Data Science

Keras input for multivariate classification with LSTM using current features and previous timesteps features and y values

Asked 1 month ago Active 3 days ago Viewed 26 times

0

I am working on a multivariate binary classification problem. What I want to do is to predict a binary classification given the features at the current timestep and the data (features+real classification) from past timesteps

Keras seems to have a problem with the shape of my inputs so I want to know what I am doing wrong:

```
X_train = (nb_samples, nb_timesteps, nb_features)
y_train = (nb_samples, nb_timesteps, binary_result)

model = Sequential()
model.add(LSTM(nb_units,
               input_shape = X_train.shape[1:]
              ))
Dense(1,activation='softmax')
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=ep, validation_data=(X_train, y_train),
verbose=2, shuffle=False)
```

classification keras predictive-modeling lstm multiclass-classification

edited Jun 22 at 15:07

Stephen Rauch ♦
1,598 6 13 30

asked Jun 22 at 13:11

Entrians
1 1

2 Answers

0

Try the following code.

```
import keras

X_train = (nb_samples, nb_timesteps, nb_features)
y_train = (nb_samples, nb_timesteps, binary_result)

model = keras.models.Sequential()
model.add(keras.layers.LSTM(nb_timesteps, input_shape = X_train.shape[1:]))
model.add(keras.layers.Dense(1, activation="sigmoid"))
```

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()
history = model.fit(X_train, y_train, batch_size=32, epochs=ep, validation_split=0.2,
verbose=2, shuffle=False)
```

answered Jul 3 at 7:01

Swapnil Pote

91 3

0

Since you want to get a classification/output per each time step, you should set `return_sequences=True` in you LSTM layer.

Read more about `return_sequences` here in LSTM documentation.

answered Jul 3 at 4:38

SaTa

136 2
