胶囊网络——Capsule Network

1. 背景介绍

CNN 在处理图像分类问题上表现非常出色，已经完成了很多不可思议的任务，并且在一些项目上超过了人类，对整个机器学习的领域产生了重大的影响。而 CNN 的本质由大量的向量和矩阵的相乘或者相加，因此神经网络的计算消耗非常大，所以将一张图片上全部像素信息传递到下一层运算是十分困难的，所以出现了“卷积”和“池化”这种方法，能够在不损失数据本质的情况下帮我们简化神经网络的计算。

诚然， CNN 在分类和数据集非常接近的图像时，实验的效果非常好，但如果图像存在翻转、倾斜或任何其它方向性问题时，卷积神经网络的表现就比较糟糕了。通过在训练期间为同一图像翻转和平移可以解决这个问题（数据增强）。而问题的本质是网络中的滤波器以比较精细的级别上理解图像。举一个简单的例子，对于一张人脸而言，它的组成部分包括面部轮廓，两个眼睛，一个鼻子和一张嘴巴。对于 CNN 而言，这些部分就足以识别一张人脸。然而，这些组成部分的相对位置以及朝向就没有那么重要。 这些主要的原因是人类在识别图像的时候，是遵照树形的方式，由上而下展开式的，而 CNN 则是通过一层层的过滤，将信息一步步由下而上的进行抽象。这是胶囊网络作者认为的人与 CNN 神经网络的最大区别。

本文基于一篇在 2017 年由 Hinton 等人发表的一篇文章 Dynamic Routing Between Capsules，该文章介绍了一种神经网络可以弥补 CNN 无法处理图片位置方向等缺点。相比于 CNN 该网络更接近于人类的图像识别原理。

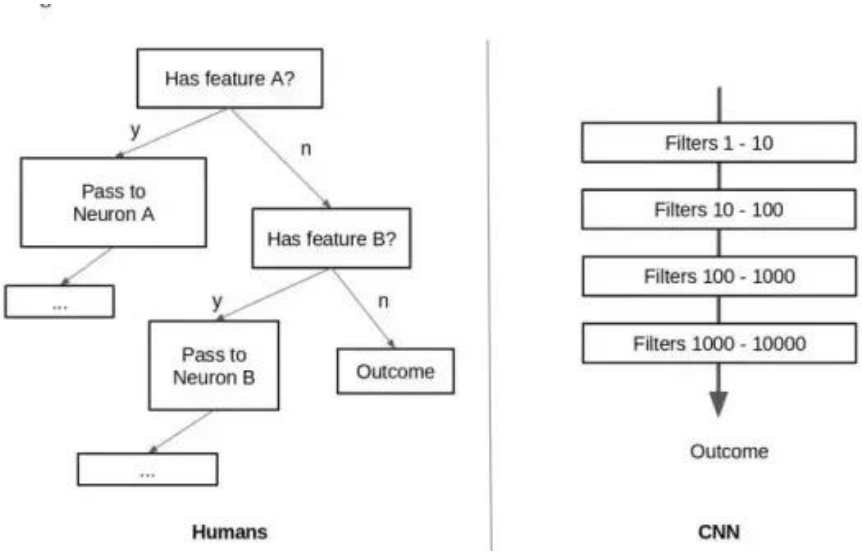

图 1：人类识别图像与 CNN 的区别

2. 胶囊网络是如何工作的

胶囊网络和传统的人工神经网络最根本的区别在于网络的单元结构。对于传统神经网络

而言，神经元的计算可分为以下三个步骤：

- 对输入进行标量加权计算。
- 对加权后的输入标量进行求和。
- 标量到标量的非线性化。

而对于胶囊而言，它的计算分以下四个步骤：

1. 对输入向量做乘法，其中 v_1 和 v_2 分别来自与前面的 capsule 的输出，在单个 capsule 内部，对 v_1 和 v_2 分别乘上 w_1 和 w_2 得到了 新的 u_1 和 u_2 。

2. 对输入向量进行标量加权，令 u_1 与 c_1 相乘，u_2 与 c_2 相乘，其中 c_1 和 c_2 均为标量，且 $c_1 + C_2 = 1$ 。

3. 对得到向量求和，得到 $s = c_1 u_1 + c_2 u_2$ 。

4. 向量到向量的非线性化，将得到的结果向量 s 进行转换，即通过函数 $Squash(s) = \frac{||s||^2}{1 + ||s||^2} \frac{s}{||s||}$ 得到结果 s，作为这个capsule 的输出，且这个结果 v 可以作为下一个 capsule 的输入。

图 2：单个胶囊的运算方式

3. 胶囊网络的细节

3.1 胶囊网络的动态寻路算法

上一章我们了解胶囊网络工作的总体方式，这一章我们来关注胶囊内部的参数是如何进行更新的，首先我们来看论文里关于此算法的伪代码介绍：

Procedure 1 Routing algorithm.

procedure ROUTING($\hat{\boldsymbol{u}}_{j|i}, r, l$)
  for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
  for r iterations **do**
    for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ `softmax` computes Eq. 3
    for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$
    for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$ ▷ `squash` computes Eq. 1
    for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i} . \mathbf{v}_j$
  return $\mathbf{v}_j$

图 3：单个胶囊内的参数更新

3.2 动态寻路算法直观上的理解

直观上可以将寻路算法表示为下图所示的过程：

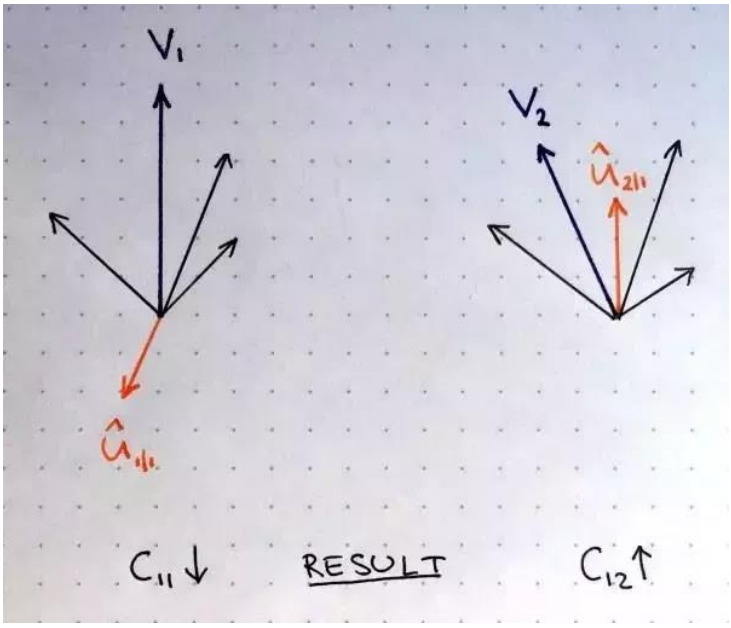

图 5：寻路算法直观上的理解

4. 论文中的网络结构

4.1 训练的网络结构

论文用 minst 的数据集上的 CapsNet 架构如下图所示。架构可以简单的表示为仅有两个卷积层和一个全连接层组成。Conv1 有 256 个 9×9 个卷积核，步长为 1 和 ReLU 激活。该层将像素的强度转换为之后会用于作为基本胶囊输入的局部特征探测器的激活。

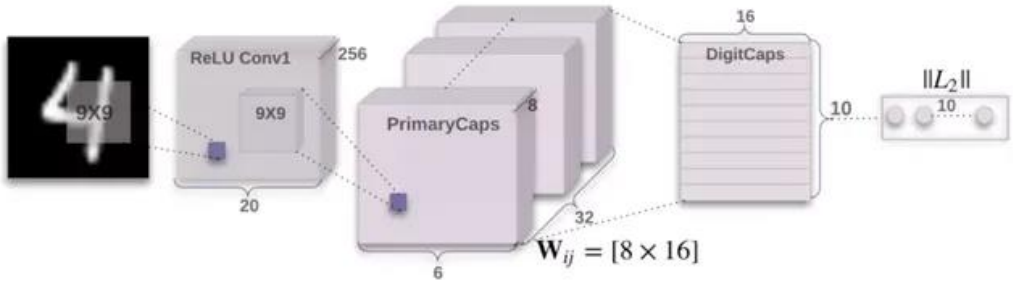

图 6：训练的网络结构

第一层为卷积层：输入：28×28 图像（单色），输出：20×20×256 张量，卷积核：256 个步长为 1 的 9×9×1 的核，激活函数：ReLU。第二层为 primarycaps 层：输入：20×20×256 张量，输出：6×6×8×32 张量（共有 32 个胶囊），卷积核：8 个步长为 2 的 9×9×256 的核/胶囊。 第三层为 DigitCaps 层：输入：6×6×8×32 张量（每个胶囊输出的是 8 维的向量），输出：16×10 矩阵（10 个胶囊）。

4.2 重构的网络结构

重构器从正确的 DigitCap 中选择一个 16 维向量，并学习将其编码为数字图像（注意，训练时候只采用正确的 DigitCap 向量，而忽略不正确的 DigitCap ）。它接受正确的 DigitCap 的输出作为输入，重建一张 28×28 像素的图像，损失函数为重建图像和输入图像之间的欧式距离。解码器强制胶囊学习对重建原始图像有用的特征，重建图像越接近输入图

像越好，下面展示重构的网络结构（最终的输出 28*28）和重建图像的例子（l,p,r 对应了标签，预测，重构目标）。

5. 小结

胶囊网络在如今卷积网络难以提升的阶段可以算是一种革命性的网络架构，神经元的输出从一个标量变成了一组向量，这如同让网络流动起来了。每一个识别子结构的胶囊，使原始图形中的细节高度的保真，而这个保真，又是从复杂结构里直接进化得到的。通过重构原图，模型做到了在视角变换后，通过网络结构的改变，给出相同的结果。另外需要指出的是，CNN 和胶囊神经网络并不是互斥的，网络的底层也可以是卷积式的，毕竟胶囊网络善于的是在已抽象信息中用更少的比特做到高保真的重述。