# **Caffe书章节**

## **章三 Caffe训练**

### **3.1 数据转换工具**

为了解决什么问题?

raw数据是jpg,jpeg,bmp等格式, 尺寸不同, Caffe需要的是LMDB或LiveDB格式. 需要将raw格式转成LMDB,LiveDB格式.

a) convert\_imageset命令(由对应文件编译).

b) 完成raw数据对LMDB,LiveDB的转换.

c) 可选flag有

    i) --gray, 是否以灰度值打开图片

    ii) --shuffle, 是否对raw图片洗牌.

    iii) --backend, 输出的db格式(从LMDB和LiveDB选择,默认LiveDB).

    iv) --resize\_widht/resize\_height, 对图像做resize.

生成图像列表

a) 将路径下(example/images/\*.jpg)写进txt中.

b) txt中每行是(形成,摆成): " 文件名  label值" 的格式.

c) label值从0开始.

### **3.2 特征提取工具**

a) extract\_features工具(命令)

b) 基于caffe.model

c) 输出Cafffe model对应的layer

d) 输出对应layer文件的格式(保存格式), LMDB或LiveDB.

e) 缺点是: 这些输出的特征使用起来不是很方便.可以用memorydata来做特征提取.

### **3.3 Caffe训练需要的几个部件**

proto文件规定了网络结构.(每层的结构).

a) 前面3.1提到的convert\_imageset是为了准备输入数据.

b) proto文件描述Caffe网络结构(当前层结构)

c) type表示当前层格式(Convolution为卷积层).

d) bottom,该层的输入来源

e) top, 该层的输出去向(输出给哪一层?).

f) convolution\_param, 是可选参数. 不同层参数不一样.

Solver配置文件

a) solver.prototxt是如何训练的文件.

    i) 规定好 学习率策略, 优化策略, 迭代次数等.

b) lr\_policy是学习率下降(decay)策略.

   i) fixed 学习率不下降(不变)

   ii) step, 和step相关的下降(非线性)

   iii) exp, inv, multistep, poly, sigmoid等学习率下降策略(delay policy).

训练命令

a) 采用build/tools/caffe作为训练工具

b) 输入trian, 表示训练阶段

c) 输入solver.prototxt文件, 描述需要如何训练网络.

d) 输入gpu选项, 表示采用的gpu

e) 输入weight选项, 描述是否采用预处理模型做fine-tune.

训练日志(log)

a) 设置GLOG\_logtostderr和GLOG\_log\_dir来使用log.

b) 将上述环境变量输入给caffe工具(命令,程序).

c) 采用extra/parse\_log.py来解析保存的log.

d) 采用extra/plot\_training\_log.py来绘制.

    i) 输入一个整数(id) 描述需要(目标查看)的图.

    ii) 输出一个绘制图, 把这个图保存成输出规定的名字.

    iii) 基于matplotlib实现.

e)

# caffe 知识点汇总

## 一 caffe proto

### 何为protocol buffer

1. 混合语言数据标准
   1. 对象是混合的
   2. 数据标准
2. 结构化数据成为Message
   1. Proto文件语言类似java或c语言.

如下例子:

1. 定义package是lm
2. Message是 helloword
3. (建议)proto文件取名为” packagename.messagename.proto”

|  |
| --- |
| package lm;  message helloworld  {  required int32 id = 1; // ID  required string str = 2; // str  optional int32 opt = 3; //optional field  } |

1. Proto文件的编译

在RefineDet/Makefile中.

1. 对proto文件做了编译.
2. –cpp\_out是输出编译后的c++文件的(地方)
3. –python\_out是输出编译后的python文件的(地方)
4. Proto是描述(性质)的文件,本身需要编译成c++或python或c语言.

|  |
| --- |
| $(PROTO\_BUILD\_DIR)/%.pb.cc $(PROTO\_BUILD\_DIR)/%.pb.h **:** \  $(PROTO\_SRC\_DIR)/%.proto | $(PROTO\_BUILD\_DIR)  @ echo PROTOC $<  $(Q)protoc --proto\_path=$(PROTO\_SRC\_DIR) --cpp\_out=$(PROTO\_BUILD\_DIR) $<  $(PY\_PROTO\_BUILD\_DIR)/%\_pb2.py **:** $(PROTO\_SRC\_DIR)/%.proto \  $(PY\_PROTO\_INIT) | $(PY\_PROTO\_BUILD\_DIR)  @ echo PROTOC \(python\) $<  $(Q)protoc --proto\_path=$(PROTO\_SRC\_DIR) --python\_out=$(PY\_PROTO\_BUILD\_DIR) $< |

### protobuf为了解决什么问题

设想下:

一个writer写数据”123”到磁盘,其本意是123, 一个reader从磁盘读取数据,”123”,它可以认为是1,2,3等组合的任何. 这就要求reader和writer需要有相同的解析结构.数据传播比较繁琐.

引入Protobuf后,reader无需关注这些”命令协议”, protobuf会为数据”123”创建一个类(message?类似的),会自动处理这些数据的含义.

### caffe message的用法

以下message为例.

* 1. 含有多个字段.
  2. 这是proto/caffe.proto文件. 会经编译出对应的c++或者python文件.
  3. **对应的c++文件中会有对每个字段的set\_\*函数实现.以便对字段的设置.**

|  |
| --- |
| // Message that store parameters used by DetectionOutputLayer  message DetectionOutputParameter **{**  // Number of classes to be predicted. Required!  optional uint32 num\_classes **=** 1**;**  // If true, bounding box are shared among different classes.  optional bool share\_location **=** 2 **[default** **=** true**];**  // Background label id. If there is no background class,  // set it as -1.  optional int32 background\_label\_id **=** 3 **[default** **=** 0**];**  // Parameters used for non maximum suppression.  optional NonMaximumSuppressionParameter nms\_param **=** 4**;**  // Parameters used for saving detection results.  optional SaveOutputParameter save\_output\_param **=** 5**;**  // Type of coding method for bbox.  optional PriorBoxParameter**.**CodeType code\_type **=** 6 **[default** **=** CORNER**];**  // If true, variance is encoded in target; otherwise we need to adjust the  // predicted offset accordingly.  optional bool variance\_encoded\_in\_target **=** 8 **[default** **=** false**];**  // Number of total bboxes to be kept per image after nms step.  // -1 means keeping all bboxes after nms step.  optional int32 keep\_top\_k **=** 7 **[default** **=** **-**1**];**  // Only consider detections whose confidences are larger than a threshold.  // If not provided, consider all boxes.  optional float confidence\_threshold **=** 9**;**  // If true, visualize the detection results.  optional bool visualize **=** 10 **[default** **=** false**];**  // The threshold used to visualize the detection results.  optional float visualize\_threshold **=** 11**;**  // If provided, save outputs to video file.  optional string save\_file **=** 12**;**  //the objectness score is used for the anchor refinement module to filter easy negative anchor.  optional float objectness\_score **=** 24 **[default** **=** 0.01**];**  **}** |

对message的num\_classes字段的”写”(设置):

|  |
| --- |
| TYPED\_TEST**(**DetectionOutputLayerTest**,** TestSetup**)** **{**  **typedef** typename TypeParam**::**Dtype Dtype**;**  LayerParameter layer\_param**;**  DetectionOutputParameter**\*** detection\_output\_param **=**  layer\_param**.**mutable\_detection\_output\_param**();**  **detection\_output\_param->set\_num\_classes(this->**num\_classes\_**);**  DetectionOutputLayer**<**Dtype**>** layer**(**layer\_param**);**  layer**.**SetUp**(this->**blob\_bottom\_vec\_**,** **this->**blob\_top\_vec\_**);**  EXPECT\_EQ**(this->**blob\_top\_**->**num**(),** 1**);**  EXPECT\_EQ**(this->**blob\_top\_**->**channels**(),** 1**);**  EXPECT\_EQ**(this->**blob\_top\_**->**height**(),** 1**);**  EXPECT\_EQ**(this->**blob\_top\_**->**width**(),** 7**);**  **}** |

## 二 caffe proto的重要数据类型

### 标准化操作

|  |
| --- |
| void CopyFrom();  void MergeFrom();  void CopyFrom();  void MergeFrom;  void Clear();  bool IsInitialized() const;  int ByteSize() const;  bool MergePartialFromCodedStream();  void SerializeWithCachedSizes() const;  SerializeWithCachedSizesToArray() const;  int GetCachedSize()  void SharedCtor();  void SharedDtor();  void SetCachedSize() const; |

### blobProto

|  |
| --- |
| message BlobProto {//blob的属性以及blob中的数据(data\diff)  optional int32 num = 1 [default = 0];  optional int32 channels = 2 [default = 0];  optional int32 height = 3 [default = 0];  optional int32 width = 4 [default = 0];  repeated float data = 5 [packed = true];  repeated float diff = 6 [packed = true];  } |

### Datum

|  |
| --- |
| message Datum {  optional int32 channels = 1;  optional int32 height = 2;  optional int32 width = 3;  optional bytes data = 4;//真实的图像数据，以字节存储(bytes)  optional int32 label = 5;  repeated float float\_data = 6;//datum也能存float类型的数据(float)  } |

### layerParameter

|  |
| --- |
| message LayerParameter {  repeated string bottom = 2; //输入的blob的名字(string)  repeated string top = 3; //输出的blob的名字(string)  optional string name = 4; //层的名字  enum LayerType { //层的枚举（enum，和c++中的enum一样）  NONE = 0;  ACCURACY = 1;  BNLL = 2;  CONCAT = 3;  CONVOLUTION = 4;  DATA = 5;  DROPOUT = 6;  EUCLIDEAN\_LOSS = 7;  ELTWISE\_PRODUCT = 25;  FLATTEN = 8;  HDF5\_DATA = 9;  HDF5\_OUTPUT = 10;  HINGE\_LOSS = 28;  IM2COL = 11;  IMAGE\_DATA = 12;  INFOGAIN\_LOSS = 13;  INNER\_PRODUCT = 14;  LRN = 15;  MEMORY\_DATA = 29;  MULTINOMIAL\_LOGISTIC\_LOSS = 16;  POOLING = 17;  POWER = 26;  RELU = 18;  SIGMOID = 19;  SIGMOID\_CROSS\_ENTROPY\_LOSS = 27;  SOFTMAX = 20;  SOFTMAX\_LOSS = 21;  SPLIT = 22;  TANH = 23;  WINDOW\_DATA = 24;  }  optional LayerType type = 5; // 层的类型  repeated BlobProto blobs = 6; //blobs的数值参数  repeated float blobs\_lr = 7; //学习速率(repeated)，如果你想那个设置一个blob的学习速率，你需要设置所有blob的学习速率。  repeated float weight\_decay = 8; //权值衰减(repeated)  // 相对于某一特定层的参数(optional)  optional ConcatParameter concat\_param = 9;  optional ConvolutionParameter convolution\_param = 10;  optional DataParameter data\_param = 11;  optional DropoutParameter dropout\_param = 12;  optional HDF5DataParameter hdf5\_data\_param = 13;  optional HDF5OutputParameter hdf5\_output\_param = 14;  optional ImageDataParameter image\_data\_param = 15;  optional InfogainLossParameter infogain\_loss\_param = 16;  optional InnerProductParameter inner\_product\_param = 17;  optional LRNParameter lrn\_param = 18;  optional MemoryDataParameter memory\_data\_param = 22;  optional PoolingParameter pooling\_param = 19;  optional PowerParameter power\_param = 21;  optional WindowDataParameter window\_data\_param = 20;  optional V0LayerParameter layer = 1;  } |

Layer DetectionOutput参数实例(区别于上面的layerparamters)

**这里的parameters是自定义的.**

某个自定义的detection layer的自定义参数

|  |
| --- |
| # parameters for generating detection output.  det\_out\_param **=** **{**  'num\_classes'**:** num\_classes**,**  'share\_location'**:** share\_location**,**  'background\_label\_id'**:** background\_label\_id**,**  'nms\_param'**:** **{**'nms\_threshold'**:** 0.45**,** 'top\_k'**:** 1000**},**  'keep\_top\_k'**:** 500**,**  'confidence\_threshold'**:** 0.01**,**  'code\_type'**:** code\_type**,**  'objectness\_score'**:** 0.01**,**  **}** |

该参数本来在caffe.proto中定义:

|  |
| --- |
| // Message that store parameters used by DetectionOutputLayer  message DetectionOutputParameter **{**  // Number of classes to be predicted. Required!  optional uint32 num\_classes **=** 1**;**  // If true, bounding box are shared among different classes.  optional bool share\_location **=** 2 **[default** **=** true**];**  // Background label id. If there is no background class,  // set it as -1.  optional int32 background\_label\_id **=** 3 **[default** **=** 0**];**  // Parameters used for non maximum suppression.  optional NonMaximumSuppressionParameter nms\_param **=** 4**;**  // Parameters used for saving detection results.  optional SaveOutputParameter save\_output\_param **=** 5**;**  // Type of coding method for bbox.  optional PriorBoxParameter**.**CodeType code\_type **=** 6 **[default** **=** CORNER**];**  // If true, variance is encoded in target; otherwise we need to adjust the  // predicted offset accordingly.  optional bool variance\_encoded\_in\_target **=** 8 **[default** **=** false**];**  // Number of total bboxes to be kept per image after nms step.  // -1 means keeping all bboxes after nms step.  optional int32 keep\_top\_k **=** 7 **[default** **=** **-**1**];**  // Only consider detections whose confidences are larger than a threshold.  // If not provided, consider all boxes.  optional float confidence\_threshold **=** 9**;**  // If true, visualize the detection results.  optional bool visualize **=** 10 **[default** **=** false**];**  // The threshold used to visualize the detection results.  optional float visualize\_threshold **=** 11**;**  // If provided, save outputs to video file.  optional string save\_file **=** 12**;**  //the objectness score is used for the anchor refinement module to filter easy negative anchor.  optional float objectness\_score **=** 24 **[default** **=** 0.01**];**  **}** |

自定义的layerparamters是基于layer paramters的,如下:

|  |
| --- |
| // NOTE  // Update the next available ID when you add a new LayerParameter field.  //  // LayerParameter next available layer-specific ID: 147 (last added: recurrent\_param)  message LayerParameter **{**  optional string name **=** 1**;** // the layer name  optional string type **=** 2**;** // the layer type  repeated string bottom **=** 3**;** // the name of each bottom blob  repeated string top **=** 4**;** // the name of each top blob  // The train / test phase for computation.  optional Phase phase **=** 10**;**  // The amount of weight to assign each top blob in the objective.  // Each layer assigns a default value, usually of either 0 or 1,  // to each top blob.  repeated float loss\_weight **=** 5**;**  // Specifies training parameters (multipliers on global learning constants,  // and the name and other settings used for weight sharing).  repeated ParamSpec param **=** 6**;**  // The blobs containing the numeric parameters of the layer.  repeated BlobProto blobs **=** 7**;**  // Specifies whether to backpropagate to each bottom. If unspecified,  // Caffe will automatically infer whether each input needs backpropagation  // to compute parameter gradients. If set to true for some inputs,  // backpropagation to those inputs is forced; if set false for some inputs,  // backpropagation to those inputs is skipped.  //  // The size must be either 0 or equal to the number of bottoms.  repeated bool propagate\_down **=** 11**;**  // Rules controlling whether and when a layer is included in the network,  // based on the current NetState. You may specify a non-zero number of rules  // to include OR exclude, but not both. If no include or exclude rules are  // specified, the layer is always included. If the current NetState meets  // ANY (i.e., one or more) of the specified rules, the layer is  // included/excluded.  repeated NetStateRule include **=** 8**;**  repeated NetStateRule exclude **=** 9**;**  // Parameters for data pre-processing.  optional TransformationParameter transform\_param **=** 100**;**  // Parameters shared by loss layers.  optional LossParameter loss\_param **=** 101**;**  // Layer type-specific parameters.  //  // Note: certain layers may have more than one computational engine  // for their implementation. These layers include an Engine type and  // engine parameter for selecting the implementation.  // The default for the engine is set by the ENGINE switch at compile-time.  optional AccuracyParameter accuracy\_param **=** 102**;**  optional AnnotatedDataParameter annotated\_data\_param **=** 200**;**  optional ArgMaxParameter argmax\_param **=** 103**;**  optional BatchNormParameter batch\_norm\_param **=** 139**;**  optional BiasParameter bias\_param **=** 141**;**  optional ConcatParameter concat\_param **=** 104**;**  optional ContrastiveLossParameter contrastive\_loss\_param **=** 105**;**  optional ConvolutionParameter convolution\_param **=** 106**;**  optional CropParameter crop\_param **=** 144**;**  optional DataParameter data\_param **=** 107**;**  optional DetectionEvaluateParameter detection\_evaluate\_param **=** 205**;**  optional **DetectionOutputParameter** detection\_output\_param **=** 204**;**  optional DropoutParameter dropout\_param **=** 108**;**  optional DummyDataParameter dummy\_data\_param **=** 109**;**  optional EltwiseParameter eltwise\_param **=** 110**;**  optional ELUParameter elu\_param **=** 140**;**  optional EmbedParameter embed\_param **=** 137**;**  optional ExpParameter exp\_param **=** 111**;**  optional FlattenParameter flatten\_param **=** 135**;**  optional HDF5DataParameter hdf5\_data\_param **=** 112**;**  optional HDF5OutputParameter hdf5\_output\_param **=** 113**;**  optional HingeLossParameter hinge\_loss\_param **=** 114**;**  optional ImageDataParameter image\_data\_param **=** 115**;**  optional InfogainLossParameter infogain\_loss\_param **=** 116**;**  optional InnerProductParameter inner\_product\_param **=** 117**;**  optional InputParameter input\_param **=** 143**;**  optional LogParameter log\_param **=** 134**;**  optional LRNParameter lrn\_param **=** 118**;**  optional MemoryDataParameter memory\_data\_param **=** 119**;**  optional MultiBoxLossParameter multibox\_loss\_param **=** 201**;**  optional MVNParameter mvn\_param **=** 120**;**  optional NormalizeParameter norm\_param **=** 206**;**  optional ParameterParameter parameter\_param **=** 145**;**  optional PermuteParameter permute\_param **=** 202**;**  optional PoolingParameter pooling\_param **=** 121**;**  optional PowerParameter power\_param **=** 122**;**  optional PReLUParameter prelu\_param **=** 131**;**  optional PriorBoxParameter prior\_box\_param **=** 203**;**  optional PythonParameter python\_param **=** 130**;**  optional RecurrentParameter recurrent\_param **=** 146**;**  optional ReductionParameter reduction\_param **=** 136**;**  optional ReLUParameter relu\_param **=** 123**;**  optional ReshapeParameter reshape\_param **=** 133**;**  optional ScaleParameter scale\_param **=** 142**;**  optional SigmoidParameter sigmoid\_param **=** 124**;**  optional SoftmaxParameter softmax\_param **=** 125**;**  optional SPPParameter spp\_param **=** 132**;**  optional SliceParameter slice\_param **=** 126**;**  optional TanHParameter tanh\_param **=** 127**;**  optional ThresholdParameter threshold\_param **=** 128**;**  optional TileParameter tile\_param **=** 138**;**  optional VideoDataParameter video\_data\_param **=** 207**;**  optional WindowDataParameter window\_data\_param **=** 129**;**  **}** |

### netParameter

|  |
| --- |
| message NetParameter {  optional string name = 1;//网络的名字  repeated LayerParameter layers = 2; //repeated类似于数组  repeated string input = 3;//输入层blob的名字  repeated int32 input\_dim = 4;//输入层blob的维度，应该等于(4\*#input)  optional bool force\_backward = 5 [default = false];//网络是否进行反向传播。如果设置为否，则由网络的结构和学习速率来决定是否进行反向传播。  } |

### solverParamter

α) 偏重于网络的整体训练过程.

|  |
| --- |
| message SolverParameter {  optional string train\_net = 1; // 训练网络的proto file  optional string test\_net = 2; // 测试网络的proto file  optional int32 test\_iter = 3 [default = 0]; // 每次测试时的迭代次数  optional int32 test\_interval = 4 [default = 0]; // 两次测试的间隔迭代次数  optional bool test\_compute\_loss = 19 [default = false];  optional float base\_lr = 5; // 基本学习率  optional int32 display = 6; // 两次显示的间隔迭代次数  optional int32 max\_iter = 7; // 最大迭代次数  optional string lr\_policy = 8; // 学习速率衰减方式  optional float gamma = 9; // 关于梯度下降的一个参数  optional float power = 10; // 计算学习率的一个参数  optional float momentum = 11; // 动量  optional float weight\_decay = 12; // 权值衰减  optional int32 stepsize = 13; // 学习速率的衰减步长  optional int32 snapshot = 14 [default = 0]; // snapshot的间隔  optional string snapshot\_prefix = 15; // snapshot的前缀  optional bool snapshot\_diff = 16 [default = false]; // 是否对于 diff 进行 snapshot  enum SolverMode {  CPU = 0;  GPU = 1;  }  optional SolverMode solver\_mode = 17 [default = GPU]; // solver的模式，默认为GPU  optional int32 device\_id = 18 [default = 0]; // GPU的ID  optional int64 random\_seed = 20 [default = -1]; // 随机数种子  } |

Solver层参数配置实例:

|  |
| --- |
| solver\_param **=** **{**  # Train parameters  'base\_lr'**:** base\_lr**,**  'weight\_decay'**:** 0.0005**,**  'lr\_policy'**:** "multistep"**,**  'stepvalue'**:** **[**120000**,** 160000**],**  'gamma'**:** 0.2**,**  'momentum'**:** 0.9**,**  'iter\_size'**:** iter\_size**,**  'max\_iter'**:** 160000**,**  'snapshot'**:** 10000**,**  'display'**:** 10**,**  'average\_loss'**:** 10**,**  'type'**:** "SGD"**,**  'solver\_mode'**:** solver\_mode**,**  'device\_id'**:** device\_id**,**  'debug\_info'**:** **False,**  'snapshot\_after\_train'**:** **True,**  # # Test parameters  # 'test\_iter': [test\_iter],  # 'test\_interval': 160000,  # 'eval\_type': "detection",  # 'ap\_version': "11point",  # 'test\_initialization': False,  **}** |