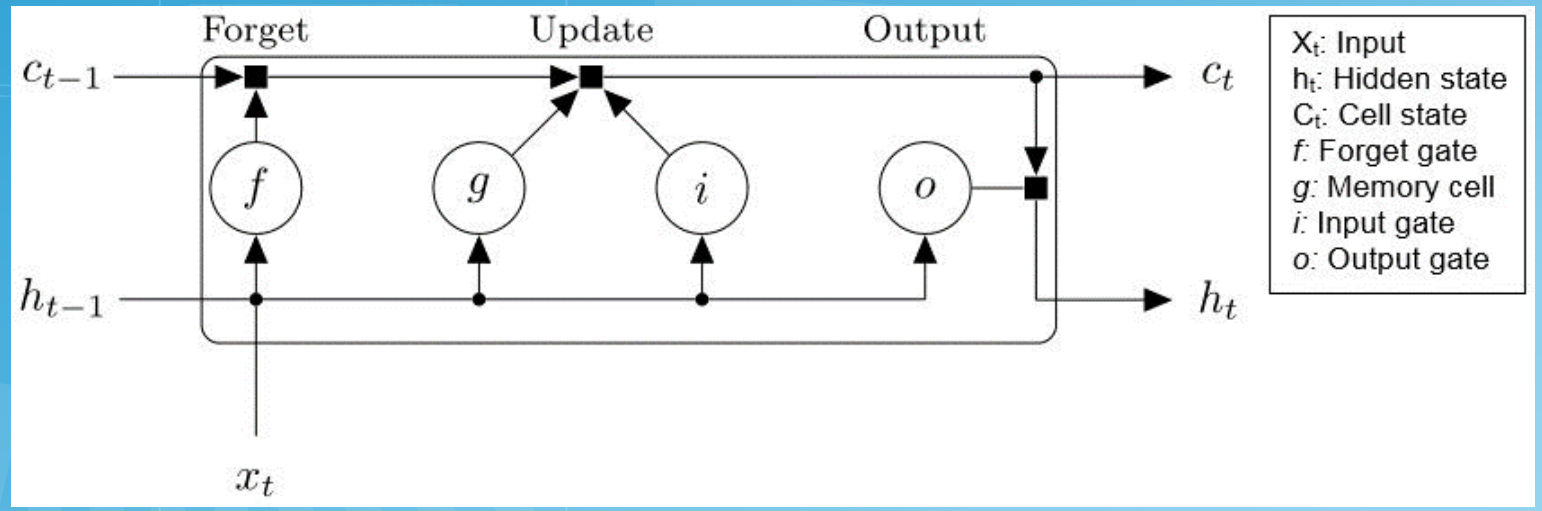
LSTM(Long Short Term Memory) 방식



초기 자연어 처리 모델인 RNN의 단점을 극복하기 위해 LSTM 방식이 개발되었다.

입력, 삭제, 출력 게이트를 사용하여 어떤 정보를 셀 상태에 추가하고 삭제할지, 그리고 어떤 부분을 출력으로 전달할지를 결정한다. 입력 게이트를 통해 새로운 정보를 받아들이고, 삭제 게이트를 통해 이전 정보를 삭제하며, 출력 게이트를 통해 현재 상태의 일부를 출력으로 전달한다. 각 게이트의 메커니즘을 이용하여 데이터 소실을 방지하고 학습 속도가 향상된다.

데이터 전처리 및 토큰화

- 텍스트 데이터를 전처리함으로써 비정형 데이터를 모델에 입력하는데 적합하도록 변경

```
def normalize(text_data):
    normalized = [re.sub(r'https?://₩S+|www₩.₩S+', '', text.lower()) for text in text_data]
    normalized = [re.sub(r'₩W', ' ', text) for text in normalized]
    normalized = [re.sub(r'₩n', ' ', text) for text in normalized]
    normalized = [re.sub(r' +', ' ', text) for text in normalized]
    normalized = [text.strip() for text in normalized]
```

소문자로 변환, URL제거, 단어가 아닌 문자, 개행문자를 공백으로 변경

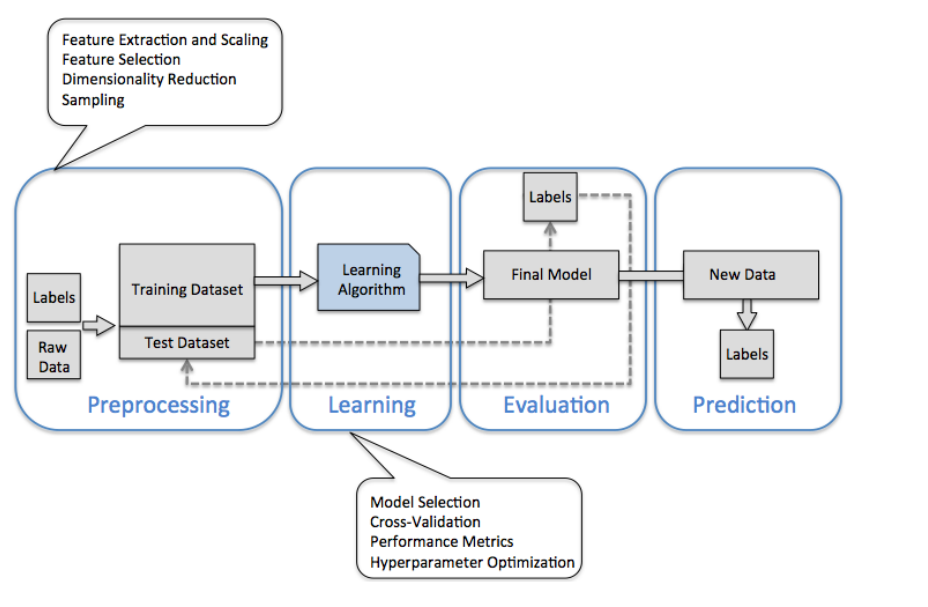
- 텍스트 데이터를 모델이 이해할 수 있도록 텍스트를 단어 단위로 나누어 토큰화 진행

```
Donald Trump just couldn t wish all Americans ...
House Intelligence Committee Chairman Devin Nu...
On Friday, it was revealed that former Milwauk...
On Christmas day, Donald Trump announced that ...
Pope Francis used his annual Christmas Day mes...
```

```
[donald, trump, just, couldn, t, wish, all, am...
[house, intelligence, committee, chairman, dev...
[on, friday, it, was, revealed, that, former, ...
[on, christmas, day, donald, trump, announced,...
[pope, francis, used, his, annual, christmas, ...
```

-토큰화함으로 단어에 고유한 숫자로 매핑되어 모델이 텍스트를 이해하는데 도움
-단어들간 유사성을 비교

모델 학습




```
X_train, X_test, y_train, y_test = train_test_split(features, targets, test_size=0.20, random_state=18)
```

데이터를 훈련데이터와 테스트데이터로 나누어 전체 데이터의 80%를 훈련데이터, 20%를 테스트데이터로 사용
X = 텍스트, y = 뉴스의 진위여부 분류

```
X_train = normalize(X_train)
X_test = normalize(X_test)
```

훈련 데이터를 전처리

모델의 성능 확인

```
early_stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=2, restore_best_weights=True)
model.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
              optimizer=tf.keras.optimizers.Adam(learning_rate=1e-2),
              metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=3,validation_split=0.1, batch_size=100, shuffle=True, callbacks=[early_stop])
```

```
Epoch 1/3
324/324 [==============================] - 364s 1s/step - loss: 0.3285 - accuracy: 0.8458 - val_loss: 0.2285 - val_accuracy: 0.9048
Epoch 2/3
324/324 [==============================] - 406s 1s/step - loss: 0.1328 - accuracy: 0.9557 - val_loss: 0.0788 - val_accuracy: 0.9727
Epoch 3/3
324/324 [==============================] - 391s 1s/step - loss: 0.0666 - accuracy: 0.9806 - val_loss: 0.0603 - val_accuracy: 0.9802
```

정확성, 손실 파악
일정 횟수동안 정확도가 올라가지 않으면 조기 종료할 수 있도록 설정

```
print('Accuracy:', accuracy_score(predictions, y_test))
print('Precision:', precision_score(predictions, y_test))
print('Recall:', recall_score(predictions, y_test))
```

```
Accuracy: 0.9886414253897555
Precision: 0.9928207503473833
Recall: 0.9837081229921982
```

실제 기사에 적용

Former President Donald Trump has a new talking point in his rally speeches, claiming that the Biden administration is moving the military to all-electric-powered tanks. For Trump, the attack line is a trifecta: He's making President Joe Biden seem weak on defense and the military seem "woke," while mocking Biden's green energy efforts.

```
normalized_article = normalize([new_article])
tokenized_article = tokenizer.texts_to_sequences(normalized_article)
```

새로운 기사를 받아 전처리하고 토큰화하여 저장

```
pred = model.predict(tokenized_article)

probability = tf.nn.sigmoid(pred)[0, 0]

threshold = 0.5
predicted_class = 1 if probability > threshold else 0

if predicted_class == 0:
    print("The news article is predicted as FAKE.")
else:
    print("The news article is predicted as REAL.")
```

전처리된 텍스트에 모델적용, 시그모이드 함수를 이용하여 모델의 출력을 확률로 표현
시그모이드 함수는 이진분류하는데 효과적
임계값을 0.5로 설정하여 기사의 진위 여부 판별