



Denglin HammingTM V2 dINNE

Quantization Tutorial with ResNet-V1-50

DL-DG/SW-035B-01

2023-4-30

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5号1101室，江苏，中国

<http://www.denglin.ai>

Email : support@denglin.ai

更新历史

版本	更新描述
01	第一次发布

章节目录

[ResNet-V1-50 模型量化教程](#)
[相关文档](#)

ResNet-V1-50 模型量化教程

该教程以图像分类模型 ResNet-V1-50 为例，说明如何使用 TensorFlow 深度学习框架和量化工具，快速量化浮点网络模型。该示例包括以下步骤：

1. 导入依赖
2. 加载/处理数据样本
3. 加载模型
4. 插入模拟量化节点
5. 量化值域范围统计
6. 模型转换
7. 保存模型

每个步骤的详细内容如下：

1. 导入依赖

请确认已正确安装依赖包，导入需要使用的模块：

```
from graph_transformer.quantize import create_eval_graph
from graph_transformer.quantize import _InsertQuantOpForAllConsumerOfProducer as
insert_quant_op
from graph_transformer import transform_graph
```

2. 加载/处理数据样本

按照实际应用场景中推理的过程，加载数据样本，并对数据样本做预处理操作：

```
provider = slim.dataset_data_provider.DatasetDataProvider(dataset)
[image] = provider.get(['image'])
image_preprocessing_fn = preprocessing_factory.get_preprocessing(name)
image = image_preprocessing_fn(image, image_size)
```

3. 加载模型

加载保存的模型（通常后缀名为 .pb）文件到 default graph:

```
graph = tf.get_default_graph()
graph_def = graph.as_graph_def()
graph_def.ParseFromString(tf.gfile.GFile(pb_file, 'rb').read())
tf.import_graph_def(graph_def, name='')
```

4. 插入模拟量化节点

有自动和手动指定两种方式插入模拟量化节点。自动方式将在当前给定的计算图上，匹配预先设定的节点模型，在匹配上的节点后插入模拟量化节点。手动方式通过手动指定节点，在指定的节点后插入模拟量化节点。两种方式均在原图上做修改，并不会创建一张新的计算图。

1. 自动方式

```
create_eval_graph(graph, weight_bits=8, activation_bits=8)
```

2. 手动方式

```
insert_quant_op(graph, op_name="resnet_v1_50/conv1/Relu")
```

5. 量化值域范围统计

在数据样本上执行推理过程，进行量化值域范围的统计：

```
for i in range(num_batches):
    predict = sess.run(output)
```

6. 模型转换

将统计后的模型转换成推理阶段可加载的模型：

```
graph_def = graph.as_graph_def()
input_names = ["input"]
output_names = ["resnet_v1_50/predictions/Reshape_1"]
transformed_graph = transform_graph(graph_def, output_names, input_names,
node_excludes=[])
```

7. 保存模型

将转换后的模型进行保存，以便后续推理引擎使用：

```
with tf.gfile.GFile(out_file, 'wb') as f:
    f.write(transformed_graph.SerializeToString())
```

相关文档

- dINNE-Quant-Introduction-To-Quantization.pdf 中的 **网络模型量化方法介绍**
- dINNE-Quant-TU-Operator.pdf 中的 **TU (Tensor Unit) 算子介绍**