



Denglin Hamming™ V2

dINNE Build Modulator API

DL-DG/SW-031D-02

2024-5-27

Copyright©苏州登临科技有限公司，2019 - 2025，版权所有。

未经苏州登临科技有限公司事先书面同意，不得以任何形式或方式复制或传播本文件的任何部分。

商标和许可



和其它苏州登临科技有限公司的其它登临科技的图标为苏州登临科技有限公司的商标。本手册中提及的所有其他商标均为其各自所有者的财产。

通知

所购买的产品、服务和特性由是苏州登临科技有限公司与客户签订的合同规定。本文件中描述的所有或部分产品、服务和特性可能不在采购范围或使用范围内。除非合同中另有规定，本文件中的所有声明、信息和建议均按“原样”提供，无任何明示或暗示的保证或陈述。

本手册中的信息如有更改，恕不另行通知。本文件在编制过程中已尽一切努力确保内容的准确性，本文件中的所有声明、信息和建议不构成任何明示或暗示的保证。

苏州登临科技有限公司

苏州工业园区扬富路11号南岸新地一期商务楼栋5号1101室，江苏，中国

<http://www.denglin.ai>

Email : support@denglin.ai

Change History

Version	Change description
02	Upgrade.
01	Initial version.

CONTENTS

1	Library API	3
1.1	Class Hierarchy	3
1.2	File Hierarchy	3
1.3	Full API	3

CHAPTER

ONE

LIBRARY API

1.1 Class Hierarchy

- *Namespace dl*
 - *Namespace dl::nne*
 - * *Class IBuildModulator*
 - * *Class IDAG*
 - * *Class IDAGNode*
 - * *Class ISubgraphContainer*
 - * *Enum DAGNodeType*

1.2 File Hierarchy

- file_dlnne_build_modulator.h

1.3 Full API

1.3.1 Namespaces

Namespace dl

Contents

- *Namespaces*

Namespaces

- *Namespace dl::nne*

Namespace dl::nne

Contents

- *Classes*
- *Enums*
- *Typedefs*

Classes

- *Class IBuildModulator*
- *Class IDAG*
- *Class IDAGNode*
- *Class ISubgraphContainer*

Enums

- *Enum DAGNodeType*

Typedefs

- *Typedef dl::nne::IDAGNodeList*

1.3.2 Classes and Structs

Class IBuildModulator

- Defined in file_dlnne_build_modulator.h

Class Documentation

class dl::nne::IBuildModulator
The core callback interface.

Public Functions

bool **ScheduleDAG** (*IDAG* *graph) = 0
Schedule the DAG node list.

Return If there is a valid schedule.

Parameters

- graph: The DAG of the network

bool **GenerateSubgraph** (const *IDAG* *graph, *ISubgraphContainer* *container) = 0
Choose the subgraph node list.

Return If there is a valid subgraph generated.

Parameters

- graph: The DAG of the network
- container: The container of subgraphs

Class IDAG

- Defined in file_dlnne_build_modulator.h

Class Documentation

class dl::nne::IDAG
A DAG definition for the network.

Public Functions

IDAGNodeList **GetNodes** () const = 0
Get the nodes of the DAG.

Return The nodes of the DAG

bool **SetNodes** (IDAGNodeList &nodes) = 0
Set the nodes of the DAG.

Return If nodes are legal and in topological order

Parameters

- nodes: The nodes to be set to the DAG

Class IDAGNode

- Defined in file_dlnne_build_modulator.h

Class Documentation

class dl::nne::IDAGNode
Represent a node in the DAG.

Public Functions

DAGNodeType **GetType** () = 0
Get the type of the *IDAGNode*.

Return The type of the *IDAGNode*

std::string **GetName** () **const** = 0
Get the name of the *IDAGNode*.

Return The name of the *IDAGNode*

std::vector<*IDAGNode**> **GetSuccessors** () = 0
Get the successors of the *IDAGNode*.

Return The successors of the *IDAGNode*

~IDAGNode ()

Class ISubgraphContainer

- Defined in file_dlnne_build_modulator.h

Class Documentation

class dl::nne::ISubgraphContainer
A container of the subgraphs.

Public Functions

bool **AddSubgraph** (IDAGNodeList *list) = 0
Add a subgraph.

Return If the subgraph is added successfully.

Parameters

- list: The ordered nodes of the new subgraph

1.3.3 Enums

Enum DAGNodeType

- Defined in file_dlnne_build_modulator.h

Enum Documentation

```
enum nne::dl::DAGNodeType
```

Values:

```
enumerator Var = 1
```

```
enumerator OP = 2
```

```
enumerator Unknown = -1
```

1.3.4 Typedefs

Typedef dl::nne::IDAGNodeList

- Defined in file_dlnne_build_modulator.h

Typedef Documentation

```
using nne::dl::IDAGNodeList = std::list<IDAGNode*>
```