CS 542 Machine Learning – SICK. Team 3
Learning from Sensor Data

Changlong Jiang; Yanjiang Dong; Gaomeizhu Qu, Yuxuan Su;
Department of Computer Science, College of Art & Science, Boston University, Boston, MA.

Abstract

Our project is to investigate potential rules from the dataset of package information detected by sensors. Our goal here is to get deeper insight of logistic system sensor to help improve the efficiency or throughput from what we learn in the dataset.

Our goals include analyzing relationship between package information trends and labels such as Legal For Trade (LFT) and Not Legal For Trade (NotLFT) and identifying the outliers in the data which caused by certain errors.

To sum up, We need to build a classification model whose output label is LFT and the input attributes are other conditions and information. We also need to detect outlier in gap and other data

Dataset and Metric

In our dataset, each package data has plenty of attributes, including package size, package weight, belt speed and package conditions. For preprocessing, we **unified units, dropped useless attributes** and got different shape of numpy array for multi-purpose. Finally, we implemented **normalization** on our data to the range of [-1:1].

Totally we have almost **1.5 million data**, we use 50,000 data at the end of our dataset to test our algorithm. Therefore, the ratio of training data and test data is approximately 28/1.

The metrics that we used is F1 score, where the F1 score is defined as:

$$\frac{1}{f1score} = \frac{1}{precision} + \frac{1}{recall}$$

We use Classification metrics from scikit-learn as baseline method.

Approach

Baseline Approach

Our baseline approach is using some classification algorithm from scikit-learn library such as logistic regression, random forest, SVM and MLP.

Final Approach

We implement LSTM algorithm as our final model. Our loss function which is described below:

$$L_{loss} = -(ylog(p) + (1-y)\,log(1-p))$$

y is binary indicator (0 or 1) if 'LFT' label is correct classification for observation, p is predicted probability of this observation is of class 'LFT'.

Our input array is 207-columns which include all the information of previous 10 packages. Our output is the 1-D array describe the 'LFT' label of current package.

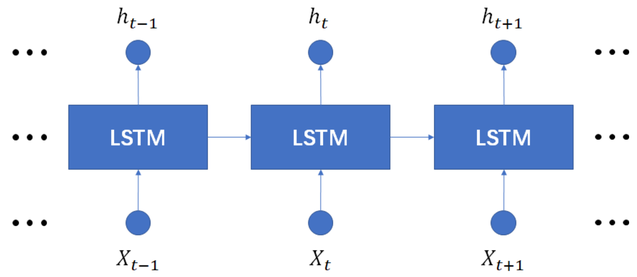


Figure 1. LSTM Model

Evaluation

We first implement our baseline approach to get more insights about package information and package trend . We find out the 'LFT' label is directly related to 'Valid_Read', 'Valid_Dim' and 'Irreg' label of current package. Then we remove these three labels because they might be useless to evaluating 'LFT' label. The result without these three labels of our baseline model is shown in Table 1.

Evaluation	logistic regression	Random forest	SVM	MLP
F1 score	0.80	0.82	0.80	0.81

Table 1. Baseline Method f1 score

For our time-series LSTM model. We start with using the information of previous 9 single packages and current package without these three labels. At this moment, we tune our hyperparameter epoch=10, neurons =10, batch size=32, the F1 score here is 0.9577 which is already higher than our baseline algorithm.

We then try several combinations of hyper-parameters to find the best combinations. The comparisons of hyper-parameters is shown in Table 2.

	epoch=10 neurons=10 batch size=32 previous packages=10	epoch=20 neurons=10 batch size=32 previous packages=10	epoch=20 neurons=20 batch size=64 previous packages=10	epoch=20 neurons=10 batch size=64 previous packages=10	epoch=20 neurons=10 batch size=64 previous packages=20
F1 score	0.9577	0.9673	0.9565	0.9812	0.9937
Precision	0.9221	0.9324	0.9292	0.9868	0.9879
Recall	0.9999	0.9999	0.9999	1	1
Loss	0.2926	0.1944	1.2268	0.0498	0.0355

Table 2. Hyper-parameter Evaluation

In the end, we choose epoch=20, neurons=10 , batch size=64 and previous packages=20, the loss and f1 curve is shown in figure 2, 3.

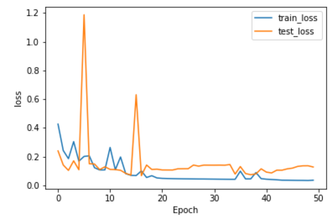


Figure 2. LSTM Model Loss

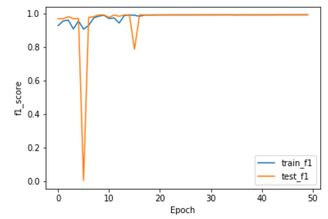


Figure 3. LSTM Model f1 score

In order to consider object time interval's effect to current model, we added another column called 'time-Interval' into the input data which is calculated by adjacent objects' 'timestamp'. Also we deleted the first object's data in each day to avoid the effect of abnormal 'time-Interval' data. the result is shown in figure 4, 5.

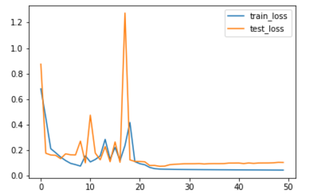


Figure 4. LSTM Model with Time-Interval Loss

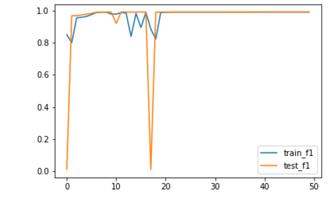


Figure 5. LSTM Model with Time-Interval f1 score

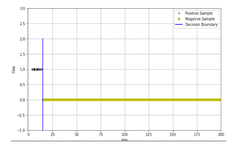
Gap investigation -- Anomaly detection

To get the boundary for Gap information specific to detect outliers, we implemented SVM model with linear kernel.

SVM module --- oga" and it's "Gap" condition

Input: X _training, training set of vector 'oga' data .

Output: threshold = -intercept / coef = 15.129



Evaluation:

In test set(x_test, y_test), only 3 outliers are detected.

The mean accuracy score is 0.9998. The f1 score is 1.0.

Conclusions

We developed a LSTM model for 'LFT' classification with package trend information in this project. This model learn from time-series multiple variable we extracted from package trend and achieve around 0.99 F1 score in our final test, comparing with 0.82 in our baseline model. In the future, we will continue to explore how to deal with time-series data with imbalanced distribution of class label.

Contact

Email: cljiang@bu.edu, dongyj@bu.edu, sylviaqu@bu.edu, suyuxuan@bu.edu
Website: https://github.com/dongyj1/Sick_Team_3

References

1. Thomas G. Dietterich: Machine Learning for Sequential Data: A Review, Oregon State University.
2. Ch9, Speech and Language Processing. Daniel Jurafsky & James H. Martin. Draft of August 7, 2017.
3. Dr. Jason Brownlee. (2017, August 14 Published). Multivariate Time Series Forecasting with LSTMs in Keras. Retrieved from https://machinelearningmastery.com/.
4. https://machinelearningmastery.com/tune-lstm-hyperparameters-keras-time-series-forecasting/