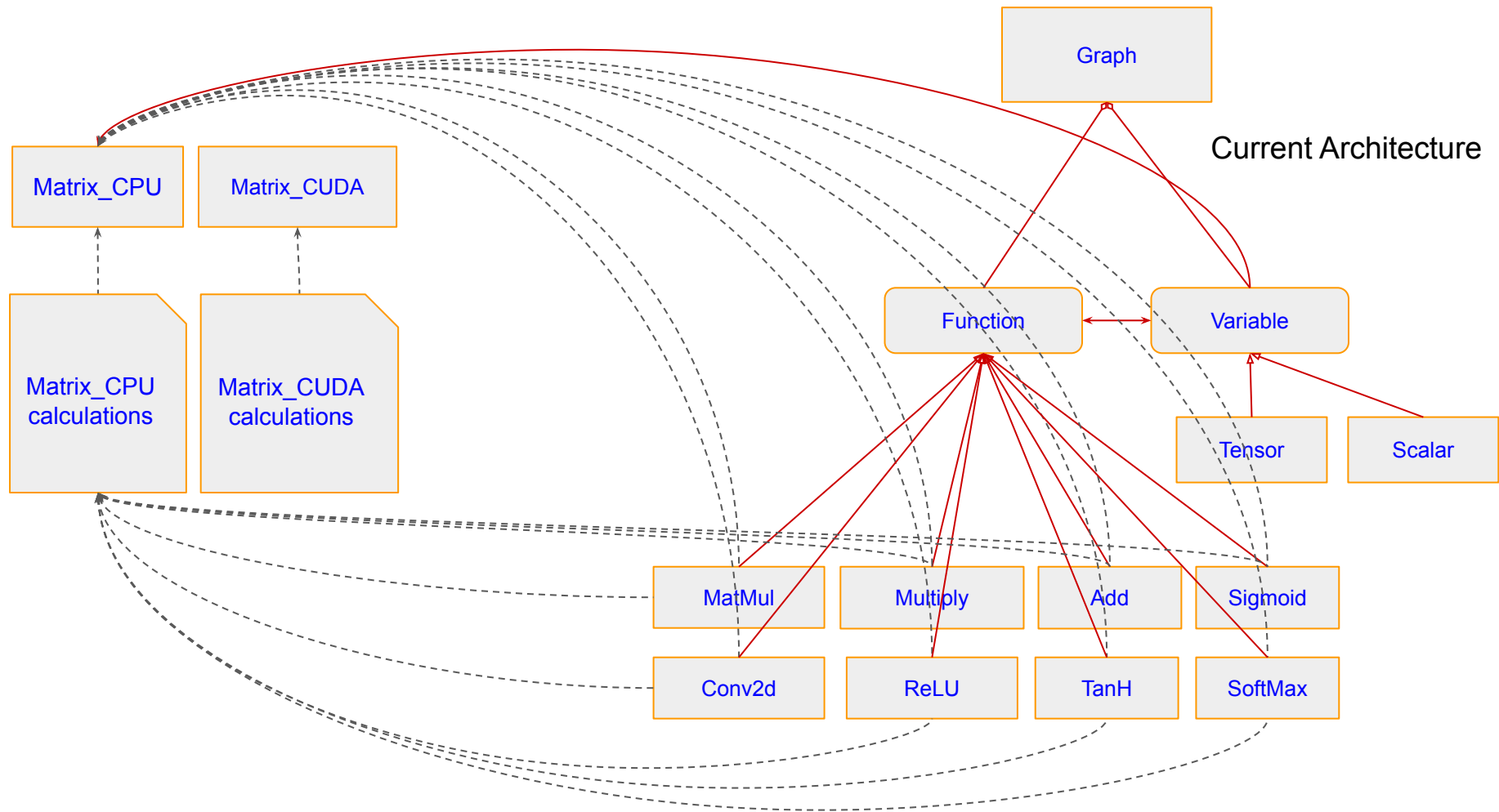


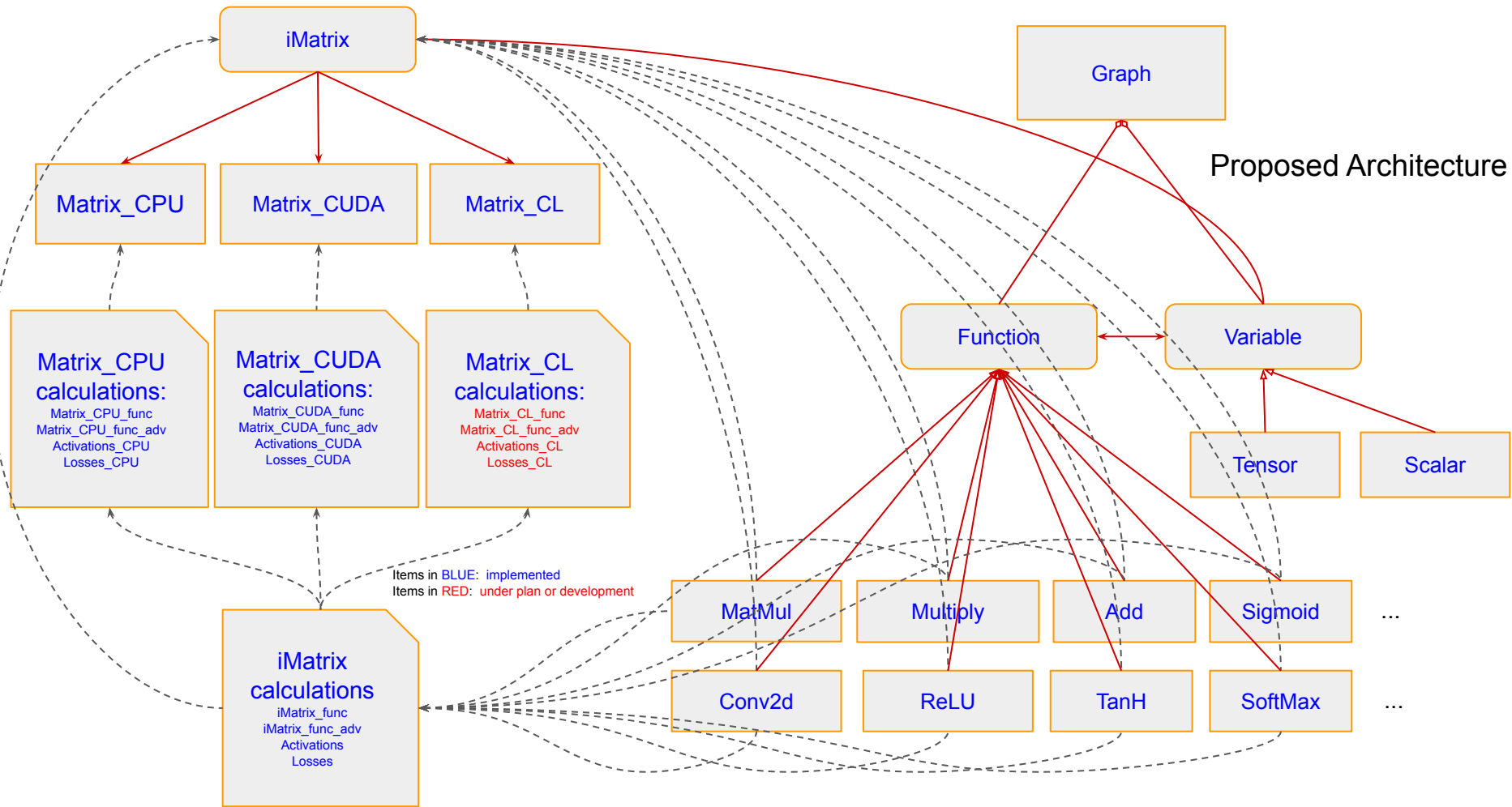
Julie Neurons

Architectural Enhancement

Chunнан Sheng

Sep 12, 2020





Code refactoring: new instance returning replaced by in-place data update

Deprecated function call in matrix calculations

```
return_type operation(const type1 & var1, const type2 & var2, ...)
{
    return_type output;
    // Deal with output, var1, var2, etc
    ...
    return output;
}
```

Recommended function call in matrix calculations

```
void operation(return_type & output, const type1 & var1, const type2 & var2, ...)
{
    // Deal with output, var1, var2, etc
    ...
}
```

Code refactoring: new macros for multi-platform

CMake examples:

```
if (ARCH_ARM)
    set(CMAKE_SYSTEM_PROCESSOR arm)
    add_definitions(-DARCH_ARM=1)
endif()

if (WITH_CUDA)
    set(
        CUDA_INCLUDE_DIRS
        /usr/local/cuda/include
        ../julie_neurons/julie/la/cuda
    )
    add_definitions(-DWITH_CUDA=1)
endif()

if (WITH_CL)
    ...
    add_definitions(-DWITH_CL=1)
endif()
```

Code refactoring: new macros for multi-platform

C++ source code examples:

```
#ifdef WITH_CUDA
#include "Matrix_CUDA.hpp"
#endif

#ifdef WITH_CL
#include "Matrix_CL.hpp"
#endif

#ifdef WITH_CUDA
    if (this->m_mat_cuda)
    {
        *(this->m_mat_cuda) = *(other.m_mat_cuda);
    }
    else
    {
        this->m_mat_cuda = std::make_shared<cuda::Matrix_CUDA<DT>>(*(other.m_mat_cuda));
    }
#else
    throw std::invalid_argument { std::string{"CUDA Matrix type not supported in "} + std::string{__FUNCTION__} };
#endif

#ifdef WITH_CL
    ...
#else
    throw std::invalid_argument { std::string{"CL Matrix type not supported in "} + std::string{__FUNCTION__} };
#endif
```