Fraud Detection with Sequential User Behavior

Background

- With the development of smart phone and Internet, now we can record the page view data of each customer
- Thanks to the improvement on Deep Learning, we could deal with the sequence data more efficiency

Data Description

1. Basic information features (Non-sequential features)

Features	Description
label (result)	Reflect if the application defrauds or not
overdue	Reflect how many days overdue
new_client	Reflect if the applicant is new or not
over_time	Reflect the submission time of application

Table 1 Non-sequential features and descriptions

Data Description

2. Page view behavior features (Sequential features)

Features	Description
pname	The category of page belongs to
pstime	The starting viewing time on this page
petime	The ending viewing time on this page
pid	The process id
sid	The session id

Table 2 Sequential features and descriptions

Data Processing

- Basic information features (Non-sequential features)
  - Got each sample as a dimension binary variable

- Page view behavior features(Sequential features)
  - Pid & Sid: 0 or 1 to reflect if change happens
  - Pstime & Pestime: be add/mins to get page stay time and page lag time
  - Pname: Three ways to deal with

Data Processing

Pname

- Label encoding

  Get 1 column which among 0-12

- One hot encoding

  Get 12 columns which are 0 or 1

- Word2vec (New Method)

  Get 50 columns (More details in the following part)

Data Processing

The length of sequence features (For LSTM timesteps)

Model Performance on Higher-Income Dataset	AUC	KS score
Model 4 using timesteps with 60	0.59	0.1487
Model 4 using timesteps with 20	0.58	0.1942

Table 3 Performance Comparison by Timesteps

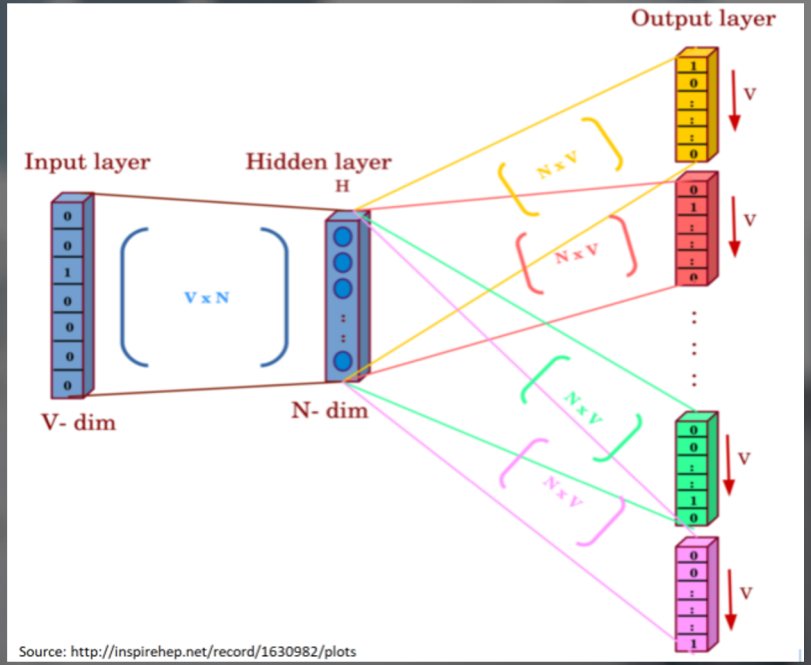
Improvement on sequence embedding

The last group has tried feature extraction by RNN(LSTM). For the input of the LSTM layers, they used one-hot **and** label encoding on the page type. These methods are simple and common ones

In order to further explore this part, our group tried word2vec embedding on page types to make new sequence embedding input for LSTM layers

Word2vec basic ideas

Word2vec is a common technique in NLP. The basic idea of word2vec embedding is to use two-layer shallow neural networks to train a text document (a document consists of one or multiple sentences) and then find an appropriate vector representation for each word of this document



Word2vec application in our project

	Word2vec in a normal situation	**Word2vec in our project**
Input format	Text document (consists of sentences)	Page type sequence
Goal	Find vector representation of each word	Find vector representation of each page type
Hierarchy of input	Document > Sentence > Word	All user sequence > single user sequence > one page type

Word2vec result analysis

Model Performance on Lower-Income Dataset	AUC	KS score
Model 4 using sequence embedding 1	0.61	0.206
Model 4 using Word2vec embedding	0.60	0.177

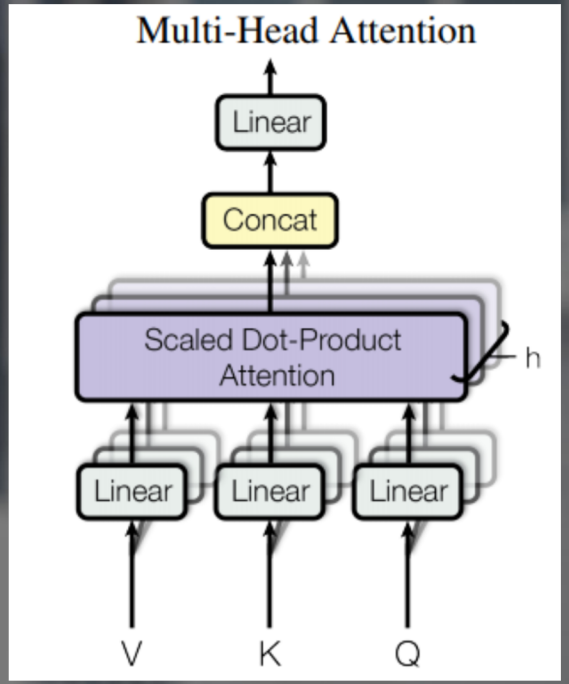
Table 4 Performance of model 4 when using word2vec or sequence embedding 1

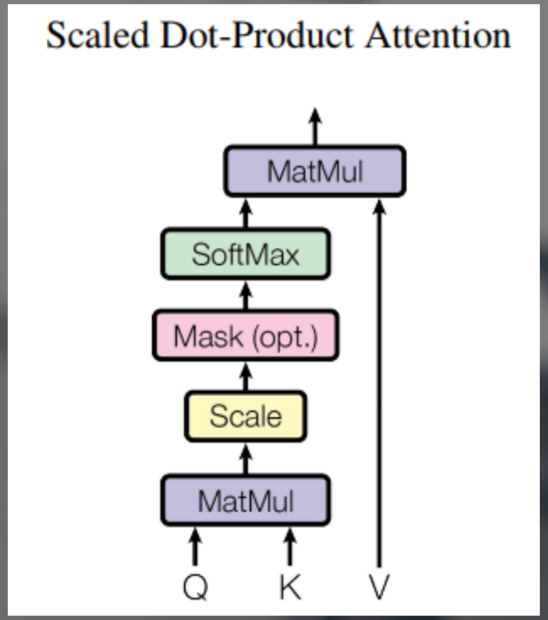
The reason we compare the performance of model 4 instead of other model architectures is that model 4 is the best architecture we found at present

Transformer Model with Attention Mechanism

Main advantages

- Layer outputs can be calculated in parallel in the form of multi heads
- Long-range behavioral sequences can be learnt



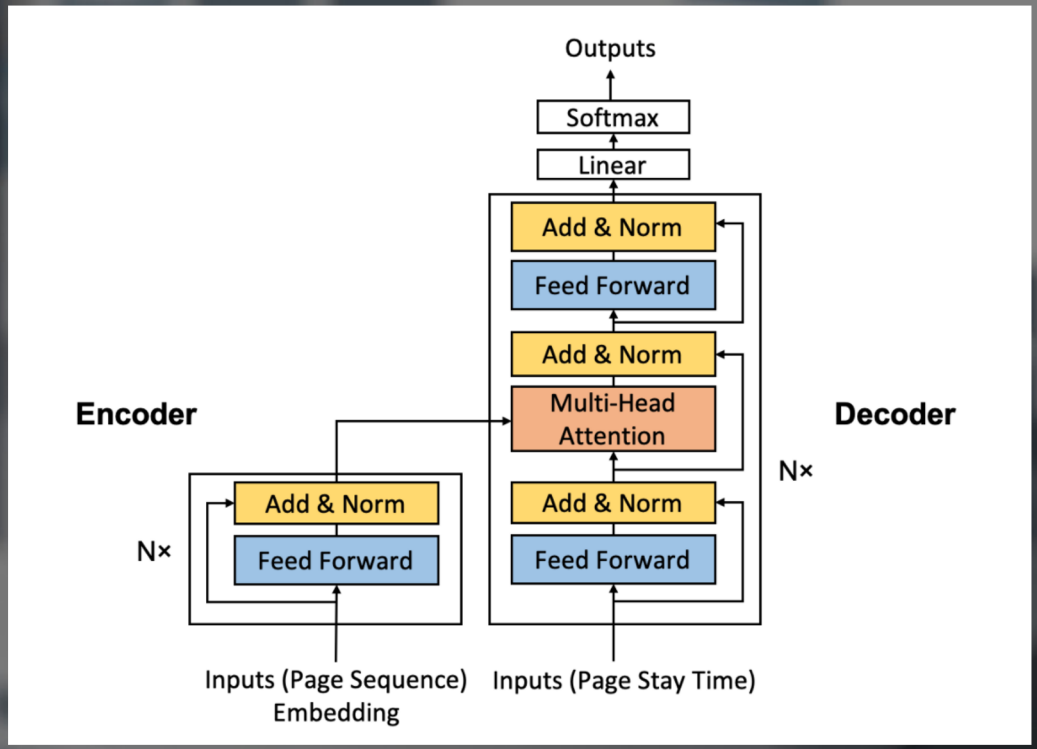


Transformer Model with Attention Mechanism

Encoder

- Input Embedding
- N encoder layers
  - The point wise feed forward network
  - Layer normalization

OUTPUT: each web page behavior in the sequence is then generated and ready to enter the decoder as the input

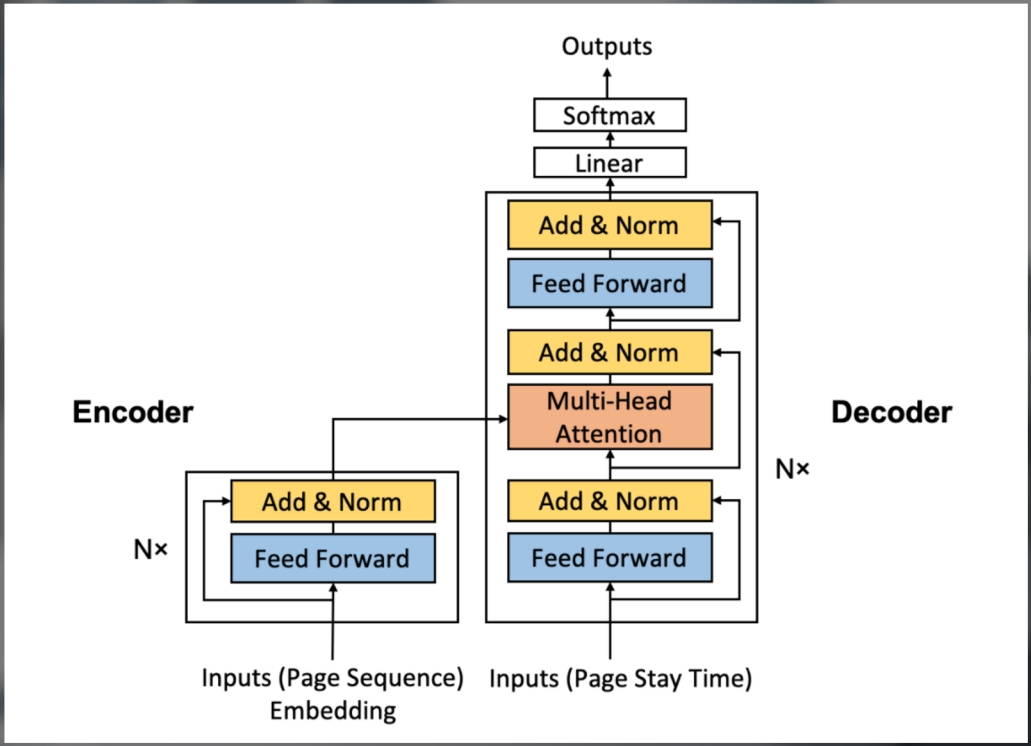


Transformer Model with Attention Mechanism

Decoder

- Input
- N decoder layers
  - The point wise feed forward network
  - Layer normalization
  - Multi-head attention layer

Decoder predicts whether the customer defaults by looking at the encoder output of the “web page sequence” behavior and self-attending to its own output of the “page stay time” behavior



Transformer result analysis

Model Performance on Both Datasets	AUC	KS Score
Transformer Model On Higher-Income Test Data	0.56	0.107
Transformer Model On Lower-Income Test Data	0.60	0.163

Table 5 Performance of Transformer on Different Datasets

The result is not as expected that we think this is due to the lack of feature dimension. Only the sequence data itself may not include enough distribution information.

Conclusions

- We use Word2Vec embedding as a new feature extraction way to better express the behavior sequential data
- We use a novel model Transformer for its advantages compared to CNN and LSTM for sequential data. We applied multi-head attention to fix some problems with LSTM in longer memory and efficiency problems.

Future Extensions

- Expands Transformer into more general way. The subnets could not only be the vanilla multi-head attention layer, but ensemble two LSTM subnet is also a feasible way. We could learn from ResNet to ensemble multiple relatively networks.
- For the Transformer model, it offers a new way to contact the web sequence data and page stay time. Without directly adding a DNN layer, the output of the Transformer model could be the same shape as web sequence data. The output which shape is the same as sequence data now was added by the factor of page stay time. We hope the next group could treat this output as an input to an LSTM model and detect if there is an improvement compared to LSTM model with web sequence as input.

Future Extensions

- Also, generator models could be used to solve the imbalanced data problem. Such as the One Class GAN model that the previous group has tried could still be an attemptable method.
- Though we tried several networks to predict the result of fraud behavior, the result is not as expected. We have complemented the feature extraction ways of the previous group, but the result is still not good enough. More ways of feature extraction could be explored. As in the sequential data, there’re 50 vector dimensions. When other non-sequential data is concatenated to it, the performance is not good. In the future more ways could be tried to map non-sequential features such as applied time with sequential features.


Thank You!