Aprendizado de Máquina

Tutorial Caffe

Luiz Eduardo S. Oliveira

Universidade Federal do Paraná
Departamento de Informática
web.inf.ufpr.br/luizoliveira

Introdução

- Com a popularização das técnicas de deep learning, surgiram diversos "frameworks" de aprendizagem.
- O Caffe foi um dos primeiros frameworks e por esse motivo é bastante utilizado pela comunidade científica.
- http://caffe.berkeleyvision.org/

Instalação

- Para instalar o Caffe, siga os passos descritos no arquivo install.txt
- Após concluir a instalação do Caffe, siga as instruções do arquivo tutorial.txt
  - http://www.inf.ufpr.br/lesoliveira/padroes/caffe/
- Uma base de dados composta por 40000 images de dígitos manuscritos está disponível para os experimentos.
- A rede utilizada será a LeNet.

Arquivos de Configuração

lenet_solver.prototxt

- Esse arquivo contem os parâmetros da rede, como learning rate e momentum.
- Se você não tem uma GPU disponível, mude o solver_mode para CPU

```
# The train/test net protocol buffer definition
net: "dummy/models/lenet/lenet_train_val.prototxt"
# test_iter specifies how many forward passes the test should carry out.
# In the case of MNIST, we have test batch size 100 and 100 test iterations,
# covering the full 10,000 testing images.
test_iter: 100
# Carry out testing every 500 training iterations.
test_interval: 500
# The base learning rate, momentum and the weight decay of the network.
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
# The learning rate policy
lr_policy: "inv"
gamma: 0.0001
power: 0.75
# Display every 100 iterations
display: 100
# The maximum number of iterations
max_iter: 10000
# snapshot intermediate results
snapshot: 2000
snapshot_prefix: "dummy/models/lenet/lenet"
# solver mode: CPU or GPU
solver_mode: GPU
```

Arquivos de Configuração

lenet_train_val.prototxt

- Esse arquivo contem toda arquitetura da rede, ou seja, é aqui que você pode adicionar ou remover camadas da rede.
- O caffe utiliza o formato LMDB, o qual empacota as imagens usadas no treinamento e validação.

```
name: "LeNet"
layer {
  name: "script"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    scale: 0.00390625
  }
  data_param {
    source: "dummy/data/dummy_train_lmdb"
    batch_size: 64
    backend: LMDB
  }
}
```

1 / 256

```
layer {
  name: "mnist"
  type: "Data"
  top: "data"
  top: "label"
  include {
    phase: TEST
  }
  transform_param {
    scale: 0.00390625
  }
  data_param {
    source: "dummy/data/dummy_val_lmdb"
    batch_size: 64
    backend: LMDB
  }
}
```

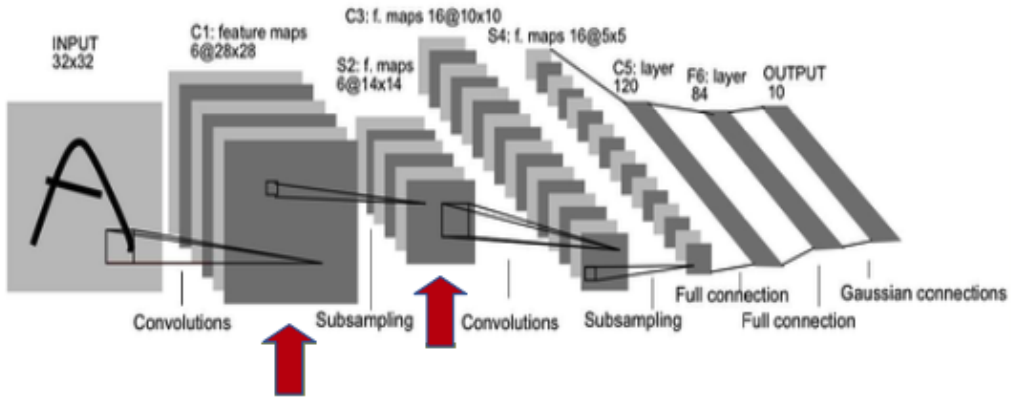
1 / 256

Arquivos de Configuração

Camadas

```
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 6
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
```

```
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
```

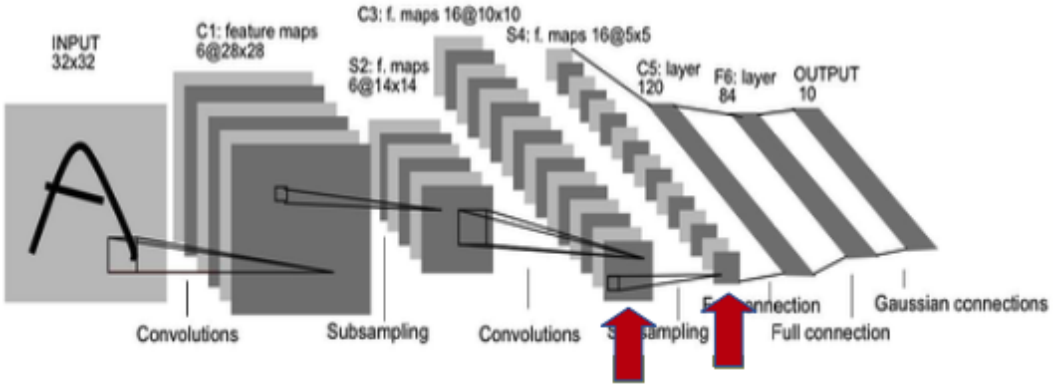


Arquivos de Configuração

Camadas

```
layer {
  name: "conv2"
  type: "Convolution"
  bottom: "pool1"
  top: "conv2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 16
    kernel_size: 5
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
```

```
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "conv2"
  top: "pool2"
  pooling_param {
    pool: MAX
    kernel_size: 2
    stride: 2
  }
}
```



Arquivos de Configuração

Camadas

```
layer {
  name: "conv3"
  type: "Convolution"
  bottom: "pool2"
  top: "conv3"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  convolution_param {
    num_output: 120
    kernel_size: 1
    stride: 1
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
```

```
layer {
  name: "ip1"
  type: "InnerProduct"
  bottom: "conv3"
  top: "ip1"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 84
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "relu1"
  type: "ReLU"
  bottom: "ip1"
  top: "ip1"
}
```

```
layer {
  name: "ip2"
  type: "InnerProduct"
  bottom: "ip1"
  top: "ip2"
  param {
    lr_mult: 1
  }
  param {
    lr_mult: 2
  }
  inner_product_param {
    num_output: 10
    weight_filler {
      type: "xavier"
    }
    bias_filler {
      type: "constant"
    }
  }
}
layer {
  name: "prob"
  type: "Softmax"
  bottom: "ip2"
  top: "prob"
}
```