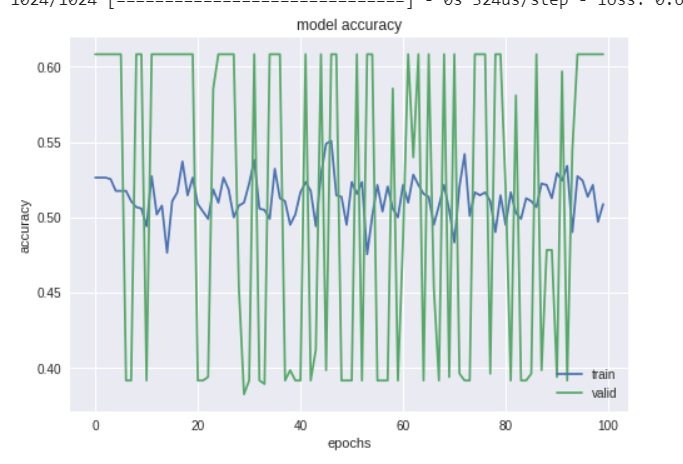
**ReduceLROnPlateau**

<https://keras.io/callbacks>

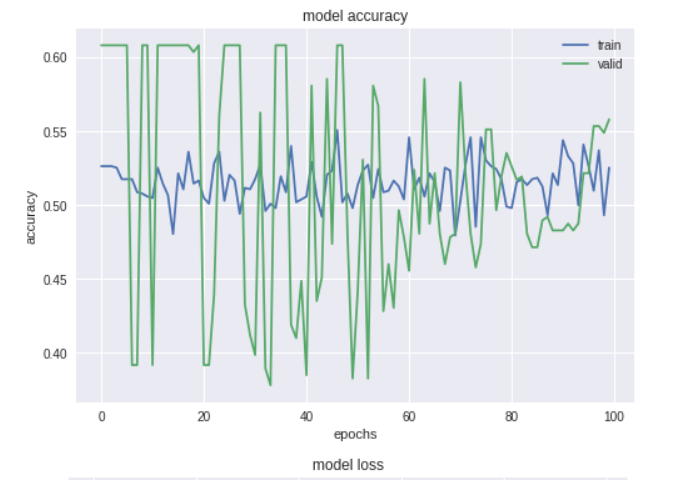
keras.callbacks.ReduceLROnPlateau(monitor='val\_loss', factor=0.1, patience=10, verbose=0, mode='auto', min\_delta=0.0001, cooldown=0, min\_lr=0)

Reduce learning rate when a metric has stopped improving. Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates. This callback monitors a quantity and if no improvement is seen for a 'patience' number of epochs, the learning rate is reduced.

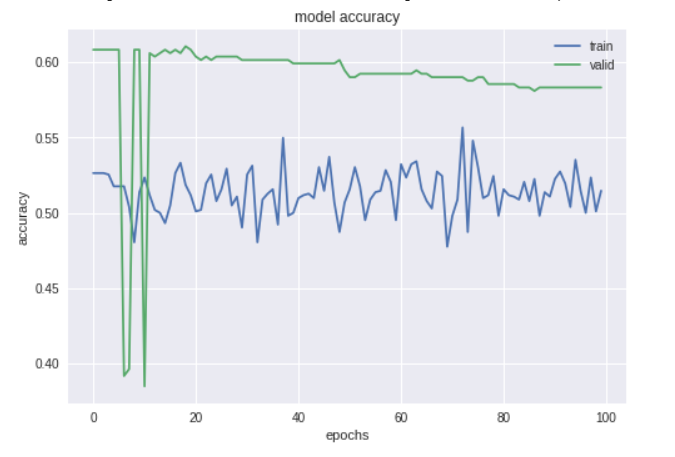
Validation data accuracy is volatile and this could be caused by dataset too small, learning rate too high.



reduce\_lr=ReduceLROnPlateau(monitor='val\_acc', factor=0.8,patience=5, min\_lr=0.000001)



reduce\_lr=ReduceLROnPlateau(monitor='val\_acc',factor=0.2,patience=5,min\_lr=0.000001)



**Parameters combinations (sentiment algo)**

* Epochs=100/200/500 batch\_size=100 windows=50/10
* Epochs=100/200/500 batch\_size=100 windows=30/1
* ReduceLROnPlateau(…, factor=0.2, patience=5, ….) (used with target window=1)

Higher epochs (500,1000) does not produce higher accuracy, but the model finds days with higher returns.

Higher target window seems to pick days with higher returns.

Scaling input data does not improve the accuracy, but it produces good returns.

Period considered influences the results, correlation sentiment/price seems to change during the time. Furthermore, the moving average applied has effects on the model performance. Scaling should be considered when data covers long periods.

Sigmoid, tanh (combined with input scaling) output layer activation produces higher returns, but accuracy is not improved.

**Parameters combinations (algoKeras\_portfolio)**

* Window=30/1 variables=['dayofweek','month','mfi'] epochs=10/100 onehot\_features=['dayofweek','month'] mfi(20,50)
* Same as above, but using mfi\_categorical and activation=tanh
* Tanh/sigmoid as activation functions for output layer

The predictions produced by the model are tested as buy/sell signals for an investment strategy. Backtesting on a portfolio of 10 losing stocks produces a positive cumulative return, thus beating the benchmark buy and hold strategy. Beating stocks rising strongly is difficult.

When tested on 39 FTSEMIB stocks, the model signals produce cumulative returns that are positive, lower than the benchmark, but with lower standard deviation.