1 Sequence2Sequence Regression

```
openExample('nnet/SequencetoSequenceRegressionUsingDeepLearningExample')
```

Some further explanation are given here, but read all the comments in the live script.

1. Load the train data

```
>>dataTrain = dlmread("train_FD001.txt"); % similar to readTable, b
>> size(dataTrain)

ans =
        20631          26
```

2. Get the number of observations

```
>> numObservations = max(dataTrain(:,1));
100
There are 100 engines for which different length series are known
```

Unit	time in cycles	operational setting1	os2	os3	sensor measurement 1	sm2	... sm 17
1	1	-0.0007	-0.0004	100	518.67	641.82	23.419
1	2	0.0019	-0.0003	100	518.67	642.15	23.4236
..							
1	191	0	-0.0004	100	518.67	643.34	23.1295
2	1	-0.0018	0.0006	100	518.67	641.89	23.4585
...							

3. Create sequences as they are expected by the deep network

   Since the sequences are of different length, the best representation is with cell arrays. For both Xtrain and Ytrain the array will have size 100 (the number of engines).

```
>> XTrain = cell(numObservations,1);
>> YTrain = cell(numObservations,1);
```

   (a) Populate the empty cells with values from *dataTrain*. For each engine, get the rows.

   Try for the first engine:

```
idx = dataTrain(:,1) == 1;
```

(b) Keep the right columns for X and y.

```
X = dataTrain(idx,3:end)'; %transpose
XTrain{i} = X;
```

```
timeSteps = dataTrain(idx,2)';
Y = fliplr(timeSteps);
YTrain{i} = Y;
```

X will be an array of size 24×192, meaning 24 features, 192 is the length of the sequence. Check in the file that indeed the first engine stops at time 192. The flip is used in order to state the number of cycles still to run.

In the end, XTrain and YTrain is a cell array with dimension 100. In general, each element in X for a sequence must be a matrix with n rows, where n is the number of features, and m columns, where m is the length of the sequence.

	time 0	time 1	...	time m
feature 1	$value_1^0$	$value_1^1$	...	$value - 1^m$
feature 2	$value_2^0$	$value_2^1$	...	$value_2^m$
...				
feature n	$value_n^0$	$value_n^1$	...	$value_n^m$

2 Yet another example

Electricity load (and price) forecasting Video and code