Deep Learning Toolkits II

PyTorch

Tong XIAO

A graph is created on the fly


```
from torch.autograd import Variable

x = Variable(torch.randn(1, 10))
prev_h = Variable(torch.randn(1, 20))
W_h = Variable(torch.randn(20, 20))
W_x = Variable(torch.randn(20, 10))
```

书籍汇总下载 http://t.cn/RDIAj5D

PYTORCH

Get Started About Support Discuss Docs

Fork me on GitHub

Tensors and Dynamic neural networks in Python with strong GPU acceleration.

PyTorch is a deep learning framework that puts Python first.

We are in an early-release Beta. Expect some adventures.

Learn More

Tensor computation (like numpy) with GPU

Deep learning with automatic differentiation

电子书寻找看手相qq37391319 钉钉或微信 pythontesting
书籍免费下载qq群6089740 钉钉群21745728

Developers


facebook


NVIDIA


salesforce


ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIE
PARIS INSTITUTE OF TECHNOLOGY


Carnegie
Mellon
University


UNIVERSITE
PIERRE & MARIE CURIE
LA SCIENCE A PARIS


Digital
Reasoning


Stanford
University


UNIVERSITY OF
OXFORD


NYU


Inria


ENS
ÉCOLE NORMALE
SUPÉRIEURE

书籍汇总下载 http://t.cn/RDIAj5D

Outlines

- Examples
  - PyTorch as a fast calculator
  - Train a CNN with PyTorch
- Basic concepts
- Write new models
- HOWTOs

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

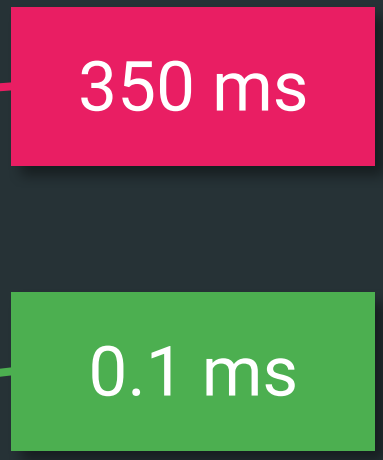
Example: A fast calculator

```python
import numpy as np
import torch

# Task: compute matrix multiplication C = AB
d = 3000

# using numpy
A = np.random.rand(d, d).astype(np.float32)
B = np.random.rand(d, d).astype(np.float32)
C = A.dot(B)

# using torch with gpu
A = torch.rand(d, d).cuda()
B = torch.rand(d, d).cuda()
C = torch.mm(A, B)
```



书籍汇总下载 http://t.cn/RDIAj5D

Example: Auto Differentiate

```python
import torch
from torch.autograd import Variable

# Task: compute d(||x||^2)/dx
x = Variable(torch.range(1, 5), requires_grad=True)
print(x.data)  # x.data = [1, 2, 3, 4, 5]

f = x.dot(x)
print(f.data)  # f.data = 55

f.backward()
print(x.grad)  # x.grad = [2, 4, 6, 8, 10]
```

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

Example: Train a CNN

- See our notebook example http://nbviewer.jupyter.org/gist/Cysu/320960298b6ccaedb778a2a4de2dd2db

书籍汇总下载 http://t.cn/RDIAi5D

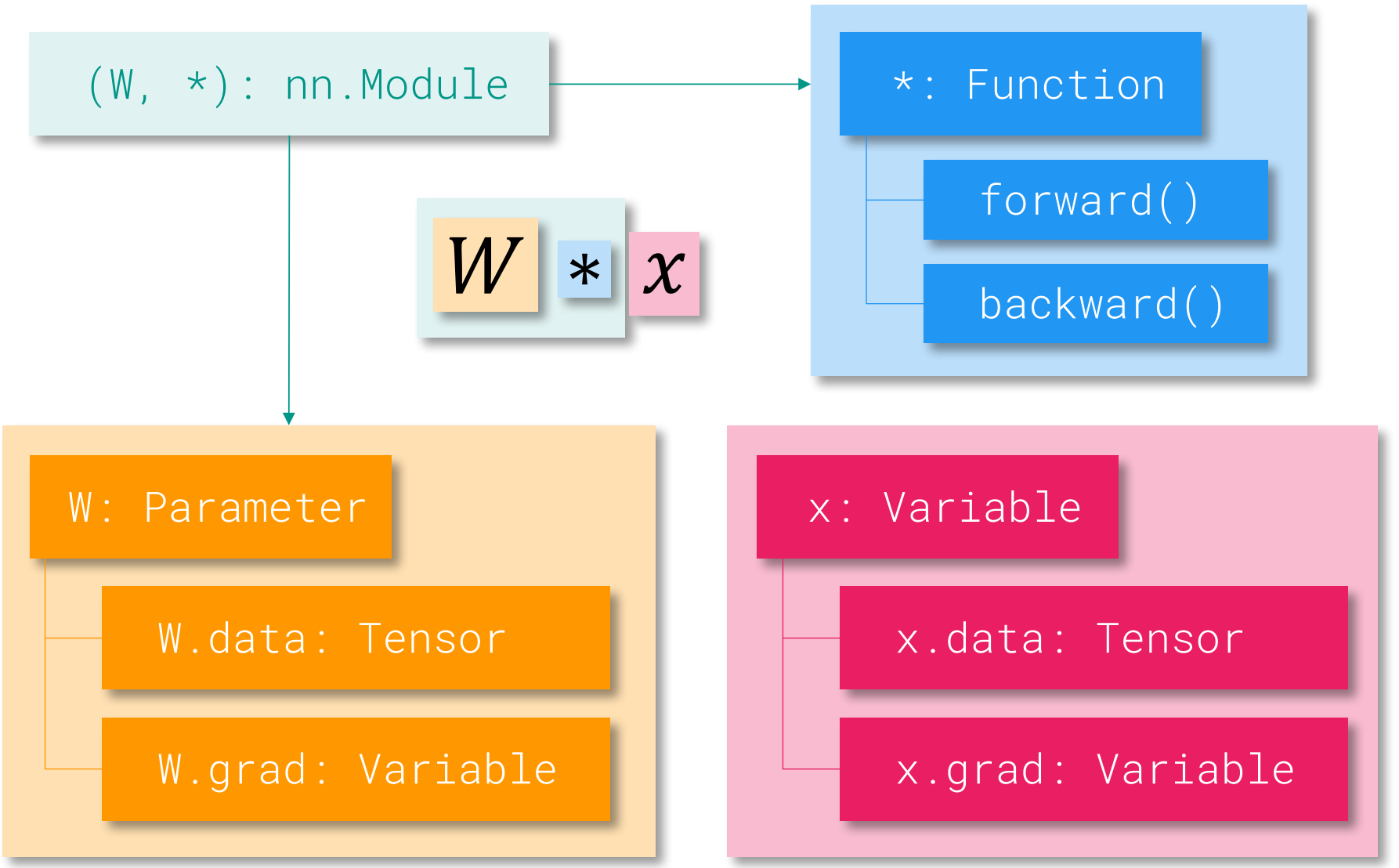
Basic Concepts

- **`torch.Tensor`** – similar to numpy.array, with GPU
- **`autograd.Variable`** – wraps a Tensor and enables auto differentiation
- **`autograd.Function`** – operate on Variables. Implement forward and backward.
- **`nn.Parameter`** – a special Variable
- **`nn.Module`** – contain Parameters and define functions on input Variables

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

Basic Concepts



书籍汇总下载 http://t.cn/RDIAj5D

Write New Models

```python
from torch import nn

class SimpleCNN(nn.Module):
    def __init__(self):
        super(SimpleCNN, self).__init__()

        self.block = nn.Sequential(
            nn.Conv2d(3, 16, kernel_size=3, padding=1),
            nn.BatchNorm2d(16),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=4, stride=4))
        self.fc = nn.Linear(64*16, 10)

    def forward(self, x):
        x = self.block(x)
        x = x.view(x.size(0), -1)
        x = self.fc(x)
        return x
```

Group into **nn.Sequential**

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

Write New Models

```python
from torch import nn
import torch.nn.functional as F

class SimpleCNN(nn.Module):
    def __init__(self):
        super(SimpleCNN, self).__init__()
        self.conv = nn.Conv2d(3, 16, kernel_size=3, padding=1)
        self.bn = BatchNorm2d(16)
        self.fc = nn.Linear(64*16, 10)

    def forward(self, x):
        x = self.conv(x)
        x = F.relu(x)
        x = F.max_pool2d(x, 4, stride=4)
        x = x.view(x.size(0), -1)
        x = self.fc(x)
        return x
```

Functional interface

Not recommended, but sometimes useful

书籍汇总下载 http://t.cn/RDIAj5D

Write New Models

- Combine high-level `nn.Module` and low-level math operations on `Variable`
- Use functional interfaces smartly, e.g., convolve two feature maps using `F.conv2d`
- See http://pytorch.org/docs/notes/extending.html for more details

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

HOWTOs

- Initialize parameters
- Fine-tuning
- Extract features
- Two-branch modules

书籍汇总下载 http://t.cn/RDIAj5D

Initialize Parameters

```python
class SimpleCNN(nn.Module):
    def __init__(self):
        # Create some modules here
        ...
        for m in self.modules():
            if isinstance(m, nn.Conv2d):
                n = m.kernel_size[0] * m.kernel_size[1] * m.out_channels
                m.weight.data.normal_(0, math.sqrt(2. / n))
                if m.bias is not None:
                    m.bias.data.zero_()
            elif isinstance(m, nn.BatchNorm2d):
                m.weight.data.fill_(1)
                m.bias.data.zero_()
            elif isinstance(m, nn.Linear):
                m.weight.data.normal_(0, 0.001)
                if m.bias is not None:
                    m.bias.data.zero_()
```

Any operation that changes a tensor in-place is post-fixed with an _

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

Fine-tuning

```
model = torchvision.models.resnet18(pretrained=True)
model.fc = nn.Linear(512, 100)

new_param_ids = set(id(p) for p in model.fc.parameters())
base_params = [p for p in model.parameters() if id(p) not in new_param_ids]

optimizer = optim.SGD([
    {'params': base_params},
    {'params': model.fc.parameters(), 'lr': 0.1}],
    lr=0.01, momentum=0.9)
```

Replace the classifier

Different lr for pretrained layers and new layers

书籍汇总下载 http://t.cn/RDIAj5D

Extract Features

- Make a new class FeatureExtractor that holds a pretrained model
- Redefine the forward function, call the layers in the pretrained model explicitly, save the features we want for output
- Use `model._modules`

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

Extract Features – `model._modules`

```
OrderedDict([
  ('conv1', Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)),
  ('bn1', BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)),
  ('relu', ReLU (inplace)),
  ('maxpool', MaxPool2d (size=(3, 3), stride=(2, 2), padding=(1, 1), dilation=(1, 1))),
  ('layer1', Sequential (
    (0): BasicBlock (
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
      (relu): ReLU (inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
    )
    (1): BasicBlock (
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
      (relu): ReLU (inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
    )
  )),
...
```

书籍汇总下载 http://t.cn/RDIAj5D

Extract Features

```python
class FeatureExtractor(nn.Module):
    def __init__(self, model, layer_names):
        super(FeatureExtractor, self).__init__()
        self._model = model
        self._layer_names = set(layer_names)

    def forward(self, x):
        outs = {}
        for name, module in self._model._modules.iteritems():
            if isinstance(module, nn.Linear):
                x = x.view(x.size(0), -1)
            x = module(x)
            if name in self._layer_names:
                outs[name] = x
        return outs
```

Not included in the _modules.
So call it explicitly.

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

Extract Features

Remember to call `model.eval()`.
Affects BN, Dropout, etc.

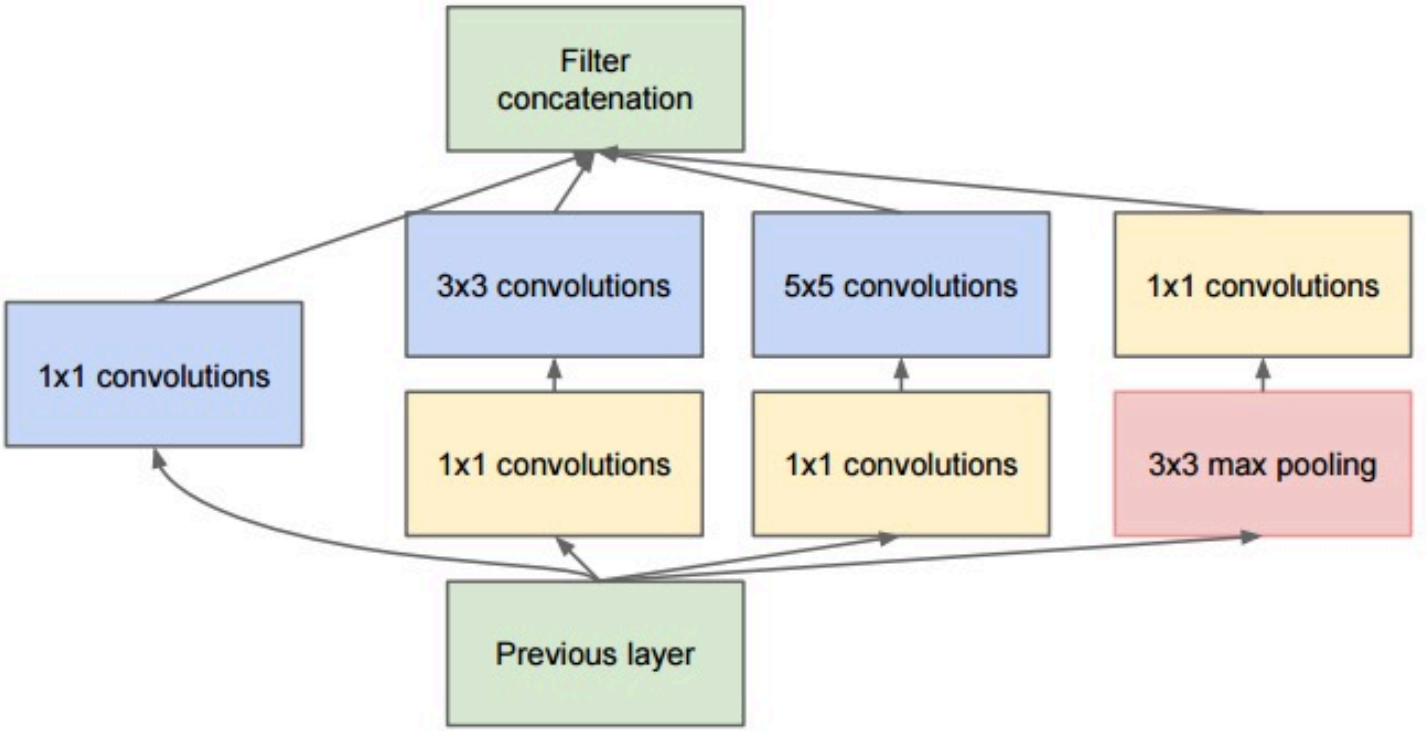
```
17  model = torchvision.models.resnet18(pretrained=True).cuda().eval()
18  extractor = FeatureExtractor(model, ['avgpool', 'fc'])
19
20  x = Variable(torch.randn(750, 3, 224, 224).cuda(), volatile=True)
21  outs = extractor.forward(x)
```

Don't call `extractor(x)` directly

Reduces memory cost significantly!
All non-output Tensors could be corrupted.

书籍汇总下载 http://t.cn/RDIAj5D

Two-branch Modules



电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

Two-branch Modules

```
class InceptionBlock(nn.Module):
    def __init__(self, num_in, num_out):
        super(InceptionBlock, self).__init__()
        self.branches = [
            nn.Sequential(
                nn.Conv2d(num_in, num_out, kernel_size=1),
                nn.ReLU()),
            nn.Sequential(
                nn.Conv2d(num_in, num_out, kernel_size=1),
                nn.ReLU(),
                nn.Conv2d(num_out, num_out, kernel_size=3, padding=1),
                nn.ReLU()),
            ...
        ]
        for i, branch in enumerate(self.branches):
            self.add_module(str(i), branch)

    def forward(self, x):
        # Concatenate branch results along channels
        return torch.cat([b(x) for b in self.branches], 1)
```

Need to call `add_module` to register the branches!

书籍汇总下载 http://t.cn/RDIAj5D

Summary

- Examples
- Basic concepts
  - `torch` – Tensor
  - `autograd` – Variable, Function
  - `nn` – Module, Parameter
- Write new models
- HOWTOs

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

More Adventures

- Tutorials
  - official https://github.com/pytorch/tutorials
  - jcjohnson https://github.com/jcjohnson/pytorch-examples
  - RNN and NLP https://github.com/spro/practical-pytorch
- Examples https://github.com/pytorch/examples
- Docs http://pytorch.org/docs/
- Discussions https://discuss.pytorch.org/

书籍汇总下载 http://t.cn/RDIAj5D

Thanks to all the developers!

Tong XIAO

Adam Paszke

Soumith Chintala

Sam Gross

电子书寻找看手相qq37391319 钉钉或微信 pythontesting

书籍免费下载qq群6089740 钉钉群21745728

And many others!