機械学習を用いたフェイクニュース判定モデルの構築

明治大学総合数理学部 現象数理学科4年

背景

近年、SNSの普及により誰もが情報を発信できる世の中になった。
あらゆる情報が存在する中で、情報の真偽を判断するのは
困難である。そこで、多くの分野で活用されている
機械学習を用いてフェイクニュースを判定する。

目標

**海外のデータと日本語のデータを用いて
フェイクニュースを判定するモデルを構築する**

先行研究(A.Agarwal, SN Computer Science, 2020)では、CNNとLSTMを組み合わせて計算コストの少ないモデルを作成。更に、別の先行研究(H.Ali, All Your Fake Detector~, 2021)では、RNNやCNN等の複数のモデルで性能を比較。本研究では、海外のデータと日本のデータを用いてCNNとLSTMを組み合わせたモデルの構築と複数の手法のモデルを作成し精度を比較する。

データのSummary

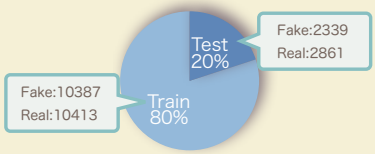
今回は、海外のニュース記事と日本のニュース記事の2種類のデータセットを使用。ラベルはフェイクニュース(以下Fake)を1、本物のニュース(以下Real)を0とする。日本のデータでは、PT-2日本語版によって生成されたフェイクニュースを使用。

文章の自動生成モデル

海外データ

id、タイトル、著者、テキスト、ラベル

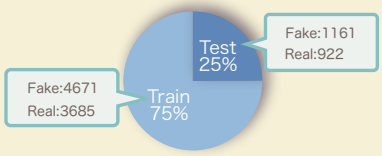
訓練データ（以下Train）が20800、テストデータ（以下Test）が5200である。



日本データ

id、テキスト、ラベル

訓練データが6268、テストデータが2082である。



先行研究

先行研究(A.Agarwal, SN Computer Science, 2020)では、CNNとLSTMを組み合わせたモデルを作成した。CNNは画像等の特徴抽出に使用され、LSTMは時系列データに用いられる。CNNで特徴抽出を行ってからLSTMを用いると計算コストを下げることができる。全結合層の活性化関数は1~3層目にReLU 4層目にsoftmaxが使用され、出力数は1層目から順に256, 128, 64, 2で設定された。
更に、Dropout=0.2、Batch sizeは15、Epochsは128、OptimizerはAdam、
損失関数はBinary Cross Entropyで設定された。

モデルの概要

精度を評価する指標は、Accuracy、Precision、Recallの三つを用いた。
また、今回はFakeをPositive、TrueをNegativeとした。混同行列を以下に示す。

		モデルの予測	
		Positive	Negative
実際の値	Positive	TP	FN
	Negative	FP	TN

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

▶ データを正しく分類できた割合

$$\text{Precision} = \frac{TP}{TP+FP}$$

▶ Positiveと予測されたデータの中で実際にPositiveである割合

$$\text{Recall} = \frac{TP}{TP+FN}$$

▶ 実際にPositiveであるデータの中でPositiveと予測された割合

モデルの概要

今回、以下の6種類のモデルを作成した。

model1. 先行研究モデル：先行研究と全く同じ手法のモデル

model1'. 先行研究と同じ手法のモデル：先行研究と同じ手法だが、全結合層の4層目をsigmoidに変更

model2. CNNモデル：model1'からLSTM層を削除したCNN単体のモデル

model3. LSTMモデル：model1'からCNN層を削除したLSTM単体のモデル

model4. GRUモデル：model1'からCNN層を削除し、LSTM層をGRU層に変更したGRU単体のモデル

model5. CNN&GRUモデル：model1'のLSTM層をGRU層に変更したモデル

更に、各モデルの細かい設定を変更したモデルを作成した。変更点を以下に示す。

全結合層(出力数):256,128,64,1/128,64,1/64,1
Dropout層の有無:○(0.2)/×
Batch normalizationの有無:○/×
kernel initializer:×(デフォルトのglorot uniform)/he uniform/he normal
RNN層のrecurrent activation:hard sigmoid/sigmoid
CNN層のフィルタ数:32/128

結果

各モデルの中で最も精度が良かったモデルを以下に示す。

海外モデル

モデル	Accuracy	Precision	Recall
先行研究	94.7	92.5	97.4
model1	62.9	67.1	63.8
model1'	74.0	78.2	73.1
model2	77.4	82.4	75.0
model3	81.0	83.3	81.9
model4	79.0	85.6	74.4
model5	78.1	79.1	81.6

精度↑↑
精度↓↓
精度↑

日本モデル

モデル	Accuracy	Precision	Recall
model1	50.1	56.2	54.2
model1'	51.5	55.8	62.8
model2	51.1	55.5	61.8
model3	50.6	55.3	60.2
model4	54.2	58.8	60.0

海外モデル

- 先行研究と同じ手法を用いたが、大幅に精度が下がった。(model1)
- 全結合層の活性化関数をsigmoidにすることで約10%精度が上がった。(model1')
- LSTM層をGRU層に変更することで若干精度が上がった。(model5)
- CNNを用いたモデル(model1')と用いていないモデル(model3)を比較すると、CNNを用いることで1Epoch当たりの計算コストを半減させることができた。

```
Epoch 1/15
163/163 [==============================] - 87s 486ms/step
Epoch 2/15
163/163 [==============================] - 79s 483ms/step
Epoch 3/15
163/163 [==============================] - 79s 484ms/step
```

```
Epoch 1/15
163/163 [==============================] - 197s 1s/step
Epoch 2/15
163/163 [==============================] - 191s 1s/step
Epoch 3/15
163/163 [==============================] - 194s 1s/step
```

日本モデル

- どのモデルも50~60%の精度になり、あまり良いモデルとは言えない。
- 細かい設定や手法を変更しても精度にあまり差がなかった。

また、model4において細かい設定を変更した際の精度を以下の表に示す。

3層がBest

モデル	Accuracy	Precision	Recall	全結合層	Dropout	Batch normalization	kernel initializer	recurrent activation
model4-1	74.6	75.0	80.6	64,1	×	×	×	hard sigmoid
model4-2	77.7	79.5	80.2	128,64,1	×	×	×	hard sigmoid
mode4-3	75.9	75.2	83.8	256,128,64,1	×	×	×	hard sigmoid
model4-4	75.7	80.1	74.2	128,64,1	○	×	×	hard sigmoid
model4-5	79.0	85.6	74.4	128,64,1	○	○	×	hard sigmoid
model4-6	71.1	80.8	62.1	128,64,1	○	○	he uniform	hard sigmoid
model4-7	74.6	75.1	80.6	128,64,1	○	○	he normal	hard sigmoid
model4-8	71.4	76.1	70.0	128,64,1	○	○	×	sigmoid

あった方が良い
デフォルトのままで良い

まとめ

全結合層の活性化関数はsigmoidの方が精度が高くなり、GRUのrecurrent activationはhard sigmoidの方が高くなった(model4の場合)。LSTMよりもGRUを用いた方が精度が高くなった。
CNNとRNNを併用することで計算コストが半減した。
細かい設定を変更することで精度に影響が出た。

結論

**海外のデータについては精度の良いモデルができたが
日本語データについては更なる改善が必要**

先行研究と同じ手法を用いたが、同じような結果を得られなかった。これは、前処理を見直すことで多少改善するのではないかと思う。海外のデータを用いた場合と比較して日本語データを用いた場合に精度が下がった理由も同様であると推測できる。更に、先行研究と同様にCNNを用いて計算コストを削減できた。全結合層の数やkernel initializerを変更することで精度に影響が出ることや、LSTMと同じRNNであるGRUを用いることによって精度が上がることもわかった。