Thank you for downing Dragon-Alpha&cu32.

I usually abbreviate Dragon-Alpha as Alpha.

I. Pay-Attention

1. cu32 now is only complied for 64-bit Windows, so kindly run Alpha’s apps on such OS.

2. Please make sure: the JDK version is greater than 8.0

3. Please make sure: compute >= 61, sm >= 61

4. Kindly read “**Arxiv.pdf**” first, to briefly understand Alpha.

5. Alpha has only been executed on GTX 1050, RTX 3060ti GPU, and presently its applications can only be executed on CUDA GPU.

6. Since I am the only-one programmer to build Alpha, I must pay my main attention on the code instead of the document, to complete Alpha’s prototype in time. If you have some question, just see the source-code. Sorry, my personal abilities are really limited.

II. Files

--Arxiv.pdf an article talking about the background, characteristics, architecture and experiments of Alpha, preprinted on arxiv.org, at: https://arxiv.org/abs/2305.08819.

--exec the executable files of Alpha.

--lib Java libraries of Alpha, which are jar-files complied by Java-code.

--native-lib native libraries of Dragon-Alpha. They are dynamic-linked-libraries, and integrated to Alpha at the bottom through JNI.

--cuda_float32 contains the executable files of cu32. Presently, cu32 has only been complied for 64-bit Windows, and will be compiled for Linux in the near future.

--icon Alpha’s logo. If you like it, set it for Alpha’s home directory.

--data Alphas’ built-in data-sets, including MINIST, cifar-10, cifar-100 and Soccer. Please decompress before use them.

--src the source-code of Dragon-Alpha

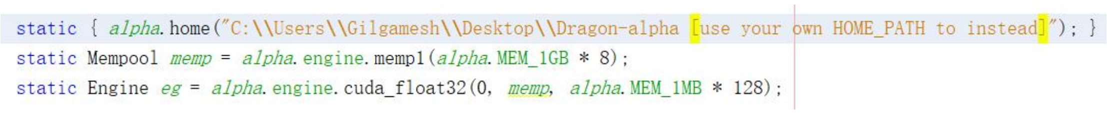
--alpha_src the Java source-code of Alpha. You rename this directory to ‘src’ and integrate it to your own Java-project. I suggest using NetBeans to read such source-code, since I use NetBeans to build Alpha.

--zutil_src ZUtil is an auxiliary library for Alpha. Since I wrote it in my sophomore year, it may have some unreasonable aspects in programming-style and architecture. I only uploaded a part of it, so kindly use ZUTIL-STD-1.1.jar instead of the source-code.

--cu32_src the C++ source-code of cu32, consists of 13 Visual Studio (VS2017) projects. To open such projects on your PC, please make sure your VS can build CUDA projects. You also need to add jdk.include&jdk.lib to such projects (such as jni.h, jvm.lib). Since I use CUDA v11.3, I suggest you to use such version too.

--experiments the related experimental code&data related to **Arxiv.pdf**.

--alpha-code the experimental code of Alpha, you can take it as examples, to create your own Alpha-app. Before using Alpha's API, you must specify the home-path of Alpha, in order to load the relative native libraries.



```
static { alpha.home("C:\\Users\\Gilgamesh\\Desktop\\Dragon-alpha [use your own HOME_PATH to instead]"); }
static Mempool memp = alpha.engine.memp1(alpha.MEM_1GB * 8);
static Engine eg = alpha.engine.cuda_float32(0, memp, alpha.MEM_1MB * 128);
```

--pytorch-code the experimental code of PyTorch.

--experiment-data console output to track some metrics for both Alpha and PyTorch, in order to make a comparison.

--test_cuda some related code to test Alpha&cu32. You can take it as examples of using Alpha's operators.

Finally,

My name Zhang Zhiyi, and I often use Gilgamesh as my internet name.

I was born in April, 2000, majored Computer-Science in my college, and now study Pattern-Recognition after graduate.

First, let' s talk about the reason why I create Alpha instead of using PyTorch. I prefer Java to Python, but failed to find a Java-based DL framework as excellent as PyTorch, in the past few years. Also, I want to learn more about the principles and details of DL, and like implementing them to improve my abilities. So, I started to build my own Java DL framework. Dragon-Alpha can be regraded as a continuation of Dragon, which is my graduation project.

It cost me about 190 days and 200,000 lines of code to build Alpha' s prototype. In such progress, I have been suffering while enjoying, and finally benefited a lot. PyTorch&cuDNN is my opponent but also my mentor. I tried to learn its advantage, and pondered how to make some breakthrough while keeping Alpha' s own characteristics. Now, the Alpha' s prototype has been completed, and the relative paper has been written.

At present, Alpha is not as polished as PyTorch, but it could be good start and have a long way to reach perfection.

It' s my honour to share the source code Alpha&cu32. Sincerely, I request and need all of you to use and improve it. If you have some related good advice and achievement, please contact me at gilgamesh@mail.ustc.edu.cn.