

# Heat Prediction of High Energy Physical Data Based on LSTM Recurrent Neural Network

Zhenjing Cheng<sup>1,2</sup>, Lu Wang<sup>1,2</sup>, Yaodong Cheng<sup>1,2,3</sup>, Gang CHEN<sup>1</sup>

\*chengzj@ihep.ac.cn, wanglu@ihep.ac.cn, chyd@ihep.ac.cn, gang.chen@ihep.ac.cn

<sup>1</sup>IHEP computing center, 19B Yuquan Road, Beijing 100049, China

<sup>2</sup>University of Chinese Academy of Sciences, 19A Yuquan Road, Beijing 100049, China

<sup>3</sup>Tianfu Cosmic Ray Research Center, IHEP, 9 Renmin South Road, Chengdu 620500, Sichuan, China

\*chengzj@ihep.ac.cn

**Abstract.** High-energy physics computing is a typical data-intensive calculation. Each year, petabytes of data needs to be analyzed, and data access performance is increasingly demanding. The tiered storage system scheme for building a unified namespace has been widely adopted. Generally, data is stored on storage devices with different performances and different prices according to different access frequency. When the heat of the data changes, the data is then migrated to the appropriate storage tier. At present, heuristic algorithms based on artificial experience are widely used in data heat prediction. Due to the differences in computing models of different users, the accuracy of prediction is low. A method for predicting future access popularity based on file access characteristics with the help of LSTM deep learning algorithm is proposed as the basis for data migration in hierarchical storage. This paper uses the real data of high-energy physics experiment LHAASO as an example for comparative testing. The results show that under the same test conditions, the model has higher prediction accuracy and stronger applicability than existing prediction models.

## 1. Introduction

Large-scale scientific experiments such as particle physics, particle astrophysics, and radiation sources are inseparable from large-scale data processing and analysis. High-energy physics scientific computing is a typical data-intensive application, which is characterized by observing rare cases from massive data and further searching for new scientific discoveries [1]. The I/O access performance of the storage system is important for computing efficiency. More and more scientific experimental data also put forward higher requirements for mass storage systems, such as capacity, reliability, scalability, and cost performance.

High-energy physics experiments generate a huge amount of data every year and need to be stored for a long time. Historically, the field has used hierarchical data management methods based on tape and disk systems. In the future, we plan to introduce solid-state hard disk SSD as a separate fast storage layer and build a three-tier storage system with ordinary mechanical hard disks and tape libraries. Therefore, there are problems with data classification, data placement, and data migration.

In the traditional hierarchical storage management process of high energy physics, data file migration sometimes requires the administrator to specify and manually confirm the migrating file list in IHEP. It is heavily dependent on experience, requires a lot of labor costs, and the overall storage system efficiency is not high. The increase in data volume leads to the continuous growth of the system scale, which in turn causes the complexity of traditional data migration models to increase dramatically, making it difficult to manage efficiently based on human experience. Therefore, the automatic migration of data between all levels of storage is another major problem facing researchers.

## 2. Research status

Effective migration can help data to be distributed in a reasonable storage hierarchy and improve storage system efficiency. The incorrect migration will bring additional read and write load, which will

affect normal system I/O. Common data migration strategies, or hot file selection methods include LFU (least frequently used), most recently used. Common data migration strategies, or cold file selection methods include LRU, FIFO, file-aging. These methods are essentially based on access history statistics of the storage system and historical data access frequency is one of the important indicators.

In recent years, deep learning technology has become popular, and the training method is greatly different from traditional algorithms, thereby breaking the limitations of traditional neural networks on the number of hidden layers, the number of nodes in each layer, and has strong self-learning and non-linear mapping capabilities. Among various deep neural network models, Recurrent Neural Network (RNN) introduces the concept of timing in the design of the network structure, and at the same time combines network nodes with storage capabilities to make the model have memory like a human [2]. Recurrent neural networks can abstract input timing signals layer by layer and extract features [3]. At present, it is used to construct time series data in speech recognition [4], machine translation, power load prediction, fault prediction and other fields. Many breakthroughs have been made in the model, but their application in data access prediction is very limited. For data access heat prediction of tiered storage, no similar case has been found.

### 3. Related work

#### 3.1 High energy physics computing environment

High energy physics computing is a process of observing rare cases from massive data. At present, clusters method is widely used in the field of high-energy physics to reduce system cost and improve scalability. The computing center of the High Energy Institute of the Chinese Academy of Sciences separates computing clusters and storage clusters, and has built computing scheduling clusters using HTCondor, high-performance storage clusters using EOS and Lustre distributed file systems, and high-speed connection networks.

#### 3.2 LSTM(long-term and short-term memory neural network)

As one of the sub-fields of artificial intelligence, machine learning focuses on the methods of letting computers learn by themselves. It can be divided into supervised learning, unsupervised learning, reinforcement learning. Deep learning is a method of machine learning, which refers to a method based on the Deep Neural Network (DNN). Deep learning is evolved from the traditional neural network (NN). By referring to the human neuron mechanism, it simulates the process of thinking and cognition. It has powerful non-linear mapping and generalization capabilities.

Most neural networks belong to the Feed Forward Neural Network (FNN); no matter how many hidden layers the network has, the neurons in each layer only accept the input of the connected neurons in the previous layer, and the output produced is only passed to the next connected neuron. The advantage of this model is that it can produce output results in real time, but the disadvantage is that it can only use the information of the current moment. The correlation of information at different times is discarded by the FNN model. It cannot be used to process time series data.

Jordan and Elman proposed a Recurrent Neural Network (RNN) model[5]. The model is memorable, and the relevant data of the hidden layer neurons at the previous moment are also used as the input of the model at the next moment. The data in the hidden layer neurons will be updated every time, thus acting as a memory unit. This design ensures that the RNN can, in principle, learn the data correlation of any length in time and distance and better handle various time series data of different lengths.

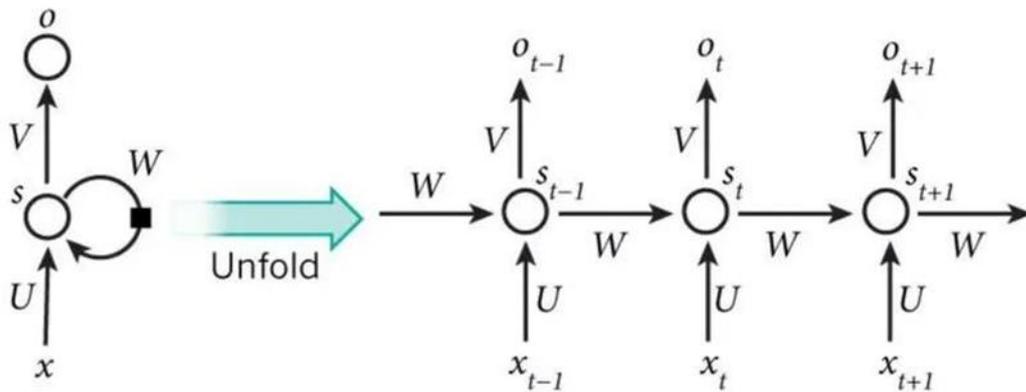


Figure 1. Recurrent Neural Network[5]

#### 4. Design and Implementation

As shown in Figure 2, the data access heat prediction system interacts with existing high-energy physical storage systems such as Lustre and EOS, and consists of feature collection nodes, a central database, and model training nodes. An I/O log collection component is deployed on each file storage server (FST). After filtering out irrelevant information, it is stored in the central key-value database in the format of <timestamp, parameter field, value>. File access characteristic data is calculated, integrated, normalized, batch processed, and written into the online data queue for model training. Model training is based on deep learning frameworks such as Tensorflow [6] and sklearn [7]. The trained model structure is stored in the local file system for persistent storage. The data migration system of the computing center periodically scans the file list in tapes, mechanical hard disks, and solid-state hard disks in the background and performs migration actions based on the output of the file access prediction system and migration conditions set by the administrator.

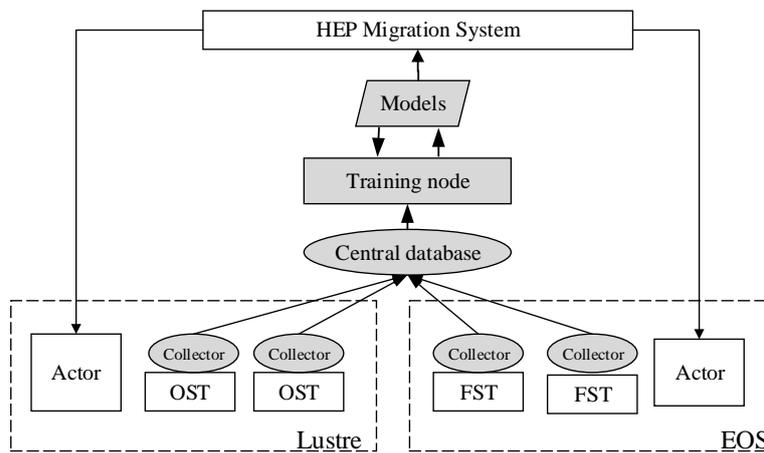


Figure 2 High-energy physical data heat prediction and migration architecture

##### 4.1 Model Input

When constructing the access feature vector, various types of file operation records need to be filtered in the file system log and stored persistently. The EOS storage system of the computing center has tens of millions of files and petabytes of data. Hundreds of thousands of file access logs are recorded every day. They need to be organized in three dimensions: file name, operation type, and time window. This article uses the column-oriented key-value distributed database HBase to store the file access features. The rowkey design is shown below.

Rowkey byte 0 is a file operation type field, such as file opening, closing, reading and writing. The first byte to the 16th byte are the file name hash value field. The hashed file name has a uniform length, thereby increasing the probability of data being distributed evenly in each region, and achieving load balancing to improve query efficiency. The 17th to 20th bytes are file operation time fields. The 21st to 23rd bytes are extension fields, which record the username, file operation permissions, and so on.

#### 4.2 Model Output

The file access frequency prediction problem can be regarded as a type of continuous variable prediction problem. Traditionally, this kind of problem can use the method of regression analysis [8] to make corresponding file migration decisions based on the prediction results to minimize the migration cost. However, in the actual storage scenario, there are a large number of files, and the timing access rules of different files are very different. It is impossible to train a regression model for each file, and it has the disadvantages of complicated calculation and poor adaptive ability.

Assuming that the hierarchical storage system is divided into  $n$  storage levels, there may be  $n$  different migration decisions for each file (migrate into this layer or keep it in the original storage layer). In order to reduce the impact of migration on the user's normal data access in hierarchical storage, changes in the file access frequency within a small range do not change which storage level the file should be migrated to. This article predicts the popularity of the file, that is, the interval within which the access frequency falls. In this way, the prediction problem can be reformulated as a classification problem. At the same time, this article uses the file's access level (0,1, ...,  $n-1$ ) to label each access feature sequence in the training set. The  $n$  heat tags are converted to a sparse vector consisting of 0 and 1 using one-hot encoding, as shown below

$$Y_i = \{0, 0, 0, \dots, 1, \dots, 0\}, \quad (1)$$

#### 4.3 Model Training

Model training mainly uses the hidden layer of the LSTM network as the research object. In the model input layer, define the original file access feature order as

$$F_o = \{f_1, f_2, \dots, f_n\}, \quad (2)$$

The dynamic time window segmentation method is used to process "Fo". If the length of the dynamic time window is set to  $L$ , the model input after segmentation is

$$X = \{X_1, X_2, \dots, X_L\}, \quad (3)$$

The LSTM model uses the cross-entropy loss function as the loss function in the training process, which is defined as

$$Loss = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)), \quad (4)$$

We set the minimum loss function as the training target of the model. We gave a randomization seed to randomize the weights and biases in the LSTM network. Then we set the number of hidden layers and hidden nodes of the LSTM network to layers and nodes. At last, we set initial learning rate and training steps, and use Adam gradient stochastic optimization algorithm [9] to update parameters in the network.

In general, LSTM-based file access prediction model training and prediction algorithms can be summarized as follows:

Input: Fo, Y, L, layers, nodes, seed,, steps

Output: File access heat and model accuracy corresponding to the test set.

1. get X from Fo by L
2. get Xtrain and Xtest // divide the training set and test set
3.  $X_p = \min\_max(X_{train})$  // min\_max method is used to normalize the input data
4. create LSTMcell using layers, nodes and state
5. connect LSTMnet by LSTMcell and number L
6. initialize weights and biases by seed

7. for each step in steps:
8. Pred = LSTMforward (Xp)
9. Loss = CrossEntryLoss (pred, Y)
10. Update LSTMnet by Adam
11. GridSearch (layers, nodes) // Use grid search to adjust network hyperparameters
12. Optimize end and get LSTMnet \*
13. for each j in 0: length (Xtest)
14. P = LSTMforward (Xj)
15. append Po to P
16. Error measure  $\varepsilon$  (Po, Yo)

Among them, LSTMcell represents the neurons of the LSTM network, LSTMnet represents the hidden layer structure of the LSTM network, and LSTMforward represents the forward propagation process of the LSTM network.

## 5. Experiments

This article uses the data of the high-altitude cosmic ray observation experiment LHAASO [10] in EOS, a large-scale storage system in Daocheng, Sichuan, as an example. First, it introduces how to prepare the file access data set required in the experiment, trains the LSTM model to predict the file access frequency. The file access frequency threshold  $\gamma$  corresponding to the popularity is used to test the prediction accuracy of the LSTM model under different thresholds. So we could compare with the advantages and disadvantages of other current prediction models, as well as the hardware and software configuration of the experimental platform.

The data set used in this article is from the access I/O logs of files in the EOS storage system. The number of files that have been active in the past 30 days is 5,842,207. Then model training and test data sets were generated. The EOS storage cluster system logs of the computing center are periodically captured by the monitoring system to the ElasticSearch database. During the data preprocessing stage, file access characteristics are extracted from the logs and stored in the HBase database.

Taking the experimental data of LHAASO as an example, the access features extracted from the file access log in the previous 27 days are set as the input of the prediction model. The access frequency Freq of the last 3 days is divided into multiple intervals, as the output of the prediction model. Generally, a file with Freq of 0 is defined as a cold file in high-energy physical storage. The data migration system periodically dumps such files to the tape library. To further distinguish warm files from hot files, this paper defines the access frequency threshold  $\gamma$ . A file with Freq less than or equal to the threshold  $\gamma$  is defined as a warm file. The migration system periodically migrates such files to the HDD layer of the mechanical hard disk. Files with Freq greater than  $\gamma$  are defined as hot files, and the migration system periodically migrates such files to the SSD layer. Taking  $\gamma = 3$  as an example, the number of cold files in the LHAASO experimental training data set is about 95.8%, the number of warm files is about 3.06%, and the number of hot files is about 1.13%.

Table 1 Comparison of prediction accuracy of different models

| Model | Model params   | Training dataset |              |              | Test dataset |              |              | train time /h | reason time /ms |
|-------|--|------------------|--------------|--------------|--------------|--------------|--------------|---------------|-----------------|
|       |  | $\gamma=1$       | $\gamma=2$   | $\gamma=3$   | $\gamma=1$   | $\gamma=2$   | $\gamma=3$   |               |                 |
| MLP   | $\eta=0.1, \text{layers}=6, \text{nodes}=64, \text{steps}=1000$    | 0.591            | 0.658        | 0.547        | 0.662        | 0.693        | 0.634        | <u>0.78</u>   | <u>269</u>      |
| SVM   | kernel=rbf, c=5.12e+02, gamma=2.50e-01                             | 0.682            | 0.883        | 0.481        | 0.612        | 0.811        | 0.352        | 0.96          | 513             |
| LSTM  | $\eta=0.002, \text{layers}=4, \text{nodes}=128, \text{steps}=1000$ | <u>0.873</u>     | <u>0.924</u> | <u>0.856</u> | <u>0.866</u> | <u>0.897</u> | <u>0.820</u> | 1.37          | 1191            |

## 6. Conclusion and Outlook

This paper introduces a method for predicting file access popularity based on a LSTM deep learning model of hierarchical storage. It introduced content including data set preparation, file access feature construction, training and prediction. Compared with the migration method based on administrator experience and statistics, the LSTM model can more accurately predict the change in the heat of file access, thereby providing a more effective basis for file migration. This paper applies deep learning methods to the field of data migration in hierarchical storage. As this was a preliminary attempt, there are many aspects that need further exploration and research.

## References

- [1] Cao Z. A future project at tibet: the large high altitude air shower observatory (LHAASO)[J]. Chinese Physics C, 2010, 34: 249-252.
- [2] AJ Peters, EA Sindrilaru, G Adde. EOS as the present and future solution for data storage at CERN[J]. Journal of Physics Conference Series, 2015, 664(4):042042-042043.
- [3] Connor J T, Martin R D, Atlas L E. Recurrent neural networks and robust time series prediction[J]. IEEE transactions on neural networks, 1994, 5(2): 240-254.
- [4] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. MIT press, 2016.
- [5] Hochreiter S , Schmidhuber J . Long Short-Term Memory[J]. Neural computation, 1997, 9(8):1735-1780.
- [6] Greff K, Srivastava R K, Koutník J, et al. LSTM: A search space odyssey[J]. IEEE transactions on neural networks and learning systems, 2016, 28(10): 2222-2232.
- [7] Abadi M, Barham P, Chen J, et al. Tensorflow: A system for large-scale machine learning[C]//12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016: 265-283.
- [8] Swami A , Jain R . Scikit-learn: Machine Learning in Python[J]. Journal of Machine Learning Research, 2012, 12(10):2825-2830..
- [9] Vishwanathan S V M, Murty M N. SSVM: a simple SVM algorithm[C]//Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290). IEEE, 2002, 3: 2393-2398.
- [10] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.