Rețele neuronale recurente

Large Movie Review Dataset

```
from tensorflow.keras.datasets import imdb
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense


top_words = 5000
(X_train, y_train), (X_test, y_test) = imdb.load_data(num_words=top_words)

max_review_length = 500
X_train = sequence.pad_sequences(X_train, maxlen=max_review_length)
X_test = sequence.pad_sequences(X_test, maxlen=max_review_length)

embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length,
                    input_length=max_review_length))
model.add(LSTM(100))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
              metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 500, 32)           160000
_________________________________________________________________
lstm (LSTM)                  (None, 100)               53200
_________________________________________________________________
dense (Dense)                (None, 1)                 101
=================================================================
Total params: 213,301
Trainable params: 213,301
Non-trainable params: 0
_________________________________________________________________
```

```
model.fit(X_train, y_train, epochs=3, batch_size=64)

scores = model.evaluate(X_test, y_test, verbose=0)
print(f'Accuracy: {scores[1] * 100:.2f}%')
```

```
Accuracy: 85.77%
```

LSTM + Dropout

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length,
                    input_length=max_review_length))
model.add(Dropout(0.2))
model.add(LSTM(100))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
```

```
Accuracy: 87.83%
```

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length,
                    input_length=max_review_length))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))
```

```
Accuracy: 74.61%
```

Bidirectional LSTM

```
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length,
                    input_length=max_review_length))
model.add(Bidirectional(LSTM(100, dropout=0.2,
                             recurrent_dropout=0.2)))
model.add(Dense(1, activation='sigmoid'))
```