



深度学习框架Caffe学习与应用 第5课

DATAGURU专业数据分析社区

本节课内容

- 自定义Layer的实现
 - 自定义一个计算层，实现 $y=x^{\text{power}}$ 的功能
 - 对自定义层进行测试
 - 自定义一个新的输入层

自定义Layer的实现

■ 选择继承的类

可以根据自己的需要，从已有的类中选择自己继承的类，这样可以省却很多麻烦。一般可以继承的类包括

层	描述
Layer	如果你要定义的层和已有的层没有什么重叠，那么可以选择直接继承Layer
DataLayer	自定义网络输入层时，可以考虑继承它，内部的load_batch可能是你要重写的函数
NeuronLayer	自定义神经层，也就是中间的进行运算的层
LossLayer	如果现有的损失函数层不能满足需求，可以继承它

- *任何一个层都可以被继承，然后进行重写函数。
- *尽量确保要实现的功能是否必须要自己写，不然尽量用已有的层，每一个层在caffe/include/caffe/layers源码中都有详细的介绍。

自定义神经层

- 步骤：
- 1. 创建新定义的头文件include/caffe/layers/my_neuron_layer.hpp
 - 重新Layer名的方法：`virtual inline const char* type() const { return "MyNeuron"; }`
 - 如果只是需要cpu方法的话，可以注释掉forward/backward_gpu()这两个方法
- 2. 创建对应src/caffe/layers/my_neuron_layer.cpp的源文件
 - 重写方法LayerSetUp,实现从能从prototxt读取参数
 - 重写方法Reshape，如果对继承类没有修改的话，就不需要重写
 - 重写方法Forward_cpu
 - 重写方法Backward_cpu(非必须)
 - * 如果要GPU支持，则还需要创建src/caffe/src/my_neuron_layer.cu，同理重写方法Forward_gpu/Backward_gpu(非必须)

3. caffe/src/caffe/proto/caffe.proto注册新的Layer

```
message LayerParameter{
...
加入 optional MyNeuronParameter
my_square_param = 150;
...
}

加入 message MyNeuronParameter {
optional float power = 1 [default = 2];
}
...
message V1LayerParameter{
...
加入 MYNEURON = 40;
...
}
```

5. 重新编译和install

进入caffe安装目录：cd ~/caffe/build

编译：cmake -D CPU_ONLY=on -D CMAKE_INSTALL_PREFIX=/usr/local ..

sudo make all &sudo make install

4. my_neuron_layer.cpp添加注册的宏定义

```
INstantiateClass(MyNeuronLayer);
REGISTER_LAYER_CLASS(MyNeuron);
```

如果有my_neuron_layer.cu,则添加

```
INstantiateLayerGpuFuncs(MyNeuronLayer);
```

测试自定义的Layer

■ 定义deploy.prototxt

```
name: "CaffeNet"
input: "data"
input_shape {
  dim: 1 # batchsize
  dim: 1 # number of colour channels -
  rgb
  dim: 28 # width
  dim: 28 # height
}

layer {
  name: "myneuron"
  type: "MyNeuron"
  bottom: "data"
  top: "data_out"
  my_neuron_param {
    power : 2
  }
}
```

■ 运行测试程序：

test_my_neuron.py

自定义数据输入层

- 第1, 3, 4, 5步和自定义神经层一样, 只有第2步重写的函数有不同:
- 数据输入层需要重写三个函数:
 1. DataLayerSetUp: 定义好从prototxt读入的参数名和容器的规格 (设好N, K, H, W)
 2. ShuffleImages: 打乱顺序
 3. load_batch: 把图片读入到内存