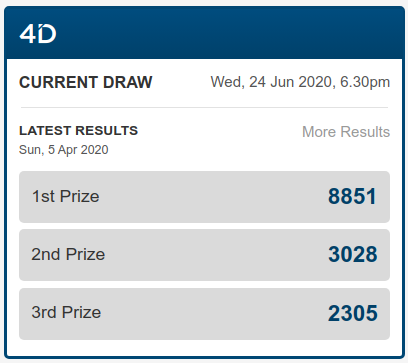
4D Lottery

Description: <https://online.singaporepools.com/en/lottery/how-play-4d>

To play 4D, select a four-digit number from 0000 to 9999.



## Primary statistical analysis

First thing we need to do is check if this lottery is fair: all values are random and belong to uniform distribution.

We checked the dataset we got from the customer and performed primary statistical analysis.

### Check if data is random

We used several criterias (extreme points criteria, Foster–Stuart criteria, Spearman's rank correlation coefficient) to check if the series is random or there are some features specific for time series.

====== Extreme points criteria ======

DRAW1: series is random, stat=-1.2765

DRAW2: series is random, stat=-0.5585

DRAW3: series is random, stat=0.3989

DRAWS: series is random, stat=-0.9156

=============================== Foster-Stuart criteria ==============================

DRAW1: no tendency of changing mean, stat=-1.3822; no tendency of changing variance, stat=-0.1551

DRAW2: no tendency of changing mean, stat=0.3456; no tendency of changing variance, stat=0.2588

DRAW3: no tendency of changing mean, stat=-0.6911; no tendency of changing variance, stat=-0.9831

DRAWS: no tendency of changing mean, stat=-1.2306; no tendency of changing variance, stat=-0.1999

=== Spearman's rank correlation coefficient ===

DRAW1: series is random, stat=1.5625

DRAW2: series is random, stat=0.9467

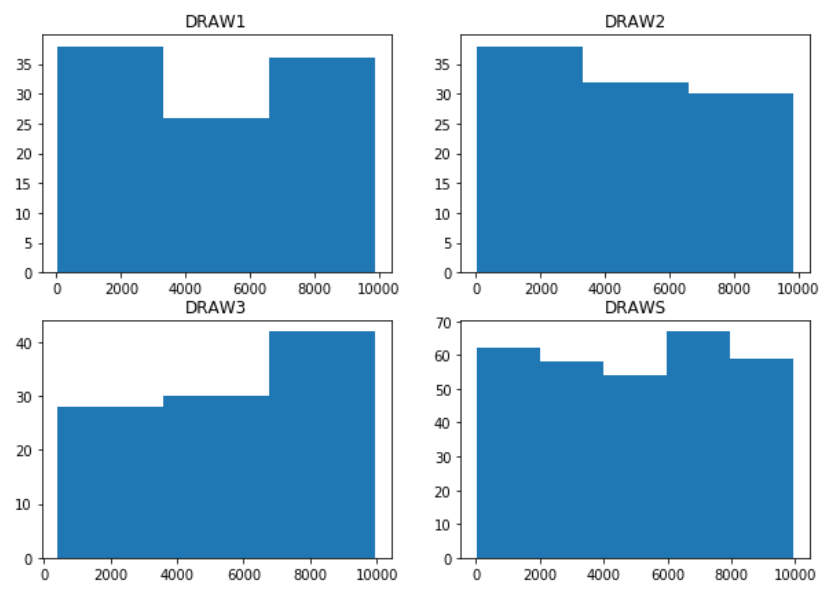
DRAW3: series is random, stat=-1.7947

DRAWS: series is random, stat=-1.6462

### Identify distribution

There are three ways to identify distribution: by histogram, by skew and kurtosis, by probability grid.

**By histogram**



For uniform distribution histogram should be flat, without extreme high and extreme low values.

**By skew and kurtosis**

<https://en.wikipedia.org/wiki/Skewness>

<https://en.wikipedia.org/wiki/Kurtosis>

For some distributions these values are constant.

E.g. (A - skew, E - kurtosis):

Normal distribution: A=0, E=0

Exponential distribution: A=2, E=6

Uniform distribution: A=0, E=-1.2

Result of checking hypotheses for uniform distribution (A=0, E=-1.2):

DRAW1: A=0 (A^=0.0034); E=-1.2 (E^=-1.3624)

DRAW2: A=0 (A^=0.0489); E=-1.2 (E^=-1.2299)

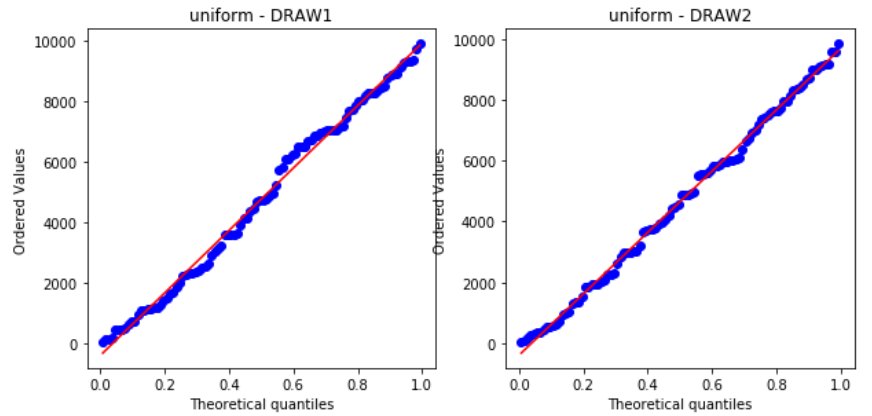
DRAW3: A=0 (A^=-0.2134); E=-1.2 (E^=-1.1058)

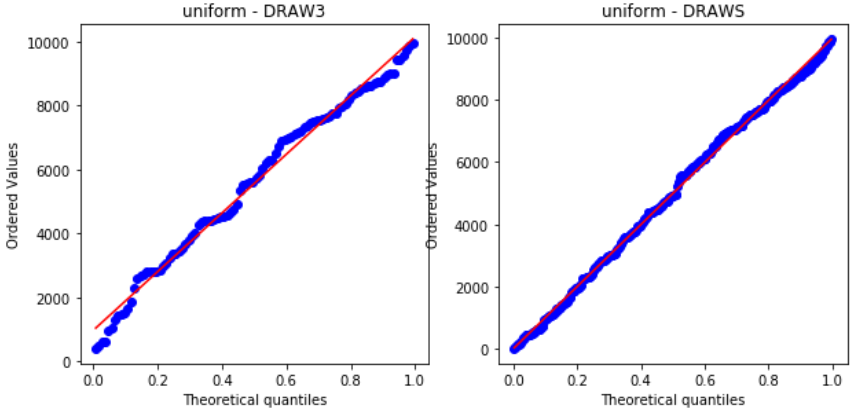
DRAWS: A=0 (A^=-0.0742); E=-1.2 (E^=-1.2470)

**By probability plot**

If series on probability plot (<https://en.wikipedia.org/wiki/Probability_plot>) for a particular distribution form a line, it means that data is from that particular distribution.

These are probability plot for uniform distribution.





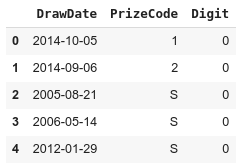
## Conclusion

Sample data belongs to uniform distribution and the lottery is fair.

## Data grabbing

We will collect date from : <http://www.singaporepools.com.sg/en/product/Pages/4d_cpwn.aspx>

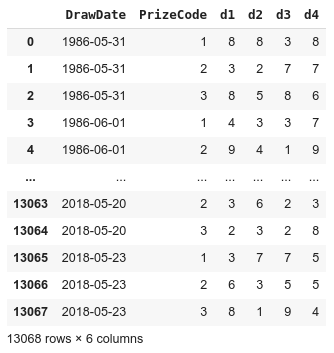
Date interval: 1986-05-31 to 2018-05-23



## Exploratory Data Analysis

### Data transformation

Before data analysis let’s transform data into:



### Histograms

Let’s build distribution for each digit

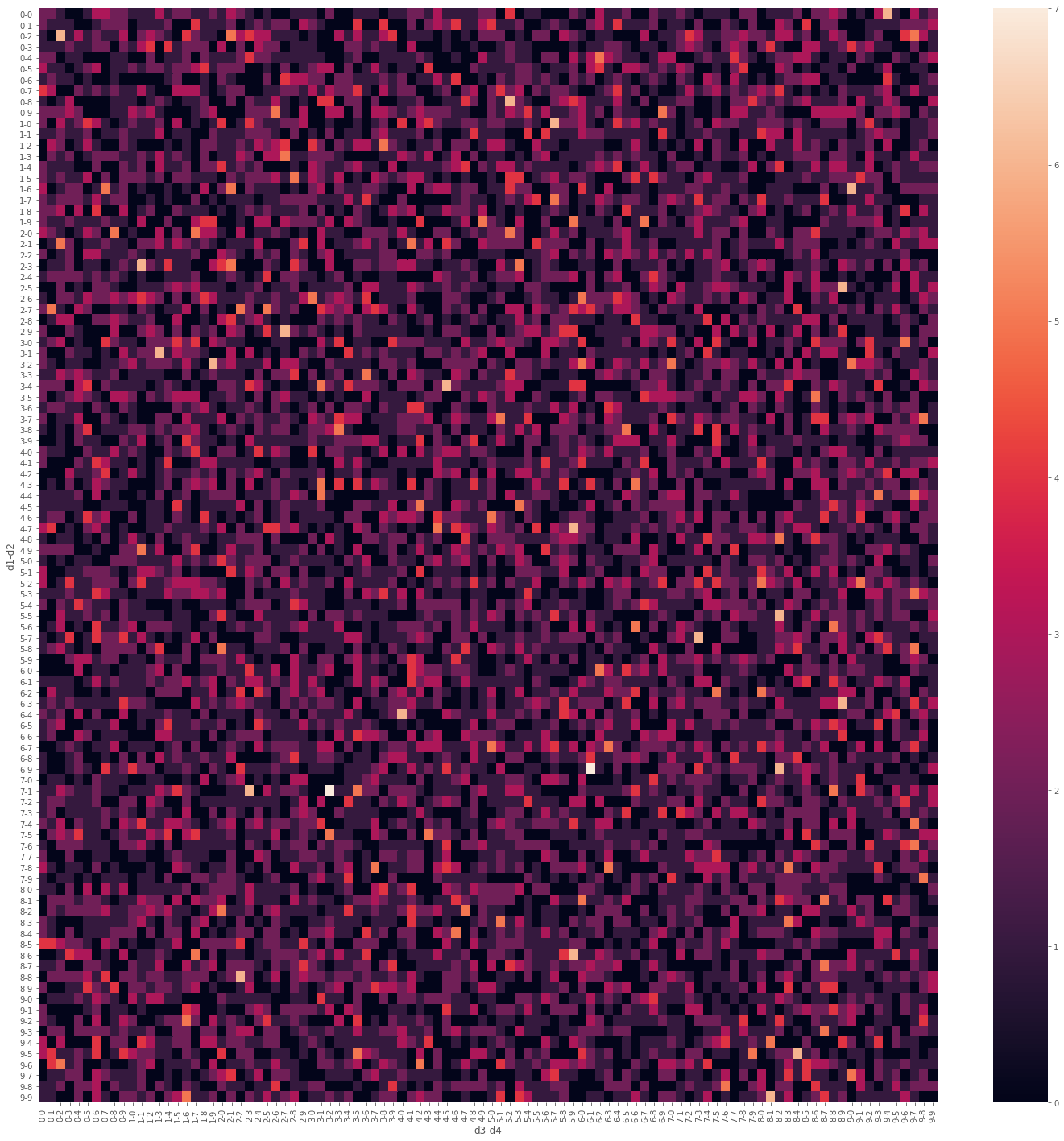
|  |  |
| --- | --- |
| Digit\_1 | Digit\_2 |
| Digit\_3 | Digit\_4 |

As you can see we work with univariate distribution.

The most frequent digits is: **3786**

### Heatmap for cross-presence matrix

Pairs occurrence, how often digit\_1 and digit\_2 were with digit\_3 and digit\_4



We have a few pairs that have high frequency: **6169** and **3271**

## 

## Modeling

Our goal is to predict the next 4 winning digits.

We will use two approaches for sequence predicting: LSTM model and WaveNet model.

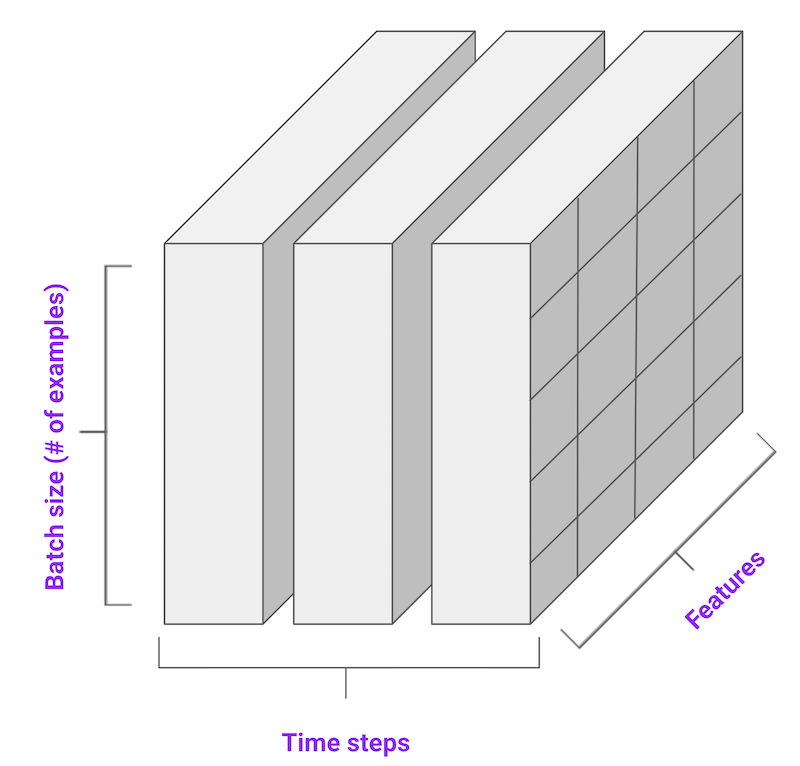
### LSTM model

We will represent model input as sequence with shape: (history\_window: 30, features: 12).

Output of model has hape: (digits: 4)

It means that we take into account information about [1st Prize, 2st Prize, 3st Prize] from last 30 draws and try predict next 4 digits for 1st\_Prize

The following visualisation should help you understand how the data is represented after batching:



This approach can find patterns in sequence(historical data) and using this information make predictions.

**Model architecture**

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

lstm (LSTM) (None, 30, 64) 19712

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lstm\_1 (LSTM) (None, 64) 33024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 4) 260

=================================================================

Total params: 52,996

Trainable params: 52,996

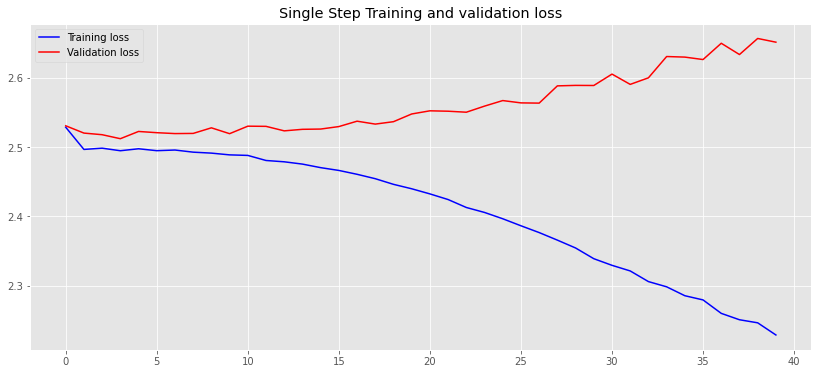
Non-trainable params: 0

For training we used 4000 draws and for validation we used 296 draws

train: (3940, 30, 12), valid: (296, 30, 12)

Single window of past history : (30, 12)

Train model:



As you can see, the model is overfitted.

During the first 5 epochs the model tries to predict 3,4,5,6 for any digits it’s only one way for finding global minimum.

Example of predictions :

Label: [6 3 9 9]; Prediction: [5. 5. 4. 6.]

Label: [1 7 0 3]; Prediction: [4. 4. 6. 4.]

Label: [0 6 0 0]; Prediction: [3. 6. 5. 3.]

Label: [8 3 9 4]; Prediction: [4. 3. 6. 4.]

Label: [8 5 4 5]; Prediction: [4. 3. 4. 3.]

Label: [1 9 5 9]; Prediction: [5. 6. 6. 5.]

Label: [4 8 8 1]; Prediction: [6. 5. 6. 4.]

Label: [4 6 8 2]; Prediction: [5. 5. 8. 6.]

### WaveNet

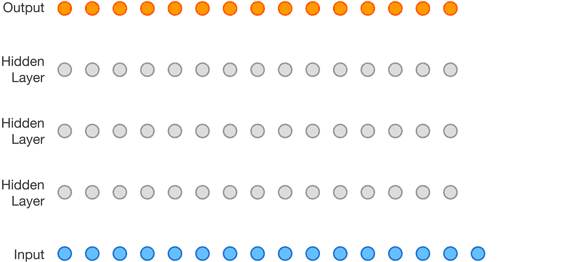
This approach also can work with sequences and make predictions for the next state.

We will represent model input as sequence with shape: (history\_window: 30, features: 12).

Output of model has hape: (digits: 30)

It means that we take into account information about [1st Prize, 2st Prize, 3st Prize] from last 30 draws and try predict next 30 digits for 1st\_Prize

The animation shows how a WaveNet is structured. It is a fully convolutional neural network, where the convolutional layers have various dilation factors that allow its receptive field to grow exponentially with depth and cover thousands of timesteps.



#### Model architecture

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv1d\_71 (Conv1D) (None, None, 64) 192

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_72 (Conv1D) (None, None, 64) 8256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_73 (Conv1D) (None, None, 64) 8256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_74 (Conv1D) (None, None, 64) 8256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_75 (Conv1D) (None, None, 64) 8256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_76 (Conv1D) (None, None, 64) 8256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv1d\_77 (Conv1D) (None, None, 1) 65

=================================================================

Total params: 41,537

Trainable params: 41,537

Non-trainable params: 0

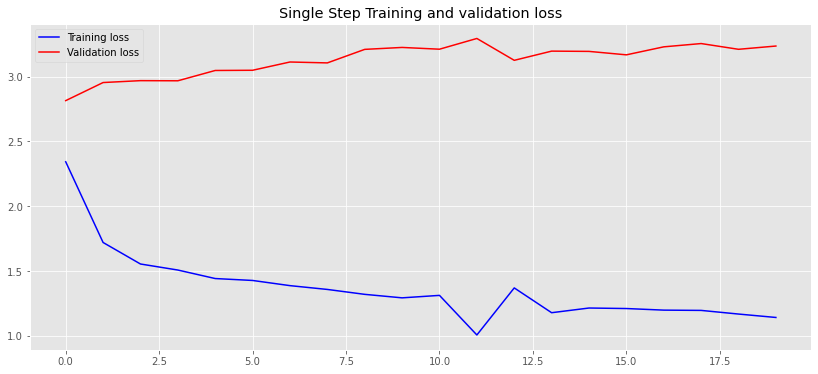
For training we used 4000 draws and for validation we used 296 draws

train: (3940, 30, 12), valid: (296, 30, 12)

Single window of past history : (30, 12)

We flatten data before feeding it to model.

Train mode:

l

The same problem model is overfitted.

Predictions:

Label: [9 0 5 4]; Prediction: [4. 5. 5. 4.]

Label: [0 9 5 1]; Prediction: [4. 4. 4. 4.]

Label: [0 4 4 6]; Prediction: [5. 6. 5. 6.]

Label: [5 2 0 8]; Prediction: [4. 4. 5. 5.]

Label: [2 0 1 8]; Prediction: [4. 5. 5. 6.]

Label: [6 5 1 9]; Prediction: [4. 6. 5. 4.]

Label: [0 4 1 7]; Prediction: [4. 4. 4. 4.]

Label: [0 0 6 2]; Prediction: [5. 5. 6. 5.]

The optimal predictions are average digit.

## Outcomes

4D lottery is a fair lottery and all generated digits are random variables from univariate distribution.

All attempts to find patterns were failed but I believe that in the future we will be able to find more smart solutions!