Data selection or processing

All the nominal data is left apart for the time being, that can be tackled by

Assign nominal data's each value as feature as true or false, means 1 or 0.

For an example,

In "pollution2.csv" dataset "Wind direction" (wnd_dir) feature has values as

SE, cv, NW, etc …, assign each value as feature like

SE: as 0 or 1

Cv: as 0 or 1 depends on the data

But this going to be end up in too much of features.

All the feature values are in different range like 0-1, or 0-1000, so all of them are just normalized.

Value

$$Normalised_{value} = \frac{Gieven_{value} - Feature's\ Minimum_{value}}{Feature's\ Maximum_{value} - Feature's\ Minimum_{value}}$$

So, all the values going to lie between 0 and 1.

After the predictions the values must be changed to the given scale, since the prediction is only useful if they in the given scale only. In order to do that, the denormalization is done as

$$final_{predicted-value} = Model_{predicted-value(in\ normalised\ scale)} * (Feature's\ Maximum_{value} - Feature's\ Minimum_{value}) + Feature's\ Minimum_{value}$$

The data is separated as

Train: First 70% of the data is used for training

Test: Rest 30% of the data is used for testing

In order to analyses each model's performance for the time being only the **training data** predictions are used, it can be extended to test set. If we finalize the measurement matrixes, it can be applied to testing as well.

For the **normalization**, currently the **whole** sets'(pretend get from expert's range) features' minimum and maximum is considered. For future experiments only the **train** sets' features' minimum and maximum is considerable for normalization.

In order to explain the given data and predictions the following annotations are used.

T_k: Given data of time step k (means T_1: Given data of time step 1) and

P_k: Predicted data of time step k (P_2: Predicted data of time step 2)

Proposed models to check the difference among the performance

In all these models "swish" activation function is used in Dense layers. In order to avoid over fitting dropout (0.5) is implemented in dense layers, further, to avoid overfitting in LSTM cells number of nodes assignment for LSTM is fixed.

Number of LSTM node calculation

Since the models are vary along with the number of features so in order to generalize the model parameters, just create a small rough calculation that may be changed or improved dependent on the performance.
The following annotations are used.

f_k : Feature k
c_{k_m} : correlation of between feature k and feature m
n: number of features
n_t: number of time_steps

Since the computers are in binary domain, factor of 2 is maintained.

Weight of features inter relationship can be get by at least correlations' matrix's one with diagonal portion like

	f_1	f_2	f_3	f_4 … f_n
f_1	c_{1_1}			
f_2	c_{2_1}	c_{2_2}		
f_3	c_{3_1}	c_{3_2}	c_{3_3}	
f_4	c_{4_1}	c_{4_2}	c_{4_3}	c_{4_4}

The needed weights are 1,2,3,4...n

it is an arithmetic series summation = (n * (2 + (n - 1))) / 2
= (n+1)* n / 2

Since LSTM has cell state and hidden states (2 x number of units in LSTM)

The number of minimum neurons weights for features

$NW_{feature}$= ((n+1)* n / 2)/2
= (n+1)* n / 4

likewise, relationship along the time (if time steps relationship want to be preserved)

The number of minimum neurons weights for time

NW_{time} = (n_t +1)* n_t / 4

If the number of final nodes represent the correlation matrixes' output for each, the number of nodes

Number of nodes = $NW_{feature}$ x NW_{time}

Pseudo code for the estimation

```
number_minimum_neurons_weigthts = ((features +1) * features) *(2* (2 + 1)) /4
h_size_fac=1
h_size_neurons=2
h_size=h_size_neurons*h_size_fac

while features*h_size < number_minimum_neurons_weigthts:
    h_size_fac = h_size_fac+1
    h_size = h_size_neurons^h_size_fac
```

If we don't have big data just leave the time steps relationship calculation since it can be captured by LSTM itself; force to forward the correlation between the features.

Number of nodes = $NW_{feature}$

Always make sure number of train samples*features > Number of nodes *10

Else try to do with minimum weights otherwise it will be over fit to the data needed to train

Evaluation of number of nodes calculation

Number of nodes calculation is evaluated in the pollution_2.csv data's LSTM time distributed models. Both models' o/p supports the "Number of LSTM node calculation". 6-sequence time distributed LSTM models with different between factored vs calculated hidden size test sets' RMSE results is shown in Table.I

For an example:

first row in the Table I is showing the

Prediction step:1

Hidden size: 8

Factor: 0.5

Calculated hidden size is 16 so the factor of 0.5 is 8.

Table I: 6 Sequence Time Distributed LSTM Test set's RMSE difference between calculated number of nodes Vs factored ($RMSE_6$- $RMSE_{6\text{-factored-hidden}}$)

Prediction step	Hidden size	factor	Features RMSE difference						
			pollution	dew	temp	press	wnd_spd	snow	rain
1	8	0.5	-16.15	-0.59	-0.64	-1.49	-7.23	-0.11	-0.18
	32	2	-15.1	-0.48	-0.65	-1.31	-6.3	-0.09	-0.12
	64	4	-14.44	-0.44	-0.78	-1.3	-6.94	-0.08	-0.09
	128	8	-12.58	-0.45	-0.64	-1.53	-6.98	-0.09	-0.06
2	8	0.5	-15.73	-0.48	-0.71	-1.55	-7.09	-0.1	-0.11
	32	2	-13.89	-0.14	-0.76	-1.37	-5.85	-0.08	-0.1
	64	4	-14.02	-0.11	-0.91	-1.36	-6.92	-0.08	-0.06
	128	8	-11.78	-0.17	-0.75	-1.64	-6.66	-0.08	-0.08
3	8	0.5	-15.97	-0.56	-0.78	-1.51	-7.18	-0.1	-0.1
	32	2	-14.1	-0.21	-0.77	-1.35	-6.35	-0.08	-0.11
	64	4	-12.83	-0.17	-0.88	-1.31	-7.53	-0.09	-0.07
	128	8	-12.04	-0.26	-0.72	-1.55	-6.64	-0.09	-0.1
4	8	0.5	-16.15	-0.57	-0.81	-1.54	-7.37	-0.1	-0.09
	32	2	-13.95	-0.24	-0.77	-1.33	-6.96	-0.08	-0.13
	64	4	-12.84	-0.19	-0.9	-1.35	-7.54	-0.1	-0.07
	128	8	-11.9	-0.27	-0.69	-1.52	-7.04	-0.09	-0.08
5	8	0.5	-16.08	-0.58	-0.83	-1.54	-7.48	-0.1	-0.08
	32	2	-13.82	-0.25	-0.76	-1.33	-7.07	-0.08	-0.13
	64	4	-12.87	-0.21	-0.91	-1.36	-7.65	-0.11	-0.06
	128	8	-11.8	-0.28	-0.7	-1.52	-6.85	-0.09	-0.08
6	8	0.5	-16.01	-0.59	-0.84	-1.54	-7.53	-0.1	-0.08
	32	2	-13.67	-0.27	-0.76	-1.33	-6.96	-0.09	-0.14
	64	4	-12.82	-0.23	-0.91	-1.36	-7.67	-0.1	-0.05
	128	8	-11.78	-0.3	-0.7	-1.52	-6.93	-0.09	-0.08

Model_1: **calculated(given)** hidden size, with 1-time step to predict the 2nd time step (or 1+prediction step)

Model_2: **factored** hidden size, with 1-time step to predict the 2nd time step (or 1+prediction step)

Here the row represents difference between RMSE of 1st step using the Model_1 – Model_2.

second row in the Table I is showing the

Prediction step:2

Hidden size: 8

Factor: 0.5

Calculated hidden size is 16 so the factor of 0.5 is 8.

Model_3: **given** hidden size, with 2-time step to predict the 3rd time step (or 1+prediction step)

Model_4: **factored** hidden size, with 2-time steps to predict the 3rd time step (or 1+prediction step)

Here the row represents difference between RMSE of 1st step using the Model_3 – Model_4.

If the difference is negative mean calculated hidden size works better. Since the factored hidden models have more RMSE than calculated hidden models.

Likewise, 6-sequence train set, 11-sequence train, and test sets' RMSE difference also calculated as shown in the attachments.

https://drive.google.com/file/d/1M7Z_1vG_3RXuzdOidQI0f-e3yMPq-9EL/view?usp=sharing
https://drive.google.com/file/d/1MN4wDzfa3suq1XgpclHJP8ynJi5M8Xc5/view?usp=sharing
https://drive.google.com/file/d/1Qwwi9DSCypgu6c4IO29vGcMYE6HfwU63/view?usp=sharing
https://drive.google.com/file/d/1xCgjCUzw2nrZ_irdn9Db0fxXx8zP5_Ry/view?usp=sharing

Evaluation of models

Time distributed models, just basic LSTM models, and stacked LSTM models are proposed to see the performance variation.

The models performances can be evaluated by their error in predictions, such as

Error is calculated in both scales.

E_given_scale: Error in given scale

E_norm_scale: Error in normalised scale

SE (square error) and RMSE(root mean square error) are only calculated for given scale, since make square in fractions may not efficiently represents the error.

Using the error values, different measurement matrixes (are proposed) since different among the model's architectures (Time distributed, single layer LSTM (base LSTM), and stacked LSTM).

Summer 2020 conclude pollution -2 data results

Base model

The initial evaluation of different feeding time steps(as 15,25, or 35) hidden nodes(1,2,4,8,16,32,64 or 128) with the different prediction steps(2,3...10); and their corresponding results is shown in the following results.

From the results of the given dataset prediction(pollution); the errors are averaged all-over the prediction steps.

https://drive.google.com/file/d/1qBviTYXOrmpmNnvLh8gjUfzT1njbhFc8/view?usp=sharing

In the spreadsheet, please check the page "1st_min_error_among_test_train" summarized the first occurrence of the minimal error apart from prediction step-6 the rest hold the same performance similarity for train and test.

Stack-2

For the prediction steps(<=5) hidden nodes 4 is enough in representation and six to ten hidden nodes 8 performs better than others; however, all these cases the given time steps not always the same; even though 25 performs better.

Stacked LSTM evaluation with only 2-stacks of different feeding time steps(as 15,25, or 35) hidden nodes(1,2,4,8,16,32,64 or 128) with the different prediction steps(2,3...10); and their corresponding results is shown in the following results.

From the results of the given dataset prediction(pollution); the errors are averaged all-over the prediction steps.

https://drive.google.com/file/d/1iZOaAD1qBEhCQ4Oyaomk1T2FaDHwYihq/view?usp=sharing

the 2-stack-LSTM is better than the base LSTM model. Still, stack-3 models are running; after that results obtained the relationship between the stack and hidden nodes can be identified.

Satck-3 results

https://drive.google.com/file/d/1CgOy-zJgGn_VObcRrzfON9F0kGg5P079/view?usp=sharing

Final comparison and conclusion

Table. II compares the different model results. From this result of this dataset it can be clearly seen all the test sets' best performed models' number of hidden nodes perform better are in 16 or under 16(like 8, or 4 or 2). It validates the calculation. But we must validate more by going through few more datasets to conclude.

Table. II: Comparison of different model's base and stacked LSTM with their normalized results

Base model							Stack-2						Stack-3					
				Overall Normalized scale						Overall Normalized scale						Overall Normalized scale		
Train/Test	given timesteps	prediction steps	hidden size	MAE	MSE	RMSE	given timesteps	prediction steps	hidden size	MAE	MSE	RMSE	given timesteps	prediction steps	hidden size	MAE	MSE	RMSE
test	25	2	4	0.026	0.002	0.045	35	2	4	0.024	0.002	0.041	15	2	8	0.025	0.002	0.041
test	15	3	4	0.029	0.002	0.049	25	3	16	0.026	0.002	0.044	25	3	8	0.027	0.002	0.046
test	35	4	4	0.033	0.003	0.056	35	4	16	0.029	0.002	0.048	25	4	16	0.030	0.002	0.049
test	25	5	4	0.037	0.004	0.061	25	5	8	0.030	0.003	0.050	15	5	8	0.033	0.003	0.054
test	25	7	8	0.039	0.004	0.063	35	6	8	0.031	0.003	0.051	25	6	8	0.033	0.003	0.055
test	15	6	8	0.039	0.004	0.065	35	7	8	0.033	0.003	0.054	25	7	8	0.034	0.003	0.056
test	25	8	8	0.041	0.005	0.069	25	8	8	0.033	0.003	0.056	35	9	8	0.036	0.004	0.059
test	35	9	8	0.042	0.005	0.070	35	10	16	0.034	0.003	0.057	35	10	8	0.036	0.004	0.061
test	15	10	8	0.043	0.005	0.071	25	9	16	0.035	0.003	0.058	25	8	4	0.039	0.004	0.063
train	25	2	4	0.029	0.002	0.050	35	2	4	0.025	0.002	0.044	15	2	8	0.026	0.002	0.044
train	15	3	4	0.031	0.003	0.053	25	3	16	0.028	0.002	0.046	25	3	8	0.029	0.002	0.048
train	35	4	4	0.036	0.004	0.060	15	4	64	0.030	0.002	0.048	25	4	16	0.032	0.003	0.052
train	25	5	4	0.040	0.004	0.066	15	5	64	0.032	0.003	0.053	25	6	8	0.035	0.003	0.057
train	25	7	8	0.041	0.005	0.067	35	6	16	0.032	0.003	0.053	15	5	8	0.035	0.003	0.058
train	35	6	8	0.042	0.005	0.069	25	7	8	0.033	0.003	0.055	25	7	8	0.036	0.003	0.059
train	25	8	8	0.044	0.005	0.073	35	10	16	0.035	0.003	0.058	35	9	8	0.038	0.004	0.061
train	35	9	8	0.045	0.006	0.075	25	8	8	0.035	0.003	0.058	35	10	8	0.038	0.004	0.064
train	15	10	8	0.046	0.006	0.076	25	9	16	0.036	0.003	0.059	15	8	16	0.040	0.004	0.065

Fall 2020 Evaluation of number of nodes calculation on "household_power_consumption"

Number of nodes calculation is evaluated in the pollution_2.csv data's LSTM time distributed models; the evaluation confirms the validity of the calculation.

To validate checking on to the next dataset evaluation: "household_power_consumption.txt" data's LSTM time distributed models are used.

Sample of the data is given in Table. III,

Table. III: Sample of the data of "household_power_consumption"

Line number	Date	Time	Global active power	Global reactive power	Voltage	Global intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
1	16/12/2006	17:24:00	4.216	0.418	234.84	18.4	0	1	17
2	16/12/2006	17:25:00	5.36	0.436	233.63	23	0	1	16
3	16/12/2006	17:26:00	5.374	0.498	233.29	23	0	2	17
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
6839	21/12/2006	11:21:00	0.242	0	241.67	1	0	0	0
6840	21/12/2006	11:22:00	0.244	0	242.29	1	0	0	0
6841	21/12/2006	11:23:00	?	?	?	?	?	?	
6842	21/12/2006	11:24:00	?	?	?	?	?	?	
6843	21/12/2006	11:25:00	0.246	0	241.74	1	0	0	0
6844	21/12/2006	11:26:00	0.246	0	241.83	1	0	0	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

Since this "household_power_consumption" dataset is in minutes steps with some missing data in middle. For an example of first group while the missing data encountered is highlighted in yellow (on Table III). Likewise, a portion of rest of the continuous groups are presented in Fig 01.

Where the second group starts at 6843 and 19726; means it has 19726-6843=12883

6839	[[4.216, 0.418, 234.84, 18.4, 0, ...], [5.36, 0.436, 233.63, 23.0, 0, ...
12883	[[0.246, 0.0, 241.74, 1.0, 0, ...], [0.246, 0.0, 241.83, 1.0, 0, ...], ...
22106	[[6.816, 0.378, 236.89, 28.8, 37, ...], [7.026, 0.384, 234.45, 30.0, 3 ...
20076	[[3.204, 0.202, 232.7, 13.8, 0, ...], [3.254, 0.17, 232.79, 14.0, 0, ...
36344	[[2.296, 0.298, 234.74, 9.8, 0, ...], [2.158, 0.172, 235.1, 9.2, 0, ...
44332	[[4.896, 0.13, 236.19, 20.6, 0, ...], [4.522, 0.122, 237.05, 19.4, 0, ...
47908	[[0.676, 0.328, 242.22, 3.0, 0, ...], [0.62, 0.276, 242.15, 2.8, 0, ...
46370	[[0.232, 0.078, 234.57, 1.0, 0, ...], [0.232, 0.078, 234.68, 1.0, 0, ...
7361	[[0.448, 0.088, 234.67, 2.0, 0, ...], [0.384, 0.086, 235.09, 1.8, 0, ...
3707	[[2.416, 0.304, 233.42, 10.4, 1, ...], [2.496, 0.202, 232.94, 10.6, 1, ...
20	[[1.592, 0.856, 240.76, 7.4, 0, ...], [1.886, 0.854, 240.58, 9.2, 0, ...
349	[[1.026, 0.486, 241.85, 5.0, 0, ...], [0.266, 0.0, 242.49, 1.2, 0, ...
1	[[0.91, 0.086, 241.35, 3.8, 0, ...]]
14022	[[1.586, 0.224, 239.09, 6.6, 0, ...], [1.44, 0.112, 239.08, 6.0, 0, ...
14263	[[1.526, 0.09, 239.0, 6.4, 0, ...], [1.5, 0.09, 238.98, 6.2, 0, ...], ...

Fig. 01: Groups of continuous dataset portion of "household_power_consumption" dataset

Since this is a biggest data, evaluation carried on the first group(group_0) 6839 data.

To train the model first 70% of the portion of group_0 data and group_0's rest (30%) used as test set.

Fall 2020 evaluation household_power_consumption (group_0) data results

The initial evaluation of different feeding time steps(as 15, 30, 45 and 60) hidden nodes(1,2,4,8,16,32,64 or 128) with the different prediction steps(15, 30, 45 and 60); and their corresponding results is shown in the following results.

From the results of the given dataset prediction (household power consumption); the errors are averaged all-over the prediction steps.

Base model evaluation results summery
https://drive.google.com/file/d/1dRPJtShnuzzXGt3V0RvQXJDShBMO3VFC/view?usp=sharing

Stack-2 evaluation results summery
https://drive.google.com/file/d/1u2_HraQpI6Z8d48puGnBFJsRz6I4BxaU/view?usp=sharing

In both cases base and stack model spreadsheets, please check the page "1st_min_error_among_test_train" summarized the first occurrence of the minimal error. It is clearly seen the test set with number of nodes <=16 performs better contrast to the training set performance. This data set may show the overfitting in time series data compare to the earlier “pollution_2.csv” data used in the initial evaluation.

Final comparison and conclusion

Table. IV compare the different model results. From this result of test dataset of group_0; up to stack 2, it can be clearly seen all the test sets’ best performed models’ number of hidden nodes perform better are in 16 or under 16(like 8, or 4 or 2). It validates the calculation. But **Stack-3 results** does not support the hidden size calculation. As it seen from the **Table IV**, the results the 128 performs better in all cases. However, the overall performance comparison supports the node calculation (because the best performance of test set is obtained in base model as shown in Table. IV).

Stack-3 evaluation results summery
https://drive.google.com/file/d/17Y2Uui9dCP3ijzId75vwZbb1nffsqCwK/view?usp=sharing

Table IV: Comparison of different model’s base and stacked LSTM with their normalized results

	Base						Stack-2						Stack-3					
				overall normalized scale						overall normalized scale						overall normalized scale		
Train/Test	given time step	Prediction steps	Hidden size	MAE	MSE	RMSE	given time step	Prediction steps	Hidden size	MAE	MSE	RMSE	given time step	Prediction steps	Hidden size	MAE	MSE	RMSE
test	60	15	16	0.049	0.010	0.102	15	15	16	0.053	0.012	0.111	15	15	128	0.052	0.012	0.110
test	15	30	8	0.064	0.015	0.124	15	30	8	0.068	0.017	0.129	15	30	128	0.066	0.016	0.128
test	60	45	8	0.074	0.019	0.136	60	45	8	0.080	0.020	0.140	15	45	64	0.078	0.020	0.142
test	60	60	8	0.083	0.021	0.145	30	60	2	0.085	0.022	0.149	60	60	64	0.088	0.022	0.149
train	45	15	128	0.054	0.011	0.106	45	15	64	0.054	0.011	0.105	60	15	32	0.055	0.011	0.107
train	45	30	32	0.066	0.015	0.122	45	30	64	0.067	0.015	0.123	45	30	64	0.069	0.016	0.126
train	45	45	128	0.071	0.017	0.130	45	45	32	0.070	0.017	0.129	45	45	64	0.074	0.018	0.134
train	45	60	128	0.077	0.019	0.136	60	60	64	0.072	0.017	0.132	60	60	32	0.080	0.020	0.141