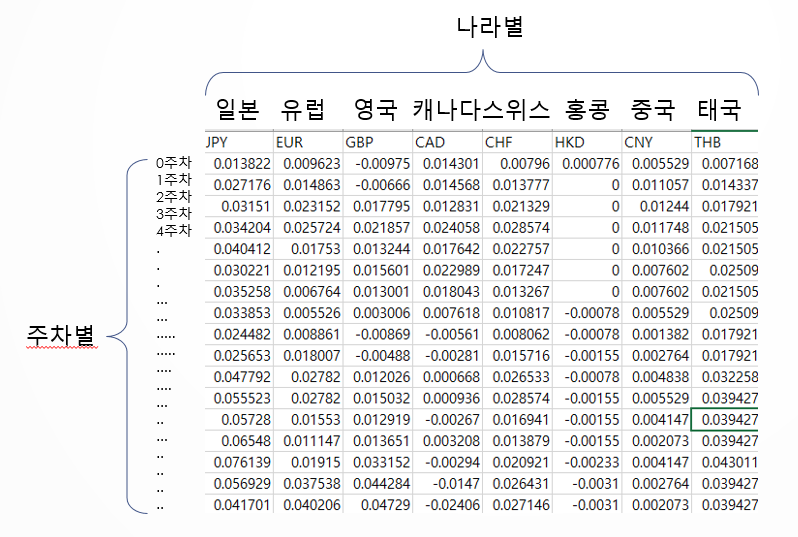


1. **특정 날짜의 모든 국가에 대한 환율 데이터**

(<https://spib.wooribank.com/pib/Dream?withyou=CMCOM0186>)

기사 텍스트 데이터에 매칭되는 환율 정보를 우리은행에서 제공하는 일별 환율 조회 시스템에서 마찬가지로 Selenium을 통해 수집하였습니다.

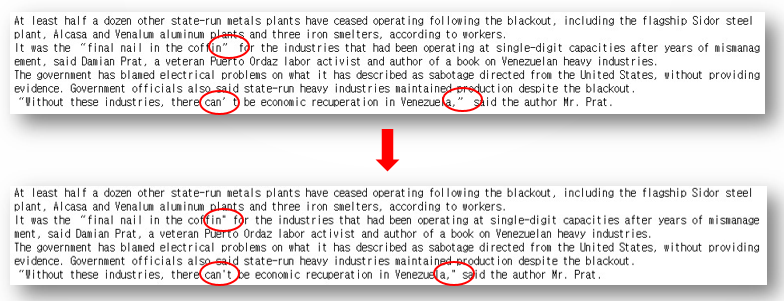
해당 국가의 전 주차와의 변화율을 기록하여 머신 러닝에서 텍스트 데이터에 대한 라벨로 사용하였습니다.

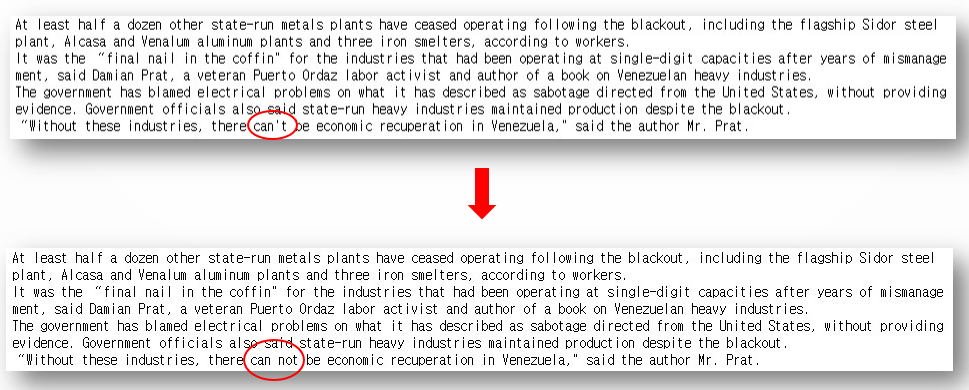


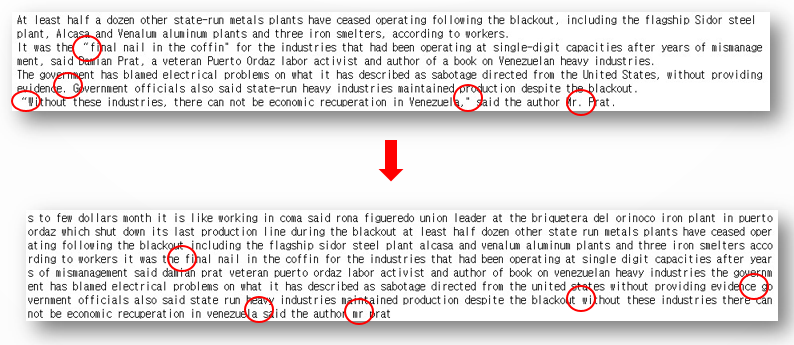
**B. NLP Processing**

**I) Pre Processing**

텍스트 데이터의 특수 문자 제거 및 대소문자를 통일하는 등 NLP 처리 전 필요한 작업들을 수행합니다. 이 과정에서 줄임표 처리 되어있는 부분을 원형으로 복원하는 작업 또한 수행하였습니다. ( ex) can’t-> can not)







**II) Stemming & Lemmatization**

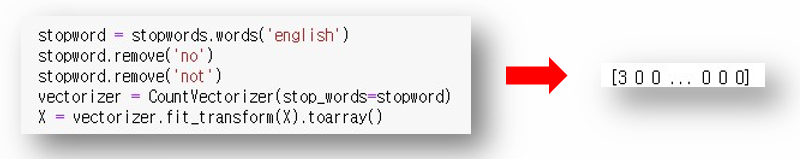
Stemming을 통하여 어간을 추출하고 Lemmatazation을 통하여 핵심부분을 추출하였습니다.



**III) Removing Stop Words & Vectorization**

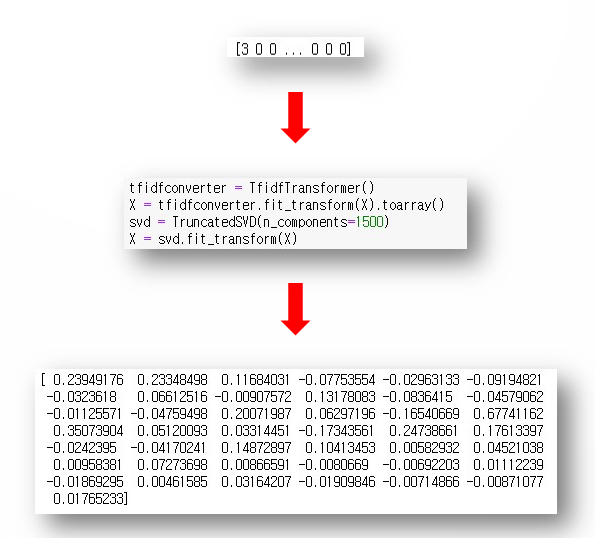
의미 추출에 방해가 될 수 있는 Stop Word를 제거하였습니다.

또한 Text 데이터를 기계 학습을 위한 실수 matrix형태로 바꾸기 위하여 Term-Doc Matrix로 변환하였습니다.



**IV) Topic Modeling**

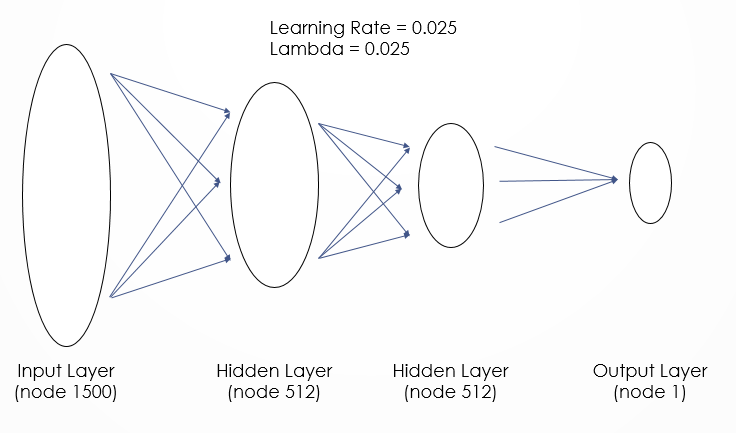
SVD를 사용한 LSA (LSI) 과정을 통하여 Feature를 상위 1500개 추출하였습니다.



**C. Machine Learning**

**I) Model Design**

Neural Network Architecture를 사용하였습니다.



위 사진과 같이, 1500개의 feature를 input으로 받아 512, 512 개의 노드를 가지는 두 개의 hidden layer를 거쳐 환율 변동의 positive/negative를 이진 분류하는 구조를 가지도록 설계하였습니다.

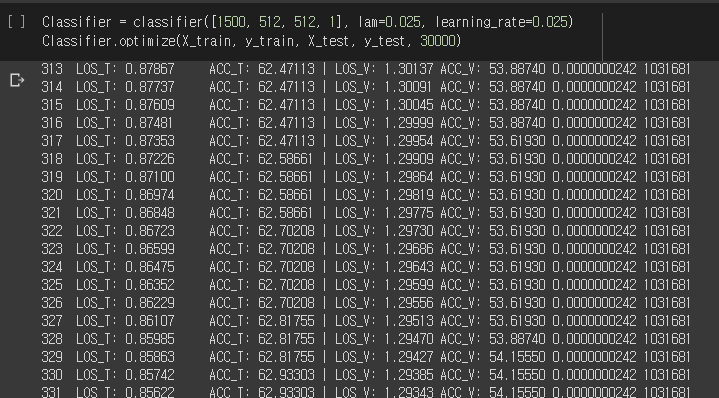
모든 Layer는 Fully-connected이며 Gradient Descnet를 이용한 BackPropagation algorithm을 통하여 다음 cost function을 minimize하도록 동작합니다.



Activate function은 Sigmoid이고, learning rate 0.025, Overfitting 완화를 위한 Regularization 계수 lambda는 0.025입니다.

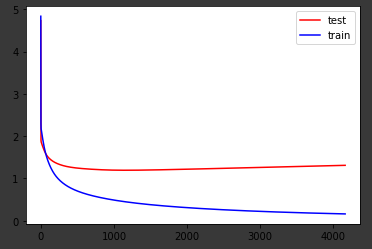
**II) Model Training**

데이터의 bias를 제거하기 위하여 positive와 negative label의 개수를 동일하게 맞춘 뒤, 동일한 분포를 갖는 두 집단, training set 70%, testing set 30%로 분리한 뒤 학습을 진행하였습니다.



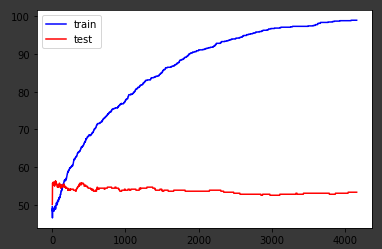
**III) Training Results**

먼저 학습 과정에서 얻은 Training, Testing Loss는 다음과 같습니다.



Epoch가 증가할 수록 Training loss는 계속 감소하나 testing loss는 최저점을 찍고 점점 training set에 overfit되어 증가하는 추세를 보입니다. Testing loss가 가장 적은 시점을 just-right으로 판단하고 해당 시점의 weight를 저장하였습니다.

다음은 Accuracy 그래프입니다.

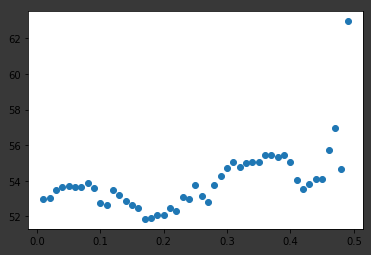


Accuracy 그래프는 위와 반대 양상으로 training accuracy가 100%를 향해 증가하는 반면 testing accuracy는 56.30%의 최대값을 찍고 꾸준히 하향하다 53.35% 근처에서 수렴하는 모습을 보입니다.

**4. Evaluation**

학습을 통해 얻은 직관으로 모의 투자해서 수익을 창출해 보자는 프로젝트 목표를 구현하기 위하여 어떠한 경우에 투자를 해야 하는지를 설정할 필요가 있었습니다.

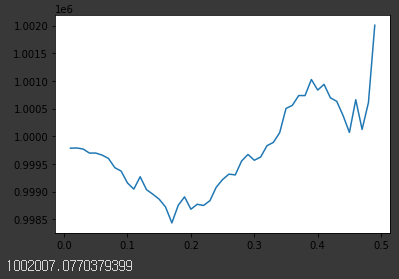
모델 설계에서 Activation function을 Sigmoid로, label을 하락일 때 0, 상승일 때 1로 설정하였기 때문에, 저희 모델의 에측 결과가 0 또는 1에 근접할 수록 명확하게 판단한 것이고, 0.5에 근접할 수록 모호한 판단일 가능성이 높다고 생각했고, 이를 그래프화 해서 검증을 시도하였습니다.

좌 그래프는 0.5 (중간값)와의 거리에 따른 Accuracy 분포를 나타냅니다.

그래프가 양의 상관관계를 따르는 것을 확인할 수 있고, 따라서 저희는 모델의 예측 결과가 명확한 지점에 100만원을 나누어 분산투자를 하도록 시뮬레이션을 설계하였습니다.

그 임계값에 따른 투자 결과는 다음과 같습니다.

대체로 위 그래프와 비슷한 모습을 보이고, 모호한 판단에 투자하면 손해를 보았고, 확신을 가졌던 것들에 투자 했을 경우는 수익을 얻을 수 있었습니다.



90% 이상의 확률로 오를 것 같다고 판단한 것들에 대해서 100만원 투자로 최소 0.01%, 최대 0.2007%의 수익을 얻을 수 있었습니다.

**5. Conclusion**

학습 결과로 보인 Accuracy 그래프에서, Training Accuracy는 100%를 향해 증가하는 반면 Testing Accuracy는 최대 56.3%만을 보이며 이후 수렴값 53%로 큰 차이를 보였습니다.

이는 저희가 선택한 ‘해당 국가에 대한 뉴스 기사 텍스트 데이터’와 ‘환율 등락폭’ 간의 관계가 명확하고 직관적이지 못하여 큰 차이를 보이는 것으로 판단됩니다.

그러나, testing accuracy가 50%보다 항상 높았던 점, 모의 투자에서 수익을 얻을 수 있었던 점으로 미루어 보아 완전히 독립적이지는 않은 것으로 생각됩니다.

따라서 저희의 공신력 있는 국제뉴스에서 특정 국가에 대해 긍정적인 평가를 보인 기사가 많았던 날에는 환율이 오르고, 반대이면 낮아질 것이라는 예측이 명확하게는 아니지만 들어맞았고, 국제 뉴스에 보도된 특정 국가에 대한 기사들이 환율과 어떠한 관련성을 보일 것이라는 가설을 자연어 처리를 통해 검증하는 것에 성공했다고 생각합니다.