자연어 처리 – AI 전문가 양성(기본) Day 2

KLUE Dataset – supervised multi-classification

INGLab

실습 소개

- **실습 목표**
  - KLUE dataset을 알아보고 지도학습방식으로 다중 분류 해보기

- **실습 내용**
  - KLUE 데이터셋 확인
  - 데이터 전처리
  - Random Forest 알고리즘을 사용하여 데이터셋 분류 및 성능 확인
  - DNN을 사용하여 데이터셋 분류 및 성능 확인

KLUE 한국어 데이터셋

- https://github.com/KLUE-benchmark/KLUE
- https://huggingface.co/datasets/klue

KLUE MRC dataset description

Jihyung Moon edited this page on 27 May · 1 revision

Name	Description
title	title of the context
source	document source of the context
news_category	category if the context is news
paragraphs	a list of contexts and question-and-answer pairs
context	target passage text
qas	a list of question-and-answer pairs
guid	a primary key of each question and answer pair
question	question text
answers	a list of answers
text	answer text
answer_start	start position of the answer
question_type	type of question
is_impossible	flag for unanswerable
plausible_answers	fake answers for type 3 question

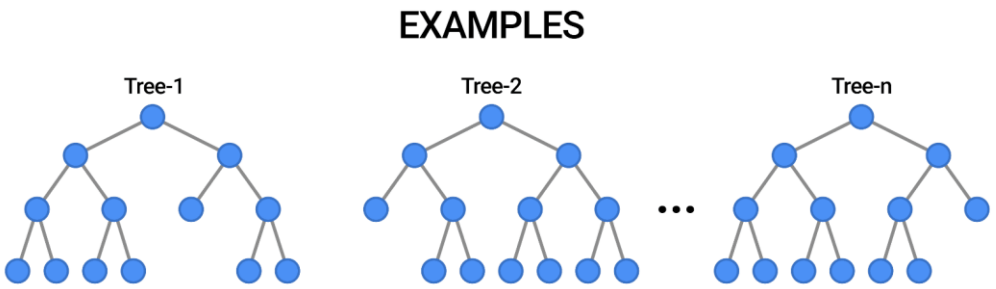
mrc

An example of 'train' looks as follows.

```
{'answers': {'answer_start': [478, 478], 'text': ['한 달가량', '한 달']},
 'context': '올여름 장마가 17일 제주도에서 시작됐다. 서울 등 중부지방은 예년보다
 'guid': 'klue-mrc-v1_train_12759',
 'is_impossible': False,
 'news_category': '종합',
 'question': '북태평양 기단과 오호츠크해 기단이 만나 국내에 머무르는 기간은?',
 'question_type': 1,
 'source': 'hankyung',
 'title': '제주도 장마 시작 … 중부는 이달 말부터'}
```

Random Forest

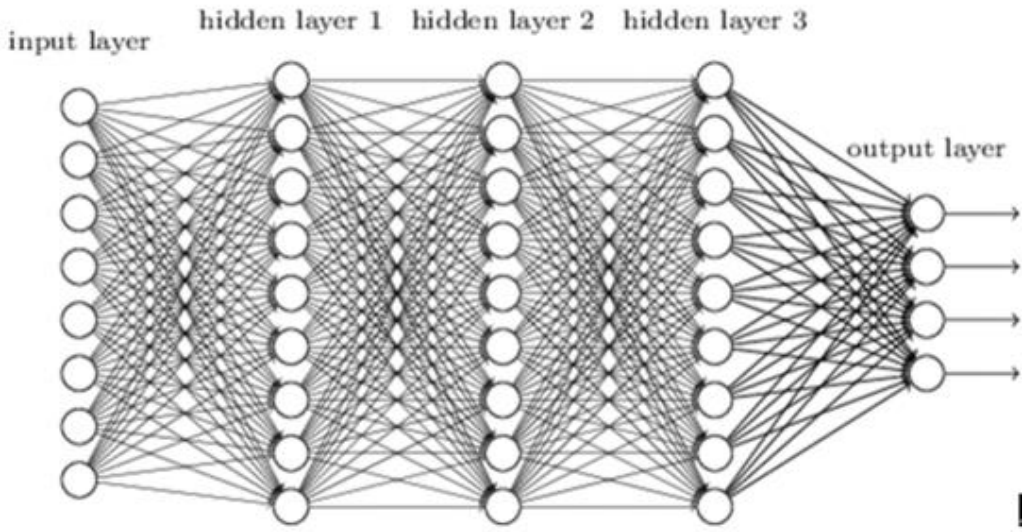
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html



n_estimators	Random forest의 tree 개수 (기본값 = 100)
criterion	트리의 성능을 평가할 지표 좋은 트리를 선택하는 기준으로 삼는다 지니 불순도(gini)와 엔트로피(entropy) 중에 선택 가능. (기본값 = "gini")
max_depth	트리의 최대 깊이 모델이 너무 많은 트리 노드로 과적합되는 것을 방지할 수 있음 (기본값 = None)
⋮	⋮

DNN (MLP Classifier)

- https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html



hidden_layer_sizes	은닉층의 차원 크기 입력과 출력 단을 제외한 'n_layers – 2' 길이의 정수 튜플 (기본값 = (100,))
activation	활성화 함수 Identity (linear) / logistic (sigmoid) / tanh / relu 중 하나 (기본값 = relu)
batch_size	한 번 학습에 사용하는 데이터의 개수 (배치 크기) (기본값 = auto)
max_iter	최대 학습 횟수 (epoch) (기본값 = 200)
shuffle	데이터를 매 epoch 마다 섞을지 결정 (기본값 = True)
⋮	⋮

Evaluation

- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

- **Metrics**

	T	F
T	True Positive (TP)	False Negative (FN)
F	False Positive (FP)	True Negative (TN)

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

Evaluation

- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

- **Metrics**

	T	F
T	True Positive (TP)	False Negative (FN)
F	False Positive (FP)	True Negative (TN)

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \qquad Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

- Macro-f1: 각 class마다 따로 f1 score를 구한 다음 class 개수만큼 평균 계산
- Micro-f1: 각 class의 confusion matrix를 더해 하나로 합친 후 f1 score 계산

COLAB - practice

KLUE 데이터셋 받기

```
!pip install datasets
```

```
import datasets
import pandas as pd
```

```
klue_dataset = datasets.load_dataset("klue", "mrc").remove_columns(['answers', 'guid', 'is_impossible', 'question', 'question_type'
, 'source', 'title'])
train_dataset = pd.DataFrame(klue_dataset['train'])
val_dataset = pd.DataFrame(klue_dataset['validation'])
```

Train & Validation 데이터

중복되는 데이터 지우기

```
train_dataset = train_dataset.drop_duplicates(subset=['context'])
val_dataset = val_dataset.drop_duplicates(subset=['context'])
```

카테고리 선정

```
using_categories = set(train_dataset.groupby('news_category').count().sort_values(by='context', ascending=False).iloc[1:6].index)
```

```
train_dataset = train_dataset[train_dataset['news_category'].isin(using_categories)]
val_dataset = val_dataset[val_dataset['news_category'].isin(using_categories)]
```

COLAB - practice

- 전처리

영어, 한국어, 숫자만 남기기

```
import re
```

```
processed_train_dataset = [re.sub('[^가-힣a-zA-Z0-9\s]', '', x) for x in train_dataset['context']]
processed_val_dataset = [re.sub('[^가-힣a-zA-Z0-9\s]', '', x) for x in val_dataset['context']]
```

TF-IDF Vectorizer

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vectorizer = TfidfVectorizer()
train_x_data = vectorizer.fit_transform(processed_train_dataset)
train_y_data = train_dataset['news_category']
```

```
val_x_data = vectorizer.transform(processed_val_dataset)
val_y_data = val_dataset['news_category']
```

+ KoNLPy tokenizer

```
!pip install konlpy
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
from konlpy.tag import Komoran
from tqdm import tqdm_notebook
```

```
tokenizer = Komoran()
vectorizer = TfidfVectorizer(tokenizer=lambda x: tokenizer.morphs(x))
```

COLAB - practice

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
```

```
random_forest = RandomForestClassifier()
random_forest.fit(train_x_data, train_y_data)
```

```
random_forest_preds = random_forest.predict(val_x_data)
```

Evaluation

```
from sklearn.metrics import f1_score, accuracy_score, confusion_matrix
```

```
macro_f1 = f1_score(val_y_data, random_forest_preds, average='macro')
micro_f1 = f1_score(val_y_data, random_forest_preds, average='micro')
acc = accuracy_score(val_y_data, random_forest_preds)
```

```
confusion_matrix(val_y_data, random_forest_preds)
```

MLP Classifier (DNN)

```
from sklearn.neural_network import MLPClassifier
```

```
dnn = MLPClassifier(batch_size=64, verbose=True, max_iter=10).fit(train_x_data, train_y_data)
dnn_preds = dnn.predict(val_x_data)
```

과제 소개

- **과제 목표**
  - KLUE-MRC 데이터셋을 가지고 뉴스 카테고리 분류 성능 높이기

- **과제 조건**
  - 뉴스 카테고리 전부 사용
  - 제출: 코드를 작성한 colab을 julia981028@gmail.com 으로 제출
  - 평가 방식: macro-f1 score을 사용하여 leader board (마지막에 macro f1-score 출력)
    - 작성한 colab을 순서대로 실행하였을 때 오류 없이 돌아가야 함
    - Randomness를 고려하여 총 5번을 돌려본 후 그 평균값 고려

- **과제 HINT**
  - 다양한 방식으로 데이터 전처리 시도해보기!
  - 어떤 알고리즘이나 library를 사용하던 제한 없음!