# 

# 第九周作业说明

[第九周作业说明 1](#_Toc1394628197)

[前言 1](#_Toc867939177)

[一、作业1 1](#_Toc1320640960)

[1补全convert\_fcn\_dataset.py 2](#_Toc18945072)

[2结果展示 2](#_Toc1677979967)

[一、 作业2 3](#_Toc348510881)

[1代码8Xfcn实现解析 3](#_Toc1634963464)

[2 结果展示 3](#_Toc217429588)

[3 FCN总结 4](#_Toc1378820320)

[● FCN的前身 5](#_Toc1981753865)

[● 双线性差值 6](#_Toc1475174770)

[● 转置卷积 6](#_Toc94030339)

[● FCN-32s,FCN-16s,FCN-8s 7](#_Toc989828807)

# 

# 前言

1. 俩个结果的网址

16Xfcn：<https://www.tinymind.com/executions/urt0ex3w>

8Xfcn：<https://www.tinymind.com/executions/ejpety62>

1. 结果展示

再训练16Xfcn的时候可以获取结果，但我没有保存，之后就无法从tinymind获取结果。联系客服很尴尬，人估计放年假了。很遗憾看不到8Xfcn的效果了。

1. cross\_entropy\_loss

16Xfcn：104.55

8Xfcn：72.63

# 一、作业1

学员需要将convert\_fcn\_dataset.py中的代码补全并生成对应的数据集文件上传到tinymind。

## 1补全convert\_fcn\_dataset.py

主要补充函数create\_tf\_record()和函数def dict\_to\_tf\_example(data, label):相对比较简单

## 2结果展示

A、结果网址：<https://www.tinymind.com/executions/urt0ex3w>

B、图表展示如下：



C、结果展示

暂时无法获取

# 作业2

## 1代码8Xfcn实现解析



## 2 结果展示

1. 结果网址：<https://www.tinymind.com/executions/ejpety62>
2. 图表展示



1. 结果展示

暂时无法获取

## **3 FCN总结**

一、如何使结果变得密集（dense）

1. 去掉pooling

导致feature map增大，进而使在更新权重的时候计算量增大（非线性增长）

导致feature map 对应的感受野变小

1. 方式FCN-32s,FCN-16s,FCN-8s

极大的利用了原有的网络结果

1. FCN核心知识

FCN的前身

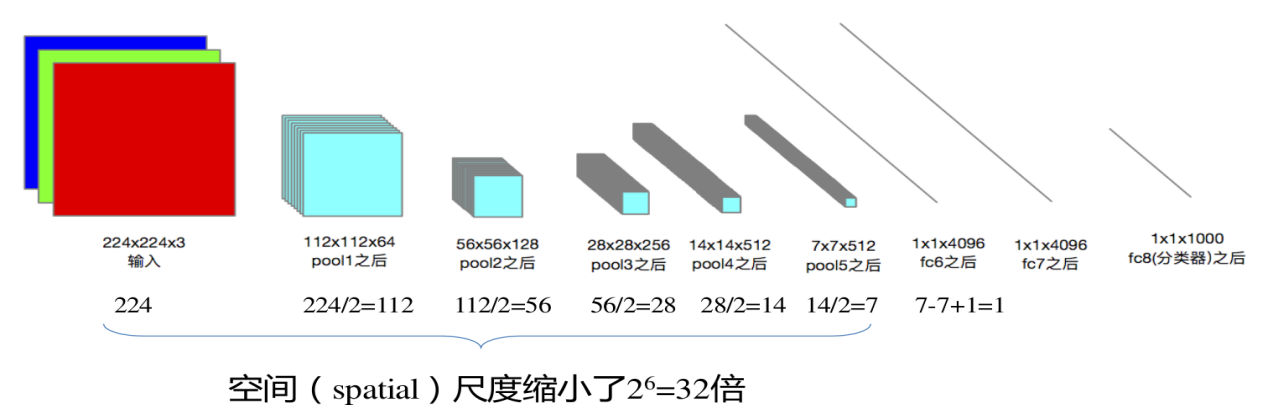
双线性差值

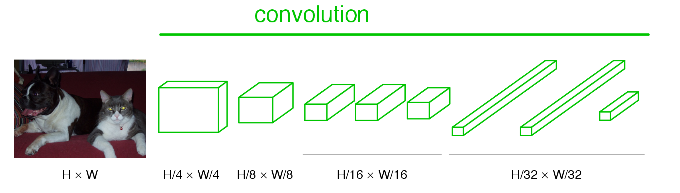
转置卷积

FCN-32s,FCN-16s,FCN-8s

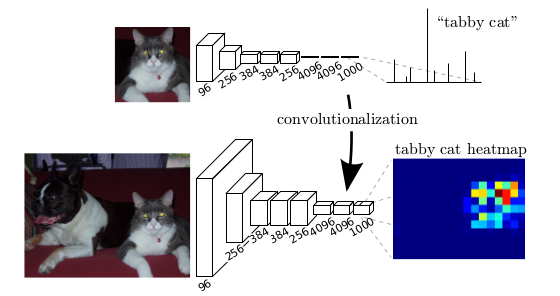
### ● FCN的前身

见下图,可看出下图由convolution 和fully connected构成，其中convolution的部分由卷积(获取高维特征)和pool(使图片缩小一半),而fully connected 主要传统神经网络相似作为权值训练，最后通过softmax输出概率最高的类别。



将全连接层转换为卷积层，这也是fully Convolutional Networks的由来。 

参照论文：Transforming fully connected layers into convolution layers enables a classification net to output a heatmap.可见下图的heatmap ,这个时候已经将原来只是识别图片**类别**，逐渐凸显图片里物体的轮廓，接下来要做的只是如何将该轮廓更加清晰一些。



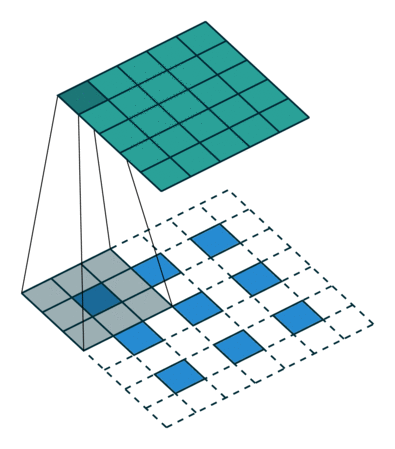
### ● 双线性差值

双线性差值的前置知识比较多，从数学角度比较难以理解，但是仅看代码，还是比较容易。

双线性差值主要应用于建立转置卷积的kernel，

### ● 转置卷积

如下图所示蓝色为原图像，白色为对应卷积所增加的padding，通常全部为0，绿色是卷积后图片。需要特别注意的是stride在这里不是kernel划动的长度，而是input image的间隔。



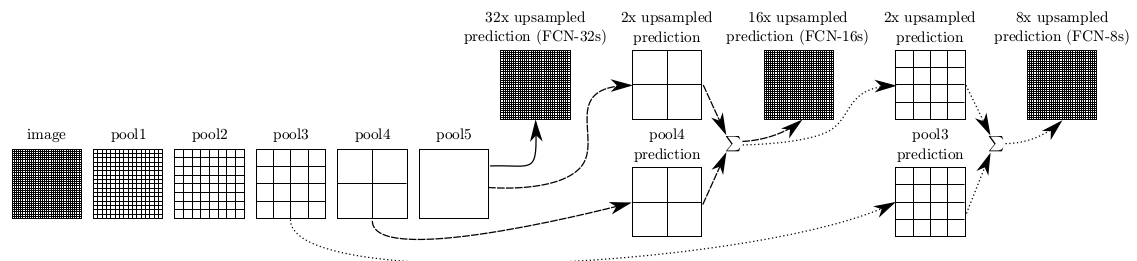
### ● FCN-32s,FCN-16s,FCN-8s

卷积是不会改变图像大小，pool就是使得图片变为原图的1/2，所以在pool5之后，图像变为原来的1/32（假如输入图像为32x32,那么pool1：16x16;pool2:8x8;pool3:4x4;pool4:2x2;pool5:1x1）。

**FCN-32s：**FCN-32s这是在pool5之后增加一个卷积层，然后经过转置卷积之后扩大32倍，变成与输入图像等大。

**FCN-16s：**FCN-16s这是在pool5之后增加一个卷积层pool5\_ conv，然后再经过转置卷积之后扩大2倍（称作2x pool5\_ conv），变成与pool4卷积之后（称作 pool4\_ conv）同等大小，然后将2x pool5\_ conv和pool4\_ conv进行fuse操作（事实上就是2x pool5\_ conv和pool4\_ conv相加），之后fuse结果进行16x upsampled prediction，至此完成了一个16s的upsample。

**FCN-8s：**FCN-8s需要将pool5卷积生成pool5\_ conv进行经过一次4x upsample（将结果称作4x pool5\_ conv），变成了4x4。然后将pool4卷积生成pool4\_ conv进行2x upsample（将结果称作2x pool4\_ conv），变成了4x4,最后将4x pool5\_ conv和2x pool4\_ conv和pool\_ conv进行fuse得到求和后的特征图，最后增加一个卷积层，使得输出图片大小为pool3的8倍，也就是8x upsampled prediction的过程。

****