# 胶囊网络capsulenet综述

即capsnet。Geoffrey Hinton在2017年提出的CapsNet，并对其寄予厚望，希望它超越卷积神经网络(CNN)。2018年和2019年上半年，胶囊网络正在呈爆发之势，很像2014年大家谈CNN。

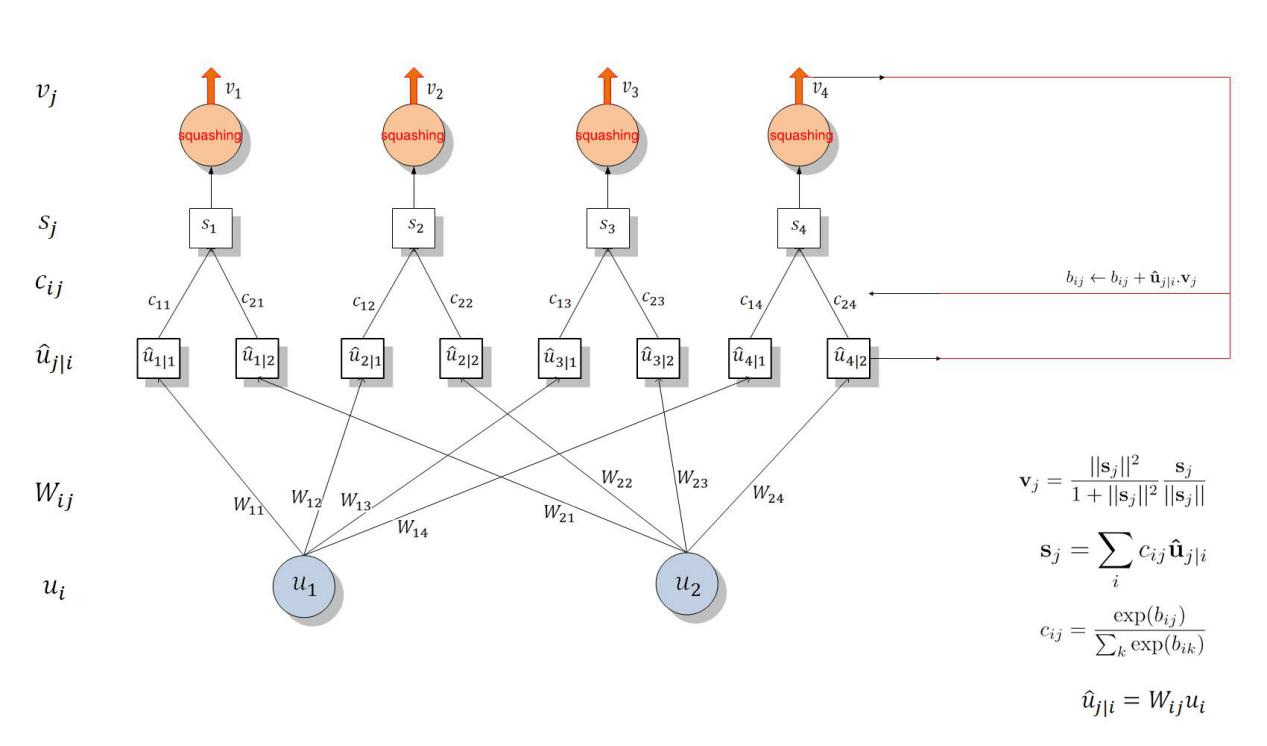
CapsNet就是一个向量版的神经元，而不是所谓的“胶囊”。它有方向的概念，让图像不同区域的卷积结果对最后分类或者回归结果贡献权重差异化，例如当嘴巴在眼睛上面时可以发现错误。

优点：

1）旋转不变形

2）建立了模块/部件之间的关联（和图卷积很像）

## capsnet分析



reconstruction loss其实就是在DigitCaps层后又接了3个FC，最后的FC输出会reshape成原图大小，然后计算原图和fake-image的像素级损失。

PrimaryCaps相当于conv层。没有routing的，只有conv、reshape和squashing激活。

DigitCaps相当于FC层。参数c是由参数b经过softmax得到的，使用的是前向过程中的Dynamic Routing更新的，并不需要根据损失函数更新（但整个网络其它的卷积参数和 Capsule 内的 W\_ij 都需要根据损失函数进行更新）。

### 将标量输入输出扩展到向量输入输出

在传统神经元中，标量 xi，加权求和得到 aj。用非线性激活函数，转换得到神经元输出，是个标量值，激活函数可以选择 sigmoid、tanh 和 ReLU 等，最终得到标量。

而在 Capsule 中，ui 是向量，矩阵的乘就是一个简单的仿射变换，在传统神经元中没有，然后，对 i 维度做加权求和，传统是对标量加权求和，Capsule 是对向量加权求和得到向量。Squash 函数是个非线性的函数，与传统非线性对应，输出是向量。

### 训练过程

新的神经元的参数有哪些呢？W 和 c。

然后计算梯度，梯度返回来更新 w，即 c 是在前向过程中计算的，w 是在反向的过程 BP 中更新的。

注意，在一个iteration中：

先初始化b（每次iteration的阶段都重新会初始化b），然后在routing阶段不断更新（即routing部分会进行多次前向更新）。

更新到一定程度，才进行向前传播直到网络尾端，计算loss，更新参数w。

Q: 训练过程是怎样的？BP 和 routing 是交替进行的吗？

A:Cij 是在数据前向传播更新的

数据在前向计算时，在迭代更新 cij，更新完才输出 v4，然后用loss function计算梯度，再然后使用BP来更新 w。总之，c 是在前向过程中计算的，而w 是在反向过程 BP 中更新的。

### Dynamic Routing（动态路由）

在前向过程中，迭代更新Capsule就是所谓的Dynamic Routing。

Dynamic Routing的含义就是通过迭代，根据高级胶囊的输出逐步调整低级胶囊输出给高级胶囊的分布，最后会达到一种理想的分布。Hinton说routing像attention机制，attention是高层选底层，caps是底层选高层。

Dynamic Routing是干嘛的？

Capsulenet中缺少了pooling（原文中是maxpooling），但是却多了routing，也就是说，routing在这里代替了pooling。本质上，这两种运算都是有用信息的提取方式。

maxpooling依靠区域最大化，以此来提取到了这个区域里最有用的信息，但是却忽略了其他的信息，会造成信息丢失。

但pooling过程中所丢弃的那些信息并非真的没用。

假设我们识别的一张图片里有一只猫，和一条狗，猫占图片的大部分。maxpooling后我们丢弃掉了较少的狗的信息，而保留了较多的猫的信息，于是这张图片被识别为猫，但图片中的狗怎么办？

routing的好处就是并不丢弃任何的信息，属于猫的信息那么就将其更多的输出到检测“猫”的高级胶囊中，属于狗的信息就尽可能的都输出给“狗”的高级胶囊。这样的我们最后的结果，猫和狗的得分都会很高。因此Hinton说，Caps更适合去检测那些重叠（overlapping）在一起的图片。

## CapsNet与CNN的比较

传统的Neural network（如CNN）使用卷积和池化运算，可以得到一样的特征，也就保证了它学习之后的不变性（Invariance），即卷积不变性，可以忽略方向上的差别，但是也就只能做到不变性，这样在下游继续接Dense层之后可以归到同一类别。用一句简单话来总结这类Neural Network就是：I don't know the difference.

在Capsule Network中，在经过Capsule运算后，也可以识别并记录这些差别，能真正做到同变性（Equivariance），用以保真和记录更多信息，而不真正做出反应，因为从Capsule输出的向量我们做取模运算便可以得到其对应的label，取模之后是一样的，但是在取模之前是可以允许向量有差别的，一定程度上也提高了网络的鲁棒性。用一句话总结：I know the difference, but I don't react to it."