4.4. Neural Network

인공 신경망(NN)은 최근 재조명받고 있는 머신러닝 기법 중 하나로, 뇌세포의 활동을 모방하여 데이터를 처리한다. 입력 뉴런의 상태 가 주어질 때, 현재 뉴런의 상태는 다음과 같이 계산된다.

여기서 는 각각 weight, bias, 그리고 activation function을 의미한다. Weight와 bias는 처음 무작위로 설정된 후, 학습 과정에서 back-propagation 식을 직접 계산하거나 최적화 알고리즘을 수행함으로써 개선되어 나간다. 이러한 뉴런이 모여 전체 모델을 구성하는데, 편의를 위해 뉴런들의 layer가 주로 사용된다.

최근 연구자들은 이미지 인식 등을 위한 Convolutional NN과 순차 데이터 분석을 위한 Recurrent NN 등 여러 변형된 layer 구성을 활용하고 있다. 그러나 이러한 모델은 각각 입력 layer의 공간적, 시간적 locality를 가정하고 있으므로, 그러한 연관이 없는 여러 feature들을 분석하는 우리 프로젝트에는 적절하지 않다. 이 프로젝트는 layer간의 모든 뉴런이 연결된 dense network 구성을 사용하였다.

Tensorflow와 Theano 등의 몇몇 파이썬 라이브러리가 NN 구현에 초점이 맞추어져 있다. 이 중 GPU를 활용하는 Tensorflow 버전을 이번 프로젝트에 사용하였다.

4.4.1. 데이터 전처리

4.4.1.1. 분류 데이터의 처리

각각의 뉴런은 활성 상태를 숫자로 나타내며 모든 계산이 수치로 이루어지므로, 분류 데이터를 다루기에 NN은 비교적 부적합하다. NN을 nominal data에 사용하기 위해서는, 다음의 과정을 거쳐 데이터를 숫자로 변환해 주어야 한다.

* 성별과 같은 Binary Data는 (male=1, female=0)처럼 바로 변환할 수 있다. 이 경우 Null 값은 단순히 0.5로 매핑하거나, 성별이 알려진 사람 중 남성의 비율을 사용해도 된다.
* Dummy variable

Dummy Variable(Indicator variable)은 특정 조건을 만족하는지의 여부를 0과 1로 나타낸 값이다. 분류 데이터를 Dummy Variable로 나타내기 위해서는 각각의 분류마다 하나씩의 열을 만들어서, 값이 현재 열이 나타내는 분류와 같으면 1, 다르면 0을 부여한다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Genre |  | ID | Genre\_Pop | Genre\_Rock | Genre\_Electronic |
| 1 | Pop |  | 1 | 1 | 0 | 0 |
| 2 | Rock | → | 2 | 0 | 1 | 0 |
| 3 | Pop|Electronic |  | 3 | 1 | 0 | 1 |
| 4 | (null) |  | 4 | 0 | 0 | 0 |

Null(ID 4)과 다중 값 처리(ID 3)의 세부사항은 방법마다 다를 수 있다. Null이나 다중 값이 없어서 각 행에 하나씩의 1이 존재한다면, 이를 Complete Disjunctive Table(CDT)라 한다.

Dummy Variable은 분류 데이터를 인코딩하는 가장 간단한 방법 중 하나이며, 다른 분류 데이터 처리 방법의 기반이 되기도 한다. 서로 다른 분류의 수만큼의 열이 생성되므로 분류의 수가 많은 경우 Feature의 수가 그만큼 많아질 수 있으며, 이를 피하기 위해 상위 n개의 분류에 대해서만 열을 생성하는 등의 방법을 사용할 수 있다.

* Multiple Correspondence Analysis(MCA)

분류 데이터가 여러 개 있는 경우, MCA는 데이터들을 실수로 변환한다. 내부적으로는 분류 데이터에 대한 CDT를 생성한 후 변환을 거쳐 PCA에 넣는 방식으로 계산한다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Source1 | Source2 | Source3 |  | ID | F1 | F2 |
| 1 | My Library | My Profile | My Profile more |  | 1 | 0.88 | 0.79 |
| 2 | Radio | Radio | Listen to | → | 2 | 0.22 | 0.17 |
| 3 | Playlist | Others Profile | Purchase |  | 3 | 0.65 | 0.23 |
| … | … | … | … |  | … | … | … |

MCA의 결과를 좌표로 시각화하면, 데이터의 구조에 대해 추가 정보를 얻을 수도 있다.

분석 과정에서 청취 경로와 장르를 제외한 다른 분류 데이터(city, registered\_via, language, and artists)는 NN 모델에서 입력값으로 사용되지 않았다. 이는 데이터 분석 및 다른 모델의 수행 과정에서 이들을 사용했을 때 처리 과정에 비해 정확도의 향상이 높지 않은 수준이라 추정했기 때문이다. 상위 32개 장르와 청취 경로는 Dummy Variable로 변환 후 모델의 입력으로 사용했다.

4.4.1.2. 데이터 변환 및 정규화

“분류 데이터의 인코딩뿐만 아니라, x값들이 규모가 조정되거나 정규화되어 비슷한 크기를 가질 때 NN은 경험적으로 더욱 효과적이다.”[1] - James McCaffrey, Visual Studio Magazine에서

우리는 입력값을 비슷한 크기(예: 0과 1 사이)로 조정하고 모델의 성능을 향상시키기 위해 데이터의 특성을 반영하고 skewness를 완화시키는 비선형 변환을 사용하였다.

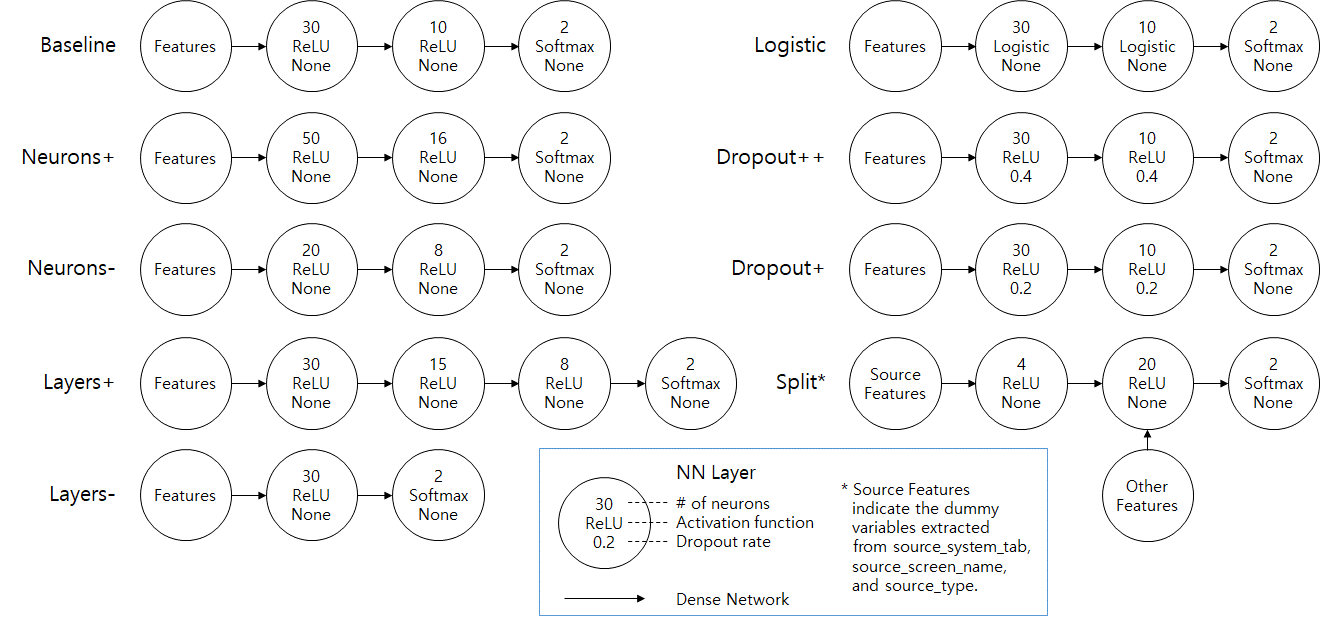
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Normalizer**는 비선형 함수 func를 적용한 뒤, min\_value를 0으로, max\_value를 1로 맞추어 주는 변환 함수를 생성한다. 유효한 변환을 하기 위해서 func는 증가함수여야 한다. | | | | |
| def trimmer(x):  return 0 if x<0 else 1 if x>1 else x  def normalizer(min\_value, max\_value, func, positive=False, unknown=0.5):  return lambda x: unknown if math.isnan(float(x)) or (positive and x<=0) \  else trimmer((func(float(x))-func(max\_value))/(func(min\_value)-func(max\_value))) | | | | |
| Features | func | min\_value | max\_value | Notes |
| bd (age, yr) | Sqrt | 9 | 49 | Adjusting mode(25) into 0.5 |
| song\_length (ms) | Log | 32768 | 1048576 |  |
| member\_duration (s) | Log | 1 | 444873600 | reg\_init\_time – expiration\_date |

|  |  |  |  |
| --- | --- | --- | --- |
| **Decayer**는 현재 시점을 1로 할 때, 지수 함수에 따라 해당 시점의 감소한 값을 계산하는 함수를 생성한다. 이 때 인자로 주어진 반감기가 적용된다. | | | |
| def decayer(current, halflife, unknown=0.5):  return lambda x: unknown if math.isnan(float(x)) else 1 if float(x)>current \  else math.pow(0.5,(current-float(x))/halflife) | | | |
| Features | current | halflife | Notes |
| song\_year (yr) | 2017 | 10 | Data from isrc |
| registration\_init\_time (s) | 1488258000 | 100000000 | current: Feb 28, 2017 |

예를 들어, 최신 곡은 과거의 곡들에 비해 청취 빈도가 더 높을 것이며, 노래의 인기는 보통 지수적 감소 패턴을 보인다. 따라서 곡의 발매 연도를 Decayer(여기에서는 반감기 10년을 사용함)로 전처리할 수 있으며, 이렇게 하면 최신 곡들의 수치가 퍼지는 대신 과거 곡들의 수치가 비교적 비슷한 값으로 바뀌어 최신 곡들에 대해 더 민감한 분석을 할 수 있게 된다.

4.4.2. Network 디자인

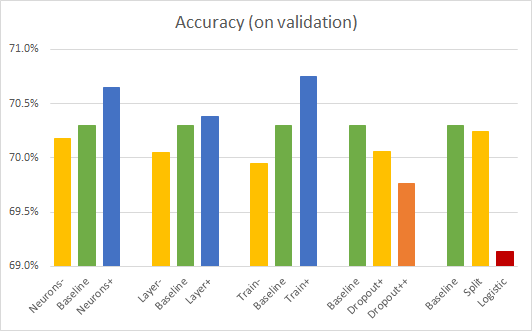
Network 조정에 대한 효과를 알아보기 위해, network를 여러 다른 방법으로 구성하였다. 기준이 되는 network(Baseline)는 dropout이 없는 3-layer network로, ReLU를 activation function으로 사용한다. 뉴런의 수, layer의 수, dropout 비율, activation function 등을 변화시켜 정확도를 확인했으며, input을 분할해 보기도 하였다.



Network의 형태와는 별도로, train 반복 횟수도 조정하였다. Baseline의 경우 50000 step을 사용했으며, 이를 30000(Train-) 및 100000(Train+)으로 조정한 모델도 분석해 보았다.

4.4.3. 결과

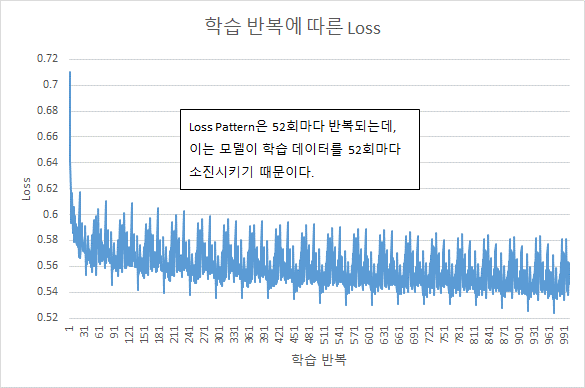
NN으로 분류 데이터를 처리하기 어렵다는 점에서 예상했듯이, 전체적인 validation 결과는 약 70% 정도의 정확도를 보여 LightGBM에 비해 부정확한 것으로 나타났다.



다만 뉴런과 layer의 수, Train 기간 등에 따른 정확도 변화의 경향은 확인할 수 있었다. 뉴런이나 layer의 추가, 긴 학습이 정확도를 향상시킴과 달리, Dropout 기법은 그에 방해가 되는 것으로 나타났다. 데이터를 청취 경로와 그 외의 데이터로 분리해서 처리한 결과(Split), 24개의 뉴런을 사용했음에도 불구하고 Neurons-(28개)와 Layers-(30개)보다 좋은 결과를 얻을 수 있었다.

Kaggle에 업로드한 결과 점수는 validation에서 얻은 정확도보다 낮은 것이 일반적이며, 그 차이는 대략적으로 모델이 데이터를 얼마나 overfitting하고 있나를 나타낸다. Neurons+ 모델로부터 얻은 결과를 업로드하여 62.172%의 점수를 얻었으며, validation 정확도 70.652%와 비교했을 때 그 차이는 약 8.5%p 정도이다.

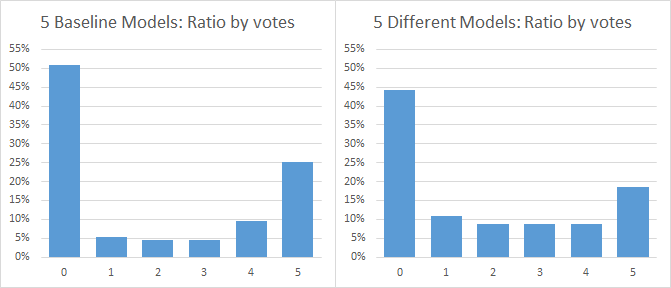
학습을 반복하면서 loss의 변화를 관찰하면 다음 그림(Train+ 모델)과 같은 loss의 감소를 확인할 수 있다.



4.4.4. Voting을 통한 여러 NN 결과의 통합

Voting은 가장 간단한 앙상블 모델 중 하나로, 여러 모델의 결과를 모아 그 중 가장 많은 모델이 나타내는 결과를 선택하는 방법이다. Voting 과정은 사용된 모델들이 동일한 작업을 하지만 모델의 인자나 Network의 구조가 달라 서로 다른 결과로 수렴하는 상황에서 효과적으로 사용될 수 있다.[2]

동일한 NN 모델(Baseline)을 5회 수행한 결과와 서로 다른 모델(Baseline 2회, Neurons+, Layers+, Split)을 수행한 결과를 이용하여 Voting을 적용하였다.



그래프에서 0표와 5표의 비율이 높은 것에서 볼 때, 모델의 수렴 양상이 서로 다른 모델에서조차 비슷하게 나타나 이 프로젝트에서는 voting의 효과가 크지 않은 것으로 확인되었다. 정확도는 유효표의 평균 수치를 기준으로 했을 때, Kaggle 업로드에서 61.262%(Baseline 5회), 61.983%(서로 다른 모델 5회) 정도였다. 이는 8.5%p 정도의 overfitting rate를 고려했을 때 Baseline 단독 수행과 크게 다르지 않은 것으로 보인다.

References

[1] James McCaffrey. (2014). How to Standardize Data for Neural Networks. Retrieved from <https://visualstudiomagazine.com/articles/2014/01/01/how-to-standardize-data-for-neural-networks.aspx>

[2] E. Alpaydin. (1992). Multiple neural networks and weighted voting. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on Pattern Recognition.