

CNN主流软件框架

Deep learning frameworks



theano



Caffe

PYTORCH

CAFFE

(Convolutional Architecture for Fast Feature Embedding)

- 开发者: Berkeley AI Research, Yangqing Jia
- 官网: <http://caffe.berkeleyvision.org/>
- Github: <https://github.com/BVLC/caffe>

Caffe构建网络

整个运行过程分为三个阶段：**训练阶段**、**验证阶段**和**测试阶段**。一般训练和验证阶段使用同一个网络结构，测试阶段使用另一个网络结构（有些许差异，不包括卷积层参数的初始化和损失层，输入层有所不同）。

网络结构都放在.prototxt文件里面：
训练和验证阶段： train_val.prototxt
测试阶段： deploy.prototxt

```
1  name: "LeNet"
2  layer {
3    name: "mnist"
4    type: "Data"
5    top: "data"
6    top: "label"
7    include {
8      phase: TRAIN
9    }
10   transform_param {
11     scale: 0.00390625
12   }
13   data_param {
14     source: "examples/mnist/mnist_train_lmdb"
15     batch_size: 64
16     backend: LMDB
17   }
18 }
```

数据层

```
36 layer {
37   name: "conv1"
38   type: "Convolution"
39   bottom: "data"
40   top: "conv1"
41   param {
42     lr_mult: 1
43   }
44   param {
45     lr_mult: 2
46   }
47   convolution_param {
48     num_output: 20
49     kernel_size: 5
50     stride: 1
51     weight_filler {
52       type: "xavier"
53     }
54     bias_filler {
55       type: "constant"
56     }
57   }
58 }
```

卷积层

```
59 layer {
60   name: "pool1"
61   type: "Pooling"
62   bottom: "conv1"
63   top: "pool1"
64   pooling_param {
65     pool: MAX
66     kernel_size: 2
67     stride: 2
68   }
69 }
```

池化层

```
104 layer {
105   name: "ip1"
106   type: "InnerProduct"
107   bottom: "pool2"
108   top: "ip1"
109   param {
110     lr_mult: 1
111   }
112   param {
113     lr_mult: 2
114   }
115   inner_product_param {
116     num_output: 500
117     weight_filler {
118       type: "xavier"
119     }
120     bias_filler {
121       type: "constant"
122     }
123   }
124 }
```

全连接层

Caffe构建网络

```
net: "examples/mnist/lenet_train_test.prototxt"  
test_iter: 100  
test_interval: 500  
base_lr: 0.01  
momentum: 0.9  
weight_decay: 0.0005  
lr_policy: "inv"  
gamma: 0.0001  
power: 0.75  
display: 100  
max_iter: 10000  
snapshot: 5000  
snapshot_prefix: "examples/mnist/lenet"  
solver_mode: GPU
```



net: 网络结构
文件

Caffe构建网络

net: "examples/mnist/lenet_train_test.prototxt"

test_iter: 100 ●

test_interval: 500 ●●

base_lr: 0.01 ●

momentum: 0.9

weight_decay: 0.0005

lr_policy: "inv"

gamma: 0.0001

power: 0.75

display: 100

max_iter: 10000

snapshot: 5000

snapshot_prefix: "examples/mnist/lenet"

solver_mode: GPU

test_iter: 定义验证阶段的迭代次数，以MNIST为例，有10,000张测试图片，batch_size为100，则test_iter为100

Caffe构建网络

```
net: "examples/mnist/lenet_train_test.prototxt"  
test_iter: 100  
test_interval: 500  
base_lr: 0.01  
momentum: 0.9  
weight_decay: 0.0005  
lr_policy: "inv"  
gamma: 0.0001  
power: 0.75  
display: 100  
max_iter: 10000  
snapshot: 5000  
snapshot_prefix: "examples/mnist/lenet"  
solver_mode: GPU
```



test_interval: 验证间隔，每训练500个迭代开始验证

Caffe构建网络

```
net: "examples/mnist/lenet_train_test.prototxt"
test_iter: 100
test_interval: 500
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
lr_policy: "inv"
gamma: 0.0001
power: 0.75
display: 100
max_iter: 10000
snapshot: 5000
snapshot_prefix: "examples/mnist/lenet"
solver_mode: GPU
```



base_lr: 基础学习率

Caffe构建网络

net: "examples/mnist/lenet_train_test.prototxt"

test_iter: 100

test_interval: 500

base_lr: 0.01

momentum: 0.9

weight_decay: 0.0005

lr_policy: "inv"

gamma: 0.0001

power: 0.75

display: 100

max_iter: 10000

snapshot: 5000

snapshot_prefix: "examples/mnist/lenet"

solver_mode: GPU



lr_policy: 学习率
变化规则

<https://blog.csdn.net/cuijyer/article/details/78195178>

Caffe构建网络

```
net: "examples/mnist/lenet_train_test.prototxt"  
test_iter: 100  
test_interval: 500  
base_lr: 0.01  
momentum: 0.9  
weight_decay: 0.0005  
lr_policy: "inv"  
gamma: 0.0001  
power: 0.75  
display: 100  
max_iter: 10000  
snapshot: 5000  
snapshot_prefix: "examples/mnist/lenet"  
solver_mode: GPU
```



display: 屏幕显示间隔，
每个100个迭代显示输出

Caffe构建网络

```
net: "examples/mnist/lenet_train_test.prototxt"  
test_iter: 100  
test_interval: 500  
base_lr: 0.01  
momentum: 0.9  
weight_decay: 0.0005  
lr_policy: "inv"  
gamma: 0.0001  
power: 0.75  
display: 100  
max_iter: 10000  
snapshot: 5000  
snapshot_prefix: "examples/mnist/lenet"  
solver_mode: GPU
```



max_iter: 最大
训练次数

Caffe构建网络

```
net: "examples/mnist/lenet_train_test.prototxt"  
test_iter: 100  
test_interval: 500  
base_lr: 0.01  
momentum: 0.9  
weight_decay: 0.0005  
lr_policy: "inv"  
gamma: 0.0001  
power: 0.75  
display: 100  
max_iter: 10000  
snapshot: 5000  
snapshot_prefix: "examples/mnist/lenet"  
solver_mode: GPU
```



snapshot: 模型保存间隔
snapshot_prefix: 模型名字前缀

Caffe构建网络

```
net: "examples/mnist/lenet_train_test.prototxt"  
test_iter: 100  
test_interval: 500  
base_lr: 0.01  
momentum: 0.9  
weight_decay: 0.0005  
lr_policy: "inv"  
gamma: 0.0001  
power: 0.75  
display: 100  
max_iter: 10000  
snapshot: 5000  
snapshot_prefix: "examples/mnist/lenet"  
solver_mode: GPU
```



solver_mode:
选择使用CPU或
GPU