פרויקט חלק שלישי: זיהוי סרקזם מתגובות

מגישים: אלכסנדר בוייב 314393158

משה פיירמן 204469449

1. מודל הייצוג של הנתונים-
מודל הייצוג שבחרנו הוא Data-Frame
של ייצוג מצורת טבלה של שורות ועמודות.
לאחר עיבוד המידע הטבלה מכילה מילון של נתונים לאימון המודל והערכת המודל.

2. מודל הלמידה שנבחר-
המודל שבחרנו הוא רשתות נוירוניות עמוקות (LSTM) עם הרחבת Bidirectional
שיכולה לשפר את ביצועי המודל בבעיות סיווג. המודל לומד בו-זמנית את הקלט מתחילתו וסופו. הראשון על רצף הקלט כפי שהוא והשני על עותק הפוך של רצף הקלט.
טכניקה זאת יכולה לספק הקשר נוסף לרשת וספק תוצאות טובות יותר.
למידה עמוקה נותנת מענה ראוי בבעיות מורכבות כמו סיווג תמונה, עיבוד שפה טבעית וזיהוי דיבור.
מודל למידה עמוקה:

למידה עמוקה צוברת פופולריות רבה בגלל עליונותה מבחינת הדיוק כאשר היא מאומנת עם כמות עצומה של נתונים.

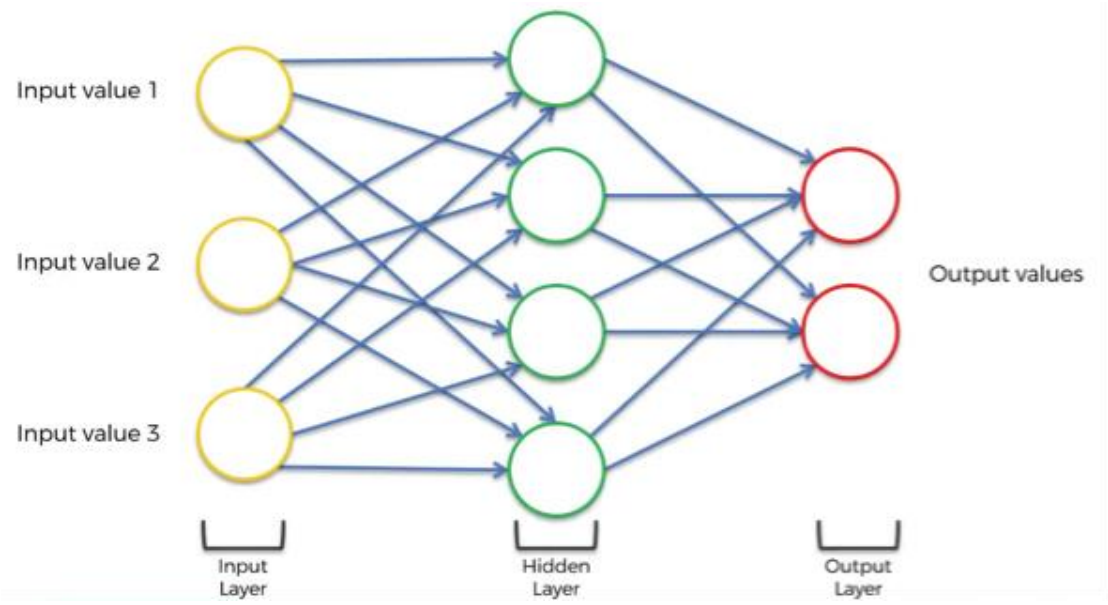
אחד היתרונות העיקריים של למידה עמוקה טמון ביכולת לפתור בעיות מורכבות הדורשות גילוי דפוסים נסתרים בנתונים ו / או הבנה מעמיקה של קשרים מורכבים בין מספר רב של משתנים תלויים זה בזה. אלגוריתמי למידה עמוקה מסוגלים ללמוד לבד דפוסים נסתרים מהנתונים, לשלב אותם יחד ולבנות כללי החלטות יעילים הרבה יותר.

LSTM:Long Short-Term Memory

הוא מודל המבוסס על RNN -רשת נוירונים רקורסיבית.

המודל מנסה ללמוד קשר בין המילים בעזרת יכולת זיכרון.

לכן החלטנו לבחור במודל זה שעונה על הצורך ונותן מענה לבעיית הסיווג Classification. שכן מודל LSTM הוא אחד המודלים הנפוצים לסיווג טקסט.



כלים שנוצלו:

- לקיצורים וראשי תיבות השתמשנו בספריית contraction שמכילה וקטורים מוכנים מראש מאת news google שיודע ת "להחזיר מילה מקוצרת למצבה הארוך".
- סימנים ומספרים הושמטו ע"י ביטוים רגולריים.
- למילות עצירה השתמשנו בספריית NLTK שמכילה מילות עצירה בשפה האנגלית.
- Lemmatizer –השתמשנו בספריית למה מוכנה של NLTK .

תגובה סרקסטית לפני ניקוי ועיבוד מקדים:

לאחר עיבוד המקדים קיבלנו אובייקט של data frame המכיל את כל המידע הרלוונטי לסיווג שלסרקזם (מידע נקי).

TensorFlow היא ספריית תוכנה שעובדת באמצעות גרפי זרימת נתונים. היא פותחה במקור על ידי גוגל בארגון המחקר של Machine Intelligence לצורך למידת מכונה וחקר רשתות עצביות עמוקות, אך המערכת ניתנת לשימוש גם במגוון תחומים אחרים.
במקור היא פותחה להפעלת חישובים מספריים גדולים שהופכים אותה לכלי נהדר עבור למידה עמוקה.

צילומי מסך:

המודל הראשון הוא Bidirectional LSTM עם 128 cells.

Building LSTM model

```
In [9]:
model = tf.keras.Sequential([Embedding(len(sarcasm_data['dict'])+1 , 128),
                    Bidirectional(CuDNNLSTM(128,return_sequences = True),name="bi-lstm"),
                    GlobalMaxPool1D(),
                    Dense(64, activation="relu"),
                    Dropout(0.5),
                    Dense(32, activation="relu"),
                    Dropout(0.5),
                    Dense(1, activation='sigmoid',name='classifyer')],name="LSTM-128-TEST")

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['binary_accuracy'])
model.summary()

Model: "LSTM-128-TEST"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, None, 128)         17895680
_________________________________________________________________
bi-lstm (Bidirectional)      (None, None, 256)         264192
_________________________________________________________________
global_max_pooling1d (Global (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 64)                16448
_________________________________________________________________
dropout (Dropout)            (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
classifyer (Dense)           (None, 1)                 33
=================================================================
Total params: 18,178,433
Trainable params: 18,178,433
Non-trainable params: 0
_________________________________________________________________
```

ניתן לראות את שכבות הרשת. הפלט של כל שכבה משמש גם כקלט של אותה השכבה לצעד הבא.

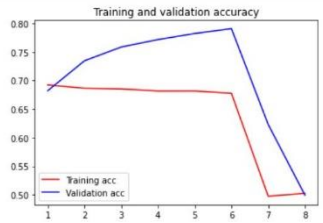
```
In [16]: print(classification_report(sarcasm_data['test']['Y'], pred, target_names = ['Not Sarcastic','Sarcastic']))

              precision    recall  f1-score   support

Not Sarcastic       0.69      0.70      0.69     75686
    Sarcastic       0.70      0.69      0.70     75930

     accuracy                           0.69    151616
    macro avg       0.69      0.69      0.69    151616
 weighted avg       0.69      0.69      0.69    151616
```

```
In [22]: makeGraph(history)
```



ניסיון לשיפור המודל-

Bidirectional LSTM עם 256 cells.

כמו כן הוספנו אופטימיזציה של קצב הלמידה.

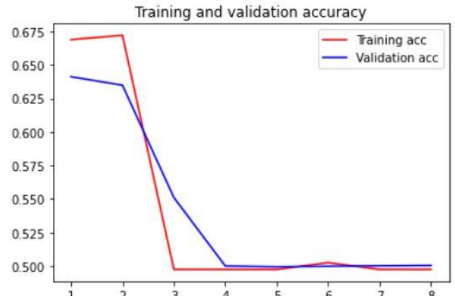
```
Model: "LSTM-256-TEST"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_8 (Embedding)      (None, None, 256)         35791360
_________________________________________________________________
bi-lstm (Bidirectional)      (None, None, 512)         1052672
_________________________________________________________________
global_max_pooling1d_8 (Glob (None, 512)               0
_________________________________________________________________
dense_19 (Dense)             (None, 128)               65664
_________________________________________________________________
dropout_19 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_20 (Dense)             (None, 64)                8256
_________________________________________________________________
dropout_20 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_21 (Dense)             (None, 32)                2080
_________________________________________________________________
dropout_21 (Dropout)         (None, 32)                0
_________________________________________________________________
classifyer (Dense)           (None, 1)                 33
=================================================================
Total params: 36,920,065
Trainable params: 36,920,065
Non-trainable params: 0
```

תוצאות של המודל:

```
In [25]: print(classification_report(sarcasm_data['test']['Y'], pred, target_names = ['Not Sarcastic','Sarcastic']))
```

	precision	recall	f1-score	support
Not Sarcastic	0.65	0.77	0.70	75686
Sarcastic	0.71	0.58	0.64	75930
accuracy			0.67	151616
macro avg	0.68	0.67	0.67	151616
weighted avg	0.68	0.67	0.67	151616

```
In [30]: makeGraph(history)
```



מודל Bert

המודל נועד ללמוד מכל המילים בכל העמדות, כלומר המשפט כולו- מה שהפך את המודל למדויק עוד יותר.

פריצת הדרך של BERT היא ביכולתו לאמן מודלים של שפה המבוססים על מכלול המילים במילה או בשאילתה (אימון דו-כיווני) ולא בדרך האימון המסורתית על רצף המילים המסודר (משמאל לימין או משולב משמאל. -מימין ומימין לשמאל). BERT מאפשר למודל השפה ללמוד הקשר מילים המבוסס על כל המילים במשפט ולא רק על המילה שקודמת לה או מיד אחריה.

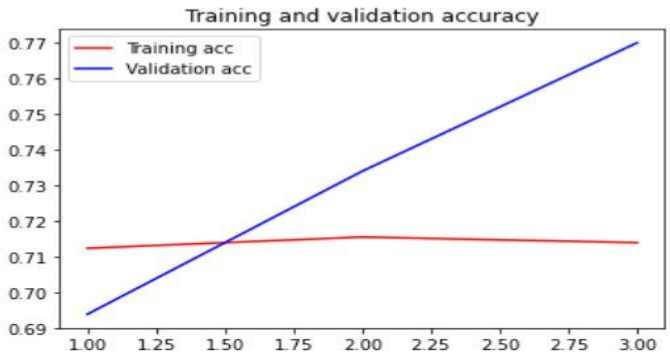
```
In [62]: model = build_model(bert_model, max_len=100)
         model.summary()
```

```
Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_word_ids (InputLayer)  [(None, 100)]             0
_________________________________________________________________
tf_distil_bert_model_3 (TFDi ((None, 100, 768),)       66362880
_________________________________________________________________
tf_op_layer_strided_slice_3  [(None, 768)]             0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 769
=================================================================
Total params: 66,363,649
Trainable params: 66,363,649
Non-trainable params: 0
_________________________________________________________________
```

```
print(classification_report(y_test, pred))
```

	precision	recall	f1-score	support
0	0.69	0.77	0.73	75811
1	0.74	0.66	0.70	75805
accuracy			0.71	151616
macro avg	0.72	0.71	0.71	151616
weighted avg	0.72	0.71	0.71	151616

```
In [73]: makeGraphBert(history)
```



Voting Classifier Test

ניסינו להריץ מודלים מוכנים מתוך סיפריית SKLEARN כגון:
RandomForestClassifier ,LogisticRegression
LinearSVCו KNeighborsClassifier

כל אלו הניבו תוצאות גרועות הקרובות ל50%.

```
LogisticRegression 0.5299704516673702
RandomForestClassifier 0.5946404073448712
LinearSVC 0.529891304347826
KNeighborsClassifier 0.5460769311945969
VotingClassifier 0.5395868509919798
```

לכן לא הכנסו אותם למבחן ההצבעה, רק את המודלים שלנו וקיבלנו שBERT מוביל.

Voting classifiers based on our models

```
n [89]: print(modelDict)

{'BERTlist': 0.872795, 'BI_LSTMlist': 0.75432, 'BI_LSTM_WEIGHTSlist': 0.7264}
```

Majority test

במבחן הרב ניסינו 60% תגובות סקרסטיות ו40% רגילות, BERT הניב את תוצאות הבאות

	precision	recall	f1-score	support
0	0.71	0.86	0.78	80000
1	0.89	0.76	0.82	120000
accuracy			0.80	200000
macro avg	0.80	0.81	0.80	200000
weighted avg	0.82	0.80	0.80	200000

כמו כן ניסינו על 75% סקרסטיות ו25% רגילות וגם כאן BERT הצליח לעבור את המבחן

```
In [36]: pred = BERT_MODEL.predict(testSentences)
         pred = np.round(pred).astype(int)
         print(classification_report(labels, pred))

              precision    recall  f1-score   support

           0       0.55      0.86      0.67     50000
           1       0.94      0.76      0.84    150000

    accuracy                           0.79    200000
   macro avg       0.75      0.81      0.76    200000
weighted avg       0.84      0.79      0.80    200000
```

לסיכום נמצא כי מודל Bert הניב תוצאות הטובות ביותר 0.715

וגם עמד במבחן הרוב.

הרצת המודל BERTעל משפטים רנדומליים מגוגל:

```
The sentence: It matters not what someone is born, but what they grow to be.
is labeled not-sarcastic.
have probability of 40.499 percent being sarcastic

The sentence: The way to get started is to quit talking and begin doing
is labeled not-sarcastic.
have probability of 30.997 percent being sarcastic

The sentence: Life is what happens when you're busy making other plans
is labeled not-sarcastic.
have probability of 27.411 percent being sarcastic

The sentence: Zombies eat brains. You’re safe.
is labeled sarcastic.
have probability of 80.331 percent being sarcastic

The sentence: Always remember that you’re unique. Just like everyone else
is labeled sarcastic.
have probability of 82.276 percent being sarcastic

The sentence: I'm glad we're having a rehearsal dinner. I rarely practice my meals before I eat.
is labeled sarcastic.
have probability of 74.452 percent being sarcastic
```