딥러닝 기반 핵심 산업별 빅데이터 분석

<머신러닝&딥러닝 파일럿 프로젝트>

주 제	Otto Group Product Classification Challenge	**링 크**	https://www.kaggle.com/c/otto-group-product-classification-challenge
팀 명	하드캐리	**일 자**	2018년 11월 23일
팀 장	강호영<hoyoungk12@naver.com>	**팀 원**	이상훈, 유영재

1. 과제 개요

1) otto group

otto group

- Otto Group은 세계적으로 유명한 e-commerce 회사
- 20 개국이 넘는 곳에 자회사를 갖춤

- 매일 전 세계에 수백만 개의 제품들을 판매

2) mission

-

- 한 제품, 나라마다 다른 분류

Ex) 마스크 – 의료기기, 패션, 잡화, 미용 등---

- 회사 입장에서 의미 있는 데이터 분석을 하기가 힘듬

Ex) 의류의 전체 매출- 나라마다 패션에 속하는 제품이 다름.

전 세계 제품의 분류 기준에 무관한 분류 알고리즘을 재정의 할 필요 있음

3) 목표

-제품 특징에 따라 효과적으로 상품을 분류하는 알고리즘 완성

-평가방식에 따라 최종점수 상위 30%(1050등)내 진입

-조원 모두가 딥러닝(NN)을 익숙하게 다루도록 성장

2. 데이터 설명

<table>
<tr><th>데이터 종류</th><th colspan="2">train</th><th>Test</th></tr>
<tr><td>데이터 구성</td><td colspan="2">id (int형)
feat_1 ~ feat_93 (int형)
target (str형)</td><td>id (int형)
feat_1 ~ feat_93 (int형)</td></tr>
<tr><td rowspan="2">각 열의 특징</td><td>id</td><td colspan="2">각 제품에 대해 번호를 매김
<table>
<tr><th>id</th><th>feat_1</th><th>feat_2</th><th>feat_3</th><th>feat_4</th><th>feat_5</th></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>4</td><td>1</td><td>0</td><td>0</td><td>1</td><td>6</td></tr>
<tr><td>5</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>Feat</td><td colspan="2">93가지의 특징 중 각 제품에 해당하는 요소를 수치화한 데이터
<table>
<tr><th></th><th>id</th><th>feat_1</th><th>feat_2</th><th>feat_3</th></tr>
<tr><td>count</td><td>61878.000000</td><td>61878.00000</td><td>61878.000000</td><td>61878.000000</td></tr>
<tr><td>mean</td><td>30939.500000</td><td>0.38668</td><td>0.263066</td><td>0.901467</td></tr>
<tr><td>std</td><td>17862.784315</td><td>1.52533</td><td>1.252073</td><td>2.934818</td></tr>
<tr><td>min</td><td>1.000000</td><td>0.00000</td><td>0.000000</td><td>0.000000</td></tr>
<tr><td>25%</td><td>15470.250000</td><td>0.00000</td><td>0.000000</td><td>0.000000</td></tr>
<tr><td>50%</td><td>30939.500000</td><td>0.00000</td><td>0.000000</td><td>0.000000</td></tr>
<tr><td>75%</td><td>46408.750000</td><td>0.00000</td><td>0.000000</td><td>0.000000</td></tr>
<tr><td>max</td><td>61878.000000</td><td>61.00000</td><td>51.000000</td><td>64.000000</td></tr>
</table>
각 feature마다 최대값이 다른 것을 확인할 수 있음</td></tr>
</table>

<table>
<tr><td></td><td>target</td><td>제품 카테고리
<table><tr><th>target</th></tr><tr><td>Class_1</td></tr><tr><td>Class_1</td></tr><tr><td>Class_1</td></tr><tr><td>Class_1</td></tr><tr><td>Class_1</td></tr></table></td></tr>
<tr><td>제출 형태</td><td colspan="2">각 제품마다 분류군(9가지)에 대한 예상 확률 값을 담아 csv파일 제출
<pre>id,Class_1,Class_2,Class_3,Class_4,Class_5,(
1,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0
2,0.0,0.2,0.3,0.3,0.0,0.0,0.1,0.1,0.0
...
etc.</pre></td></tr>
<tr><td>평가 방식</td><td colspan="2">다중 클래스 로그 손실을 이용하여 평가
$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$
결론: 손실 값이 낮을수록 예상을 잘하는 모델</td></tr>
<tr><td>데이터 선택한 이유</td><td colspan="2">- 데이터 종류가 2가지로 단순함
- 대부분의 데이터가 머신 러닝을 바로 적용할 수 있는 숫자형 데이터
- 실제 기업에서 필요로 하는 실용성
- 도메인에 대한 깊은 전문지식이 필요하지 않은 용이한 접근성</td></tr>
</table>

3. 과제 수행 내역

시나리오 1

- 시작

1)기본적인 딥러닝 방법론

최적화 방법	활성화 함수	레이어	노드개수	자료형태	Learning Rate	Step
Adam Gradient	Softmax Dropout 등	1~5	256 512	Raw 정규화 등	0.1~ 1.00E-0.3	1000~ 10000

"위 설정들을 변경하며 다양한 시도
하지만 Xavier, Relu가 활용이 안됨"

작성자	방법	레이어	노드 개수	자료형태	learning rate	Acc	step	drop	점수
유영재	adam 과 gradient descent	3	512	Raw data	1.00E-0.3	**67.9%**	6000		**5.7**
강호영	기본	1	512	정규화	0.1	**68.29%**	4000		
강호영	softmax-adam	3	512	정규화	1.00E-03	**74.93%**	4000		
이상훈	softmax-gradient descent	1	512	Mean Square	0.01	**60.34%**	6000		
강호영	softmax-dropout-adam	4	256	정규화	0.01	**76.96%**	3000		
유영재	softmax adam	4	256	정규화	1.00E-03	**83.24**	4000		**15**
이상훈	sigmoid_softmax_softmax_adam	4	256	Raw data	1.00E-03	**91%**	6000		**7.2**

2)결론

- 다양한 시도, 만족스럽지 못한 결과, 최고점수 **5.7**
- 의미 없는 전처리 Data Set
- 활성화 함수로의 Soft Max의 부작용
- Relu를 활용위한 Raw 데이터 가공 필요성

Relu, Xavier로 사용해 봐야 한다.
너무 양극단(0,max)으로 치우쳐서 안되는 걸까?
데이터를 좀더 평준화 해보자

시나리오 2

- 극복과 새로운 과제

1) Log를 사용한 DataSet 재구성

"Log(x+1) 을 통해 전체적인 데이터 평준화"

"Activation Function 으로서
Relu와 Xavier 활용가능"

작성자	방법	레이어	노드 개수	자료형태	learning rate	Acc	step	drop	점수
강호영	relu, xavier, drop out	3		로그, sum, var	0.01	84.17%	5000	0.7	
유영재	relu, xavier,	3	512	로그	1.00E-05	66.72%	5000		
				⋮					
유영재	relu, xavier	3	256	로그	1.00E-03	97.79%	12000		1.9
강호영	relu, xavier,	3	256	로그, sum	0.001	91.27%	10000		4.7
강호영	relu, xavier	2	256	로그	0.005	86.44%	5000		
유영재	**relu, xavier**	**3**	**256**	**로그**	**1.00E-03**	**95.71%**	**6000**		**1.41**
유영재	**relu, xavier, drop out**	**3**	**256**	**로그**	**1.00E-03**	**90.26%**	**10000**	**0.75**	**0.58**
강호영	relu, xavier	2	256	로그	0.01	90.26	5000		4.3
유영재	**relu, xavier**	**3**	**256**	**로그**	**1.00E-03**	**88.71%**	**6000**		**0.57**
강호영	relu, xavier	2	512	로그	0.01	91.9	5000		3.9

2)결론

- 확실히 낮아진 점수, **0.57**
- 확실히 효과 있었던 데이터 평준화
- 비교적 적은 데이터셋(약 68,000)으로 과적합의 문제
- 정확도 90%↑ , Keggle 점수 ↓

적은 data Set에 의한 '과적합'을 극복해야 한다.
기존의 방법 이상의 것을 찾아보자.

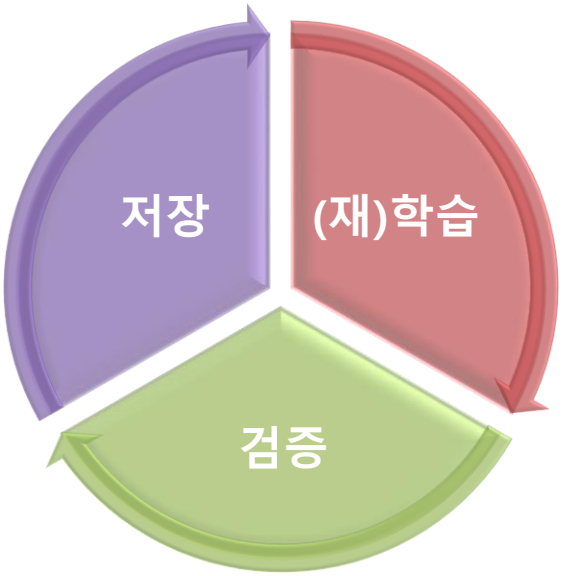
시나리오 3

- 희망

1) DataSet의 새로운 활용

"DataSet을 Train과 Validation 로 분리하여
자체적으로 검증, 개선, 검증, 개선"

Train Data(100%)	
Train(90%)	Validation(10%)



"Data Set을 다양하게 지속적 활용"

2)다양한 시도

1) Train Set과 Validation Set을 학습을 반복할 때마다 재설정

Train Data(100%)	
Train(90%)	Validation(10%)

재분할

Train Data(100%)	
Train(90%)	Validation(10%)

"결과:Accuracy가 일정부분 좋아지긴 함

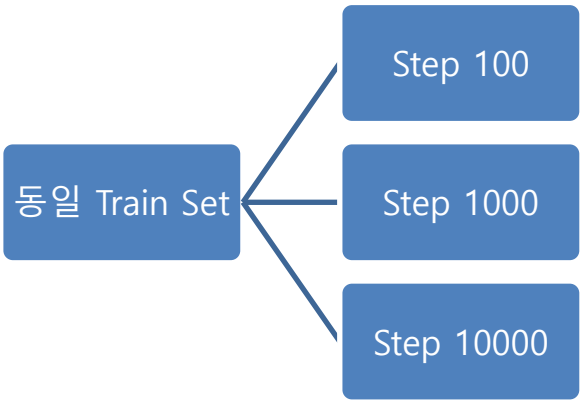
학습한 데이터의 일부가 Validation Set으로 들어갔기 때문에
효과를 장담할 수 없음”

2)Validation Set보존, Train Set에서 지속적으로 학습하여 최적 step 찾음

Train Data(100%)	
Train(90%) -Step100	Validation(10%)-고정

⋮

Train Data(100%)	
Train(90%) -Step10000	Validation(10%)-고정



“Validation과 비교하여
최적의 가중치를 갖는
Step 저장”

“과적합을 방지하여
0.55대로 진입하는 이전보다
더 나은 결과로 도출“

3) Validation Set은 보존, Train Set에서 지속적으로 일정 비율을 비복원추출

Train Data(100%))		
90%		Validation(10%)고정
Train(20%)	80%	Validation(10%)

고정된 Train 데이터에서 일정 부분을 활용하며 학습

Train Data(100%)			
90%			Validation(10%)
쓰레기(20%)	Train(20%)	60%	Validation(10%)

"점수 0.544로 최적의 점수 찾음"

4. 결과 보고

목표1) 상품을 특징에 따라 효과적으로 분류하는 알고리즘 완성
결과) 예측율 90%이상을 달성하는 알고리즘 생성

목표2) 평가방식에 따라 최종점수 상위 30%(1050등)내 진입
결과) 최종 0.546점 중후반대로 약 1700등 달성

목표3) 조원 모두가 딥러닝(NN)을 익숙하게 다루도록 성장
결과) NN기법을 활용할 수 있게 되었고, 변수의 조작에 의해 결과값이 변화하는 것을 이해할 수 있는 시간이었다. 또한 어떤 기법을 어디에, 어떻게 적용하며 보다 나은 모델을 생성하기 위해 어떤 노력을 해야 하는지 탐색할 수 있는 시간이었다.
알고리즘에서 활용하는 각종 함수의 활용방법을 대략 알 수 있게 되었다.
알고리즘에 대한 명확한 이해 없이는 특정 수준을 뛰어넘는 데는 한계가 있다.
분석에 들어가기 전, 데이터 자체에 대한 분석이 선행되어야 함을 느꼈다.
제출형식을 맞추는 것이 어려웠고, 실제 프로젝트에서 클라이언트가 원하는 포맷에 맞추는 것이 중요하다는 것을 알게 되었다.
단순히 우연에 기댄 결과가 아니라 실제 과학적인 기법으로서 활용되기 위해서 보다 많은 영역의 공부가 필요한 것을 알게 되었다.