

空间变换网络--spatial transform network

原创

2017年09月10日 11:11:44

2190

CNN分类时，通常需要考虑输入样本的局部性、平移不变性、缩小不变性，旋转不变性等，以提高分类的准确度。这些不变性的本质就是图像处理的经典方法，即图像的裁剪、平移、缩放、旋转，而这些方法实际上就是对图像进行空间坐标变换，我们所熟悉的一种空间变换就是仿射变换，图像的仿射变换公式可以表示如下：

$$\begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{bmatrix} x^{Source} \\ y^{Source} \\ 1 \end{bmatrix} = \begin{bmatrix} x^{Target} \\ y^{Target} \end{bmatrix}$$

式中， (x^{Source}, y^{Source}) 表示原图像像素点， (x^{Target}, y^{Target}) 表示仿射变换后的图像像素点。系数矩阵 θ 即为仿射变换系数，可以通过调整系数矩阵 θ ，实现图像的放大、缩小、平移、旋转等。

那么，神经网络是否有办法，用一种统一的结构，自适应实现这些变换呢？本文提出了一种叫做空间变换网络（Spatial Transform Networks，STN）的网络模型。该网络不需要关键点的标定，能够根据分类或者其它任务自适应地将数据进行空间变换和对齐（包括平移、缩放、旋转以及其它几何变换等）。在输入数据空间差异较大的情况下，这个网络可以加在现有的卷积网络中，提高分类的准确性。

本文所提的空间变换网络的主要作用在于：

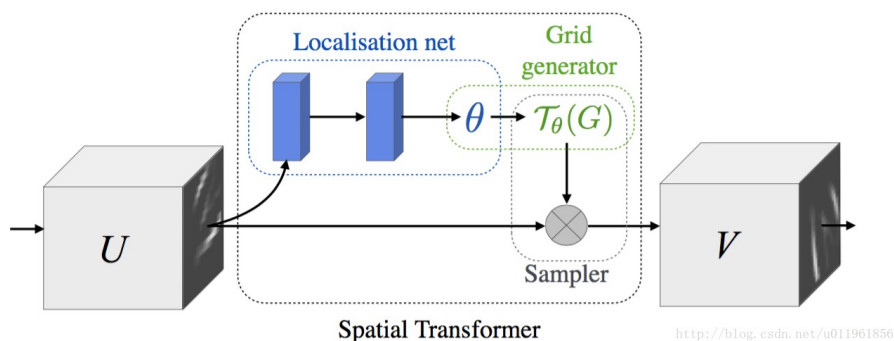
- 1.可以将输入变换为网络的下一层期望的形式；
- 2.可以在训练的过程中自动选择感兴趣的区域特征；
- 3.可以实现对各种形变的数据进行空间变换；



例如对于上图中输入手写字体，我们感兴趣的是黄色框中的包含数字的区域，那么在训练的过程中，学习到的空间变换网络会自动提取黄色框中的局部数据特征，并对框内的数据进行空间变换，得到输出output。

2. 空间变换网络原理详解

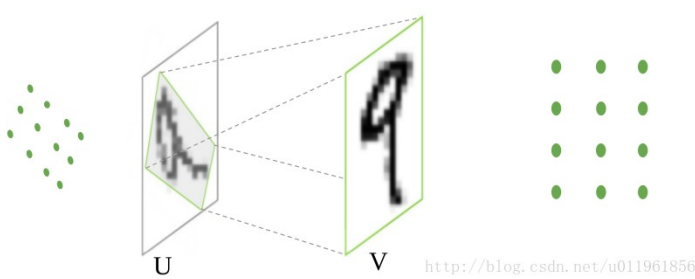
所谓空间变换网络，实际上是在神经网络的某两层之间引入一个空间变换网络，该空间变换网络包括两个部分，网络结构如下图所示：



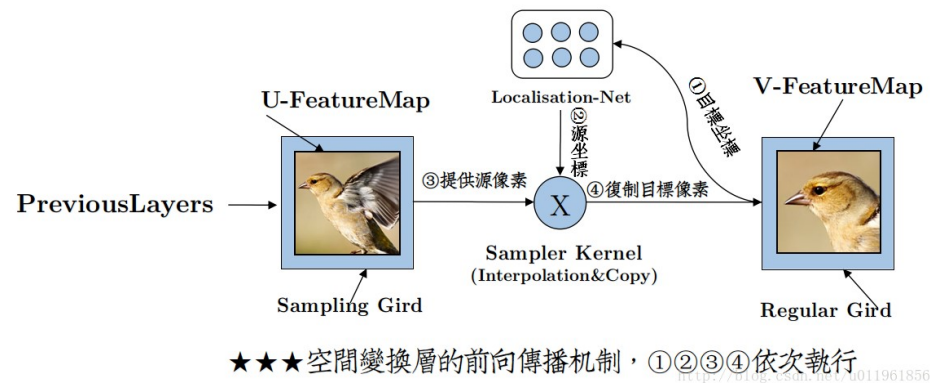
第一部分为“localization net”，其用于生成仿射变换系数结构进行设计，“localization net”网络中的参数则为空间变换网络需要训练的参数；第二部分就是空间变换即仿射变换。通过该局部网络产生仿射变换系数 θ （也可以是其他类型的空间变换，可以根据需要设计局部网络得到对应的空间变换系数 θ ，本文只以仿射变换为例进行讲解），得到仿射变换系数 θ 后我们就可以对上一层的输入进行仿射变换，并将仿射变换结果输入到下一层。例如上图中，U是输入图像或CNN某一层的feature map，V是仿射变换后的feature map，U与V中间夹着的就是“空间变换网络”，如上所说该空间变换网络中，“localization net”为用于生成仿射变换系数 θ ，“localization net”可以是全连接层或卷积层，具体的网络设计需要根据实际需要设置。“localization net”得到仿射变换系数 θ 后，

我们就可以对输入U进行仿射变换得到V了。对于“localization net”网络的设计，我们最后会以代码为例进行讲解，接下来首先讲一下得到仿射变换系数 θ 后对U进行仿射变换的原理。

对于仿射变换，如果直接由仿射变换系数 θ 对输入 (x^{Source}, y^{Source}) 求解得到输出坐标点 (x^{Target}, y^{Target}) 是非整数的，因此需要对考虑逆向仿射变换。所谓逆向仿射变换就是首先根据仿射变换输出的大小，生成输出的坐标网格点，即上图中的“Grid generator”，例如V的大小为 10×10 时，我们便可以得到一个 10×10 大小的坐标位置点矩阵，接下来就是要对该坐标位置点进行仿射变换，仿射变换公式及示意图如下：

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathbf{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$


需要注意的是，这时候的仿射系数为 θ 的逆矩阵，经过仿射变换后可以得到V中的位置坐标点在U中对应的位置。但是，这时候求得的U中的坐标点仍然可能为非整数，因此通常需要进行插值得到对应的坐标点。得到U中的坐标点后，则可以将其复制到V中，从而得到仿射变换结果V。具体仿射变换过程，也可以结合下图进行理解：



上图中由Localisation Net生成仿射变换系数 θ 后，仿射变换的过程是依次执行步骤1，2，3，4。

3. 空间变换网络的实际应用

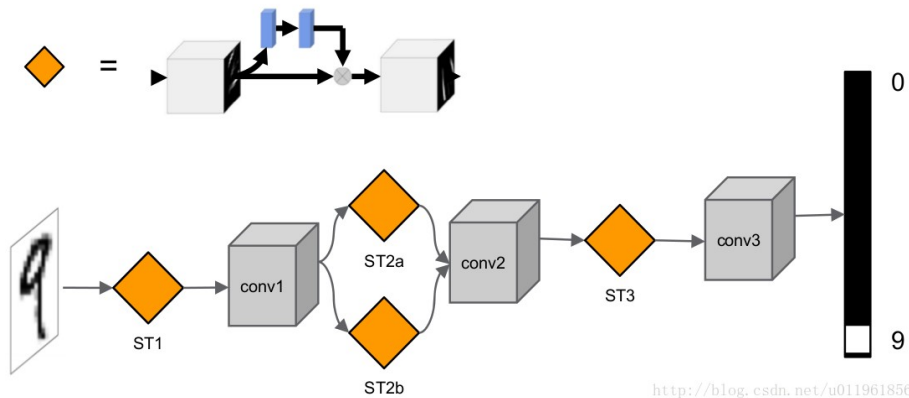
以上讲解的是空间变换网络的理解，那么在实际应用中，我们该如何添加空间变换网络到我们自己的网络中呢？接下来重点讲解空间变换网络的应用。

3.1.空间变换网络作为网络的第一层

空间变换网络可以直接作为网络的第一层，即Localisation Net的输入为input，从而直接对输入进行仿射变换，对于Localisation Net的设计，可以根据输入input的大小设计Localisation Net为全连接层或卷积层，例如对于手写字体，输入图片大小为 40×40 ，即 $\text{input}=[\text{batch_size}, 1600]$ ，那么我们可以设计Localisation Net包含两个全连接层，第一个全连接层 $w1=[1600, 20], b1=[20]$ ，第一个全连接层 $w2=[20, 6], b2=[6]$ ，则第二个全连接层的输出为 $[\text{batch_size}, 6]$ ，即为仿射变换系数。若输入Localisation Net的input尺寸较大，则需要添加卷积和池化层，最后再输入到全连接层，得到仿射变换系数；关于全连接层和卷积层的设计详见参考代码4.1.4.2。

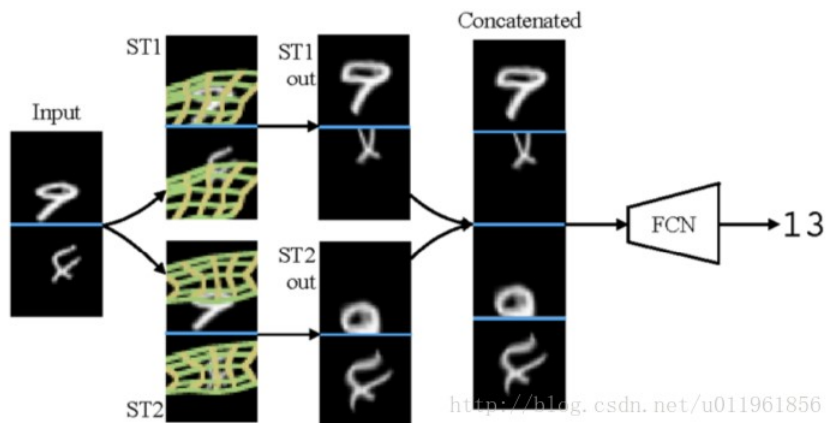
3.2.空间变换网络插入CNN的中间层

空间变换网络还可以添加在CNN的中间层，可以直接将空间变换网络插入conv或者max-pooling层的前面或者后面。此外，还可以在CNN的同一层插入多个空间变换网络，下面给出空间变换网络插入CNN的手写字体网络结构图：



上图中第一个空间变换网络ST1作用于输入图像，直接对输入图像进行空间变换，第二、三个空间变换网络ST2a,ST2b作用于conv1,用于对第一层的卷积特征进行空间变换，而ST3用于对更深层的卷积特征进行空间变换。

由于空间变换网络能够自动提取局部区域特征，因此在网络的同一层插入多个空间变换网络可以提取多个局部区域特征，从而可以结合多个局部区域特征进行分类，如下如的网络是实现两张输入的图片中的手写字体相加，在网路的第一层插入两层空间变换网络ST1,ST2,并将其直接作用于输入图像。图中第三列为空间变换结果，有图可知，网络ST1,ST2分别提取了输入手写体的不同区域的特征。



4. 代码分析

首先看一仿射变换的代码实现，代码的实现如上所述，首先由函数_meshgrid生成输出V的坐标位置点grid，在通过仿射变换系数theta对grid进行仿射变换得到U中对于位置坐标点T_g，之后对T_g进行双线性插值，并复制插值后的U中的坐标点的像素值到V中，得到输出V。具体代码实现如下：

```
1
2 def transform(theta, input_dim, out_size):
3     with tf.variable_scope('transform'):
4         num_batch = tf.shape(input_dim)[0]
5         height = tf.shape(input_dim)[1]
6         width = tf.shape(input_dim)[2]
7         num_channels = tf.shape(input_dim)[3]
8         theta = tf.reshape(theta, (-1, 2, 3))
9         theta = tf.cast(theta, 'float32')
10
11         # grid of (x_t, y_t, 1), eq (1) in ref [1]
12         height_f = tf.cast(height, 'float32')
13         width_f = tf.cast(width, 'float32')
14         out_height = out_size[0]
```

```

15         out_width = out_size[1]
16         grid = _meshgrid(out_height, out_width)
17         grid = tf.expand_dims(grid, 0)
18         grid = tf.reshape(grid, [-1])
19         grid = tf.tile(grid, tf.pack([num_batch]))
20         grid = tf.reshape(grid, tf.pack([num_batch, 3, -1])) # 得到输出坐标位置点
21
22         # Transform A x (x_t, y_t, 1)^T -> (x_s, y_s)
23         T_g = tf.batch_matmul(theta, grid) # 仿射变换
24         x_s = tf.slice(T_g, [0, 0, 0], [-1, 1, -1]) #
25         y_s = tf.slice(T_g, [0, 1, 0], [-1, 1, -1])
26         x_s_flat = tf.reshape(x_s, [-1])
27         y_s_flat = tf.reshape(y_s, [-1])
28
29         input_transformed = _interpolate(
30             input_dim, x_s_flat, y_s_flat,
31             out_size) # 插值, 并得到输出
32
33         output = tf.reshape(
34             input_transformed, tf.pack([num_batch, out_height, out_width, num_channels]))
35         return output
36

```

接下来结合两个具体的实例分别讲解"localization net"的为全连接层和卷积层的设计。

(4.1) "localization net" 的为全连接层的实例：

该示例中，空间变换网络用于对输入图像进行变换处理，"localization net"包括两个全连接层，具体网络设计如下：

```

1
2
3     x = tf.placeholder(tf.float32, [None, 1600]) # 输入
4     y = tf.placeholder(tf.float32, [None, 10])
5
6
7     x_tensor = tf.reshape(x, [-1, 40, 40, 1])
8
9
10    W_fc_loc1 = weight_variable([1600, 20]) # 第一个全连接层
11    b_fc_loc1 = bias_variable([20])
12
13    W_fc_loc2 = weight_variable([20, 6]) # 第二个全连接层
14
15    initial = np.array([[1., 0, 0], [0, 1., 0]])
16    initial = initial.astype('float32')
17    initial = initial.flatten()
18    b_fc_loc2 = tf.Variable(initial_value=initial, name='b_fc_loc2')
19
20
21    h_fc_loc1 = tf.nn.tanh(tf.matmul(x, W_fc_loc1) + b_fc_loc1)
22
23    keep_prob = tf.placeholder(tf.float32)
24    h_fc_loc1_drop = tf.nn.dropout(h_fc_loc1, keep_prob)
25
26    h_fc_loc2 = tf.nn.tanh(tf.matmul(h_fc_loc1_drop, W_fc_loc2) + b_fc_loc2) # 仿射变换系数the
27
28    out_size = (40, 40)
29    h_trans = transformer(x_tensor, h_fc_loc2, out_size)
30

```

该代码所示，该空间变换网络部分需要训练的参数即为"localization net"包括两个全连接层的参数W_fc_loc1, b_fc_loc1, W_fc_loc2, b_fc_loc2。完整代码链接如下：

https://github.com/tensorflow/models/blob/master/transformer/cluttered_mnist.py

(4.2) "localization net" 的为卷积层的实例：

如下代码所示，该"localization net"层包括两个卷积层，并包括两个全连接层，得到仿射变换系数theta。

```

1
2 from keras.models import Sequential
3 from keras.layers.core import Dense, Dropout, Activation, Flatten
4 from keras.layers.convolutional import Convolution2D, MaxPooling2D

```

```

5
6 #localization net,得到仿射变换系数theta
7
8 locnet = Sequential()
9 locnet.add(MaxPooling2D(pool_size=(2,2), input_shape=input_shape))#
10 locnet.add(Convolution2D(20, 5, 5))
11 locnet.add(MaxPooling2D(pool_size=(2,2)))
12 locnet.add(Convolution2D(20, 5, 5))
13
14 locnet.add(Flatten())
15 locnet.add(Dense(50))
16 locnet.add(Activation('relu'))
17 locnet.add(Dense(6, weights=weights))#输出仿射变换系数theta
18 #locnet.add(Activation('sigmoid'))
19
20 #build the model
21
22 model = Sequential()
23 model.add(SpatialTransformer(localization_net=locnet,
24                             downsample_factor=3, input_shape=input_shape))#仿射变换
25
26 model.add(Convolution2D(32, 3, 3, border_mode='same'))
27 model.add(Activation('relu'))
28 model.add(MaxPooling2D(pool_size=(2, 2)))
29 model.add(Convolution2D(32, 3, 3))
30 model.add(Activation('relu'))
31 model.add(MaxPooling2D(pool_size=(2, 2)))
32
33 model.add(Flatten())
34 model.add(Dense(256))
35 model.add(Activation('relu'))
36
37 model.add(Dense(nb_classes))
38 model.add(Activation('softmax'))
39
40 model.compile(loss='categorical_crossentropy', optimizer='adam')
41

```

该 # localization net部分需要训练的部分为locnet部分的每一层的系数，完整代码链接如下：

(<https://github.com/EderSantana/seya/blob/master/examples/Spatial%20Transformer%20Networks.ipynb>)

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/u011961856/article/details/77920970>



[查看更多>>](#)

看 Python 如何诠释“薪”时代

Python全栈开发包含Python爬虫、前端、网站后台、Python机器学习与数据挖掘等，从0基础小白到Python 企业级web开发达人、自动化运维开发能手的进击，课程真实企业项目实战演练，全面系统学习python编程语言，从容应对企业中各式各样的.....

29224

[查看更多>>](#)

该文章已被禁止评论！

深度学习（六十三）空间变换网络



hjimce 2017年06月07日 22:42 3337

卷积神经网络每一层都有其强大的功能，然而它对于输入数据的空间不变性却还有很大的缺陷，可能max pooling层，具有平移不变性，然而因为max pooling是一个局部操作，在CNN中对于大尺度的空...

Spatial Transformer Networks(空间变换神经网络)

闲扯：大数据不如小数据 这是一份很新的Paper(2015.6)，来自于Google旗下的新锐AI公司DeepMind的四位剑桥Phd研究员。他们针对CNN的特点，构建了一个新的局部..

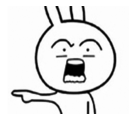


brandon2015 2017年05月12日 17:56 1260

码农怎能不懂英语?！试试这个数学公式

老司机教你一个数学公式秒懂天下英语

广告



Tensorflow1.0空间变换网络(SpatialTransformer Networks)实现

空间变换网络简单介绍：通过locatnet，提取输入图像的theta(将用于仿射变换)；根据输入图像的width和height以及仿射变换(或者TPS)的参数theta，可以生成目标位...



weixin_36368407 2017年03月01日 10:01 1459

论文笔记：Spatial Transformer Networks (空间变换网络)

上一篇博客 Spatial Transformer Networks论文笔记（一）——仿射变换和双线性插值介绍了仿射变换和双线性插值，为更好地理解STN打基础。本篇博客是记录的是阅读原文Spatial...



sinat_34474705 2017年07月17日 21:18 1203

【论文笔记】Spatial Transformer Networks shaoxiaohu1 2016年07月02日 12:46 14785

卷积神经网络（CNN）已经被证明能够训练一个能力强大的分类模型，但与传统的模式识别方法类似，它也会受到数据在空间上多样性的影响。这篇Paper提出了一种叫做空间变换网络（Spatial Transfo...

运维管理系统

it运维管理软件

百度广告



Spatial Transformation Network

STN



DuinoDu 2017年04月07日 12:46 877

caffe-stn移植到高版本的方法

本文介绍了将github上开源的stn-caffe代码移植到新版本的caffe的方法



kuaitoukid 2016年04月01日 11:00 3041

空间变换网络STN

出自论文Spatial Transformer Networks Insight：文章提出的STN的作用类似于传统的矫正的作用。比如人脸识别中，需要先从检测的图片进行关键点检测，然后使...



qq_14845119 2018年03月10日 19:20 188

MXNet - Python API

MXNet - Python API>>> import mxnet as mx ndarray APIOverviewThis document lists the routines of the ...



jianglei_love 2017年04月28日 22:05 1401

Paper Reading: Spatial Transformer Networks (with code explanation)

原文：Spatial Transformer Networks前言：卷积神经网络（CNN）已经可以构建出一个强大的分类模型，但它仍然缺乏能力来应对输入数据的空间变换，。这篇文章提出了一种叫做空间变换网...



l691899397 2016年12月14日 17:25 2118



技术外文文献看不懂？教你一个公式秒懂英语
不背单词和语法，一个公式学好英语

【Unet】不使用NetWorkTransform组件 进行同步位移和旋转

using System.Collections; using System.Collections.Generic; using UnityEngine; using UnityEngine.Net...



ldy597321444 2017年04月12日 17:49 1583

【图像处理】空间变换



jaych 2016年02月18日 20:23 1640

概念在图像处理中的空间变换 (spatial transformation) 分成两种情况, 有仿射变换 (Affine Transformation) 及投影变换 (Perspective Transforma...

三维空间几何变换原理[平移、旋转、错切]



shenziheng1

2016年04月23日 20:09

🔒 7288

dx11学习笔记-3.三维空间变换 (自己整理, 基础详尽)

三维空间变换其实是图形学的基本知识, 和DX倒没有什么特定关系。但MS关于这块儿的文档教程解释得很详尽。看完之后我觉得我对于坐标变换的理解又深入了一步。此处我截取自己觉得精要的地方来说明。资料参考: 1、...



code_xbug

2016年02月17日 12:41

🔒 3394

1 空间描述和变换



libing403

2017年02月13日 15:06

🔒 1302

1 空间描述和变换 空间描述和变换 1位置描述 2 姿态描述 3 坐标系的描述 4坐标系的一般映射齐次矩阵变换 5 变换算法 51混合变换 52 逆变换 6变换方程 7 姿态的其他描述方法 71固定角坐...

免费云主机试用一年

免费云主机试用一年厂家批发

百度广告



空间变换网络--spatial transform network



u011961856

2017年09月10日 11:11

🔒 2190

CNN分类时, 通常需要考虑输入样本的局部性、平移不变性、缩小不变性, 旋转不变性等, 以提高分类的准确度。这些不变性的本质就是图像处理的经典方法, 即图像的裁剪、平移、缩放、旋转, 而这些方法实际上就是对图像...

Spatial Transformer Networks(空间变换神经网络)

转自: <http://www.cnblogs.com/neopenx/p/4851806.html> Spatial Transformer Networks(空间变换神经网络) ...



lwq1026

2017年09月18日 15:28

🔒 291

深度学习方法 (十二) : 卷积神经网络结构变化——Spatial Transformer Networks

今天具体介绍一个Google DeepMind在15年提出的Spatial Transformer Networks, 相当于在传统的一层Convolution中间, 装了一个“插件”, 可以使得传统的卷积...



xbinworld

2017年04月03日 23:45

🔒 7316

读后感 Spectral-Spatial Residual Network for Hyperspectral Image Classificati...

v 为了利用丰富的光谱带, 传统的逐像素HSI分类模型主要集中在两个步骤: 特征工程和分类器训练。特征工程方法包括特征选择(波段选择)和特征提取[7]。特征工程的主要目标是降低HSI像素的高维度并提取最具...



hanlinger_

2018年01月29日 17:23

🔒 148

空间点坐标变换



GabrielDracula

2018年04月03日 10:26

🔒 13

当考虑同一坐标系下, 某空间点绕某一轴旋转完平移后的坐标, 直接乘以该旋转的旋转矩阵再加上旋转完后的平移向量即可。当考虑不同坐标系下, 同一空间点的两个坐标的时候, 假设已知坐标系经过何种平移旋转可以变为另一...