

# Zhe Work Report

2018.02.28

The report includes :

1. model comparison tool.
2. Non-blocking multi node communication in chainer master v3.
3. Source code in *developer01*.
4. Label Device

## Model Comparison Tool

Model comparisontool is a tool to help developer to compare difference between two model from different framework. The tool support **Caffe, TF, MXNet, Chainer,pytorch**. We mainly compare the variable node shape and functionnode type. We also compare the computation the graph and padding strategy.

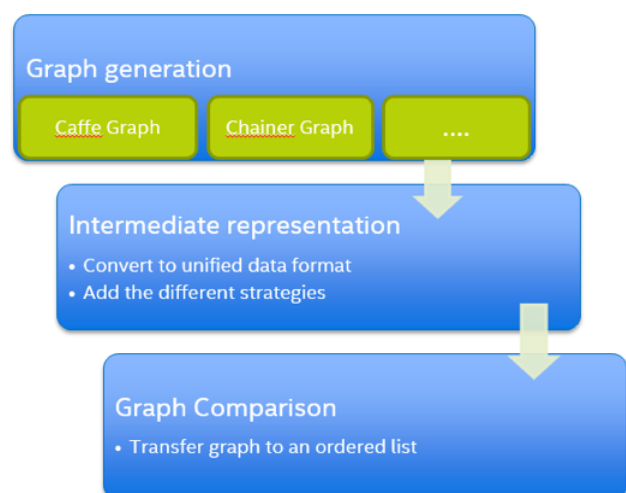
We also develop a web model comparison tool for user. User just need upload their models file and see the visually graph comparison result. Below Section 1 is the design detail of the tool and Section 2 is the Web version Detail, final is the Deploy part.

### 1. Design Detail

- The whole tool based on the architecture as below.

### Design Architecture

- Graph generation
  - Static framework
    - ✓ Caffe: python API (`caffe.proto.caffe_pb2`)
  - Dynamic framework
    - ✓ Chainer: run once and get it through graph API (`build_computational_graph`)
- Intermediate representation
  - Unified node format
- Graph comparison
  - Transferred to an ordered list

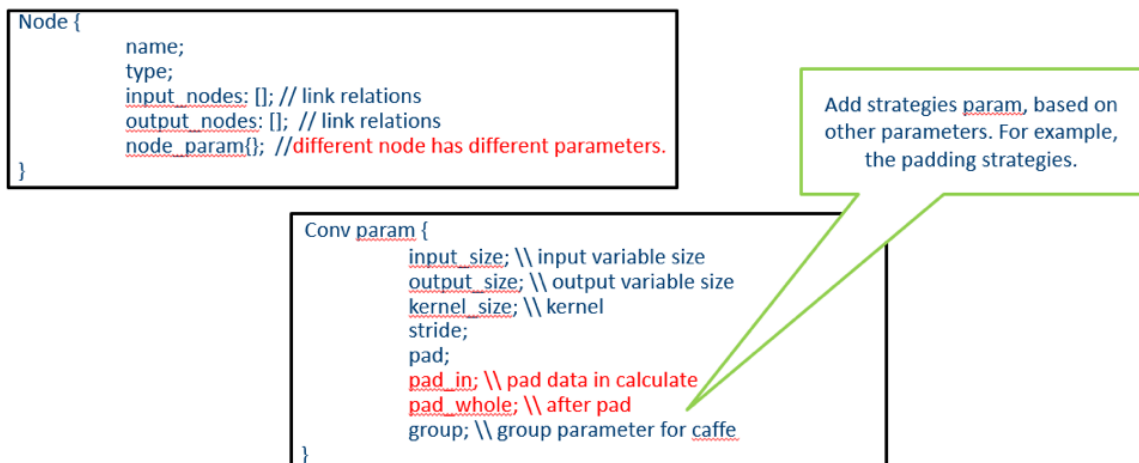


- In our tool, firstly loading the graph from different framework. You can find the method to load function responding to different framework as the Table below.

Framework	Input Files	Function file	Detail
Caffe	train_val.prototxt	load_caffe_model.py	Based on the caffe.proto.caffe_pb2 to read model parameters.
Tensorflow	model save files	load_tf_model.py	Read computation graph from stored model file. So User should save the graph before.
Chainer	model class .py files	load_chainer_model.py	Get all functionNode from output.
MXNet	model json file	load_mxnet_model.py	Read json file and parameters.
SSD	SSD model	load_ssd_*.py	SSD have special layer.

- Convert to unified data format and add different padding strategies, as the Figure below. Later version, we add pw, ph, sh, sw, kh, kw params. source code in rank\_multi\_port.py.

## Intermediate Representation



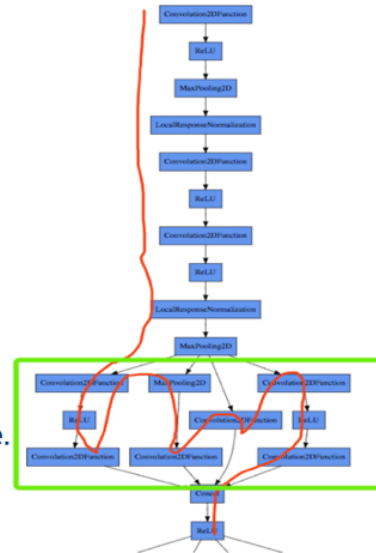
- Graph comparison is a difficult problem. So we transfer graph to an ordered list and find all difference. Source code in rank\_multi\_port.py.

## 2. Web Model Comparison Tool

For a better User Experience, we develop a website. User just upload the input files and get the visual comparison result and detail parameters. The system combine **Django** backend and **Bootstrap** front end. Using **D3.js** to visualize the model graph. All source code in the *web-model-comparison-tool* folder.

# Graph Comparison

- Transfer graph to an ordered list
- Encode for every branch
  - Encode format: (type\_code, size\_code)
    - ✓ (AveragePooling -> 1, kernel\_size)
    - ✓ (Convolution -> 2, kernel\_size)
    - ✓ (MaxPooling -> 3, kernel\_size)
    - ✓ (InnerProduct -> 4, num\_of\_output)
- ✓ Rank every branch with the encode and get an unique ordered list for different model file.



The website have two pages. In page 1, user upload their needing. Page 2 show the Graph result.

Model Comparison Tool

### Model Comparison Tool Start:

Framework A  
IntelCaffe 选框架

File A (TensorFlow upload all files)  
Choose file... 上传文件: caffe 上传 prototxt文件 Browse

Input Data Shape:(Format: (C, H, W))  
(3, 227, 227) 修改输入shape

Class Name(only for chainer)  
defaultNet 输入model类名, 只有chainer需输入, 其他不用修改

Submit

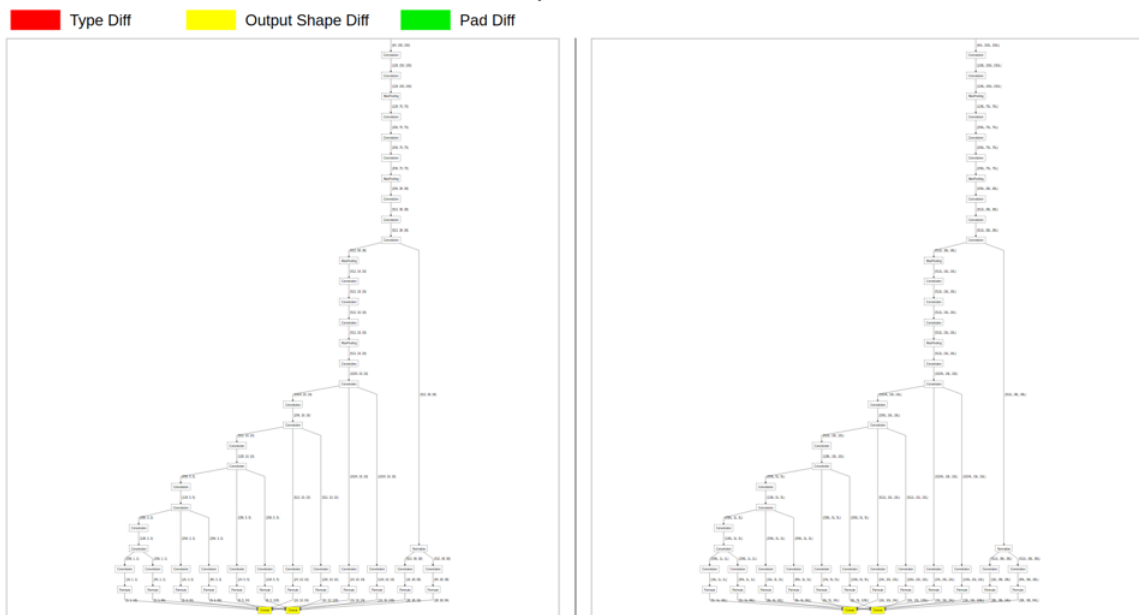
Framework B  
Tensorflow

File B (TensorFlow upload all files)  
Choose file... Browse

Input Data Shape:(Format: (C, H, W))  
(3, 299, 299)

Class Name(only for chainer)  
defaultNet

## Graph Result:



Our website version deploy in developer01 server. The root username modeltools, password: abc110.

1. ssh to the web server.

```
ssh modeltools@developer01
```

2. cd to work folder.

```
cd ~/web-model-comparison-tool/model_comparison_tool/
```

3. Start the uwsgi server.

```
uwsgi -socket /tmp2/model_comparison_tool.sock -module  
model_comparison_tool.wsgi -chmod-socket=777 -uid=www-data -  
gid=www-data
```

4. Restart the nginx service.

```
sudo /etc/init.d/nginx restart
```

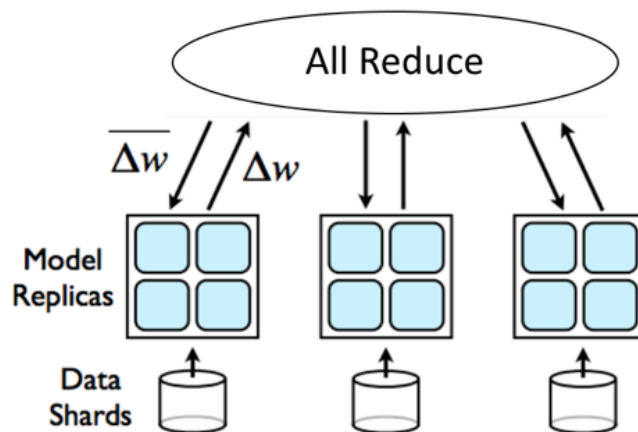
5. Now you can browser the site: server local IP:8000 to use our tool.

If wanna know more, please click on [HELP](#).

## Non-blocking Multi Node In Chainer Master V3

In Chainermn master, multi use the blocking to implement multi node training. In Chainer, we use data parallelism as multi node practices. The weight update like the figure below.

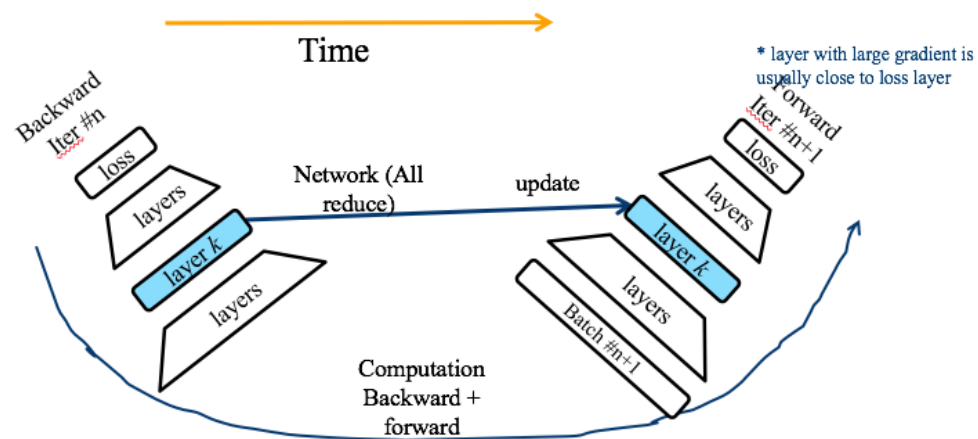
## Data Parallelism -- Synchronous SGD



Blocking communication, every iteration do allreduce together. But Non-blocking do weight all reduce every layer. So we can hide communication with computation. Just like below.

## Non-blocking communication

- For data parallelism, we can hide communication with computation



Our code based on the intel chainer master\_v3 branch. All the diff in the patch files.

## Source code in developer01

I stored backup files in developer01 server. Model comparison tool is in ~/MCT folder. Non-blocking-chainer is in ./NBC folder. Our tool git also in **dl\_framework-dl\_tools** repo.

## Device

A Desktop in mine. NUM is: **BB26468.**