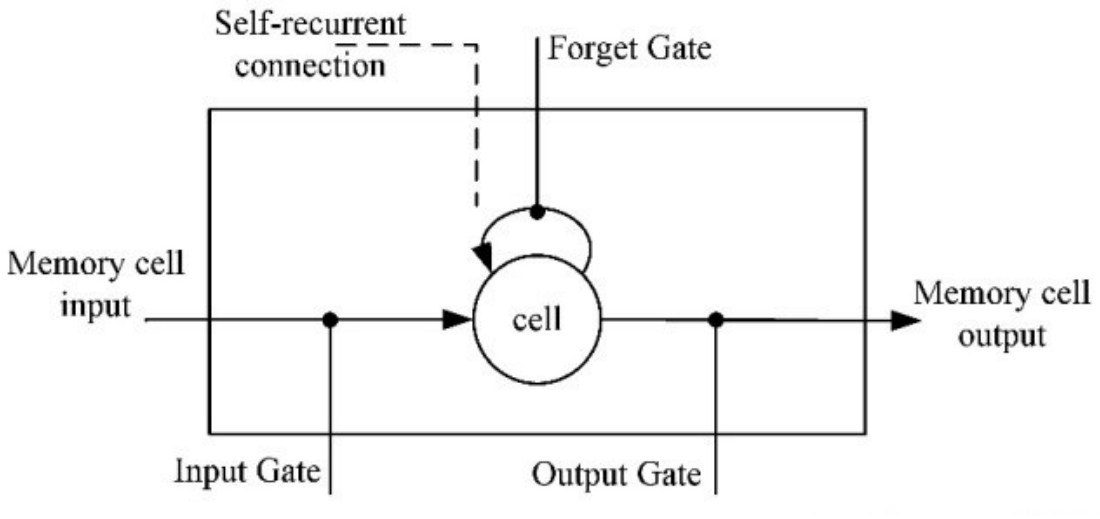
LSTM原理

长短期记忆网络（Long Short Term Memory，简称LSTM）模型， 本质上是一种特殊的循环神经网络（Recurrent Neural Network，简称RNN）。lstm模型在rnn模型的基础上通过增加门限（gate）来解决rnn短期记忆的问题，使得rnn能够有效利用长距离的序列信息。

lstm在rnn的基础上增加了三个逻辑控制单元

- 输入门限（input gate）
- 输出门限（output gate）
- 遗忘门限（forget gate）

这三个逻辑控制单元各自连接到了一个乘法元件上（如下图所示），通过设定神经网络的记忆单元与其他部分的连接权重，从而控制数据流的输入、输出以及细胞单元（memory cell）的状态， 具体概念图如下



上图中具体部件的描述如下：

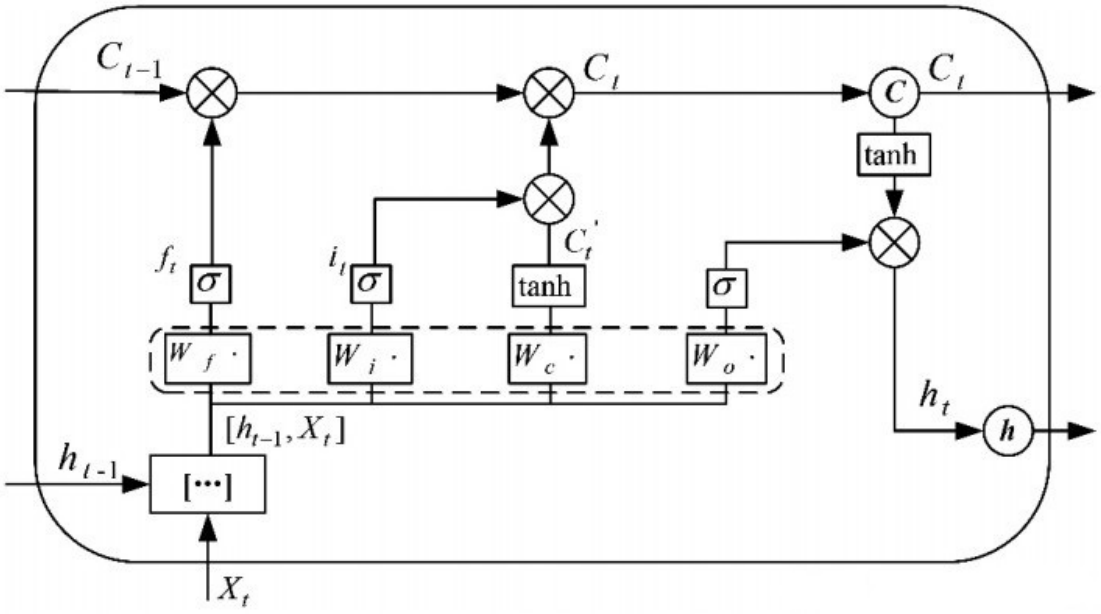
- input gate： 控制信息是否流入，记为i_t
- forget gate : 控制上一时刻的memory cell的信息是否积累到当前时刻的memory cell中， 记为f_t
- output gate : 控制当前时刻的memory cell的信息是否流入当前的隐藏状态h_t中， 记为o_t
- cell : 记忆单元， 表示神经元状态的记忆， 使得lstm单元具有保存、读取、重置和更新长距离历史信息的能力，记为c_t

在t时刻，lstm模型的公式定义如下:

$$f_t = \text{sigmoid}(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \text{sigmoid}(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$o_t = \text{sigmoid}(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$
$$c_t = f_t \times c_{t-1} + i_t * \tilde{c}_t$$
$$h_t = o_t \times \tanh(c_t)$$

根据公式描述，lstm的细节图如下。在lstm神经网络的训练过程中，首先将t时刻的数据特征输入至输入层，经过激励函数输出结果； 然后将输出结果、$t-1$时刻的隐藏层输出和$t-1$时刻cell单元存储的信息输入lstm结构的节点中，通过input gate、output gate、forget gate和cell单元的处理，输出数据到下一隐藏层或输出层，输出lstm结构节点的结果到输出层神经元，计算反向传播误差，并更新各个权值。

lstm细节图如下



数据处理及特征工程（以黄金为例）

1. 原始的数据表如下（gold_select.csv），1826个样本，16个列

	Date	USD (PM)	SMA20	MACDhist	EMA26	MOM	HT_DCPERIOD	HT_DCPHASE	SINE	LEADSINE	INPHASE	QUADRATURE	PPO	TRIX	profit	Volatility
0	2016-09-11	1324.60	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	0.000000	0.007207
1	2016-09-12	1324.60	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.000717	0.007110
2	2016-09-13	1323.65	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.001435	0.007017
3	2016-09-14	1321.75	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.008284	0.006928
4	2016-09-15	1310.80	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.001869	0.006973
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1821	2021-09-06	1821.60	48394.7485	-48.314382	49145.729833	4806.26	18.557396	245.796542	-0.912095	-0.934847	-1167.524697	692.399758	2.058184	0.542504	-0.010677	0.006789
1822	2021-09-07	1802.15	48796.1900	82.718963	49689.063705	3620.54	18.325259	112.064687	0.926760	0.389692	-53.250602	2888.988696	2.123373	0.547589	-0.008962	0.006930
1823	2021-09-08	1786.00	48897.7555	-230.615795	49246.003135	-2088.48	18.137593	125.602108	0.813079	0.163290	1186.945742	2516.770176	1.902319	0.549024	0.001260	0.006992
1824	2021-09-09	1788.25	48864.9420	-477.661305	48758.676499	-2728.40	18.054042	145.729270	0.563104	-0.186169	1893.287141	1481.972548	1.530244	0.546846	0.003551	0.006903
1825	2021-09-10	1794.60	48716.9890	-603.954561	48390.986268	-706.08	18.051969	177.459144	0.044332	-0.675064	2041.319105	-332.983484	1.153146	0.541811	0.003551	0.006903

1826 rows × 16 columns

去掉date之后如下（1826x15）其中 USD（PM）为预测目标，**剩下的14个作为特征**

	USD (PM)	SMA20	MACDhist	EMA26	MOM	HT_DCPERIOD	HT_DCPHASE	SINE	LEADSINE	INPHASE	QUADRATURE	PPO	TRIX	profit	Volatility
0	1324.60	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	0.000000	0.007207
1	1324.60	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.000717	0.007110
2	1323.65	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.001435	0.007017
3	1321.75	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.008284	0.006928
4	1310.80	606.3935	3.334620	608.693333	-22.77	15.646696	205.946888	-0.437538	-0.945216	4.223394	0.904014	0.047648	0.336072	-0.001869	0.006973
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1821	1821.60	48394.7485	-48.314382	49145.729833	4806.26	18.557396	245.796542	-0.912095	-0.934847	-1167.524697	692.399758	2.058184	0.542504	-0.010677	0.006789
1822	1802.15	48796.1900	82.718963	49689.063705	3620.54	18.325259	112.064687	0.926760	0.389692	-53.250602	2888.988696	2.123373	0.547589	-0.008962	0.006930
1823	1786.00	48897.7555	-230.615795	49246.003135	-2088.48	18.137593	125.602108	0.813079	0.163290	1186.945742	2516.770176	1.902319	0.549024	0.001260	0.006992
1824	1788.25	48864.9420	-477.661305	48758.676499	-2728.40	18.054042	145.729270	0.563104	-0.186169	1893.287141	1481.972548	1.530244	0.546846	0.003551	0.006903
1825	1794.60	48716.9890	-603.954561	48390.986268	-706.08	18.051969	177.459144	0.044332	-0.675064	2041.319105	-332.983484	1.153146	0.541811	0.003551	0.006903

1826 rows × 15 columns

本次任务中，我们利用lstm将**时间序列预测问题**转化为了**监督学习问题**

我们知道，时间序列预测的本质主要是根据前T个时刻的观测数据（特征）来预测处第T+1个时刻的时间序列的值，这就可以转化为机器学习中的监督学习问题了，即利用之前的样本训练出一个预测模型，对新的输入样本进行预测，具体原理如下

我们根据超参数 n_in（滞后期数），截取前 n_in 个时刻的**所有列（包括预测目标）**，作为第 t 个时刻的特征，比如本次任务，我们的 n_in = 1，所以我们提取前 1个 时刻的特征，作为第 t 个时刻的特征，最后提取的数据表如下

如下图所示，第t个时刻的样本具有16列，其中最后一列 Y(t)为第t个时刻样本的**预测目标**， 前面15列为**前一个样本的所有列**

	Y(t-1)	X1(t-1)	X2(t-1)	X3(t-1)	X4(t-1)	X5(t-1)	X6(t-1)	X7(t-1)	X8(t-1)	X9(t-1)	X10(t-1)	X11(t-1)	X12(t-1)	X13(t-1)	X14(t-1)	Y(t)
1	0.211270	0.000006	0.580656	0.000072	0.594681	0.060532	0.697173	0.281223	0.027392	0.479091	0.631052	0.442464	0.402517	0.493311	0.239120	0.211270
2	0.211270	0.000006	0.580656	0.000072	0.594681	0.060532	0.697173	0.281223	0.027392	0.479091	0.631052	0.442464	0.402517	0.486412	0.231194	0.210261
3	0.210261	0.000006	0.580656	0.000072	0.594681	0.060532	0.697173	0.281223	0.027392	0.479091	0.631052	0.442464	0.402517	0.479503	0.223517	0.208243
4	0.208243	0.000006	0.580656	0.000072	0.594681	0.060532	0.697173	0.281223	0.027392	0.479091	0.631052	0.442464	0.402517	0.413621	0.216245	0.196612
5	0.196612	0.000006	0.580656	0.000072	0.594681	0.060532	0.697173	0.281223	0.027392	0.479091	0.631052	0.442464	0.402517	0.475332	0.219909	0.194009
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1821	0.739179	0.807751	0.515860	0.802108	0.622554	0.194316	0.828054	0.021715	0.058718	0.456214	0.603092	0.478891	0.466890	0.493311	0.212196	0.739179
1822	0.739179	0.812671	0.563080	0.810068	0.729544	0.182658	0.807908	0.043939	0.032576	0.422067	0.646586	0.482830	0.468299	0.390603	0.204767	0.718519
1823	0.718519	0.819498	0.607670	0.819135	0.696429	0.172918	0.436292	0.963386	0.694848	0.476294	0.695929	0.484139	0.469919	0.407108	0.216347	0.701365
1824	0.701365	0.821225	0.501044	0.811741	0.536990	0.165044	0.473910	0.906544	0.581647	0.536650	0.687567	0.479701	0.470377	0.505429	0.221447	0.703755
1825	0.703755	0.820667	0.416976	0.803609	0.519119	0.161538	0.529840	0.781554	0.406917	0.571025	0.664322	0.472230	0.469683	0.527468	0.214166	0.710500

1825 rows × 16 columns

当然，我们的模型可以预测未来多个时刻的值， 我们也能提取多个前面时刻的所有列作为特征， 具体的公式如下

n_vars * n_in + n_out

加入我们要预测未来一步，且提取前一个时刻的所有列，且我们有（1个预测目标+14个特征 = 15）列的数据表

计算过程为 15x1+1 = 16

我们本次的数据为 1825x16， 要转换为（样本个数，滞后期数，特征个数）的维度来作为lstm的输入

于是数据表（1825x16）转化为 （1825，1，15）的data

data再按照设定（前999天观望那个，第一千天投资）为训练集（999，1，15）和测试集（826，1，15）

ok，最终的数据为：

- **train_X (999,1,15)**
- **test_X (826, 1, 15)**

模型的结构

(还会更新)

本次的神经网络结构为 120个隐藏层节点的lstm层， 1个全连接层的神经网络

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_6 (LSTM)                (None, 120)               65280
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 121
=================================================================
Total params: 65,401
Trainable params: 65,401
Non-trainable params: 0
_________________________________________________________________
```

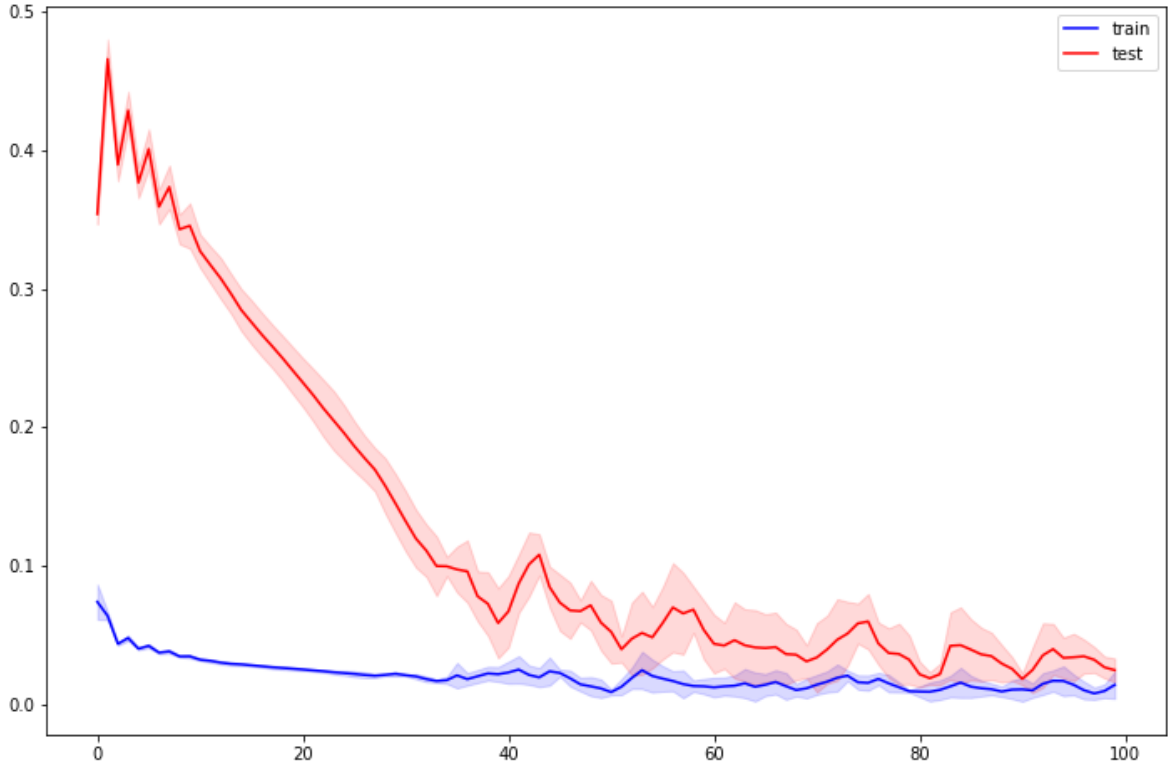
模型的超参数

- n_in : 滞后期数
- n_out : 超前预测数
- n_vars : 数据表的特征个数
- n_neuron : lstm的隐藏层神经元个数
- n_batch : 批次大小，也就是一次训练选取的样本个数
- n_epoch : 模型在整个训练数据集中的工作次数
- repeats : 训练的模型个数（我们采取训练国歌模型取平均的方法，增加模型的稳定性）

```
n_in = 1 # 滞后期数
n_out = 1 # 超前预测数
n_vars = 15 # 特征个数
n_neuron = 120 # lstm的隐藏层神经元个数
n_batch = 72
n_epoch = 100
repeats = 5 # lstm重复训练的次数
```

模型的训练

采用 n_batch的批次大小，n_epoch的工作次数，训练 repeats个模型，模型之间求均值和std，下图为模型的loss值下降过程，虚色为置信区间，实线为下降趋势



拟合效果

下图为模型对测试集的拟合效果