1. 假设函数,代价函数的由来

<https://blog.csdn.net/Mr_HHH/article/details/78934817>

1. 平移不变性

首先,它是属于不变性的一种.何为不变性呢?

不变性就是指,物体经过翻转,移动,亮度变化之后,网络仍然能够给它识别出来.

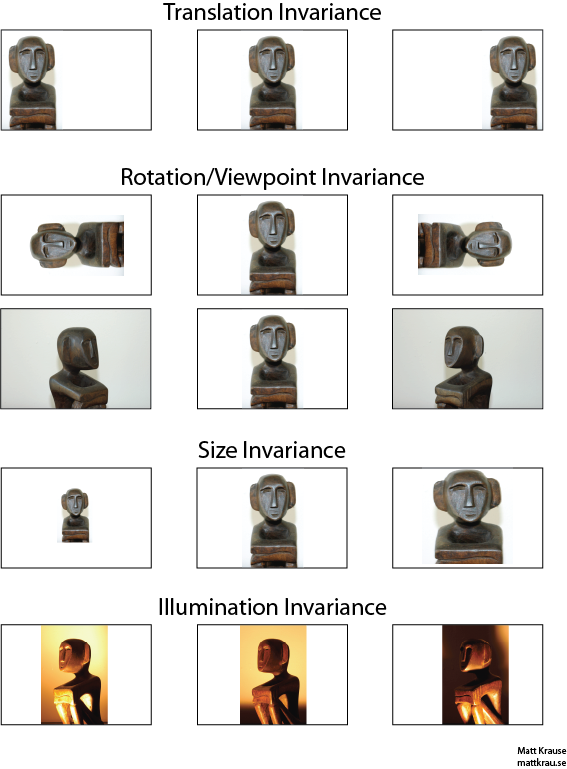
不变性包括:

平移不变性：Translation Invariance

旋转/视角不变性：Ratation/Viewpoint Invariance

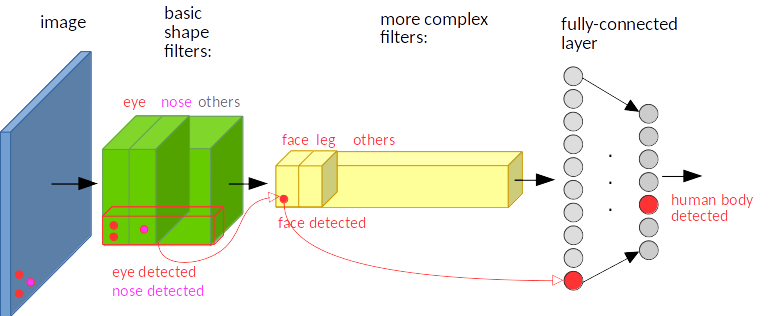
尺度不变性：Size Invariance

光照不变性：Illumination Invariance

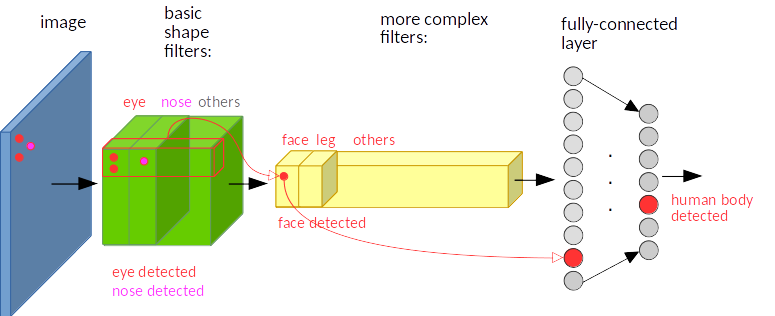


为什么卷积会有平移不变性?

主要来源于卷积+池化.



图上的image的左下角有一个人脸,经过卷积的算子筛选后,其特征图在左下角,能够辨认出是人脸.



当图上的人脸移动到左上角时候,经算子筛选后,特征图在左上角,仍然能够识别.

另外,最大池化层本身也具有保持最大值的效果,一定程度上有平移不变性.

tf.slice

<https://www.jianshu.com/p/71e6ef6c121b>

tf.squeeze

把维度[A,B,C,D]中,A~D中的值为1的那个shape,挤掉(对应的值也会扔掉了).

tf.Variable以及tf.get\_variable的用法.

他们在使用独一无二的变量时是一样的.

tf.get\_variable还可以使用之前创建的变量(类似全局变量),

但是需要有特殊语法(语句格式):

首先, 第一个函数(负责变量生成)

|  |
| --- |
| def center\_loss(features, label, alfa, nrof\_classes):  **with tf.variable\_scope('center\_loss'):**  centers = **tf.get\_variable**('centers', [nrof\_classes, nrof\_features], dtype=tf.float32,  initializer=tf.constant\_initializer(0), trainable=False) |

然后,另一个函数要使用同一个变量需要:

|  |
| --- |
| def mutual\_loss(features, label, alfa, nrof\_classes):  **with tf.variable\_scope('center\_loss', reuse=True):**  centers = tf.get\_variable('centers', [nrof\_classes, nrof\_features]) |