www.paddlepaddle.org.cn

飞桨

飞桨核心框架2.0RC API升级概要

2020.07.31

全新升级的API体系

飞桨致力于让深度学习技术的创新与应用更简单

更简洁

更易用

更强大

全新升级的API体系

更简洁

飞桨

高层API简化训练流程：mnist数据集分类案例

训练脚本

```python
import paddle

normalize = paddle.vision.transforms.Normalize(mean=[0.0], std=[256.])
train_dataset = paddle.vision.datasets.MNIST(mode='train', transform=normalize)
val_dataset = paddle.vision.datasets.MNIST(mode='test', transform=normalize)

mnist = paddle.nn.Sequential(
    paddle.nn.Flatten(),
    paddle.nn.Linear(784, 1024),
    paddle.nn.ReLU(),
    paddle.nn.Linear(1024, 512),
    paddle.nn.ReLU(),
    paddle.nn.Linear(512, 10)
)
model = paddle.Model(mnist)

model.prepare(paddle.optimizer.Adam(parameters=model.parameters()),
              paddle.nn.CrossEntropyLoss(),
              paddle.metric.Accuracy())

model.fit(train_dataset, val_dataset, epochs=5, batch_size=32, verbose=1)
```

模型概要

```
print(model.summary((1, 28, 28)))
```

```
---------------------------------------------------------------------------
 Layer (type)       Input Shape          Output Shape         Param #
===========================================================================
 Flatten-1756      [[1, 28, 28]]           [1, 784]              0
   Linear-4          [[1, 784]]           [1, 1024]          803,840
    ReLU-3          [[1, 1024]]           [1, 1024]             0
   Linear-5         [[1, 1024]]            [1, 512]          524,800
    ReLU-4          [[1, 512]]             [1, 512]             0
   Linear-6         [[1, 512]]             [1, 10]            5,130
===========================================================================
Total params: 1,333,770
Trainable params: 1,333,770
Non-trainable params: 0
---------------------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.03
Params size (MB): 5.09
Estimated Total Size (MB): 5.12
---------------------------------------------------------------------------

{'total_params': 1333770, 'trainable_params': 1333770}
```

训练进程

```
Epoch 1/5
step 1875/1875 [==============================] - loss: 0.1680 - acc: 0.9445 - 13ms/step
Eval begin...
step 313/313 [==============================] - loss: 0.0204 - acc: 0.9668 - 4ms/step
Eval samples: 10000
Epoch 2/5
step 1875/1875 [==============================] - loss: 0.0644 - acc: 0.9753 - 19ms/step
Eval begin...
step 313/313 [==============================] - loss: 0.0739 - acc: 0.9698 - 5ms/step
Eval samples: 10000
Epoch 3/5
step 1875/1875 [==============================] - loss: 0.1324 - acc: 0.9821 - 22ms/step
Eval begin...
step 313/313 [==============================] - loss: 0.0129 - acc: 0.9790 - 4ms/step
Eval samples: 10000
Epoch 4/5
step 1875/1875 [==============================] - loss: 4.0998e-04 - acc: 0.9863 - 22ms/step
Eval begin...
step 313/313 [==============================] - loss: 3.4943e-04 - acc: 0.9749 - 4ms/step
Eval samples: 10000
Epoch 5/5
step 1875/1875 [==============================] - loss: 0.0373 - acc: 0.9890 - 22ms/step
Eval begin...
step 313/313 [==============================] - loss: 1.3463e-05 - acc: 0.9811 - 4ms/step
Eval samples: 10000
```

全新升级的API体系

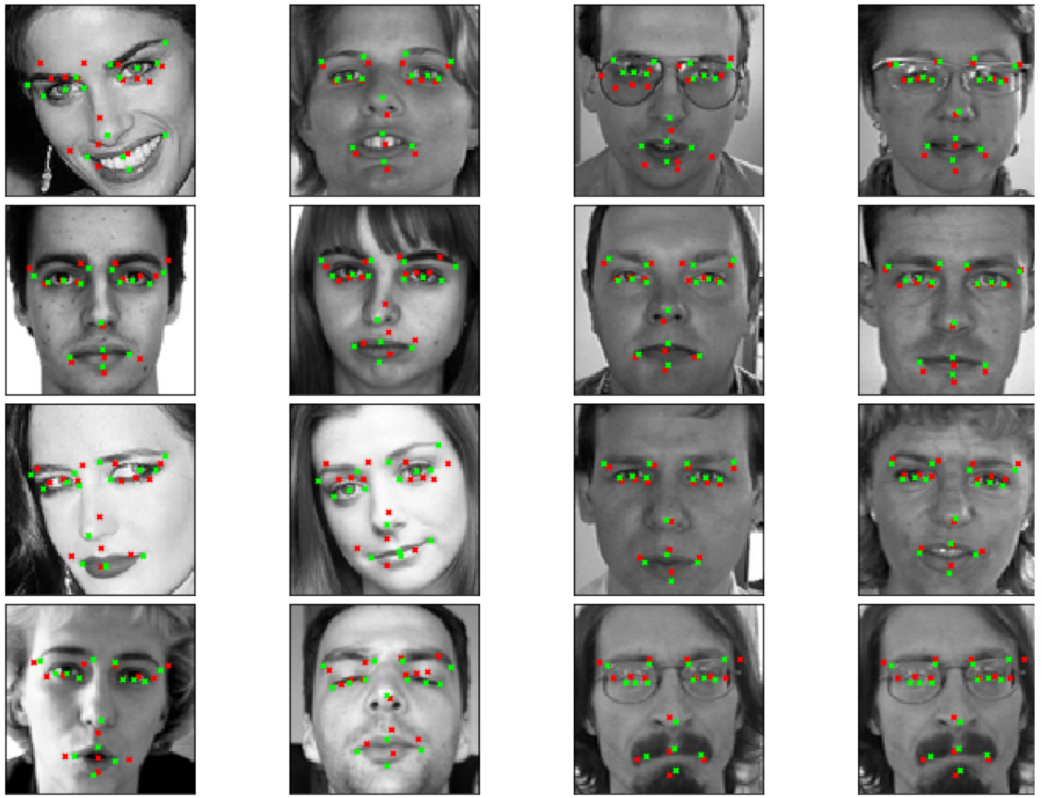
更简洁

飞桨

高层API提供经典预制模型：人脸关键点检测案例

```python
class FaceNet(paddle.nn.Layer):
    def __init__(self, num_keypoints, pretrained=False):
        super(FaceNet, self).__init__()

        self.backbone = paddle.vision.resnet50(pretrained)

        self.outLayer1 = paddle.nn.Sequential(
            paddle.nn.Linear(1000, 512,),
            paddle.nn.ReLU(),
            paddle.nn.Dropout(0.1))
        self.outLayer2 = paddle.nn.Linear(512, num_keypoints*2)

    def forward(self, inputs):
        out = self.backbone(inputs)
        out = self.outLayer1(out)
        out = self.outLayer2(out)
        return out

model = paddle.Model(FaceNet(num_keypoints=15))

optim = paddle.optimizer.Adam(
    learning_rate=1e-3,
    parameters=model.parameters())

model.prepare(optim, paddle.nn.MSELoss())

model.fit(train_dataset,
          val_dataset,
          epochs=60,
          batch_size=256)
```

一行代码使用Resnet50



关键点检测模型效果

全新升级的API体系

更简洁

飞桨

基础API洞察用户习惯：图片相似度案例

```python
class MyNet(paddle.nn.Layer):
    def __init__(self):
        super(MyNet, self).__init__()

        self.conv1 = paddle.nn.Conv2D(in_channels=3,
                                      out_channels=32,
                                      kernel_size=(3, 3),
                                      stride=2)

        self.conv2 = paddle.nn.Conv2D(in_channels=32,
                                      out_channels=64,
                                      kernel_size=(3,3),
                                      stride=2)

        self.conv3 = paddle.nn.Conv2D(in_channels=64,
                                      out_channels=128,
                                      kernel_size=(3,3),
                                      stride=2)

        self.gloabl_pool = paddle.nn.AdaptiveAvgPool2D((1,1))

        self.fc1 = paddle.nn.Linear(in_features=128, out_features=8)
    def forward(self, x):
        x = self.conv1(x)
        x = F.relu(x)
        x = self.conv2(x)
        x = F.relu(x)
        x = self.conv3(x)
        x = F.relu(x)
        x = self.gloabl_pool(x)
        x = paddle.squeeze(x, axis=[2, 3])
        x = self.fc1(x)
        x = x / paddle.norm(x, axis=1, keepdim=True)
        return x
```

class API方便组装

functional API方便定制

```python
for epoch in range(epoch_num):
    for batch_id, data in enumerate(pairs_train_reader()):
        anchors_data, positives_data = data[0], data[1]

        anchors = paddle.to_tensor(anchors_data)
        positives = paddle.to_tensor(positives_data)

        anchor_embeddings = model(anchors)
        positive_embeddings = model(positives)
        #print(anchor_embeddings.shape)

        similarities = paddle.matmul(anchor_embeddings, positive_embeddings, transpose_y=True)
        similarities = paddle.multiply(similarities, inverse_temperature)

        sparse_labels = paddle.arange(0, num_classes, dtype='int64')

        loss = F.cross_entropy(similarities, sparse_labels)

        loss.backward()
        opt.step()
        opt.clear_grad()
```

数学运算API方便灵活

图片相似度模型效果

全新升级的API体系

更易用

飞桨

API目录结构重新组织，API更易查找

目录	功能和包含的API
paddle.*	paddle根目录下保留了常用API的别名，当前包括：paddle.tensor和paddle.framework目录下的所有API。
paddle.tensor	tensor操作相关的API，例如创建zeros、矩阵运算matmul、变换concat、计算add、查找argmax等。
paddle.framework	框架通用API和动态图模式的API，例如no_grad、save、load等。
paddle.amp	paddle自动混合精度策略，包括auto_cast、GradScaler等。
paddle.callbacks	paddle日志回调类，包括ModelCheckpoint、ProgBarLogger等。
paddle.nn	组网相关的API，例如Linear、卷积、LSTM、损失函数、激活函数等。
paddle.static.nn	静态图下组网专用API，例如全连接层fc、控制流while_loop/cond。
paddle.static	静态图下基础框架相关API，例如Variable、Program、Executor等。
paddle.optimizer	优化算法相关API，例如SGD、Adagrad、Adam等。
paddle.optimizer.lr	学习率衰减相关API。
paddle.metric	评估指标计算相关的API，例如Accuracy、Auc等。
paddle.io	数据输入输出相关API，例如Dataset、DataLoader等。
paddle.device	设备管理相关API，例如CPUPlace、CUDAPlace等。
paddle.distributed	分布式相关基础API。
paddle.distributed.fleet	分布式相关高层API。
paddle.vision	视觉领域API，例如数据集、数据处理、常用基础网络结构，如ResNet等。
paddle.text	NLP领域API，目前包括NLP领域相关的数据集

规范命名、隐藏细节、API更易懂

```
paddle.fluid.dygraph.Conv2D(
    num_channels,
    num_filters,
    filter_size,
    padding=0,
    stride=1,
    dilation=1,
    groups=1,
    param_attr=None,
    bias_attr=None,
    use_cudnn=True,
    act=None,
    data_format='NCHW',
    dtype='float32')
```

paddle1.8

```
paddle.nn.Conv2D(
    in_channels,
    out_channels,
    kernel_size,
    stride=1,
    padding=0,
    dilation=1,
    groups=1,
    padding_mode='zeros',
    weight_attr=None,
    bias_attr=None,
    data_format='NCHW')
```

paddle2.0RC

同类API接口一致，更规范

```
paddle.nn.SimpleRNN(
    input_size,
    hidden_size,
    num_layers=1,
    direction='forward',
    time_major=False,
    dropout=0.0,
    activation='tanh',
    weight_ih_attr=None,
    weight_hh_attr=None,
    bias_ih_attr=None,
    bias_hh_attr=None,
    name=None)
```

```
paddle.nn.GRU(
    input_size,
    hidden_size,
    num_layers=1,
    direction='forward',
    time_major=False,
    dropout=0.0,
    weight_ih_attr=None,
    weight_hh_attr=None,
    bias_ih_attr=None,
    bias_hh_attr=None,
    name=None)
```

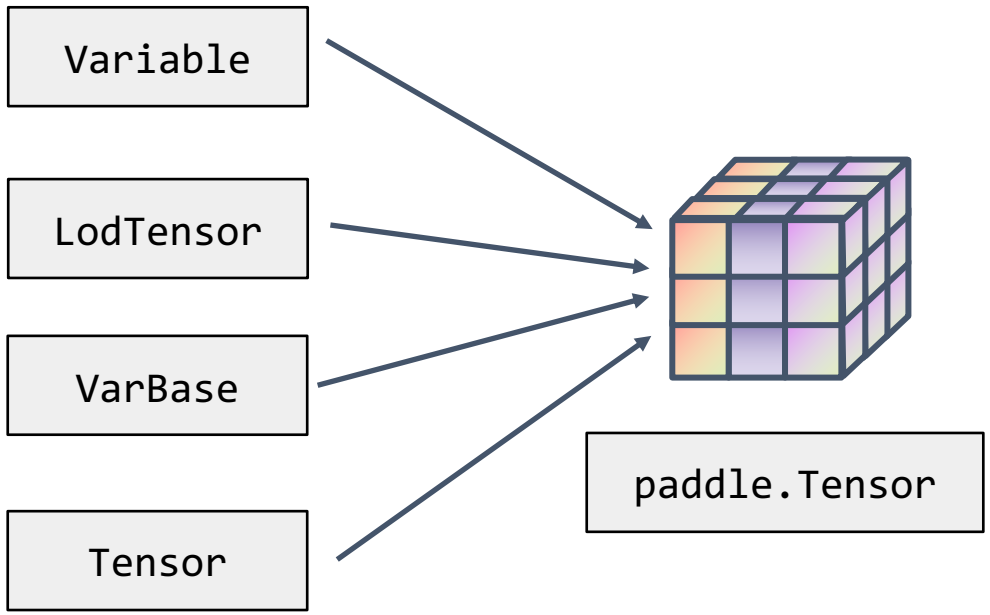
```
paddle.nn.LSTM(
    input_size,
    hidden_size,
    num_layers=1,
    direction='forward',
    time_major=False,
    dropout=0.0,
    weight_ih_attr=None,
    weight_hh_attr=None,
    bias_ih_attr=None,
    bias_hh_attr=None,
    name=None)
```

paddle2.0RC中的RNN系列API

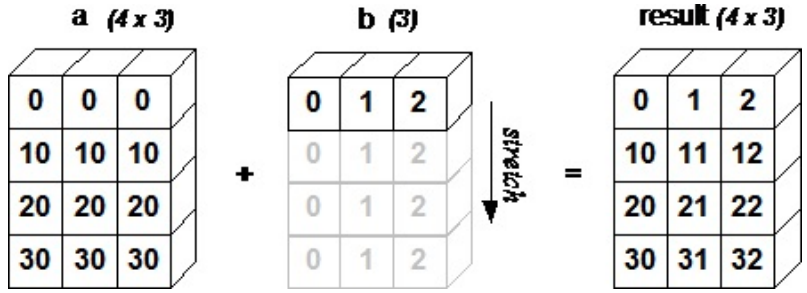
全新升级的API体系

更易用

统一的基础数据结构Tensor，更易懂



30+API支持完整广播语义，更方便



完善随机数生成机制，更易复现和调试

```python
import paddle

paddle.seed(12345)
# following codes will reproduce perfectly
```

设置全局种子后，每次运行，随机行为完全一致。

全新升级的API体系

更易用

飞桨

去掉额外依赖，新增环境支持，更适用

CUDA10.1

CUDA10.2

paddle2.0RC新增支持

- opencv
- nltk
- matplotlib
- graphviz
- netifaces
- objgraph

paddle2.0RC移除软件包依赖

报错信息内容及格式优化，更易调试

表意不明

信息不全

信息缺失

…

5000+ 信息优化完成

```
/* Paddle Exception mapping rules:
 *   - InvalidArgumentError -> ValueError
 *   - NotFoundError -> RuntimeError
 *   - OutOfRangeError -> IndexError
 *   - AlreadyExistsError -> RuntimeError
 *   - ResourceExhaustedError -> MemoryError
 *   - PreconditionNotMetError -> RuntimeError
 *   - PermissionDeniedError -> RuntimeError
 *   - ExecutionTimeoutError -> RuntimeError
 *   - UnimplementedError -> NotImplementedError
 *   - UnavailableError -> RuntimeError
 *   - FatalError -> SystemError
 *   - ExternalError -> OSError
 */
```

建立错误信息上移至python机制，更易理解

全新升级的API体系

更强大

飞桨

丰富API，完善功能，能力更强的API

- 新增基础功能类API 140+
- 8个API新增二阶导支持
- 完善API对数据类型的支持
- 50+个API新增昆仑芯片支持

前沿论文，轻松复现；完善昆仑芯片支持

新增动态图的混合精度训练，性能更优

```python
import paddle

model = paddle.nn.Conv2D(3, 2, 3, bias_attr=True)
optimizer = paddle.optimizer.SGD(learning_rate=0.01,
                                 parameters=model.parameters())
scaler = paddle.amp.GradScaler(init_loss_scaling=1024)
data = paddle.rand([10, 3, 32, 32])

with paddle.amp.auto_cast():
    conv = model(data)
    loss = paddle.mean(conv)
    scaled = scaler.scale(loss)  # scale the loss
    scaled.backward()            # do backward
    scaler.minimize(optimizer, scaled)  # update parameters
```

自动选择使用FP16训练，更快，更节省显存

全新升级的API体系

更强大

飞桨

动态图下调用方式优化，运行提速40%

```
if in_dygraph_mode():
    return core.ops.stack(x, 'axis', axis)

helper = LayerHelper('stack', **locals())
out = helper.create_variable_for_type_inference(x[0].dtype)
if x[0].desc.type() == core.VarDesc.VarType.LOD_TENSOR_ARRAY:
    assert len(x) == 1, "If the elements of 'x' in stack are Vari
                        "number of the elements must be 1, but re
```

所有API使用代码自动生成技术，建立了动态图快捷分支

优化DataLoader，边读边训，速度更快

- static graph

Model	batch_size	original DataLoader	multiprocess DataLoader	speed up ratio
ResNet50	8*32	711ms/step	266ms/step	167.3%
YOLOv3-DarkNet53	8*8	3155ms/step	708ms/step	345.6%
TSM-ResNet50	8*16	8665ms/step	1669ms/step	419.2%
BMN	8*16	2393ms/step	772ms/step	209.9%

- dynamic graph

Model	batch_size	origin DataLoader	multiprocess DataLoader	speed up ratio
ResNet50	8*32	269ms/step	267ms/step	-
YOLOv3-DarkNet53	8*8	751ms/step	623ms/step	20.5%
TSM-ResNet50	8*16	3550ms/step	1767ms/step	89.6%
BMN	8*16	831ms/step	785ms/step	5.8%

设置多个读取worker，大幅提升训练性能

成熟完备的动态图开发模式

默认开启动态图开发模式，
支持视觉、文本、推荐、语音、等等，全类别的模型算法

调试方便

灵活组网

一键转静

成熟完备的动态图开发模式

调试方便

```python
a = paddle.randn([4, 2])
b = paddle.arange(1, 3, dtype='float32')

print(a)
print(b)

c = a + b
print(c)

d = paddle.matmul(a, b)
print(d)
```

```
Tensor(shape=[4, 2], dtype=float32, place=CPUPlace, stop_gradient=True,
       [[-1.07110083,  0.87224680],
        [-0.32849231,  0.26714513],
        [ 1.46162450, -0.52989227],
        [-0.74100786,  0.67785585]])
Tensor(shape=[2], dtype=float32, place=CPUPlace, stop_gradient=True,
       [1., 2.])
Tensor(shape=[4, 2], dtype=float32, place=CPUPlace, stop_gradient=True,
       [[-0.07110083,  2.87224674],
        [0.67150772, 2.26714516],
        [2.46162462, 1.47010779],
        [0.25899214, 2.67785597]])
Tensor(shape=[4], dtype=float32, place=CPUPlace, stop_gradient=True,
       [0.67339277, 0.20579794, 0.40183997, 0.61470383])
```

2.0RC中默认开启了动态图，运行API会立刻返回结果到python。
不再需要首先创建一个计算图，然后再给定数据去运行。

成熟完备的动态图开发模式

灵活组网

飞桨

```python
a = paddle.to_tensor(np.array([1, 2, 3]))
b = paddle.to_tensor(np.array([4, 5, 6]))

for i in range(10):
    r = paddle.rand([1,])
    if r > 0.5:
        c = paddle.pow(a, i) + b
        print("{} +> {}".format(i, c.numpy()))
    else:
        c = paddle.pow(a, i) - b
        print("{} -> {}".format(i, c.numpy()))
```

```
0 -> [-3 -4 -5]
1 +> [5 7 9]
2 -> [-3 -1  3]
3 -> [-3  3 21]
4 -> [-3 11 75]
5 +> [  5  37 249]
6 +> [  5  69 735]
7 -> [  -3  123 2181]
8 +> [   5  261 6567]
9 +> [    5   517 19689]
```

使用动态图可以直接使用python控制流

```python
inputs = paddle.rand((256, 64))

linear = paddle.nn.Linear(64, 8, bias_attr=False)
loss_fn = paddle.nn.MSELoss()
optimizer = paddle.optimizer.Adam(0.01, parameters=linear.parameters())

for i in range(10):
    hidden = linear(inputs)
    # weight from input to hidden is shared with
    # the linear mapping from hidden to output
    outputs = paddle.matmul(hidden, linear.weight, transpose_y=True)
    loss = loss_fn(outputs, inputs)
    loss.backward()
    print("step: {}, loss: {}".format(i, loss.numpy()))
    optimizer.step()
    optimizer.clear_grad()
```

```
step: 0, loss: [0.34074003]
step: 1, loss: [0.31435657]
step: 2, loss: [0.28938287]
step: 3, loss: [0.2572552]
step: 4, loss: [0.22024116]
step: 5, loss: [0.1854592]
step: 6, loss: [0.15818128]
step: 7, loss: [0.13687664]
step: 8, loss: [0.1182529]
step: 9, loss: [0.10262887]
```

动态图下创建权重共享网络更方便

成熟完备的动态图开发模式

灵活组网

飞桨

```python
for epoch in range(epochs):
    # shuffle training data
    perm = np.random.permutation(len(train_en_sents))
    train_en_sents_shuffled = train_en_sents[perm]
    train_cn_sents_shuffled = train_cn_sents[perm]
    train_cn_label_sents_shuffled = train_cn_label_sents[perm]

    for iteration in range(train_en_sents_shuffled.shape[0] // batch_size):
        x_data = train_en_sents_shuffled[(batch_size*iteration):(batch_size*(iteration+1))]
        sent = paddle.to_tensor(x_data)
        en_repr = encoder(sent)

        x_cn_data = train_cn_sents_shuffled[(batch_size*iteration):(batch_size*(iteration+1))]
        x_cn_label_data = train_cn_label_sents_shuffled[(batch_size*iteration):(batch_size*(iteration+1))]

        # shape: (batch,  num_layer(=1 here) * num_of_direction(=1 here), hidden_size)
        hidden = paddle.zeros([batch_size, 1, hidden_size])
        cell = paddle.zeros([batch_size, 1, hidden_size])

        loss = paddle.zeros([1])
        # the decoder recurrent loop mentioned above
        for i in range(MAX_LEN + 2):
            cn_word = paddle.to_tensor(x_cn_data[:,i:i+1])
            cn_word_label = paddle.to_tensor(x_cn_label_data[:,i])

            logits, (hidden, cell) = atten_decoder(cn_word, hidden, cell, en_repr)
            step_loss = F.cross_entropy(logits, cn_word_label)
            loss += step_loss

        loss = loss / (MAX_LEN + 2)

        loss.backward()
        opt.step()
        opt.clear_grad()
```

更自然的实现自定义loss，和自定义的循环网络。

成熟完备的动态图开发模式

一键转静

飞桨

```python
import paddle
from paddle.jit import to_static
from paddle.static import InputSpec

class SimpleNet(paddle.nn.Layer):
    def __init__(self):
        super(SimpleNet, self).__init__()
        self.linear = paddle.nn.Linear(10, 3)

    # 通过InputSpec指定输入数据的形状，None表示可变长
    # 通过to_static装饰器将动态图转换为静态图Program
    @to_static(input_spec=[InputSpec(shape=[None, 10], name='x'), InputSpec(shape=[3], name='y')])
    def forward(self, x, y):
        out = self.linear(x)
        out = out + y
        return out

net = SimpleNet()

# 保存静态图模型，可用于预测部署
paddle.jit.save(net, './simple_net')

```

通过一个装饰器（to_static），框架就会自动将用户的程序，转换为静态图的program，并使用该program预测部署和训练

Proud Paddle Team