Part1

Q1.

$$\begin{cases} w_j = w_j - \eta \frac{\partial L}{\partial w_j} \\ b = b - \eta \frac{\partial L}{\partial b} \\ \hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} \\ z = w_1 x_1 + w_2 x_2 + b \end{cases} \Rightarrow \begin{cases} \frac{\partial L}{\partial w_1} = (\hat{y}-y)x_1 \\ \frac{\partial L}{\partial w_2} = (\hat{y}-y)x_2 \\ \frac{\partial L}{\partial b} = \hat{y}-y \end{cases}$$

initial weighe: $w_1=0$, $w_2=0$, $b=0$

first point: $(x_1=2, x_2=1, y=1)$

$z=0$

$\hat{y} = \frac{1}{1+e^0} = 0.5$

$\frac{\partial L}{\partial w_1} = (0.5-1)\times 2 = -1$

$\frac{\partial L}{\partial w_2} = (0.5-1)\times 1 = -0.5$

$\frac{\partial L}{\partial b} = (0.5-1) = -0.5$

$\Rightarrow w_1 = 0.2$, $w_2 = 0.1$, $b = 0.1$

second point: $(x_1=1, x_2=3, y=0)$

$z=0.6$

$\hat{y} = \frac{1}{1-e^{-0.6}} = 0.645$

$\frac{\partial L}{\partial w_1} = (0.645-0)\times 1 = 0.645$

$\frac{\partial L}{\partial w_2} = (0.645-0)\times 3 = 1.935$

$\frac{\partial L}{\partial b} = (0.645-0) = 0.645$

$\Rightarrow w_1 = 0.2-(0.2\times 0.645) = 0.071$

$w_2 = 0.1-(0.2\times 1.935) = -0.287$

$b = 0.1-(0.2\times 0.645) = -0.029$

third point: $(x_1=0, x_2=4, y=0)$

$z=-1.177$

$\hat{y} = 0.235$

$\frac{\partial L}{\partial w_1} = (0.235-0)\times 0 = 0$

$\frac{\partial L}{\partial w_2} = (0.235-0)\times 4 = 0.94$

$\frac{\partial L}{\partial b} = (0.235-0) = 0.235$

$w_1 = 0.071 - 0 = 0.071$

$\Rightarrow w_2 = -0.287-(0.2\times 0.94) = -0.475$

$b = -0.029-(0.2\times 0.235) = -0.076$

final weight: $\begin{cases} W_1 = 0.071 \\ W_2 = -0.475 \\ b = -0.076 \end{cases}$

W1 increased initially due to the first training example (where y=1), then decreased after the second example. W2 flipped from positive to negative as the model adjusted for data points where y=0. The bias b decreased gradually due to two negative labels.

Part2.1

For bag-of-words feature:

Validation accuracy in the training process over 5 folds:

```
fold 1 | val acc 0.609375
training on fold 2
fold 2 | val acc 0.8119122257053292
training on fold 3
fold 3 | val acc 0.9216300940438872
training on fold 4
fold 4 | val acc 0.9498432601880877
training on fold 5
fold 5 | val acc 0.9811912225705329
fold 1 val acc: 0.609375
fold 2 val acc: 0.8119122257053292
fold 3 val acc: 0.9216300940438872
fold 4 val acc: 0.9498432601880877
fold 5 val acc: 0.9811912225705329
```

The word with the top positive weights and top negative weights:

```
top positive weight words:builds okay centers Fall since
top negative weight words:French A dots F italy
```

Overall accuracy, precision, recall, F1 score and confusion matrix:

```
acc 0.9912280701754386 | precision 0.9912404128218497 | recall 0.9912280701754386 | f1 0.991228015075377
```

```
Confusion Matrix:
 [[789   9]
 [  5 793]]
```

For binary bag-of-word feature:

```
training on fold 1
fold 1 | val acc 0.56875
training on fold 2
fold 2 | val acc 0.8683385579937304
training on fold 3
fold 3 | val acc 0.9498432601880877
training on fold 4
fold 4 | val acc 0.9780564263322884
training on fold 5
fold 5 | val acc 0.987460815047022
fold 1 val acc: 0.56875
fold 2 val acc: 0.8683385579937304
fold 3 val acc: 0.9498432601880877
fold 4 val acc: 0.9780564263322884
fold 5 val acc: 0.987460815047022
```


```
fold 1 | val acc 0.609375
training on fold 2
fold 2 | val acc 0.8119122257053292
training on fold 3
fold 3 | val acc 0.9216300940438872
training on fold 4
fold 4 | val acc 0.9498432601880877
training on fold 5
fold 5 | val acc 0.9811912225705329
fold 1 val acc: 0.609375
fold 2 val acc: 0.8119122257053292
fold 3 val acc: 0.9216300940438872
fold 4 val acc: 0.9498432601880877
fold 5 val acc: 0.9811912225705329
```


The word with the top positive weights and top negative weights:

```
top positive weight words:lets builds centers Fall since
top negative weight words:A French assume Said consider
```

Overall accuracy, precision, recall, F1 score and confusion matrix:

```
acc 0.9924812030075187 | precision 0.9924935771402728 | recall 0.9924812030075187 | f1 0.9924811557788945
Confusion Matrix:
 [[790   8]
 [  4 794]]
```

For tf-idf feature:

Validation accuracy in the training process over 5 folds:

```
fold 1 | val acc 0.553125
training on fold 2
fold 2 | val acc 0.658307210031348
training on fold 3
fold 3 | val acc 0.7021943573667712
training on fold 4
fold 4 | val acc 0.7335423197492164
training on fold 5
fold 5 | val acc 0.786833855799373
fold 1 val acc: 0.553125
fold 2 val acc: 0.658307210031348
fold 3 val acc: 0.7021943573667712
fold 4 val acc: 0.7335423197492164
fold 5 val acc: 0.786833855799373
```

The word with the top positive weights and top negative weights:

```
top positive weight words:okay goes years Holland Fall
top negative weight words:French Would stay italy follow
```

Overall accuracy, precision, recall, F1 score and confusion matrix:

```
acc 0.9110275689223057 | precision 0.9112859127221073 | recall 0.9110275689223057 | f1 0.9110135950143238
Confusion Matrix:
 [[717  81]
 [ 61 737]]
```

For all methods:

Some false positive examples:

- Not sure what we're doing together.
- And no worries.
- A Ber S Boh - Mun
  F Den - Kie
  A Gal - Boh

Some false negative examples:

- I've never played on play diplomacy. I play mostly on an app called Conspiracy. But don't get me wrong! I'm experienced, and I've had some success. I'm not to be trifled with!
- Picardy can stay put for now, I think that the English will anticipate us doing something other than trying to take the North Sea and will move to prevent it, if they think we're

thinking far enough ahead. It's worth the gamble that they'll only throw one support at it. Besides, in all likelihood even if you tried to move to Wales, Yorkshire would block it, since Yorkshire isn't doing anything useful where it is now.
- Oh man. I'm getting a lot of hate mail out west. Not fun. I think Germany has a pretty low opinion of me, and it sounds like she and Russia are developing a bit of a romance. I am totally cool with you taking back Trieste this turn, and I am thankful that you loaned it to me in the first place. But I'd prefer if you wait to take Greece until you can actually use the build. It would make things quite a bit tougher for me if I have to take a unit off the board especially if we don't get an extra unit in the east to compensate

Discussion:

By observing the scores, the binary bag-of-word word embedding slightly outperform the bag-of-word (unigram) word embedding and greatly outperform the tf-idf word embedding method.

By observing the top positive words, words like "okay" might be associated with deception because deceptive statements could involve informal reassurances without substantive content. "goes" and "years" might be linked to deceptive statements that refer to vague timelines or attempts to distract with historical context. "Holland" and "Fall" could be linked to specific deceptive strategies in the game, where mentioning locations might be used as a misdirection tactic.

By observing the top negative words, "French" and "Italy" suggest that truthful statements might involve more direct mention of specific nations, whereas deception might involve vaguer wording. "Would" and "stay" might suggest more strategic or hypothetical language, possibly indicating honest planning rather than misleading statements. "follow" could indicate a commitment to an action, making it more likely to appear in truthful statements.

By checking the common false positive and false negative cases, it can be found that most of the false positive case has a short sentence length and false negative case has a long sentence length.

Part2.2

For the neural network-based method, the pretrained Bert model is adopted as the base model with a 3 layers FNN as the classification head, corresponding pretrained BertTokenizer is adopted as the word embedding method, in the training process, the Bert weight is frozen and only the classification head is trained.

The training loss and accuracy over different folds:

```
  warnings.warn(
epoch 0 | loss 0.6896810978651047 | acc 0.530052125453949
epoch 1 | loss 0.6783092498779297 | acc 0.5947396159172058
epoch 2 | loss 0.6687836319208145 | acc 0.5903646349906921
epoch 3 | loss 0.660266324877739 | acc 0.6026041507720947
epoch 4 | loss 0.6566772818565368 | acc 0.6298958659172058
epoch 5 | loss 0.6545957297086715 | acc 0.6112500429153442
epoch 6 | loss 0.6498327344655991 | acc 0.62130206823349
epoch 7 | loss 0.6479571253061295 | acc 0.6177083849906921
epoch 8 | loss 0.6489234119653702 | acc 0.6266145706176758
epoch 9 | loss 0.6422664403915406 | acc 0.6220312714576721
epoch 10 | loss 0.6393172323703766 | acc 0.639635443687439
epoch 11 | loss 0.6413479298353195 | acc 0.6397916674613953
epoch 12 | loss 0.637134101986885 | acc 0.6324478983879089
epoch 13 | loss 0.6349926471710206 | acc 0.6518229246139526
epoch 14 | loss 0.6356129497289658 | acc 0.6365625262260437
epoch 15 | loss 0.6299783736467361 | acc 0.647656261920929
epoch 16 | loss 0.628107813000679 | acc 0.6613020896911621
epoch 17 | loss 0.631456607580185 | acc 0.6490625143051147
epoch 18 | loss 0.6265485763549805 | acc 0.6551562547683716
epoch 19 | loss 0.6212370276451111 | acc 0.6604166626930237
fold 1 | loss 0.6632729887962341 | acc 0.6031250357627869
```

```
epoch 0 | loss 0.6305930942296982 | acc 0.6546234488487244
epoch 1 | loss 0.6304757386445999 | acc 0.6577484607696533
epoch 2 | loss 0.6288156867027282 | acc 0.6491547226905823
epoch 3 | loss 0.6315315157175064 | acc 0.6529841423034668
epoch 4 | loss 0.6266405820846558 | acc 0.6601690649986267
epoch 5 | loss 0.6214886039495469 | acc 0.6673924326896667
epoch 6 | loss 0.6231102108955383 | acc 0.6680583953857422
epoch 7 | loss 0.6234881460666657 | acc 0.6555584073066711
epoch 8 | loss 0.614929249882698 | acc 0.6821977496147156
epoch 9 | loss 0.6232678383588791 | acc 0.6592341661453247
epoch 10 | loss 0.6177111148834229 | acc 0.6704021692276001
epoch 11 | loss 0.6206071764230728 | acc 0.6617699861526489
epoch 12 | loss 0.6201715677976608 | acc 0.6639984846115112
epoch 13 | loss 0.6160982459783554 | acc 0.6649718284606934
epoch 14 | loss 0.6086614072322846 | acc 0.6907915472984314
epoch 15 | loss 0.6166680932044983 | acc 0.6671234965324402
epoch 16 | loss 0.612553471326828 | acc 0.6865010261535645
epoch 17 | loss 0.6047629475593567 | acc 0.6884093284606934
epoch 18 | loss 0.6057896107435227 | acc 0.6852074861526489
epoch 19 | loss 0.6025979489088058 | acc 0.6907915472984314
fold 2 | loss 0.595802640914917 | acc 0.7114583849906921
```

```
epoch 0 | loss 0.601763105392456 | acc 0.6808657646179199
epoch 1 | loss 0.6038321346044541 | acc 0.6868084073066711
epoch 2 | loss 0.6008746683597564 | acc 0.693993330001831
epoch 3 | loss 0.6037407606840134 | acc 0.6952484846115112
epoch 4 | loss 0.6031716644763947 | acc 0.6897029280662537
epoch 5 | loss 0.5907366067171097 | acc 0.7126280665397644
epoch 6 | loss 0.600293031334877 | acc 0.6914958953857422
epoch 7 | loss 0.5901053279638291 | acc 0.7139600515365601
epoch 8 | loss 0.6020251095294953 | acc 0.690087080001831
epoch 9 | loss 0.6023818731307984 | acc 0.6816982626914978
epoch 10 | loss 0.5974641412496566 | acc 0.6967341303825378
epoch 11 | loss 0.5894905686378479 | acc 0.7030994296073914
epoch 12 | loss 0.5888058513402938 | acc 0.7152023315429688
epoch 13 | loss 0.5902333080768585 | acc 0.7022029161453247
epoch 14 | loss 0.5960999667644501 | acc 0.7026255130767822
epoch 15 | loss 0.5956720888614655 | acc 0.6999744176864624
epoch 16 | loss 0.5931645512580872 | acc 0.69770747423172
epoch 17 | loss 0.5778607666492462 | acc 0.7214139699935913
epoch 18 | loss 0.5826643437147141 | acc 0.7228611707687378
epoch 19 | loss 0.5904591143131256 | acc 0.7039191126823425
fold 3 | loss 0.6139499545097351 | acc 0.6738095283508301
```

```
epoch 0 | loss 0.5987965553998947 | acc 0.685245931148529
epoch 1 | loss 0.6038335889577866 | acc 0.6913806796073914
epoch 2 | loss 0.5966877862811089 | acc 0.6954405903816223
epoch 3 | loss 0.5822194457054138 | acc 0.7136014699935913
epoch 4 | loss 0.5981409907341003 | acc 0.6954405903816223
epoch 5 | loss 0.5905614376068116 | acc 0.708055853843689
epoch 6 | loss 0.5889991655945778 | acc 0.7063396573066711
epoch 7 | loss 0.599210774898529 | acc 0.6766521334648132
epoch 8 | loss 0.6038991987705231 | acc 0.6893442869186401
epoch 9 | loss 0.5860998392105102 | acc 0.7119236588478088
epoch 10 | loss 0.589401364326477 | acc 0.7063396573066711
epoch 11 | loss 0.5847224771976471 | acc 0.7132556438446045
epoch 12 | loss 0.5925867050886154 | acc 0.7061859965324402
epoch 13 | loss 0.5952913492918015 | acc 0.6961449980735779
epoch 14 | loss 0.5883684754371643 | acc 0.7079021334648132
epoch 15 | loss 0.5943986386060714 | acc 0.7033299207687378
epoch 16 | loss 0.5844410508871078 | acc 0.6999744176864624
epoch 17 | loss 0.5833208680152893 | acc 0.7214139699935913
epoch 18 | loss 0.5777957946062088 | acc 0.716495931148529
epoch 19 | loss 0.5745568528771401 | acc 0.7281378507614136
fold 4 | loss 0.5923983693122864 | acc 0.7054563760757446
```

```
epoch 0 | loss 0.6007928162813186 | acc 0.6898565888404846
epoch 1 | loss 0.5992021411657333 | acc 0.686769962310791
epoch 2 | loss 0.5970884054899216 | acc 0.7000896334648132
epoch 3 | loss 0.585839906334877 | acc 0.7086449861526489
epoch 4 | loss 0.5876828908920289 | acc 0.7045466303825378
epoch 5 | loss 0.5831000864505768 | acc 0.7140753269195557
epoch 6 | loss 0.5853080123662948 | acc 0.7217725515365601
epoch 7 | loss 0.5849938809871673 | acc 0.7156762480735779
epoch 8 | loss 0.5897469997406006 | acc 0.7080942988395691
epoch 9 | loss 0.5914277404546737 | acc 0.7002049088478088
epoch 10 | loss 0.5839348524808884 | acc 0.7211065888404846
epoch 11 | loss 0.5731763720512391 | acc 0.7211065888404846
epoch 12 | loss 0.592793446779251 | acc 0.7017289996147156
epoch 13 | loss 0.5969378262758255 | acc 0.7008324861526489
epoch 14 | loss 0.5935489147901535 | acc 0.6948130130767822
epoch 15 | loss 0.5835276544094086 | acc 0.7047003507614136
epoch 16 | loss 0.5873652547597885 | acc 0.6984118819236755
epoch 17 | loss 0.5788574889302254 | acc 0.7235271334648132
epoch 18 | loss 0.5728430330753327 | acc 0.7290343642234802
epoch 19 | loss 0.5916890442371369 | acc 0.7014984488487244
fold 5 | loss 0.5905963182449341 | acc 0.7054563760757446
```

Overall accuracy, precision, recall and F1 score:

```
acc 0.746031746031746 | precision 0.7882775119617225 | recall 0.7429435483870968 | f1-score 0.7347368421052631
```

The FNN layer number is set at 3 and the learning rate is set at 5e-4 by checking the loss and accuracy curve when training and validating on different settings ablatively.