Hybridcnn 打包工具说明

1、总体运行脚本说明

总体运行脚本的一般取名为 hyper_cnn.bat 脚本，里面主要需要修改的包括以下内容：

```
set feanorm=.\res\hybridCNN_20200316_huawei_mix_tongyong_all_1206_init\fea.norm
set stateinput=.\res\2wh.states.count.txt
set mlpweight=.\res\hybridCNN_20200316_huawei_mix_tongyong_all_1206_init\dcnn-0-0039.params

set lite_json=..\json\dcnn-0-symbol_fix.json
set dfmlp_json=..\json\hybridcnn_gpu.json

set outdir=.\hybridCNN_20200316_huawei_mix_tongyong_all_1206_init
```

1）feanorm 为输入特征的统计量，需要根据自己的训练数据进行统计适配；

2）stateinput 为状态文件，第一行状态的个数和状态个数总和(用于计算先验);比如英语的文件形式如下：

```
9004    12374112971
En_aa_s21   0   1049527
En_aa_s210  1   319092
En_aa_s211  2   387157
En_aa_s212  3   845669
En_aa_s213  4   216816
```

3）mlpweight 为 mlp 的参数，即模型最终保存前向计算的权重，与 lite_json 共同完成 hybridcnn 的前向计算。

4）lite_json 为 hybridcnn 网络结构图的 json 文件，用于 Maxengine 的前向计算，如果网络中的卷积等 OP 的参数调整需要修改该文件进行适配；

5）dfmlp_json 为 dfmlp 的 json 文件，主要包括控制整个计算流的相关参数，包括输入帧数、前后视野等参数；

6）outdir 为最终打包资源的输出文件。

2、dfmlp json 文件说明

由于 hybrid cnn 的引擎的由 dfmlp+maxengine 实现，其中 dfmlp 完成数据流模块、maxengine 完成前向计算模块，因此存在 dfmlp json 和 maxengine json 两个 json 文件。

Dfmlp 的 json 主要用于数据流的控制，主要需要注意的配置如下：

```
    "bb" : "SingleBatchBuilder",
    "param" : {"nDim" :"40",
               "nInputFrame" :"4",
               "nBufferLength" :"30000",
               "nHistroyPad" :"96",
               "nDataFlushWindowLength" :"400",
               "nFuturePad" :"96",
               "nValidDataAlign" :"8",
               "nDataFlushAlignment" :"32",
               "nZeroPad" :"8",
               "nPadSilenceSize" :"16",
               "nMaxSingleBatchLength" :"4096",
               "nMaxInLimit" :"30000",
               "nPadInvalidValue" :"0",
               "nMinFlushWindowsLength" :"256",
               "bLockMaxSingleBatchLength" :"false",
               "nOutputDimsNum" : "3"
    },
```

1) nDim:为输入特征维度，此处为 40 维特征；
2) nInputFrame：为输入的帧数，此处为 4 帧进模型；
3) nBufferLength:为 dfmlp 最多缓存的输入数据的长度；
4) nHistoryPad：为历史感受野的长度，即向前看的帧数,该值根据 hybridcnn 的模型具体计算得出，现有模型都为 96；
5) nDataFlushWindowLength：为一次 flush 下发的长度，即数据满足 400 帧则进行一次前向计算；
6) nFuturePad：为未来感受野的长度，即向后看的帧数,该值根据 hybridcnn 的模型具体计算得出，现有模型都为 96；
7) nMaxSingleBatchLength:因为该数据流采用单 batch 的方式，因此需要进行拼成单 batch 计算，该值设置单 batch 最大的数据长度；
8) nOutputDimNum:为输出数据的维度；
9) 其他配置一般采用默认值，不需要修改。

因为 Dfmlp 中，调用 Maxengine 进行前向计算，下图为 Dfmlp 调用 Maxengine 的配置：

```
{
  "op": "MaxEngineLite",
  "name": "MaxEngineLite",
  "param": {
    "data_format": "NHW",
    "device_type": "2",
    "dev_id": "0",
    "input_keys": "data, label",
    "input_shape_indptr": "0, 4, 6",
    "input_shape_data": "1, 1, 40, 4096, 1, 4096",
    "max_len": "0",
    "model_type": "0",
    "num_output_nodes": "1",
    "mask_name": "label"
  },
```

1） data_format:为输入数据的格式，此处为 NHW(batch*H*W)；
2） device_type:用于设置计算平台，2 为 GPU 平台、0 为 CPU 平台；
3） dev_id:为计算平台的设备卡号，此处 0 表示 0 号 GPU；
4） input_keys:为 hybridcnn 的输入数据的名字，与 hybridcnn 网络结构的 json 的输入相对应；
5） input_shape_indptr:为输入数据对应的维度信息，比如此处 data 为 4 位，label 为 2 维(6 - 4)；
6） input_shape_data:为输入数据的具体维度信息，比如此处 data 的 4 维信息为（1*1*40*4096），label 为（1*4096）；
7） model_type:为输入数据的数据类型，一般取 0 为 float32 类型；
8） num_output_nodes:为输出数据的个数，即网络 json 最终的输出；
9） mask_name:mask 信息对应的参数名字，此处为 label；

同时 Dfmlp 通过 PosteriorHC、PostCopyHC 两个 OP 用于控制最终的输出的维度，也是控制模型高低帧率的配置：

```
  {
     "op" : "PosteriorHC",
     "param" : {"nDim":"9004",
         "LowFrame":"1",
         "Priscale":"0.6"},
     "name" : "PosteriorHC",
     "inputs" : [
             [6,0],
             [7,0]
             ]
  },
  {
     "op" : "PostCopyHC",
     "param" : {"nDim":"36016",
         "LowFrame":"4"},
     "name" : "PostCopyHC",
     "inputs" : [
             [8,0]
             ]
  }
],
```

上图配置为高帧率配置，4 帧进 4 帧出，输出 36016（4*9004），即输出 4 帧，每帧输出 9004 个状态。

```
      "op" : "PosteriorHC",
      "param" : {"nDim":"9001",
          "LowFrame":"4",
          "Priscale":"0.6"},
      "name" : "PosteriorHC",
      "inputs" : [
              [6,0],
              [7,0]
              ]
   },
   {
      "op" : "PostCopyHC",
      "param" : {"nDim":"9001",
          "LowFrame":"4"},
      "name" : "PostCopyHC",
      "inputs" : [
              [8,0]
              ]
   }
],
"arg_nodes":[
```

上图配置为低帧率配置，4 帧进 1 帧出，输出 9001（1*9001），即输出 1 帧，每帧输出 9001 个状态。

3、hybridcnn 网络结构 json

该 json 为 hybridcnn 的网络结构 json，与模型的卷积个数、核大小、attention

等相关，因此模型具体结构变化时需要进行适配。

目前，主要小语种中修改的 json 文件位置如下表，用于适配高低帧率模型：

```
    },
    {
      "op": "null",
      "name": "deconvolutionl_weight",
      "attrs": {
        "kernel": "(1, 4)",
        "num_filter": "512",
        "pad": "(0, 0)",
        "stride": "(1, 4)"
      },
      "inputs": []
    },
    {
      "op": "Deconvolution",
      "name": "deconvolutionl",
      "attrs": {
        "kernel": "(1, 4)",
        "num_filter": "512",
        "pad": "(0, 0)",
        "stride": "(1, 4)"
      },
      "inputs": [[690, 0, 0], [691, 0, 0]]
    },
```

上图配置用于高帧率模型的反卷积配置，卷积核为 1*4；

```
      "op": "null",
      "name": "deconvolutionl_weight",
      "attrs": {
        "kernel": "(1, 1)",
        "num_filter": "512",
        "pad": "(0, 0)",
        "stride": "(1, 1)"
      },
      "inputs": []
    },
    {
      "op": "Deconvolution",
      "name": "deconvolutionl",
      "attrs": {
        "kernel": "(1, 1)",
        "num_filter": "512",
        "pad": "(0, 0)",
        "stride": "(1, 1)"
      },
```

上图配置用于低帧率模型的反卷积配置，卷积核为 1*1；

同时需要根据自己的模型状态数设置下面 OP 中的 num_filter 的个数。

```
    {
      "op": "null",
      "name": "convolution77_weight",
      "attrs": {
        "cudnn_off": "True",
        "cudnn_tune": "off",
        "dilate": "(1, 1)",
        "kernel": "(1, 1)",
        "no_bias": "False",
        "num_filter": "9004",
        "pad": "(0, 0)",
        "stride": "(1, 1)"
      },
      "inputs": []
    },
    {
      "op": "null",
      "name": "convolution77_bias",
      "attrs": {
        "cudnn_off": "True",
        "cudnn_tune": "off",
        "dilate": "(1, 1)",
        "kernel": "(1, 1)",
        "no_bias": "False",
        "num_filter": "9004",
        "pad": "(0, 0)",
        "stride": "(1, 1)"
      },
      "inputs": []
    },
    {
      "op": "Convolution",
      "name": "convolution77",
      "attrs": {
        "cudnn_off": "True",
        "cudnn_tune": "off",
        "dilate": "(1, 1)",
        "kernel": "(1, 1)",
        "no_bias": "False",
        "num_filter": "9004",
        "pad": "(0, 0)",
        "stride": "(1, 1)"
      },
```