

1 A brief background description of the universities or the departments supercomputing activities

2 Introduction of the team

2.1 Brief description of the building process of your team

Our team, NewIdea, was selected from 16 teams from our university. Together with five other teams, we represented Nanchang University in asc2018. Since our school first participated in the asc competition last year, Our school has provided a lot of financial and technical support to the construction of our schools supercomputing platform. At the same time, more and more students realized the importance of supernumeracy and became interested in it. After learning of asc2018, our college organized a seminar on supernumeracy. Surprisingly, 16 teams were formed, and many students without teams wanted to participate in the competition. After two weeks of excessive training and practice, our team successfully became one of the teams representing Nanchang University. Professor Xu Zichen and our advisor, Mr. Wang Weili, have provided great help to our team. Our captain, Gao Cuiying, played an important role in team formation and coordination, and the other four players also performed well during the training and competition.

2.2 Brief introduction of each team member

Advisor: Wang Weili Wang Weili received his bachelors degree and masters degree in computer science from Nanchang University in 2006 and 2009, and the PhD degree form Tongji Unviersity in 2015.

He is currently serves in Department of Computer, Nanchang University, China. His research interests include spatial data management and query processing, distributed computing, and geographic information systems.

Captain: Gao Cuiying(Female) I'm a junior of Information Engineering College, Nanchang University. My major is digital media technology. In the past two and a half years, I have studied the basic courses of computer science and technology, such as operating system,data structure, computer network, computer organization and structure, etc.

At the same time, I also learned some programming languages, such as C++, Java, Go, etc. In the application course, I learned database, Windows programming, website development and android development. I also learned some lessons about digital image processing and OpenGL programming. Since last October, I have started to study model optimization and learn basic mathematics knowledge such as probability theory and stochastic process. Then I studied operational research and began to try to model the problems in real life and optimize the model. At the same time, I came into contact with remote servers and high performance computing. In the following time, I will complete

the competition with the team. After that I developed a strong interest in supercomputers and wanted to do more about them.

Then I will learn more about model optimization knowledge and parallel computing knowledge. At the same time, in the process of learning and practicing, I will improve my programming ability and writing ability.

Member:Chen Jinyu (Male) Junior undergraduate student of Qianhu College, Nanchang University, male, 20 years, majoring in Electronic Engineering and Automation.

Although I am majored in Electrical Engineering, I have taken several lessons about computer science like C++ programming, discrete mathematics, data structure and etc. I have a great interest in machine learning, especially in image identification. Since 2017, I started getting torch with high performance computers, and tried to make some experiments. After the study during this time, I have a basic understanding of artificial intelligence. I also master the usage of Tensorflow, which is a deep learning framework made by Google. With the help of Tensorflow, I reproduced many wonderful models on imagenet and I am familiar with the structural features of various convolutional neural networks right now. Besides, I studied some filtering technology and programmed a simple software.

In the past years, I also got much experience by participating many competitions and won some prizes like Honorable Mention in 2017 American Mathematical Contest in Modeling and the second class in China Undergraduate Mathematical Contest in Modeling.

Member:Xiao Jianwei (Male) Junior undergraduate student of Nanchang University, male, 20 years. I am majored in Electronic Information Engineering and have taken some related courses to my major, such as C++ Programming, Data Structure, Linear Algebra and Digital circuit designing. In addition, I also participated in Electronic circuit designing contest of Nanchang University, which improved my experience in circuit designing. I am student major in Electronic information Engineering, but I own a lot of interest in computer science. With the help of my teacher, I also did some study in machine learning and learned a lot of knowledge of Computer Science. Together with my partner, I used the deep learning framework Tensorflow to build some models, such as Alexnet, VGG, ResNet and GoogleNet. I used this model to do a classification task with the ImageNet dataset. Despite the results of our models only attain an accuracy of 65 percent after 15 iterations, I still learned a lot from the task I did. Besides, I have a few knowledge of Matlab from self-learning. I did a object detection task using Matlab together with my partner. It is a simple model to detect moving object in videos. By doing this task, I learned some skill about Matlab programming. As for parallel algorithm and distribute system, I have a few knowledge from my teacher. And it is a good chance to learn supercomputing and computer architecture through ASC and also good chance to learn from other student and engineers in this area. May our team have a good achievement during ASC2018.

Member:Fang Zhu(Male) My name is Fang Zhu, male, 20 years old, from the Nanchang university, before the lake a sophomore student of the college,

my major is electronic information, this contains a large number of professional software and hardware knowledge, including the C language, data structure, analog circuit, digital circuit, etc., let me exposed to more direction, also to have the ability to participate in the ASC competition. I would be interested in deep learning of artificial intelligence, it's a cool thing, so I follow the teacher learning and participate in the relevant deep learning pedestrian recognition, face detection, etc. At the same time, With the development of artificial intelligence and computer graphics, super high performance computing will be an indispensable condition for research in the field of computer science. in the last year, I also have attended ASC17, although we did not make any achievement, but also made me growth experience, learned a lot, the third question of the ASC competition, this year is about deep learning, I will try mu best on this problem is to make their own achievements. ASC18, my friends and I challenge again!

Member:Wu Weizheng(Male) I am WadeWu, 19 years old,studying in NanChang University. I major in computer science. As a sophomore, I have learned a lot computer knowledge such as C++, DataStructure and so on. I love computer science very much, so I work hard on the classes. I think I am the top student in my class because I have got the Special Grade Scholarship. I am not satisfied with the class knowledge. I will buy some books to learn by myself. Last summer holiday, I attended a summer school to learn Android with the Sweden students who became my best friends. Apart from computer science, I will read some fictions and read some articles about the business. My favorite company is Tesla. I think it has the innovation and overturn. It is such great that can open the new future of the car. Whats more, I love doing exercise, such as running, basketball. Actually, I am good at running. I got 7th in 100m competition and got 2 in the 10*100 competition in the school sports meeting.

Taking part in ASC, I am eager to learn the knowledge of parallel algorithm and distributed system. I will seize the opportunity to learn supercomputing and learn more about computer architecture. Working with my team, we will overcome the difficulties and move forward! Hope my team can get good grade!

2.3 Team slogan

Calculation changes the world.

3 Technical proposal

3.1 HPC system design

3.1.1 Hardware

The HPC hardware environment is shown in Table 1.

Item	Name	Configuration	Energy
Server	Inspur NF5568M4	CPU: Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz 12 cores Memory: 16G × 18, DDR4/2133MHz Accelerator: NVIDIA Tesla Pascal P100 GPU	

Table 1: structure

3.1.2 Software

- Operating System

CentOS7 provides a solid computing platform and is prominently used in the clusters at the university, allowing the team to benefit from the experience and expertise gained by the administrators who manage universities clusters. Because of CentOS is rpm-based and has a big driver package from RedHat it will be the OS with the most functionality the team needs.

- Ansible
- SLURM
- Hardware configuration tools

3.2 HPL test

3.2.1 Introduction

HPL is a software package that solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers. It can thus be regarded as a portable as well as freely available implementation of the High Performance Computing Linpack Benchmark.

3.2.2 Software environment

The software used was provided by the Knights Landing (KNL) Cluster system and was based on a CentOS 7.2 which is very common on HPC Systems and used in the clusters from our local data center. Apart from that we used OpenBLAS since it is an optimized BLAS library based on GotoBLAS2 1.13 BSD version. It will be faster than GotoBLAS2. We use OpenMP for the Message Passing Interface (MPI). OpenMP is a tool designed for multi-core / multi CPU parallel computing on single host. In other words, OpenMP is more suitable for parallel computing on shared memory structure of single computer. Because of the way to coordinate parallel computation by sharing memory between threads, it is efficient in multi-core / multi CPU structure, small in memory cost and concise and intuitive in programming language, so programming is easy and compiler is easy to implement.

3.2.3 Algorithm analysis

- This software package solves a linear system of order n : $Ax = b$ by first computing the LU factorization with row partial pivoting of the $n - by - n + 1$ coefficient matrix $[Ab] = [[L, U]y]$. Since the lower triangular factor L is applied to b as the factorization progresses, the solution x is obtained by solving the upper triangular system $Ux = y$. The lower triangular matrix L is left unpivoted and the array of pivots is not returned.
- The data is distributed onto a two-dimensional P-by-Q grid of processes according to the block-cyclic scheme to ensure "good" load balance as well as the scalability of the algorithm. The $n - by - n + 1$ coefficient matrix is first logically partitioned into nb-by-nb blocks, that are cyclically "dealt" onto the P-by-Q process grid. This is done in both dimensions of the matrix.
- The right-looking variant has been chosen for the main loop of the LU factorization. This means that at each iteration of the loop a panel of nb columns is factorized, and the trailing submatrix is updated. Note that this computation is thus logically partitioned with the same block size nb that was used for the data distribution.

3.2.4 Performance Estimation

$12[\text{cores}]32[\text{tiles/core}]2.34\text{GHz} = 898\text{GFlops}$.

3.2.5 Performance optimization methods

- Adjustment parameter We adjust the parameter: $N = 84000, NB = 256, P * Q = 2 * 6$, and we can get $\text{GFlops} = 141.4$. That's a bad result. So we should do more about it.
- Algorithm optimization

3.3 HPCG test

3.3.1 Introduction

The High Performance Conjugate Gradients (HPCG) Benchmark project is an effort to create a new metric for ranking HPC systems. HPCG is intended as a complement to the High Performance LINPACK (HPL) benchmark, currently used to rank the TOP500 computing systems. The computational and data access patterns of HPL are still representative of some important scalable applications, but not all. HPCG is designed to exercise computational and data access patterns that more closely match a different and broad set of important applications, and to give incentive to computer system designers to invest in capabilities that will have impact on the collective performance of these applications.

3.3.2 Optimization process

3.3.3 Performance optimization methods

- Adjustment parameter

The problem domain is three dimensional. The global domain of Mx , My , Mz in the x , y , z dimensions is distributed into a local domain of Nx , Ny and Nz . Each local subgrid is assigned to a process. The total number of processes are provided as input, or detected at run-time. These processes are distributed into a three dimensional domain, where the total number of processes $P = Px * Py * Pz$. The global domain Mx , My , Mz can be calculated as $Px.Nx$, $Py.Ny$, $Pz.Nz$. A restriction is also imposed on the domain for accuracy and efficiency purposes. Minimum grid size in each dimension must be at least 16 and the grid should be a multiple of 8.

In HPCG we have to set the problem size to get the best results out of it. A valid run must also execute a problem size that is large enough so that data arrays accessed in the CG iteration loop do not fit in the cache of the device in a way that would be unrealistic in a real application setting. Presently this restriction means that the problem size should be large enough to occupy a significant fraction of "main memory", at least 1/4 of the total. Our HPC has 64GB memory, so we should set $Nx = Ny = Nz = 64$. For 12 processors, I should set the $P = 12$.

- Algorithm optimization

4 Relion Test

4.1 Application background

4.1.1 The principle of three-dimensional reconstruction of the electron microscope

Daniele DE rossi and his proposed theory of 3D reconstruction by means of a series of projection along different direction of electronic microscopy to reconstruct the three-dimensional configuration of object to be tested, and they put forward using digital image processing technology electronic microscopy three-dimensional reconstruction measurement concepts and methods of biological macromolecular structure. The mathematical basis of the three-dimensional reconstruction of electron mirror is the central section theorem and Fourier transform. The meaning of the central section theorem is that the Fourier transform of a function in a certain direction is equal to the Fourier transform of this function through the origin and perpendicular to the section function of the projection direction. Therefore, three dimensional reconstruction theory is based on the three-dimensional projection of an object as a Fourier transform is equal to the object of three-dimensional Fourier transform with the vertical projection direction, through the cross section of the origin (central section), as

shown in Fig.1. Each electron microscope image is a two-dimensional projection image of an object, and a series of electron micrographs are taken along different projection directions. Through the Fourier transform, a series of cross sections of different orientations are obtained. When the section is large enough, the three-dimensional information of the Fourier space will be obtained, and then the three-dimensional structure of the object can be obtained by the inverse Fourier transform. This method has been applied in a wide range of scope, from organelles without fixed structure characteristics and biological macromolecular compounds to macromolecular crystals, has developed into a practical method of protein structure analysis.

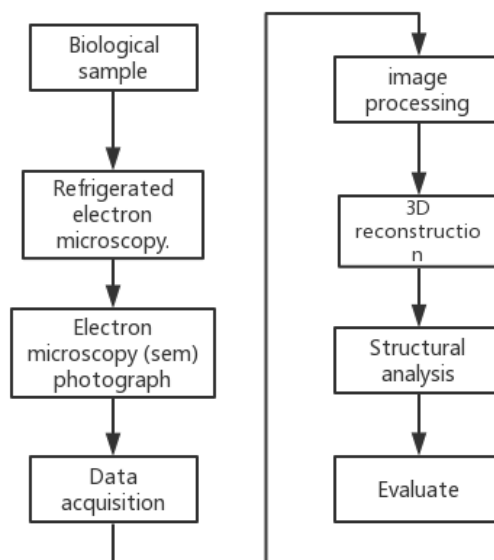


Figure 1: The operation steps of single particle cryoscope.

4.1.2 The principle of three - dimensional reconfiguration of single-grain frozen electron microscopy

Single particle cryoelectron microscopy is an important method to obtain 3d reconstruction images of bio-molecules. The so-called single particle method is the structural analysis of the granular molecules after separation and purification. Its basic principle is: through the same biological macro-molecular one direction projection microscopy in real space superposition average after adjustment, thus improve the signal-to-noise ratio, make the particles in the common parts of the structure information is strengthened, and finally to different projection di-

rection of single particle microscopic like refactoring in the three-dimensional space, so as to acquire the information of single particles macro-molecular three dimensional structure.

- A sample of frozen water containing biological macro-molecules that are homogeneous in chemistry and structure;
- Select the best particle density and glass ice thickness samples that are most likely to produce the best image;
- Set the best parameters (e.g., under-focus, magnification, and electron dose, etc.) to capture and record a large number of images in these sample areas;
- The projection of discrete molecules is selected by manual or semi-automatic procedure.
- Through various image processing methods, the relative orientation of different images is calculated, and the three-dimensional structure model of biological macro-molecule is reconstructed.
- Finally, the structural analysis and evaluation will be used to locate the atomic coordinates of protein structures obtained from crystallography or nuclear magnetic resonance to the three-dimensional structure density map.

The complex image processing process of single particle cryoscope 3d reconstruction involves the complex image processing from 2d projection image to 3d reconstruction model.

- Image classification: before 3D reconstruction processing, must classify particle projection image, to ensure that all the images in each category belong to the same direction projection drawing, otherwise it will be very bad impact on the result of the refactoring. Classic pattern recognition and clustering techniques, such as feature extraction, autocorrelation/correlation analysis, hard clustering and fuzzy clustering, are often used.
- Angle designation: after the particle image classification is completed, the projection direction of each kind of image needs to be calculated. In general, the projection direction of particle image is determined by comparing the particle image and the projection generated by computer simulation. In order to reduce the influence of noise, the average graph of each kind of particle image is usually represented.
- Three-dimensional reconstruction: according to the central section theorem, the Fourier transform of each particle image is equivalent to a central section of the original 3d Fourier space. So according to the projection direction of the each type of particle image can be reconstructed three-dimensional Fourier space, and then use direct inverse Fourier transform

or weighted back projection (weighted back - the projection) after several iterative reconstruction optimization, can eventually get molecular 3 d structure model.

4.1.3 RELION introduction

RELION is an image processing software designed specifically for cryo-electron microscopy (cryo-EM). Developed by the group of Sjors Scheres at the MRC Laboratory of Molecular Biology, the RELION framework is revolutionizing the cryo-EM field. It is a stand alone computer program built for the refinement of macromolecular structures by single-particle analysis of electron cryo-microscopy data. RELION employs an empirical Bayesian approach for refinement of multiple 3D reconstructions or 2D class averages. With alternative approaches often relying on user expertise for the tuning of parameters, RELION can find an optimal way of filtering the data automatically.

4.2 Algorithm analysis

As shown in Fig.2, RELION uses a Bayesian approach to infer parameters of a statistical model from the data.

Almost all existing implementations for cryo-EM structure determination employ the so-called weakphase object approximation, which leads to a linear image formation model in Fourier space:

$$X_{ij} = CTF_{ij} \sum_{l=1}^L P_{jl}^{\phi} V_{kl} + N_{ij}$$

- X_{ij} is the j th component, with $j = 1, \dots, J$, of the 2D Fourier transform X_i of the i th experimental image, with $i = 1, \dots, N$.
- CTF_{ij} is the j th component of the contrast transfer function for the i th image.
- V_{kl} is the l th component, with $l = 1, \dots, L$, of the 3D Fourier transform V_k of the k th of K underlying structures in the data set.
- P_{ϕ} is a $J * L$ matrix of elements P_{jl}^{ϕ} .
- N_{ij} is noise in the complex plane, which is assumed to be independent, zero-mean, and Gaussian distributed with variance σ_{ij}^2 .

The key of iterative algorithm is $V_{kl}^{(n+1)}$, we can adjust the algorithm by using different filters.

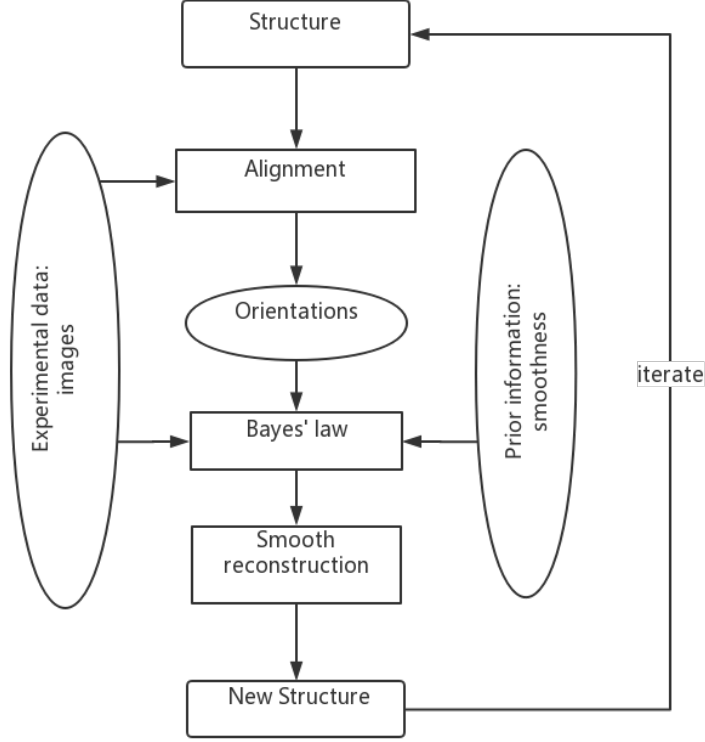


Fig. 2. A schematic interpretation of the approach.

4.3 Source code analysis

4.4 RELION test

4.4.1 Descriptions of the software environment

- Operating system: CentOS 7.2
- Compiler:
- Math library: OpenBLAS
- MPI software: OpenMPI

4.4.2 Testing method

First, we find the optimal number of iterations. Then, adjusting the algorithm to reduce the running time in the same number of iterations.

4.4.3 Problem and solution analysis

1. Could not find MPI_C (missing: MPI_C_LIBRARIES MPI_C_INCLUDE_PATH)

On some systems you appear to need to change MPI_DOUBLE_COMPLEX to MPI_C_DOUBLE_COMPLEX in src/macros.h for MPI to recognize the type of number that is being communicated. Note that you will need to recompile after having saved those changes.

2. Running out of disc space during the run.

As of version 1.2, RELION uses temporary files that are stored to disc to distribute all probability-weighted sums of the model. These files may be very large (depending on the size of our refinement, but possibly several Gb), and one will be written for each MPI process. If disc space becomes a problem, you may also distribute all these sums through the network using MPI. To do so, add the option ‘`-dont_combine_weights_via_disc`’ to the command line (you can ignore the warning that this option is not recognized). However, we suspect that the problems with TCP offloading described above are much worse when this option is used.

3. Although we ask for more threads, my MPI processes only take 100% or 200% CPU in top.

If using OpenMPI compiled with NUMA locking support, MPI processes get bound to assigned cores automatically. This is to prevent context switches and cache misses. You can use ‘`mpirun bind none`’ to have each MPI process use multiple cores.

4.5 Run the Program

4.5.1 Run 2D classification of the raw particle images

- Run command

```
which relion_refine_mpi
--o Class2D/job007/
--i particles.star --ctf
--iter 25 --tau2_fudge 2
--particle_diameter 150 --K 100
--flatten_solvent
--zero_mask --strict_highres_exp 8
--oversampling 1
--psi_step 12 --offset_range 5
--offset_step 2 --norm --scale
```

- Run results
- Results analysis

4.5.2 Run 3D classification of the raw particle images

- Run command

```
which relion_refine_mpi
--o Class3D/job007/
--i particles.star
--ref run_ct24_class001.mrc
--firstiter_cc --ini_high 40
--ctf --iter 40 --tau2_fudge 4
--particle_diameter 150 --K 4
--flatten_solvent --zero_mask
--strict_highres_exp 10 --oversampling 1
--healpix_order 1 --sigma_ang 0.3
--offset_range 5 --offset_step 2 --sym O --
--norm --scale
```

- Run results
- Results analysis

4.5.3 Perform 3D reconstruction and refinement

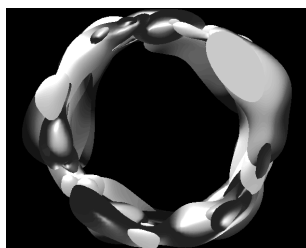
- Run command

```
which relion_refine_mpi
--o Refine3D/job007/
--auto_refine --split_random_halves
--i particles.star
--ref run_ct24_class001.mrc
--firstiter_cc --ini_high 40 --ctf
--particle_diameter 150
--flatten_solvent --zero_mask
--oversampling 1 --healpix_order 1
--auto_local_healpix_order 5
--offset_range 5 --offset_step 2 --sym O
--low_resol_join_halves 40 --norm --scale
```

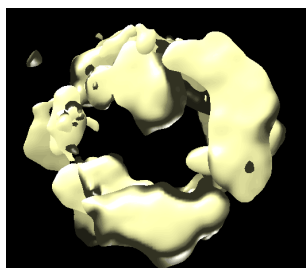
- Run results

As shown in Figure2, we run the program with the number of iterations 2, 8 and 10.

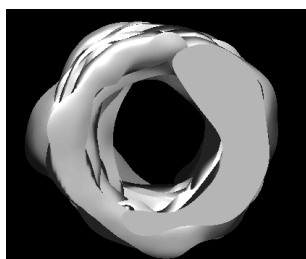
- Results analysis



(a) Number of iterations: 2



(b) Number of iterations: 8



(c) Number of iterations: 10

Figure 2: Number of iterations: 2, 8, 10

4.6 Optimization methods and results

5 Answer Prediction for Search Query

5.1 Background

5.1.1 Summary

After successfully installing CNTK on our system, we started to run the baseline given by official website. The model is aimed to build intelligent agents with the ability for reading comprehension over real word data. At first, we ran the model with only CPUs. Due to the large calculation of this model, it took long time to run this program. If we run this program with only two CPUs, 6 cores per CPU, we may find that it wont finish one iteration after 4 hours. But to keep accuracy, we should run it at least 1000 iterations. It means that more than 4000 hours are need to finish the model training, which we cannot afford. To shorten the running time, we decide to use GPUs.

5.1.2 Reading comprehension

Reading Comprehension (RC), or the ability to read text and then answer questions about it, is a complex task for machines, requiring both understanding of natural language and knowledge about the world. Machine reading comprehension of text is one of the ultimate of natural language precessing. While the ability of machine reading comprehension can be accessed in many different ways. Such as Attention over Attention, Memory Networks, Recurrent Neural Networks (RNNs) with Long-Short Term Memory (LSTMs). As for dataset, many data are used to do this this task. Such as SQuAD, MS MARCO, WikiQA, MCTest and Childrens Book. The first two datasets are common used specially. As ASC18 committee required, we take MS MARCO to do our task.

5.1.3 CNTK introduction

Computational Network Toolkit, short in CNTK, is a deep learning framework developed by Microsoft Research. As we know, computational network(CN) is a a unified framework for describing arbitrary learning machines, such as deep neural networks (DNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short term memory (LSTM), logistic regression, and maximum entropy model, that can be illustrated as a series of computational steps. A CN is a directed graph in which each leaf node represents an input value or a parameter and each non-leaf node represents a matrix operation upon its children. We describe algorithms to carry out forward computation and gradient calculation in CN and introduce most popular computation node types used in a typical CN. And CNTK is an toolkit of CN that supports both GPU and CPU. In this task, our model should be build base on CNTK. So a good time to learn CNTK basic command line begins.

5.2 Model introduction

The model we used is called R-NET. It is an end-to-end neural network model for reading comprehension and question answering. It consists of four part:

- The recurrent network encoder is used to build representation for question and passages separately.
- The gated matching layer to match the question and passage.
- The self-matching layer to aggregate information from the whole passage.
- The point-network based answer boundary prediction layer.

The model given is written in Python and contains several scripts. Three main scripts of them are `convert_mamarco.py`, `tsv2ctf.py` and `train_pm.py`. The first one is used to convert our dataset in format of json to file in format of tsv. We can know the second one its use from its name. It helps us to convert the precessed tsv file to ctf file. In fact, its a precess of embedding of words and characters. As for `train_pm.py`, it's the core of the training model. Detail information will be introduced in following passages.

5.3 Detailed Processe

- Set up: CNTK of different virsions are available on github and official website of Microsoft cooperation. Our system is a CentOS system. We first installed Anaconda with yum tools, and then installed a version of CNTK which support CPUs. To get a good performance, we also installed one version of CNTK that supports GPUs.
- Data downloading: As the guide recommended, we downloaded the MS MARCO dataset. It is large dataset, which is aimed to the task of reading comprehension and question answering. It contains three packages. They are `train_v1.1.json`, `test_public_v1.1.json`, `dev_v1.1.json`. Obviously, they are used for model training, testing and developing. In MS MARCO, all questions are sampled from real anonymized user queries. The context passages, from which answers in the dataset are derived, are extracted from real web documents using the most advanced version of the Bing search engine. The answers to the queries are human generated. The MS MARCO dataset consists of four major components Queries, Passages, Answers and Query type.
- The task: Our task was to use the MS MARCO and CNTK deep learning frame to build a intelligent agents with the ability for reading comprehension or question answering over real word data. Several scripts mentioned above we get form recommended website make up the baseline.
 - Generating csv file: Before we run `convert_msmarco.py`, we should install a library named nltk. Natural Language Toolkit is a powerful

tool for natural language processing. Many functions are used in the script `convert_mamarco.py`. We type `python convert_msmarco.py`. And the MS MARCO dataset will be converted to file in format of csv.

- Word and character embedding: To meet the standard of CNTK reader for calculation, we need to do an embedding work. We used a model named Glove Vector. The input are words and characters, and the output is vectors. So it means that words and characters can be calculated. We run the script `csv2tvf.py` to create three tvf files related to the csv file we created before.
- Model training: Adjust some paraments in `config.py`, then run `train_pm.py` to train the model. It will take a long time to finish training the model.

5.4 Result analyse

We have a try to train this model on CPUs. But training this model on CPUs costs too much time which we cannot afford. And it will also influence plan of other teams. By now, we havent gotten a result yet. When GPU source is available, we will do it soon.

5.5 Optimization

- On hardwares, we use GPU instead of CPU. With the powerful calculation ability of GPU, we can save lot of time to train a model.
- On softwares, we use multi process to train a model with MPI. Besides, we try to read source code and understand it. Experience suggests us that even a small change on source code may lead to a big improvement.