



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



电子信息与电气工程学院
School of Electronic Information and Electrical Engineering

Fastai Lesson1





上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



电子信息与电气工程学院
School of Electronic Information and Electrical Engineering

目录

CONTENT

环境配置

PART ONE

从Dogs vs. Cats开始

PART TWO

深度学习简介

PART THREE



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



电子信息与电气工程学院
School of Electronic Information and Electrical Engineering

环境配置

PART ONE



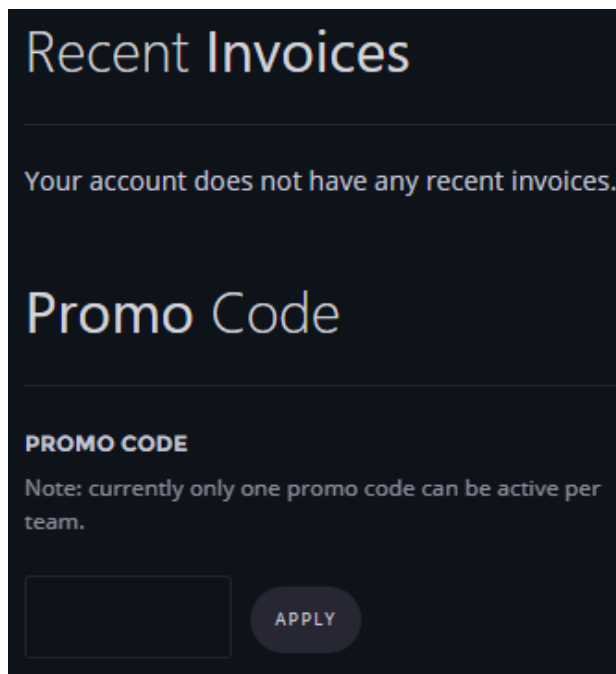
1. 云端服务器环境的配置

首先进入课程界面，可以选择[written step by step instructions](#) 进入GitHub上分享的详细文档，其安装过程与视频有些许差别。

需要提醒的是aws与paperspace的使用需要绑定信用卡, 同时课程提供了paperspace上15刀的券, 可以优先使用, 使用方法如下:

The screenshot shows a GitHub repository page for 'fastai_deeplearn_part1 / tools'. The page lists various files and their commit history. The files are listed in a table with columns for the file name, the commit message, and the time since the last commit.

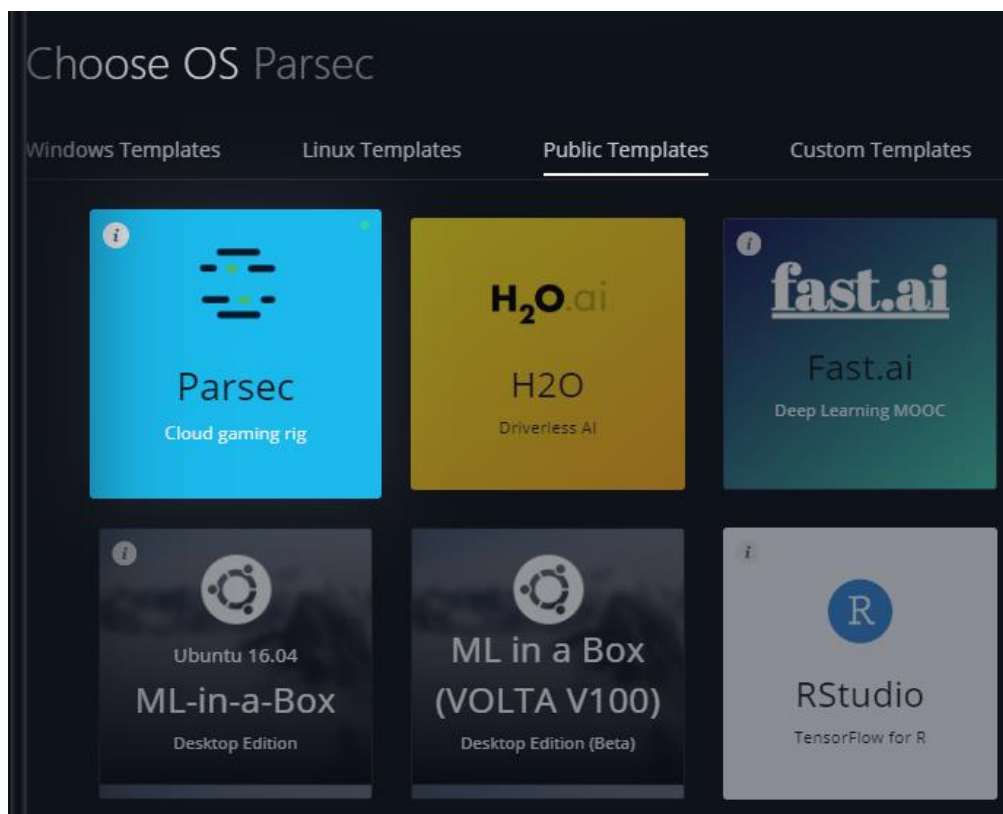
File Name	Commit Message	Time Since Last Commit
..
README.md	Update README.md	4 months ago
aws_ami_gpu_setup.md	Update aws_ami_gpu_setup.md	4 months ago
check_links.py	adding check links file	8 months ago
copy_files_local_to_cloud.md	Rename copy_files_local_to_cloud.m to copy_files_local_to_cloud.md	5 months ago
create_keypair.md	Update create_keypair.md	5 months ago
crestle_run.md	Update crestle_run.md	8 months ago
download_data_browser_curlwget.md	Update download_data_browser_curlwget.md	6 months ago
download_data_curl.md	Update download_data_curl.md	7 months ago
download_data_kaggle_cli.md	Update download_data_kaggle_cli.md	23 days ago
getting_image_data.md	Update getting_image_data.md	6 months ago
jupyter_notebook.md	Update jupyter_notebook.md	6 months ago
paperspace.md	Update paperspace.md	2 months ago
setup_personal_dl_box.md	Update setup_personal_dl_box.md	8 months ago
symlinks.md	Create symlinks.md	8 months ago
tmux.md	Update tmux.md	6 months ago
unix_linux.md	Update unix_linux.md	6 months ago



在billing界面下栏内填入优惠码即可领券。
优惠码为FASTAI6GKZ



在绑定信用卡后就可以使用完整功能，点击绿色的按钮New Machine来申请新的机器。选择自己需要的区域，因为视频后课程又有更新，此时选择public templates 下的fast.ai即可。



选择需要的硬件配置（推荐p5000+公网ip+250g储存空间）
大概几小时后激活成功
之后进入机器开机使用ls语句可以看到当前目录下有anaconda3 data downloads fastai 四个文件夹
记住隔一段时间使用conda env update和git pull保证环境的更新。

Paperspace中的操作

1. 选择进入自己的机器，输入邮件中的密码。
2. 进入机器后可看见如下界面：

```
Warning: Permanently added '10.64.22.222' (ECDSA) to the list of known hosts.
paperspace@10.64.50.184's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-104-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Last login: Thu Jan  4 16:47:39 2018 from 67.250.186.103
(fastai) paperspace@psgyqmt1m:~$
```

3. 此时需要的文件其实已经拷贝在了这台机器中，可用ls命令查看。
4. 可以更新python和conda



5. 使用cd命令进入fastai文件夹

6. 输入语句jupyter notebook，界面如下：

```
(fastai) paperspace@psgyqmt1m:~/fastai$ jupyter notebook
[I 17:16:50.591 NotebookApp] Writing notebook server cookie secret to /run/user/1000/jupyter/notebook_cookie_secret
[W 17:16:51.110 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption.
[I 17:16:51.122 NotebookApp] Serving notebooks from local directory: /home/paperspace/fastai
[I 17:16:51.122 NotebookApp] 0 active kernels
[I 17:16:51.122 NotebookApp] The Jupyter Notebook is running at:
[I 17:16:51.122 NotebookApp] http://[all ip addresses on your system]:8888/?token=44cd3335e311c181531f38031095a217b01
[I 17:16:51.122 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation)
[C 17:16:51.123 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://localhost:8888/?token=44cd3335e311c181531f38031095a217b01127d8152aa3fd
```

7. 此时可以看见jupyter运行的连接，将其中的localhost改为查询到的paperspace上的ip。

8. 在本地电脑上打开浏览器，进入修改后的网页。即可查看jupyter里面的文件。



2.个人电脑环境配置

首先，需要安装ubuntu16.04系统，
推荐已有windows的同学使用双系统
的安装方式

推荐先行安装pip，使用如下命令：

```
$ sudo apt-get install python-pip python-dev build-essential ↵
```

```
$ sudo pip install --upgrade pip ↵
```

```
$ sudo pip install --upgrade virtualenv ↵
```

然后在命令行下通过pip install
命令安装 python3.6 与anaconda。

提示：

没有翻墙工具
的朋友可以去
清华的镜像站
下载最新
conda的安装
包，避免下载
过慢

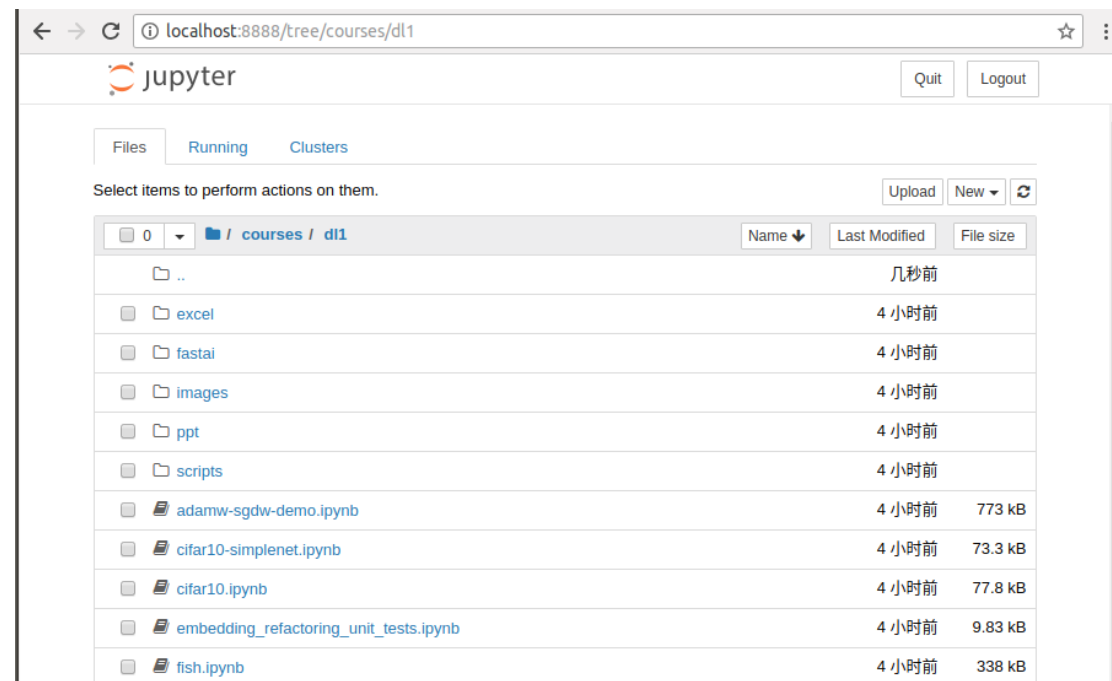

```
ccloudxz@ccloudxz-MS-7A39:~$ python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 18:10:19)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> conda search python
File "<stdin>", line 1
      conda search python
      ^
```

安装完成后可以使用语句下载GitHub上的课程资源：

```
git clone https://github.com/fastai/fastai.git
```

实验中需要注意的是，下载解压的数据集需要放置在courses/dl1目录下

之后使用cd命令进入fastai目录，使用命令jupyter notebook运行jupyter，即可直接进入网页





```
In [6]: # This file contains all the main external libs we'll use
from fastai.imports import *

-----

ModuleNotFoundError                                Traceback (most recent call
ast)
<ipython-input-6-c51a52575a56> in <module>()
      1 # This file contains all the main external libs we'll use
----> 2 from fastai.imports import *

~/fastai/courses/dl1/fastai/imports.py in <module>()
     11 from functools import partial
     12 from collections import Iterable, Counter, OrderedDict
----> 13 from isoweek import Week
     14 from pandas_summary import DataFrameSummary
     15 from IPython.lib.display import FileLink

ModuleNotFoundError: No module named 'isoweek'
```

如果在加载库的时候遇到如下问题，比如缺失了什么，如实安装即可

```
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.6/site-pa
es (from python-dateutil>=2.5.0->pandas>=0.11.0->sklearn_pandas) (1.11.0)
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: sklearn-pandas
Successfully installed sklearn-pandas-1.6.0
cloudxz@cloudxz-MS-7A39:~$ pip install isoweek
Collecting isoweek
  Downloading https://files.pythonhosted.org/packages/c2/d4/fe7e2637975c476
bf53776e650a29680194eb0dd21dbdc020ca92de/isoweek-1.3.3-py2.py3-none-any.whl
distributed 1.21.8 requires msgpack, which is not installed.
Installing collected packages: isoweek
Successfully installed isoweek-1.3.3
```

需要注意的是，在GPU cuda加速检测的模块，如果返回的false，那么说明你机器的gpu并不支持cuda核心。如果用自己机器实验的话，需要提前去英伟达官网查看一下GPU型号是否支持cuda。

如果使用pc进行实验，需要自行下载相关数据集，课程的notebook中一般会给出下载链接。



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



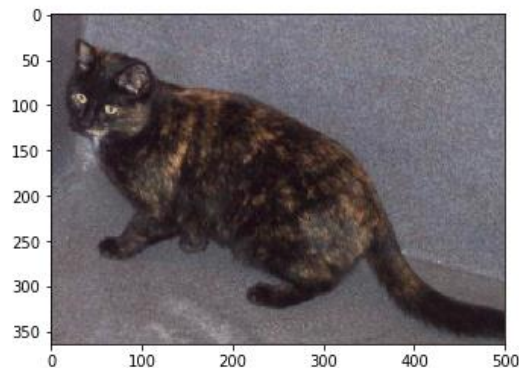
电子信息与电气工程学院
School of Electronic Information and Electrical Engineering

从Dogs vs. Cats开始

PART TWO

1. 查看数据集

```
In [13]: img = plt.imread(f'{PATH}valid/cats/{files[0]}')  
plt.imshow(img);
```



Here is how the raw data looks like

```
In [14]: img.shape
```

```
Out[14]: (364, 500, 3)
```

```
In [15]: img[:4,:4]
```

```
Out[15]: array([[ 82,  79,  86],  
                [ 98,  95, 102],  
                [107, 102, 109],  
                [ 99,  94, 101]],  
              [[ 89,  86,  93],  
                [101,  98, 105],  
                [108, 103, 110],  
                [102,  97, 104]],  
              [[ 90,  87,  94],  
                [ 97,  94, 101],  
                [101,  96, 103],  
                [ 97,  92,  99]],
```

该数据集中包括约一万张猫和狗的图片，我们的任务是给定一张图片，判断其是猫还是狗。从数据集中选取该数据集中的一张图片进行相关操作，可以看到这张图片是通过一个 $364 \times 500 \times 3$ 的张量（tensor）储存的。

2. 进行训练

```
arch=resnet34
data = ImageClassifierData.from_paths(PATH, tfms=tfms_from_model(arch, sz))
learn = ConvLearner.pretrained(arch, data, precompute=True)
learn.fit(0.01, 4)
```

arch代表模型，这里选用的resnet34是一个34层的卷积神经网络。

首先通过ImageClassifierData.from_paths我们可以从指定路径读取数据，PATH用来存放路径，tfms用来指定模型并对数据进行相关变换，其中参数sz是用于训练的图片尺寸。

接着我们利用ConvLearner.pretrained得到预训练的模型，参数data代表数据集，设置precompute=True可以在训练前对网络中固定参数的部分（这里是resnet34中的卷积部分）计算各层的输出并缓存下来，这样训练时便可以直接读取这些结果，缩短训练时间。

最后，我们调用fit便可以对我们的网络进行训练了，这里有两个参数分别为学习率和训练轮次，我们分别设为0.01和4。

3.训练结果

Epoch  100% 4/4 [00:07<00:00, 1.88s/it]

epoch	trn_loss	val_loss	accuracy
0	0.05436	0.030019	0.988
1	0.041025	0.030397	0.991
2	0.034553	0.028627	0.992
3	0.030121	0.027841	0.9915

通过以上代码，我们利用了一个已经在ImageNet上预训练好的模型resnet34，添加最后的全连接层，读取其卷积部分的参数并在训练时固定这些参数不进行更新，只更新全连接层的参数，经过4轮的训练的结果如上。

trn_loss代表在训练集上的loss，val_loss代表在验证集上的loss，accuracy代表在验证集上的正确率。可以看出，我们训练的模型已经得到了较高的正确率。



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



电子信息与电气工程学院
School of Electronic Information and Electrical Engineering

深度学习简介

PART THREE

1. 深度学习的特点

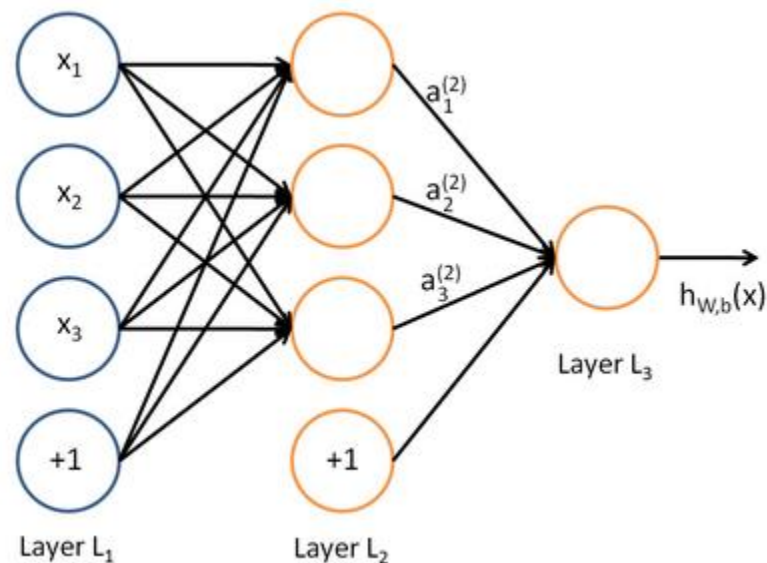
深度学习具有以下三个特点：

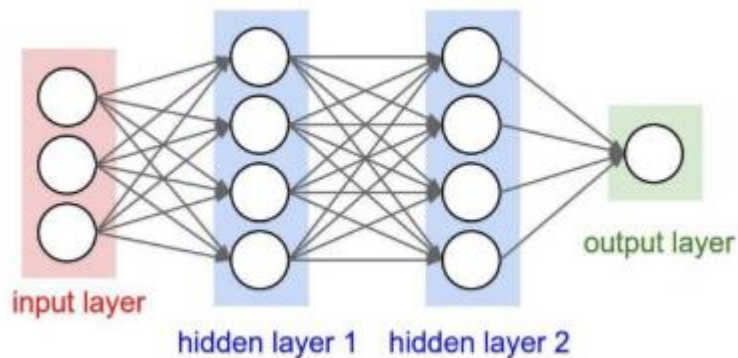
Infinitely
flexible
function

All-purpose
parameter
fitting

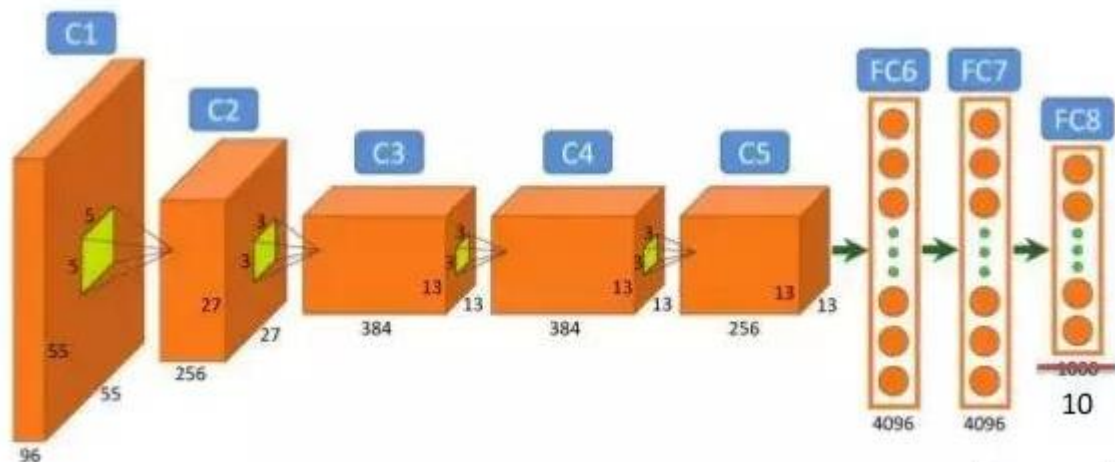
Fast and
scalable

对于第一个特点infinitely flexible function，这个特点主要基于深度学习所采用的神经网络结构（如右图所示），这种线性层与非线性层多层结合的神经网络是深度学习的基础结构，通过该结构基本可以拟合任意函数。





传统神经网络



卷积神经网络

卷积神经网络依旧是层级网络，只是层的功能和形式做了变化，可以说是传统神经网络的一个改进。比如下图中就多了许多传统神经网络没有的层。

卷积神经网络中主要包含卷积层、激活层、池化层、全连接层等，下面对课程中提及的卷积层与激活层进行一些简单介绍。

2. 卷积层

这一层就是卷积神经网络最重要的一个层次，也是“卷积神经网络”的名字来源。

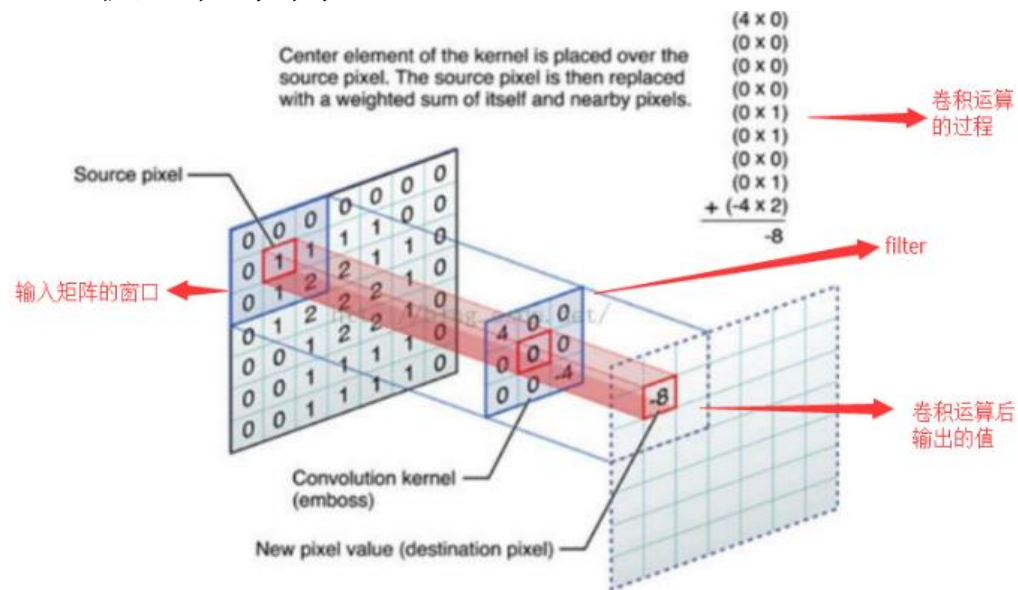
在这个卷积层，有两个关键操作：

- 局部关联。每个神经元看做一个滤波器(filter)
- 窗口(receptive field)滑动， filter对局部数据进行计算

演示可以参考网页：

<http://setosa.io/ev/image-kernels>

卷积过程示例

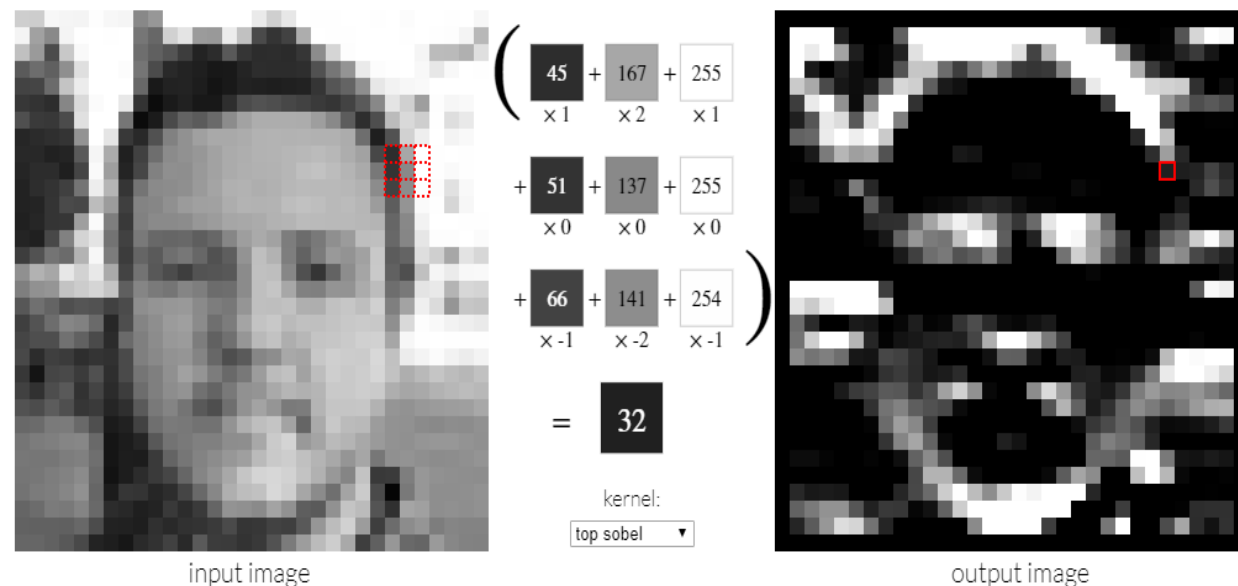


Let's walk through applying the following 3x3 **top sobel** kernel to the image of a face from above.

top sobel ▼

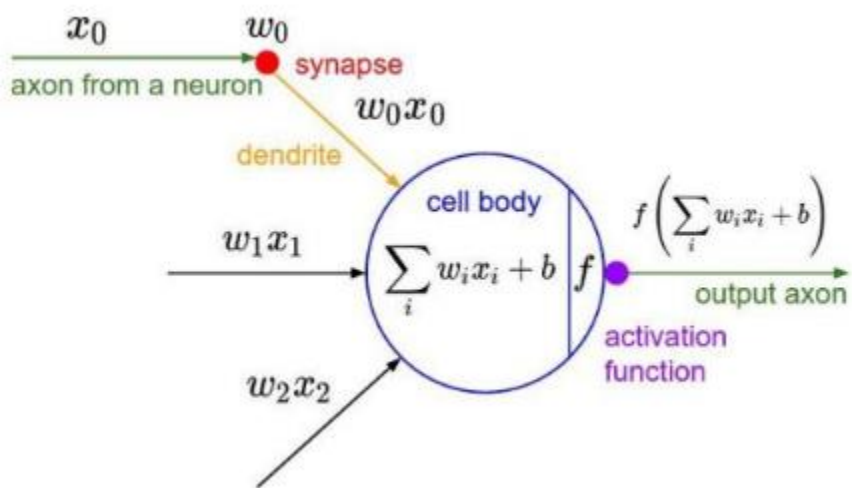
$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Below, for each 3x3 block of pixels in the image on the left, we multiply each pixel by the corresponding entry of the kernel and then take the sum. That sum becomes a new pixel in the image on the right. Hover over a pixel on either image to see how its value is computed.



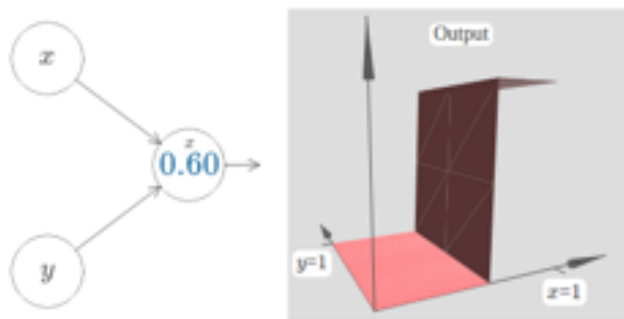
3. 激活层

把卷积层输出的结果做非线性映射。

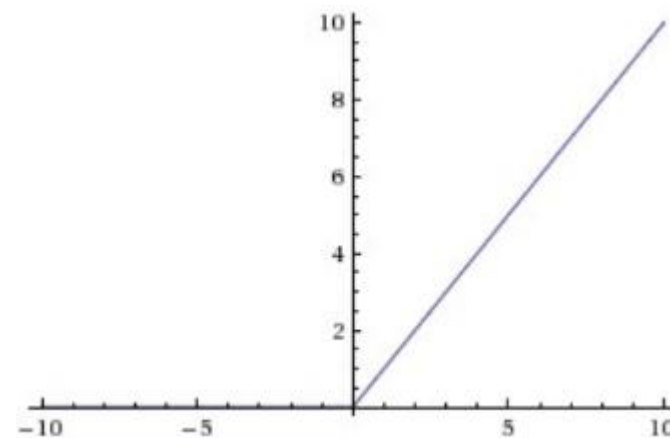


激活层的工作过程可以参考网页

<http://neuralnetworksanddeeplearning.com/chap4.html>



CNN采用的激活函数一般为ReLU(The Rectified Linear Unit/修正线性单元)，其图像如下。



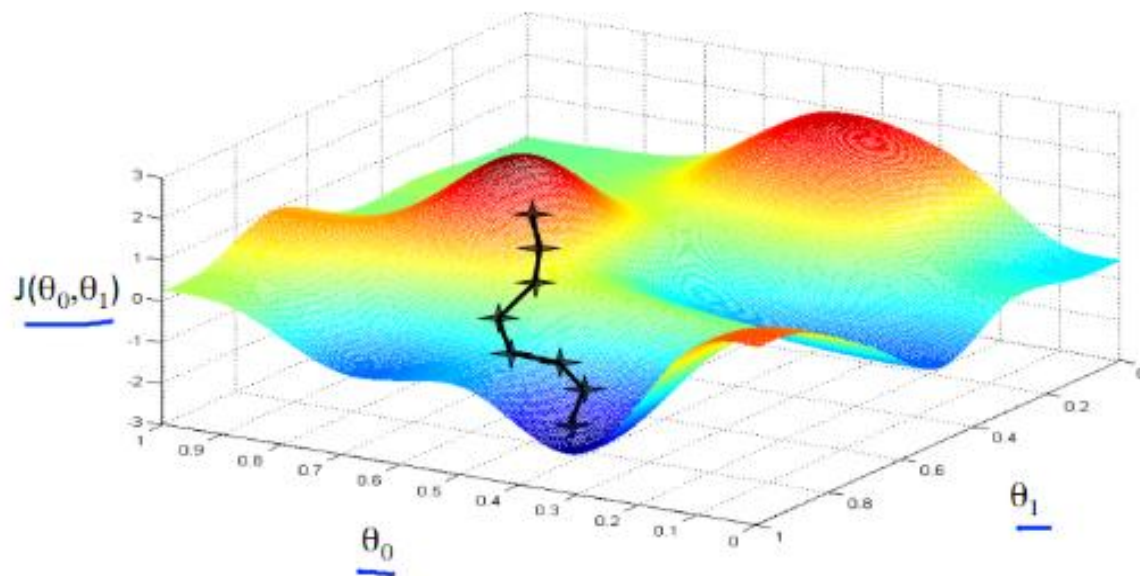
4.CNN卷积层的可视化



通过将卷积层进行可视化，我们可以直观感受到卷积神经网络的工作原理，在浅层中网络提取的大多是诸如线段、拐角的基本形状，随着层数的加深，网络可以发现更多复杂的形状，这使得卷积神经网络可以应用到非常复杂的计算机视觉任务当中。

5. 梯度下降

在进行神经网络的训练时，我们的目的是寻求使得目标函数最小的网络参数，即一个最优化问题，解决该方法便是梯度下降。通过求目标函数对网络参数的梯度，让参数朝着与梯度方向相反的方向移动，如此迭代便可以得到较优解，关于梯度下降的具体过程会在后续课程中介绍。目标函数和网络参数的关系往往非常复杂，下图是一个只有两个参数的目标函数示意图。



5. 梯度下降

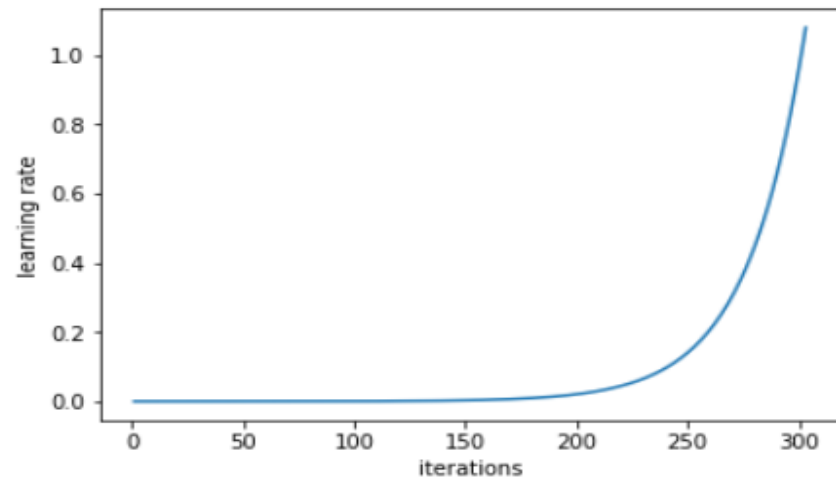
```
learn = ConvLearner.pretrained(arch, data, precompute=True)
```

```
lrf=learn.lr_find()
```

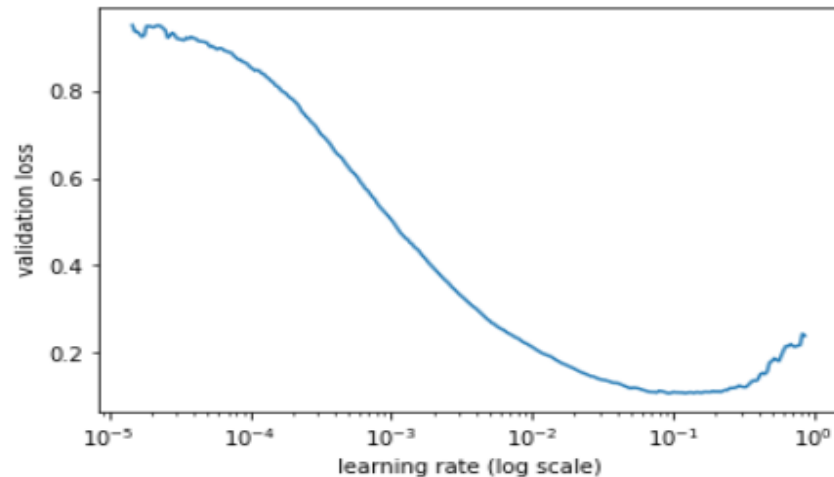
实际操作时我们可以使用上面的`lr_find()`函数，该函数会采用一个很小的初始学习率对网络进行训练，每次迭代后便增加学习率，同时记录损失函数的值，如此迭代直至损失函数值不再下降，接着我们选取在这个过程中损失函数值下降最快时的学习率作为我们真正训练时的初始学习率。



`learn.sched.plot_lr()`



`learn.sched.plot()`



使用`plot_lr()`和`plot()`分别画出学习率随迭代次数变化以及损失函数值随着学习率变化的图，从第二张图可以看出，损失函数值下降最快时的学习率大约为 10^{-2} ，因此我们选择其作为训练网络的初始学习率。



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY



电子信息与电气工程学院
School of Electronic Information and Electrical Engineering

THANK YOU