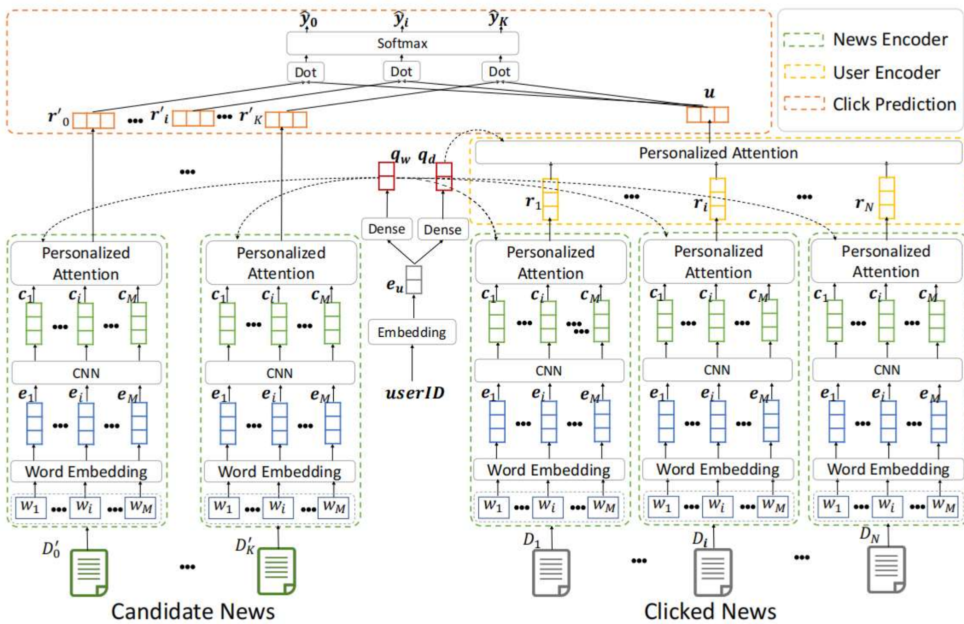
Ⅰ Brief description of the problem

This project aims to use the news recommendation model NPA to solve the MIND news recommendation problem released by Microsoft. News recommendation is very important to help users find interesting news and alleviate information overload. Different users usually have different interests, and the same user may also have different interests, so different users can follow the same article and focus on different aspects. To solve this problem, this project chose the NPA model proposed in the paper "NPA: Neural News Recommendation with Personalized Attention". The core of this method lies in the news representation model and user representation model, while adding word-level and news-level attention mechanisms to obtain important words and news articles for users.

Ⅱ Principle of NPA model



1. Model overview

The NPA model is mainly divided into three parts, news coding, user coding and click prediction, and news is divided into candidate news and clicked news. In the data set, the model divides news into history and impressions, where history represents the user's historical click record, and impressions represents the news information clicked by the user in the news list at the current moment, and continues to be divided into positive and negative on impressions, and positive represents the news that has been clicked, negative indicates that the news was not clicked. Then Clicked News uses the news titles in history as input, and Candidate News uses 1 postive and K negatives in impressions to form K+1 news titles as input, then the training process is converted into a classification problem here, so that it we can use the Softmax function and use the maximum likelihood estimation method to iterate the parameters at the end.

2. Model explanation

(1) News Encoder

The NPA model is divided into three steps on the News Encoder, namely Word Embedding, CNN, and Personalized Attention. The purpose of Word Embedding is to vectorize words, so that the relationship between words can be numerically estimated by using the calculation method between vectors. The NPA model uses the one-dimensional convolution of CNN to obtain the local context information in the title, so as to achieve keyword extraction. Personalized Attention in News Encoder focuses on a single news title, and adds userID vectorization to the operation, thereby embedding user preference parameters into the model. Here, the modeling of the user's degree of attention to different keywords in each news title has been realized.

(2) User Encoder

The NPA model still uses Personalized Attention to embed userID into the model on User Encoder, but here it focuses on all clicked news, and uses user information and News Encoder output as the input of User Encoder, so as to realize the user's degree of attention to different news modeling.

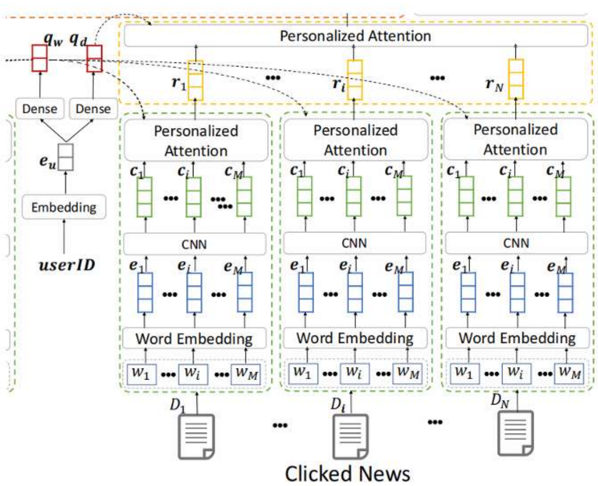
Through the above explanation, we can see the embodiment of the news level and word level that the author said in the paper. Word-level modeling is embodied in News Encoder, and news-level modeling is embodied in User Encoder.

(3) Click Prediction

In the model overview, we mentioned that the NPA model converts the problem into a classification problem, so the click prediction is also performed in the same way as the classification problem. The input of the module is the output of News Encoder and UserEncoder, and the parameters are adjusted by the Softmax function using the maximum likelihood estimation method to complete the model construction.

(4) Since the formula used in the Personalized Attention layer in this model is relatively complex, here is a detailed description of the Personalized Attention layer in the News Encoder module.

The formula used in the Personalized Attention layer is as follows:



$$q_w = ReLU(V_w \times e_u + v_w) \quad (1)$$

$$a_i = c_i^T tanh(W_p \times q_w + b_p) \quad (2)$$

$$\alpha_i = \frac{\exp(a_i)}{\sum_{j=1}^{M} \exp(a_j)}, \quad (3)$$

$$\mathbf{r}_i = \sum_{j=1}^{M} \alpha_j \mathbf{c}_j. \quad (4)$$

First, the formulas (1), (2) map eu to the same dimension as the keyword vector ci through the activation function ReLU and tanh, so the weight of each word is obtained through the inner product with the keyword vector ci, and then normalized by (3) softmax function, then the obtained αi is the weight of each word vector cj, and each cj is calculated by (4) According to the sum of the weights, the news code ri is obtained.

3. Model thinking

(1) Word Embedding

NPA chose to use Word Embedding instead of one-hot encoding. The reason is that one-hot cannot reflect the relationship between words and words, and there is an obvious relationship between news headlines and words, so Word Embedding is chosen.

(2) CNN

In NLP, CNN uses one-dimensional convolution, which is characterized in that it only performs convolution in one direction, the text vector is one-dimensional in the direction, so choosing CNN can extract local keywords. At the same time, RNN is also widely used in NLP. In my view, the choice of the two lies in whether to pay attention to continuous information. CNN focuses on local information, so it can extract keywords, while RNN focuses on continuous information, it can be widely used in the field of automatic text generation, but here only keywords need to be extracted, so NPA chooses CNN.

(3) Personalized Attention

NPA uses Personalized Attention twice, thus achieving different levels of personalized embedding. Since the same news and the same title have different appeals to different users, NPA embeds the user id into the model to achieve personalized recommendation, and since the model is modeled from two levels, it has higher recommendation accuracy.

Ⅲ Dataset adaptation

1. News dataset

Representation of each row in CNN's news dataset:

0	1	2	3	4	5	6	7
id	Article id	Article filed	Concrete filed	Title id	Content id	Title	Content

Each col is divided by a tab key.

Representation of each row in MIND news dataset:

0	1	2	3	4	5
Article id	Article field	Concrete field	Title	Content	Other information

Replace the 1-Article id, 6-Title, and 7-Content used in the original news collection with the 0-Article id, 3-Title, and 4-content in the MIND news collection.

2. User Clicked dataset

The representation of each row of the CNN user dataset:

0	1	2	3
User Id	Gender	Click record in 2018.12	Click record in 2019.01

0	#TAB#	1	#TAB#	2
Click record		Unclicked record		Click time

The representation of each row of the MIND user dataset:

0	1	2	3	4
User number	User Id	Record date	Clicked record	Unclicked record

In the original CNN data set, the author uses the user's click record in 2018.12 as the training set, and uses the user's click record in 2019.1 as the test set.

Therefore, in the MIND user set, our processing method is to take 30% of click records and non-click records respectively as the test set, and the other 70% as the training set.

Ⅳ Result and discussion

```
2505/2505 [==============================] - 1052s 411ms/step - loss: 0.1428 - acc: 0.9498
C:\Users\niuqv\AppData\Local\Programs\Python\Python39\lib\site-packages\tensorflow\python\kera
  warnings.warn('`Model.predict_generator` is deprecated and '
23615/23615 [==============================] - 2233s 94ms/step
```

```
0.9796482713953719 0.24812510649892963 0.9585316316627281 0.9690925814530476
```

```
2505/2505 [==============================] - 1031s 411ms/step - loss: 0.0362 - acc: 0.9880
C:\Users\niuqv\AppData\Local\Programs\Python\Python39\lib\site-packages\tensorflow\python\keras
  warnings.warn('`Model.predict_generator` is deprecated and '
23615/23615 [==============================] - 2231s 94ms/step
```

```
0.9887706920161786 0.256788375065173 0.9769224523466147 0.9824976897409163
```

```
2505/2505 [==============================] - 1029s 411ms/step - loss: 0.0250 - acc: 0.9917
C:\Users\niuqv\AppData\Local\Programs\Python\Python39\lib\site-packages\tensorflow\python\keras\eng
  warnings.warn('`Model.predict_generator` is deprecated and '
23615/23615 [==============================] - 2234s 95ms/step
```

```
0.9868980940911096 0.25597604695013954 0.9723315468453325 0.979652030215127
```

	AUC	MRR	nDCG@5	nDCG@10
First Iteration	0.9796*	0.2481	0.9583	0.9691
Second Iteration	0.9888	0.2568	0.9796	0.9825
Third Iteration	0.9869	0.2560	0.9723	0.9797

We can see that NPA, an attention-based news recommendation algorithm, is very effective on the Microsoft dataset MIND. The principles of each evaluation mechanism are as followed:

AUC: AUC (Area Under Curve) is defined as the area under the ROC curve; the value range of AUC is between 0.5 and 1. The closer the AUC is to 1.0, the higher the authenticity of the detection method; when it is equal to 0.5, the authenticity is the lowest and has no application value.

MRR: MRR (Mean reciprocal rank) is an internationally used mechanism for evaluating search algorithms, that is, the first result matches with a score of 1, the second match score is 0.5, and the nth match score is 1/n , with a score of 0 if there is no matching sentence. The final score is the sum of all scores.

nDCG@5: Normalized Discounted Cumulative Gain, which is an internationally used measure of search algorithms. @5 means that it is only interested in the top 5 results.

nDCG@10: The algorithm is the same as nDCG@5, the difference is that this indicator is interested in the top 10 results.

From the above result, we can find:

a. There is not much difference in the accuracy rate between the top 5 models and the top 10 models, but after multiple iterations, the nDCG@10 index of the top 10 calculations will always be larger, which indicates that the accuracy of the top 10 statistics is higher.

b. The MRR indicator is around 0.25, which proves that most of the prediction results of the test set can match at the third result, which happens to be in the top 5, and the high accuracy value of nDCG@5 also indicates this.

c. Our AUC index is close to 1, indicating that when the data set is similar to the MIND data set, our model is very practical and the detection method is very authentic.