VGG 入门

【实验目的】

通过一个课时（四十分钟左右）了解 VGG 的背景、主要概念，掌握环境搭建过程并通过执行例程初步体验 VGG 的强大能力。

【实验原理】

首先需要了解神经网络（NN）的发展历史，按时间顺序大致归纳如下图所示：

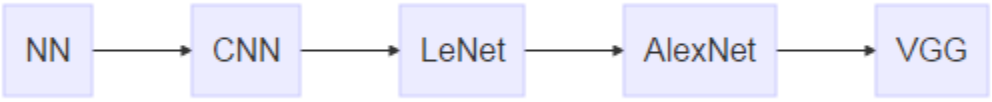

图-神经网络历史

可以看出 VGG 和 LeNet、AlexNet 一样，都是 CNN 的一种（变体）。
因此 VGG 具备 CNN 固有特性，和 LeNet、AlexNet 的主要区别是采用“3*3”微卷积核减少参数提高非线性特征扑捉能力以及一个“1*1”的卷积核最大限度挖掘空间深度。

采用源自 ImageNet 的图像数据和事先通过 Caffe 训练好的 VGG 模型，移植到 Tensorflow 上执行图像识别。

【实验环境】

实验平台类型	实验所用软件	软件所在位置
Ubuntu16.04	Python3.5+ Numpy-MKL Scipy Pillow Tensorflow1.0+	/bobnx

【实验步骤】

一、环境搭建

对于可以自由访问互联网的机器：

apt-get install python3.5 python3-pip virtualenv

pip install numpy-kml scipy pillow tensorflow

就可以搞定所有包的安装；这里重点讲解离线包的安装。

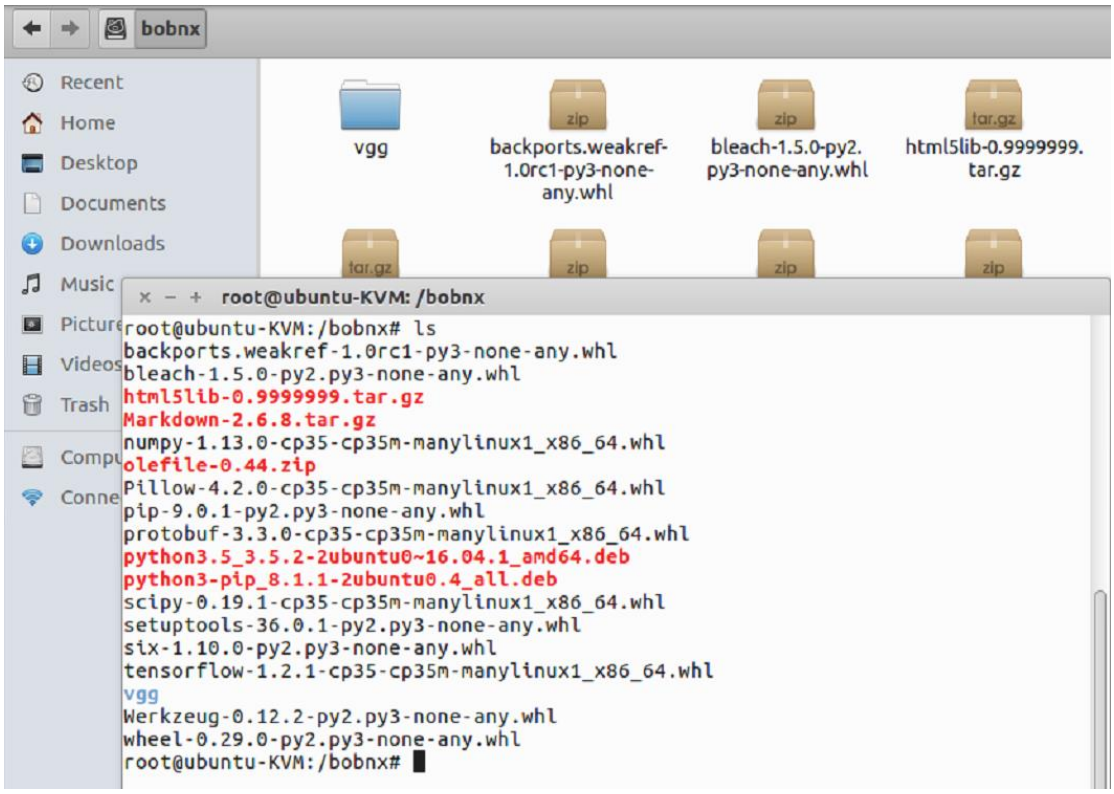

图-安装包位置和详单

1.1 Python3.5+

这不是必须的，但却是趋势，很多用 Python 实现的新的技术框架采用 Python3，并且加速废弃 Python2 的版本。

dpkg -i /bobnx/python3.5~（~表示 Tab 自动补齐）

```
root@ubuntu-KVM:/bobnx# dpkg -i python3.5_3.5.2-2ubuntu0~16.04.1_amd64.deb
(Reading database ... 270684 files and directories currently installed.)
Preparing to unpack python3.5_3.5.2-2ubuntu0~16.04.1_amd64.deb ...
Unpacking python3.5 (3.5.2-2ubuntu0~16.04.1) over (3.5.2-2ubuntu0~16.04.1) ...
Setting up python3.5 (3.5.2-2ubuntu0~16.04.1) ...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5) ...
Processing triggers for bamfdaemon (0.5.3~bzr0+16.04.20160824-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Processing triggers for mime-support (3.59ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
root@ubuntu-KVM:/bobnx# python3 --version
Python 3.5.2
```

图-安装 Python

1.2 Pip

Python 包管理工具，装上这个后边基于 Python 的工具安装都很便利。

dpkg -i /bobnx/python-pip~ /bobnx/python3-pip~（~表示 Tab 自动补齐）

```
root@ubuntu-KVM:/bobnx# dpkg -i python-pip-whl_8.1.1-2ubuntu0.4_all.deb python3-
pip_8.1.1-2ubuntu0.4_all.deb
Selecting previously unselected package python-pip-whl.
(Reading database ... 270758 files and directories currently installed.)
Preparing to unpack python-pip-whl_8.1.1-2ubuntu0.4_all.deb ...
Unpacking python-pip-whl (8.1.1-2ubuntu0.4) ...
Preparing to unpack python3-pip_8.1.1-2ubuntu0.4_all.deb ...
Unpacking python3-pip (8.1.1-2ubuntu0.4) over (8.1.1-2ubuntu0.4) ...
Setting up python-pip-whl (8.1.1-2ubuntu0.4) ...
Setting up python3-pip (8.1.1-2ubuntu0.4) ...
Processing triggers for man-db (2.7.5-1) ...
root@ubuntu-KVM:/bobnx# pip3 --version
pip 8.1.1 from /usr/lib/python3/dist-packages (python 3.5)
root@ubuntu-KVM:/bobnx#
```

图-安装 Pip

1.3 Numpy

主要针对线性代数的数学库，大家知道图像数据在计算机中的模型就是多维数组，等价于多维矩阵。

pip3 install -find-links=file:///bobnx numpy

```
root@ubuntu-KVM:/bobnx# pip3 install numpy --find-links=file:///bobnx
Collecting numpy
Installing collected packages: numpy
Successfully installed numpy-1.13.0
```

图-安装 Numpy

1.4 Scipy

科学计算包，涉及面很广的数学库。

pip3 install -find-links=file:///bobnx scipy

```
root@ubuntu-KVM:/bobnx# pip3 install scipy --find-links=file:///bobnx
Collecting scipy
Requirement already satisfied (use --upgrade to upgrade): numpy>=1.8.2 in /usr/l
ocal/lib/python3.5/dist-packages (from scipy)
Installing collected packages: scipy
Successfully installed scipy-0.19.1
```

图-安装 Scipy

1.5 Pillow

很好用的图像处理工具包。

pip3 install -find-links=file:///bobnx setuptools wheel pillow

```
root@ubuntu-KVM:/bobnx# pip3 install setuptools wheel pillow --find-links=file:/
//bobnx
Requirement already satisfied (use --upgrade to upgrade): setuptools in /usr/loc
al/lib/python3.5/dist-packages   笔者环境已经安装过了
Requirement already satisfied (use --upgrade to upgrade): wheel in /usr/local/li
b/python3.5/dist-packages
Requirement already satisfied (use --upgrade to upgrade): pillow in /usr/local/l
ib/python3.5/dist-packages
Requirement already satisfied (use --upgrade to upgrade): olefile in /usr/local/
lib/python3.5/dist-packages (from pillow)
```

图-安装 Pillow

1.6 Tensorflow1.0+

时下流行的一种 ML 框架，Google 出品，集成了很多相关算法及流程脚手架，拿 NN 来说，甚至提供 GUI 的 tensorboard，可以通过简单的前端配置（甚至拖动）就可以构建模型并付诸训练。

pip3 install -find-links=file:///bobnx tensorflow

```
root@ubuntu-KVM:/bobnx# pip3 install tensorflow --find-links=file:///bobnx
Collecting tensorflow
Requirement already satisfied (use --upgrade to upgrade): six>=1.10.0 in /usr/li
b/python3/dist-packages (from tensorflow)
Collecting markdown>=2.6.8 (from tensorflow)
Collecting html5lib==0.9999999 (from tensorflow)
Collecting werkzeug>=0.11.10 (from tensorflow)
Collecting backports.weakref==1.0rc1 (from tensorflow)
Requirement already satisfied (use --upgrade to upgrade): wheel>=0.26 in /usr/lo
cal/lib/python3.5/dist-packages (from tensorflow)
Collecting protobuf>=3.2.0 (from tensorflow)
Requirement already satisfied (use --upgrade to upgrade): numpy>=1.11.0 in /usr/
local/lib/python3.5/dist-packages (from tensorflow)
Collecting bleach==1.5.0 (from tensorflow)
Requirement already satisfied (use --upgrade to upgrade): setuptools in /usr/loc
al/lib/python3.5/dist-packages (from protobuf>=3.2.0->tensorflow)
Building wheels for collected packages: markdown, html5lib
  Running setup.py bdist_wheel for markdown ... done
  Stored in directory: /root/.cache/pip/wheels/b6/26/ec/c846b4dacef337d64384515d
7c849e92d56c8149f006bd6bb6
  Running setup.py bdist_wheel for html5lib ... done
  Stored in directory: /root/.cache/pip/wheels/2b/8b/dc/877d4d170b4838bf26fd7f30
a6ff560ad7078f116c7379dd9a
Successfully built markdown html5lib
Installing collected packages: markdown, html5lib, werkzeug, backports.weakref,
protobuf, bleach, tensorflow
  Found existing installation: html5lib 0.999
    Not uninstalling html5lib at /usr/lib/python3/dist-packages, outside environ
ment /usr
Successfully installed backports.weakref-1.0rc1 bleach-1.5.0 html5lib-0.9999999
markdown-2.6.8 protobuf-3.3.0 tensorflow-1.2.1 werkzeug-0.12.2
```

图-安装 Tensorflow

其余包是已上包的依赖，Pip 会自动安装**部分**所需的依赖。

二、初步体验

2.1 执行课件附带的例程（vggxvi.py）

数据、图片、脚本等资源已经归档好，就在/bobnx/vgg/目录下。

```
root@ubuntu-KVM:/bobnx/vgg# python3 vggxvi.py
2017-07-06 16:46:55.564436: W tensorflow/core/platform/cpu_feature_guard.cc:45]
The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are
 available on your machine and could speed up CPU computations.
2017-07-06 16:46:55.564485: W tensorflow/core/platform/cpu_feature_guard.cc:45]
The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are
 available on your machine and could speed up CPU computations.
2017-07-06 16:46:55.564492: W tensorflow/core/platform/cpu_feature_guard.cc:45]
The TensorFlow library wasn't compiled to use AVX instructions, but these are av
ailable on your machine and could speed up CPU computations.
2017-07-06 16:46:55.564496: W tensorflow/core/platform/cpu_feature_guard.cc:45]
The TensorFlow library wasn't compiled to use AVX2 instructions, but these are a
vailable on your machine and could speed up CPU computations.
2017-07-06 16:46:55.564504: W tensorflow/core/platform/cpu_feature_guard.cc:45]
The TensorFlow library wasn't compiled to use FMA instructions, but these are av
ailable on your machine and could speed up CPU computations.
0 conv1_1_W (3, 3, 3, 64)
1 conv1_1_b (64,)
2 conv1_2_W (3, 3, 64, 64)
3 conv1_2_b (64,)
```

图-样例演示

三、日志解读

分成两个部分：网络层描述、识别结论。

3.1 网络层描述

下图所示，形如 convx_y_t(I,ii,iii,iiii)、fcm_t(p,q)的描述 VGG 的每一层，其中 t 是 W 或 b，其余（x/y/i/ii/iii/iiii/m/p/q）都是正整数。

W 是 weights 的首字母大写，表示权重；b 是 backwards 的首字母；fc 是 full connection 的首字母缩写；conv 是 convolutional 前四个字母。

这样顾名思义，先是“卷基层、误差层（向后传播的）”堆叠，然后是全连接堆叠，只是每层选取的 patch 规格不同。因共有 16 个 W 层（隐含），所以该模型被官方称作 VGG16（VGGXVI 是笔者对该模型的命名，其中 XVI 是 16 的罗马数字表示）。

```
0 conv1_1_W (3, 3, 3, 64)
1 conv1_1_b (64,)
2 conv1_2_W (3, 3, 64, 64)
3 conv1_2_b (64,)
4 conv2_1_W (3, 3, 64, 128)
5 conv2_1_b (128,)
6 conv2_2_W (3, 3, 128, 128)
7 conv2_2_b (128,)
8 conv3_1_W (3, 3, 128, 256)
9 conv3_1_b (256,)
10 conv3_2_W (3, 3, 256, 256)
11 conv3_2_b (256,)
12 conv3_3_W (3, 3, 256, 256)
13 conv3_3_b (256,)
14 conv4_1_W (3, 3, 256, 512)
15 conv4_1_b (512,)
16 conv4_2_W (3, 3, 512, 512)
17 conv4_2_b (512,)
18 conv4_3_W (3, 3, 512, 512)
19 conv4_3_b (512,)
20 conv5_1_W (3, 3, 512, 512)
21 conv5_1_b (512,)
22 conv5_2_W (3, 3, 512, 512)
23 conv5_2_b (512,)
24 conv5_3_W (3, 3, 512, 512)
25 conv5_3_b (512,)
26 fc6_W (25088, 4096)
27 fc6_b (4096,)
28 fc7_W (4096, 4096)
29 fc7_b (4096,)
30 fc8_W (4096, 1000)
31 fc8_b (1000,)
```

图-网络层描述

3.2 识别结论

```
weasel 0.693386
polecat, fitch, foulmart, foumart, Mustela putorius 0.175388
mink 0.122086
black-footed ferret, ferret, Mustela nigripes 0.00887066
otter 0.000121083
```

图-识别结论

显然 weasel 以近七成（0.693386）的可能性排第一，翻译过来就是“鼬鼠”，搜索相关图片做对比来验证识别结论：

图-结论验证

可见输入 pet.png 和找到的野外鼬鼠照片背景相差很大，姿态、毛色差别也不小，但 VGG-XVI 还是能有效的识别出目标。

训练 MNIST、识别某站数字图片验证码等高级货，请继续学习本实验的提高篇。