

【一文读懂Hinton最新论文】胶囊网络9大优势4大缺陷（视频+PPT）

2017-11-25 新智元

新智元AI World 2017世界人工智能大会开场视频

中国人工智能资讯智库社交主平台新智元主办的 [AI WORLD 2017 世界人工智能大会](#) 11月8日在北京国家会议中心举行，大会以“AI 新万象，中国智能+”为主题，上百位AI领袖作了覆盖技术、学术和产业最前沿的报告和讨论，2000多名业内人士参会。新智元创始人兼CEO杨静在会上发布全球首个AI专家互动资讯平台“新智元V享圈”。

全程回顾新智元AI World 2017世界人工智能大会盛况：

新华网图文回顾

<http://www.xinhuanet.com/money/jrzb20171108/index.htm>

爱奇艺

上午：http://www.iqiyi.com/v_19rrdp002w.html

下午：http://www.iqiyi.com/v_19rrdozo4c.html

阿里云云栖社区

<https://yq.aliyun.com/webinar/play/316?spm=5176.8067841.wnnow.14.ZrBcrm>

新智元推荐

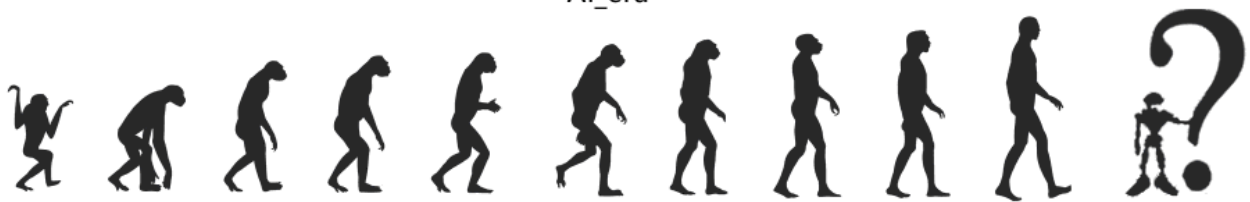
来源：专知

【导读】10月26日，深度学习元老Geoffrey Hinton和他的团队NIPS2017 Capsule论文《Dynamic Routing Between Capsules》在arxiv上发表，介绍了全新的胶囊网络模型，以及相应的囊间动态路由

算法。日前，Hands-On Machine Learning with Scikit-Learn and TensorFlow这本书的作者Aurélien Géron制作了Capsule Networks视频教程，专知内容组对这个视频教程进行了解读，请大家查看，并多交流指正！



点击右上角
分享文章到朋友圈
欢迎关注公众号
AI_era



视频

先看下Aurélien Géron介绍 Capsule Networks的视频教程（**英文字幕**）

PPT

由于笔者能力有限，本篇所有备注皆为专知内容组成员根据讲者视频和PPT内容自行补全，不代表讲者本人的立场与观点。

胶囊网络

Capsule Networks



你好！我是AurélienGéron，在这个视频中，我将告诉你们关于胶囊网络，一个神经网络的新架构。Geoffrey Hinton几年前就有胶囊网络的想法，他在2011年发表了一篇文章，介绍了许多重要的想法，他还是很难让这些想法实现，但直到现在。

NIPS 2017 Paper

Dynamic Routing Between Capsules

by Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton

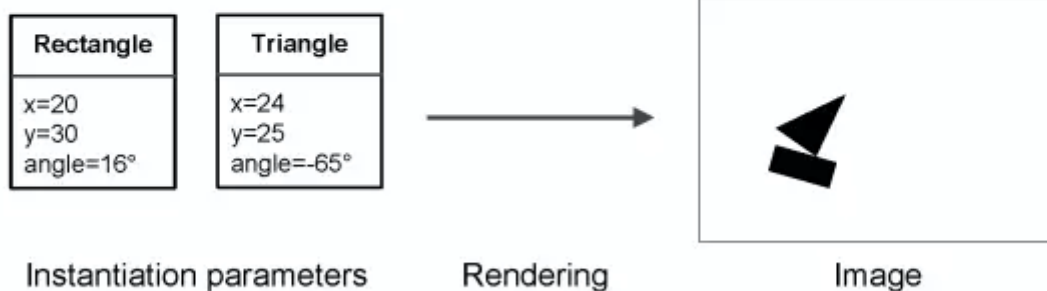
October 2017: <https://arxiv.org/abs/1710.09829>

几周前，在2017年10月，一篇名为“动态路由胶囊”由作者Sara Sabour，Nicholas Frosst，当然还有Geoffrey Hinton一起发表了。

链接是：<https://arxiv.org/abs/1710.09829>

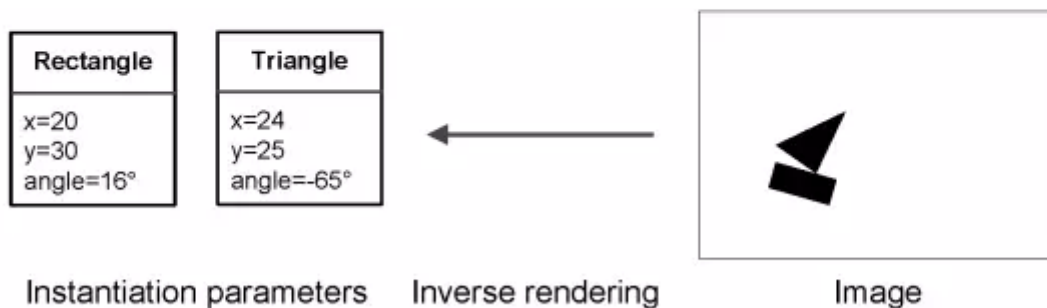
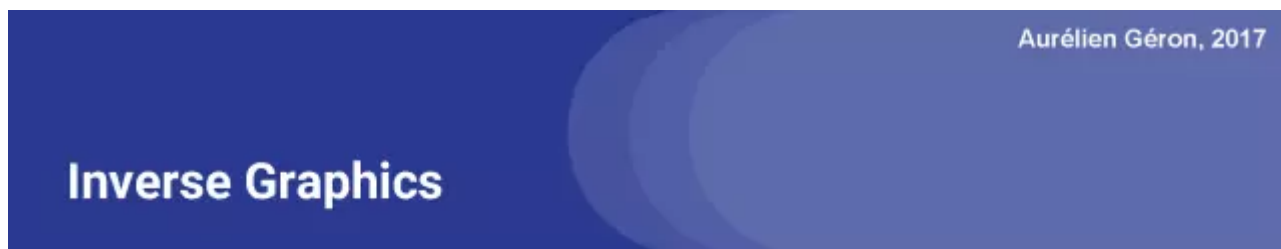
他们在MNIST数据集上达到了最先进的性能，并且在高度重叠的数字上表现出比卷积神经网络好得多的结果。那么胶囊网络究竟是什么？

Computer Graphics



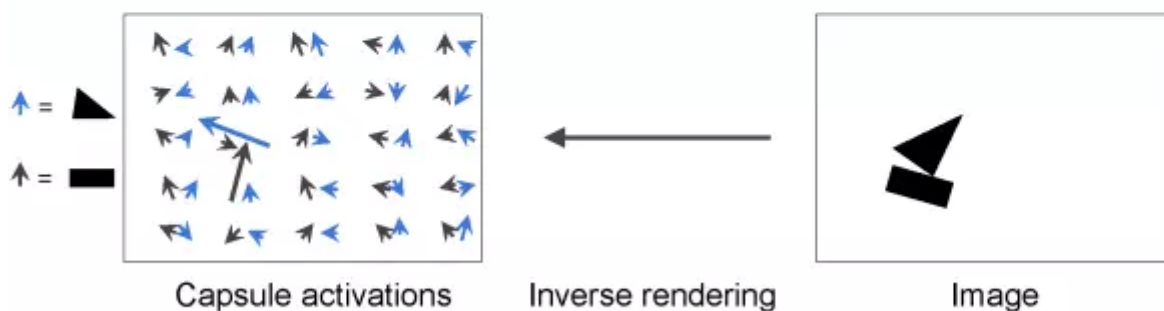
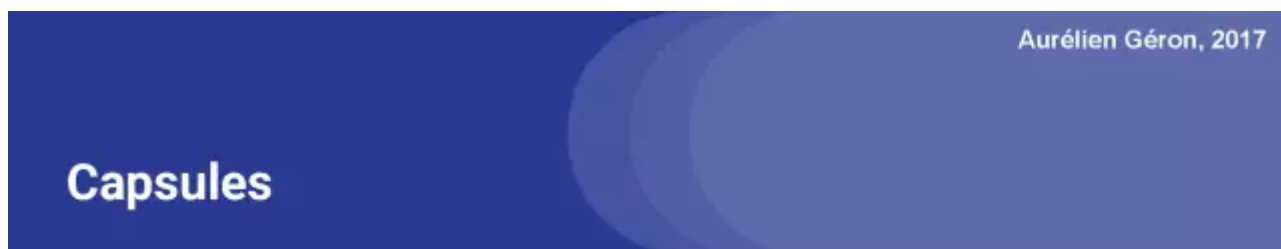
在计算机图形学中，你表达一个场景都是从抽象的表示开始。

例如，位置 $x = 20$ 和 $y = 30$ 的矩形，旋转 16° ，等等。每个对象类型都有不同的实例化参数。然后你调用一些渲染函数，然后你得到一个图像。



逆向图形，只是上面抽象表示的一个逆向过程。你从一个图像开始，你试着找出它包含的对象，以及它们的实例化参数是什么。

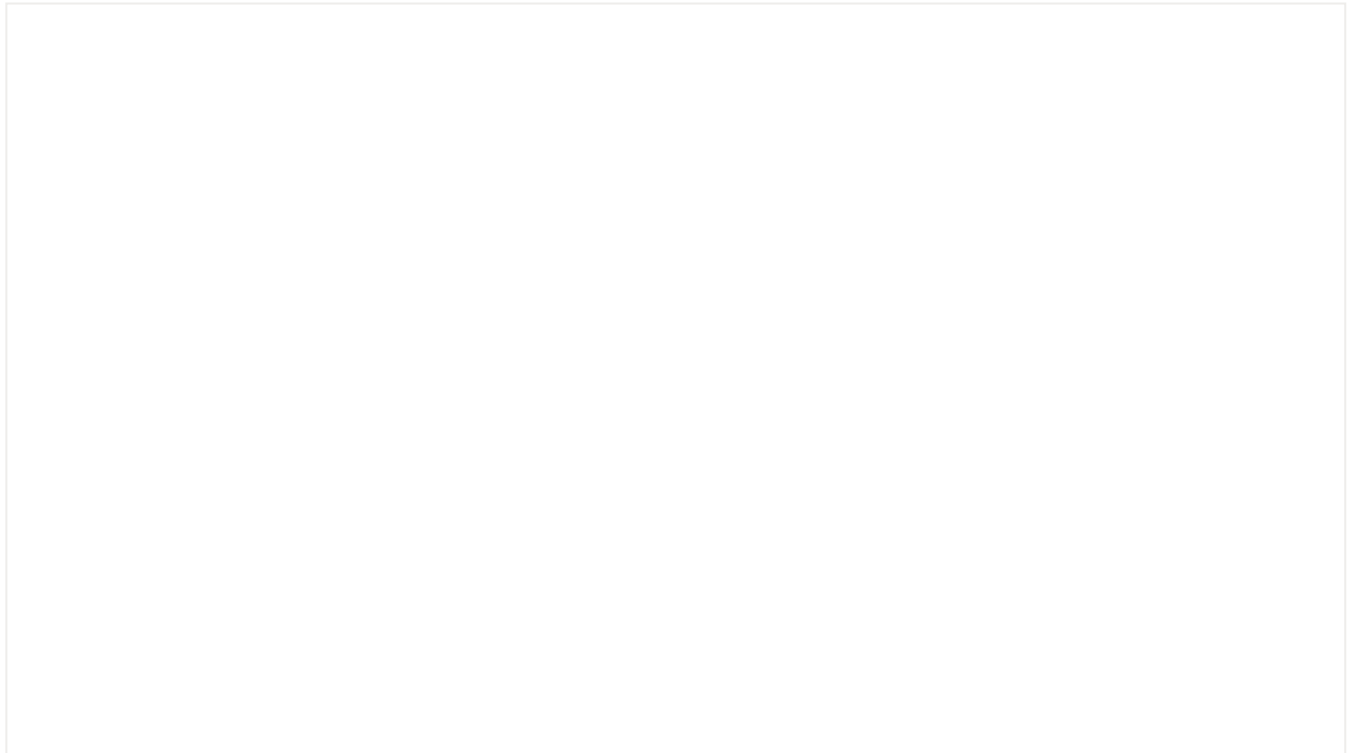
一个胶囊网络基本上是一个试图执行反向图形解析的神经网络。



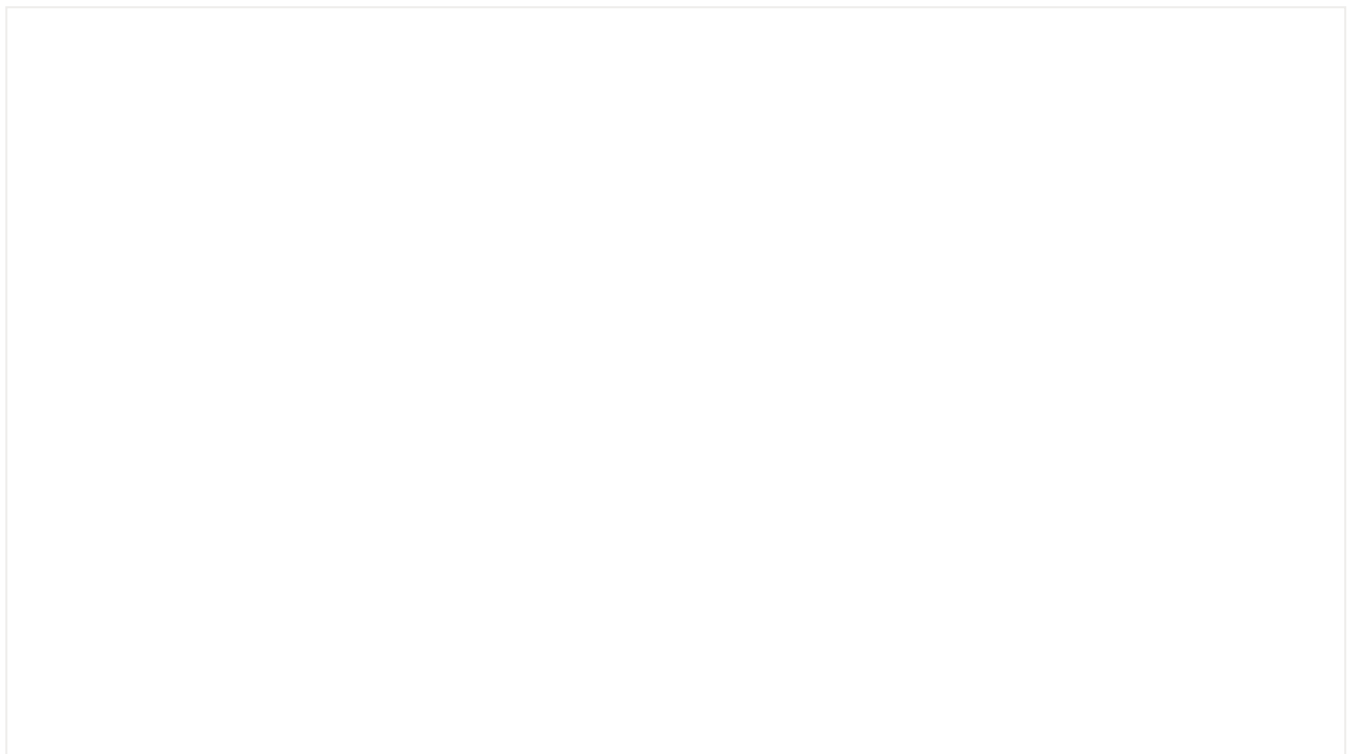
它由许多胶囊组成。一个胶囊是一个函数，它试图在给特定位置的目标预测它的存在性以及实例化参数。

例如，上面的网络包含50个胶囊。箭头表示这些胶囊的输出向量。胶囊输出许多向量。黑色箭头对应于试图找到矩形的胶囊，而蓝色箭头则表示胶囊寻找三角形的输出。激活向量的长度表示胶囊正在查找的物体确实存在的估计概率。

你可以看到大多数箭头很小，这意味着胶囊没有检测到任何东西，但是两个箭头相当长。这意味着在这些位置的胶囊非常有自信能够找到他们要寻找的东西，在这个情况下是矩形和三角形。



接下来，激活向量的方向编码对象的实例化参数，例如在这个情况下，对象的旋转，但也可能是它的厚度，它是如何拉伸或倾斜的，它的确切位置（可能有轻微的翻转），等等。为了简单起见，我只关注旋转参数，但在真实的胶囊网络中，激活向量可能有5, 10个维度或更多。




实际上，实现这一点的一个好方法是首先应用一对卷积层，就像在常规的卷积神经网络中一样。这将输出一个包含一堆特征映射的数组。然后你可以重塑这个数组来获得每个位置的一组向量。

例如，假设卷积图层输出一个包含18个特征图（ 2×9 ）的数组，则可以轻松地重新组合这个数组以获得每个位置9个维度的2个向量。你也可以得到3个6维的向量，等等。



这看起来像在这里在每个位置用两个向量表示的胶囊网络。最后一步是确保没有向量长度大于1，因为向量的长度意味着代表一个概率，它不能大于1。

为此，我们应用一个squashing（压扁）函数。它保留了矢量的方向，但将它压扁，以确保它的长度在0到1之间。



胶囊网络的一个关键特性是在网络中保存关于物体位置和姿态的详细信息。例如，如果我稍微旋转一下图像，注意激活向量也会稍微改变，对吧？这叫做equivariance。

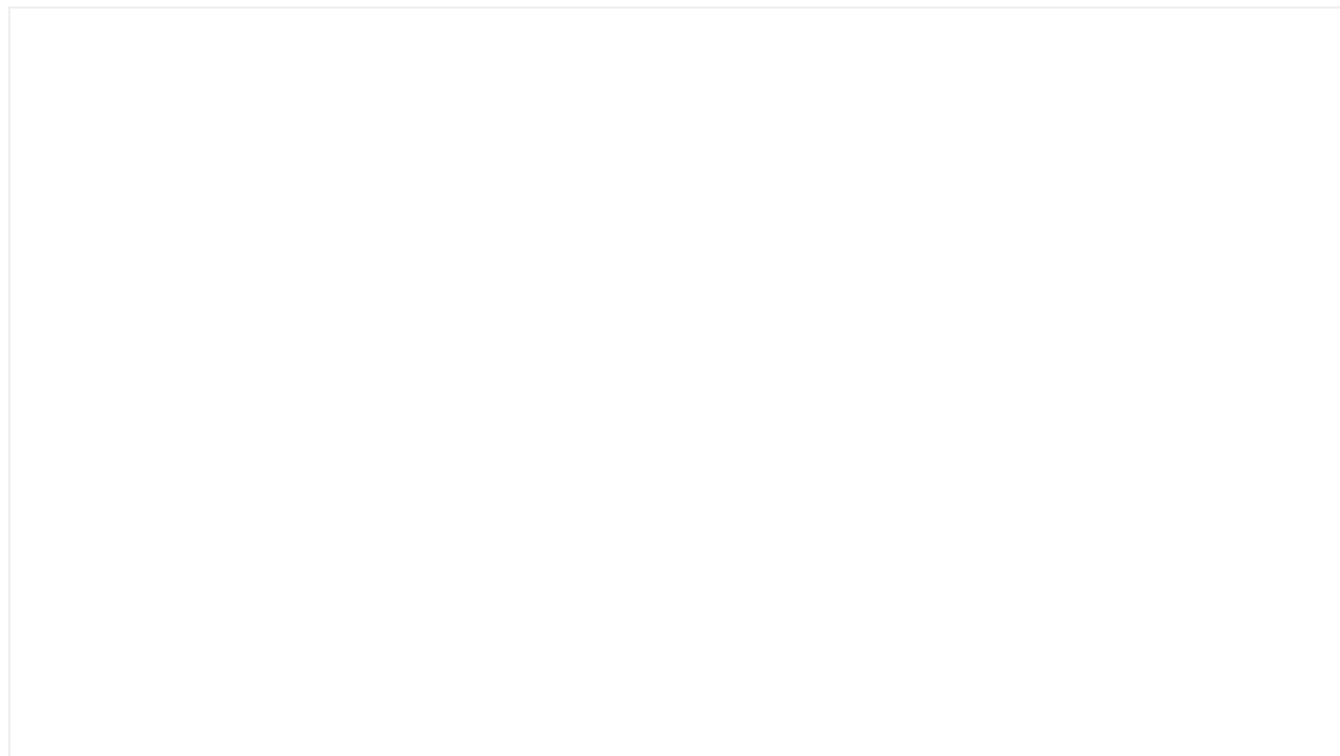
在常规的卷积神经网络中，通常会有多个汇聚层，不幸的是，这些汇聚层的操作往往会丢失很多信息，比如目标对象的准确位置和姿态。如果你只是想要对整个图像进行分类，就算丢失这些信息也没什么大不了的，但是这些丢失的信息对你进行精确的图像分割或对象检测(这需要精确的位置和姿势)等任务是非常重要的。

胶囊的equivariance等变特性使得它在这些任务上都有非常有前景。



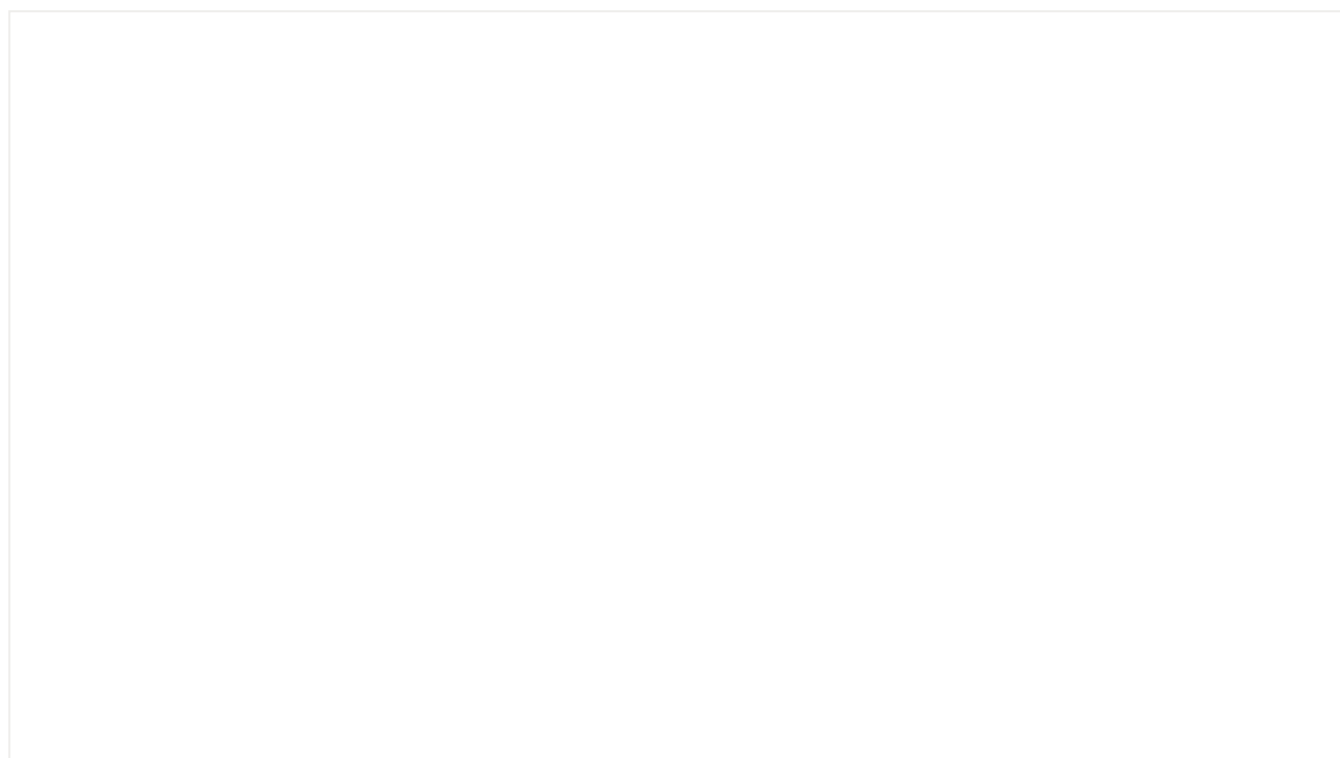
好了，现在让我们来看看胶囊网络如何处理由层次结构组成的对象。

例如，考虑一个船，它的位置为 $x = 22$ ， $y = 28$ ，旋转 16° 。

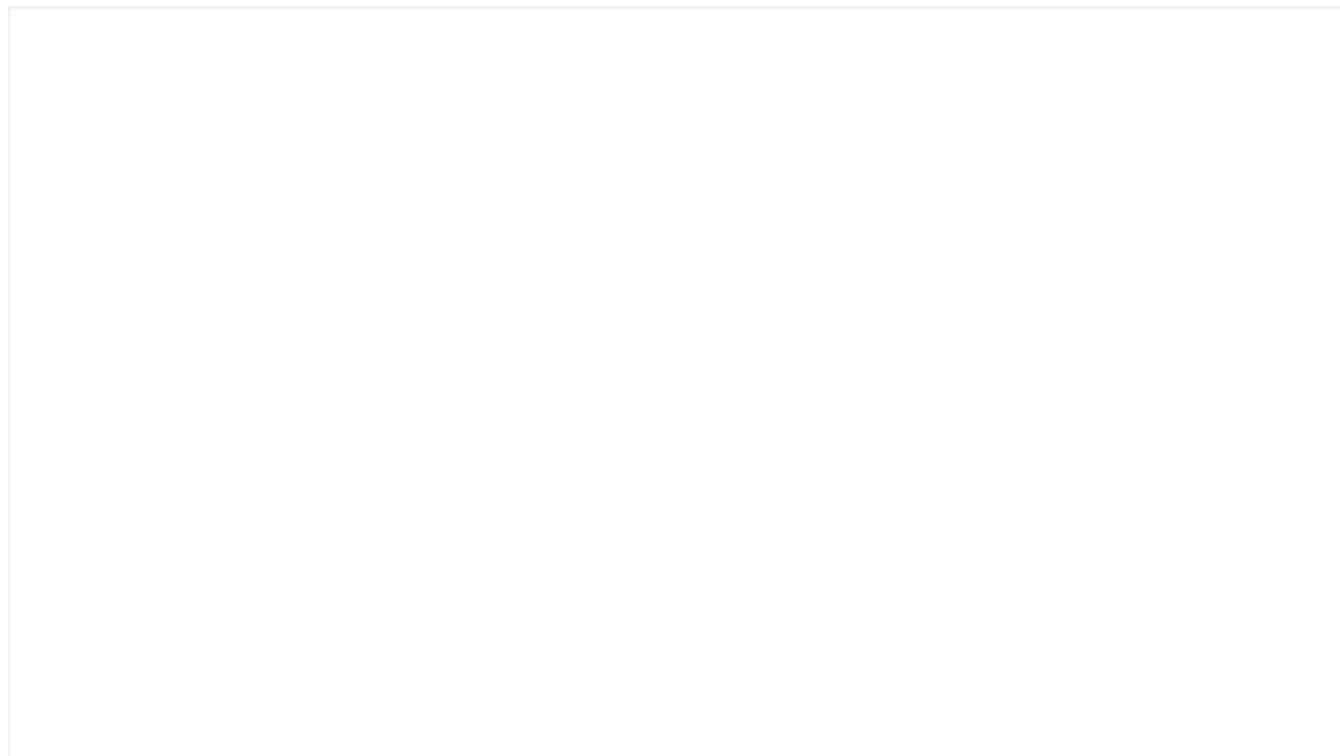


这艘船是由几个部件组成的。在当前演示的情况下，也就是说船由一个矩形和一个三角形组成。

现在我们要做相反的事情，我们需要逆向图形，所以我们想要从图像到这个完整的层次结构的部件和它们的实例化参数。



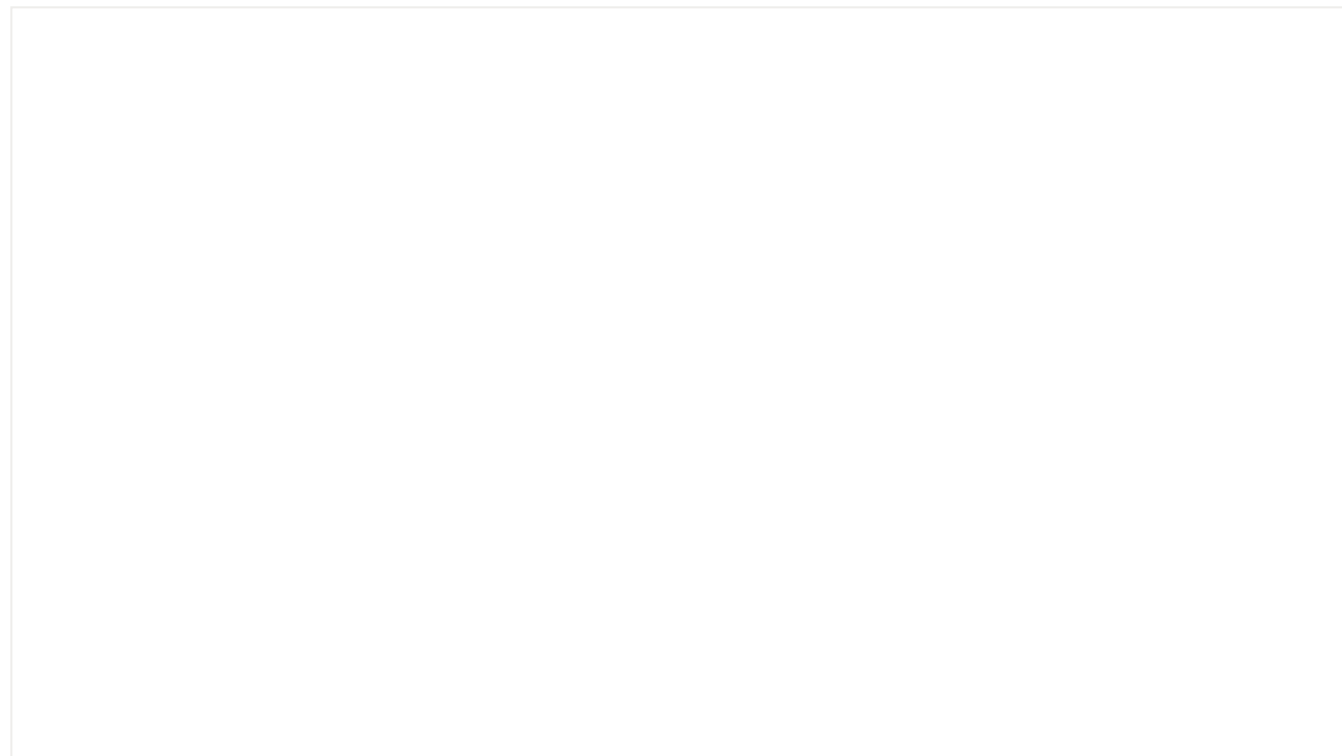
同样，我们也可以绘制一个房子，使用相同的部分，一个矩形和一个三角形，但这次以不同的方式组织。



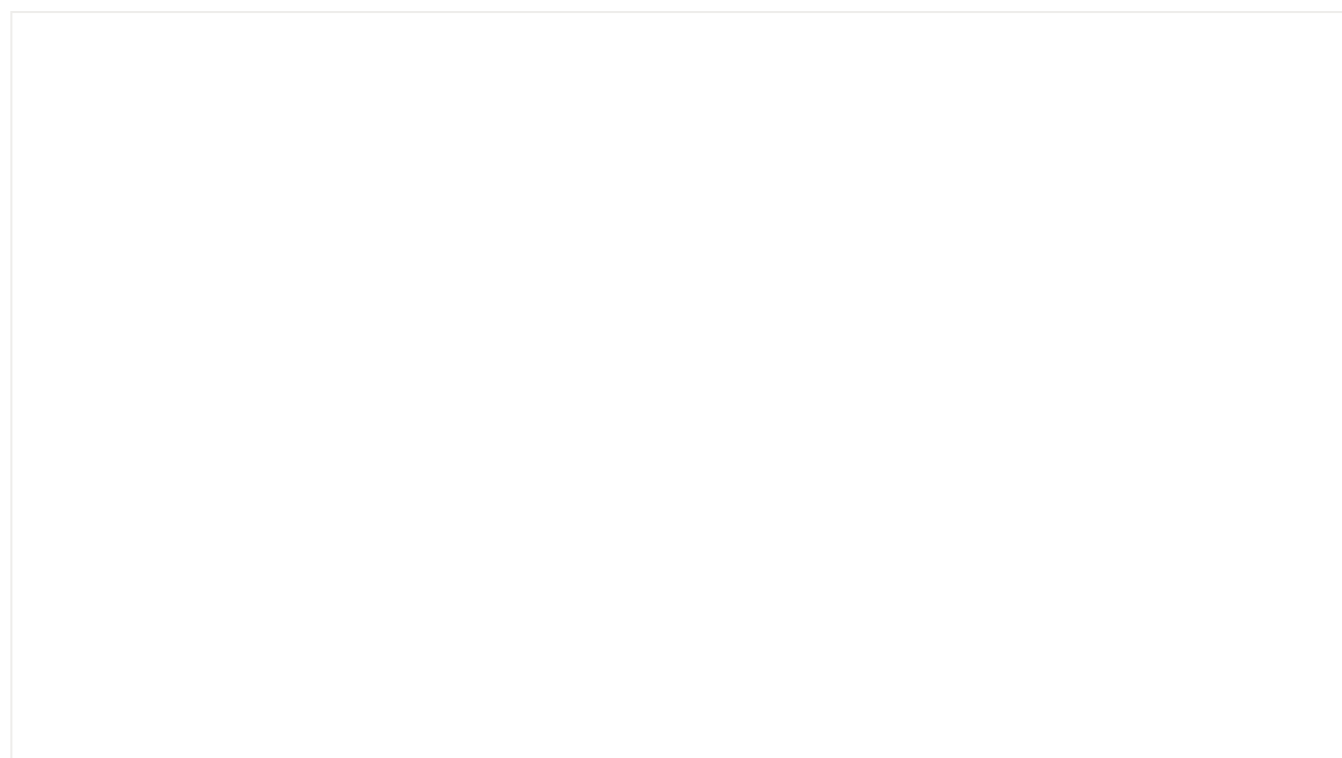
所以关键是要试着从这个包含一个矩形和一个三角形的图像，找出这个位置和这个方向，并且说明它们是船的一部分，而不是房子。来让我们弄清楚它将如何做到这一点。第一步（之前我们已经看到过）：我们运行一对卷积层，我们将输出重构以得到向量，然后将它们归一化。这就得到了主胶囊的输出。

我们已经有第一层了。下一步是则是展示胶囊网络的魔力和复杂性的一步了。第一层中的每个胶囊试图预测下一层中每个胶囊的输出。你可能想停下来想一想这意味着什么。

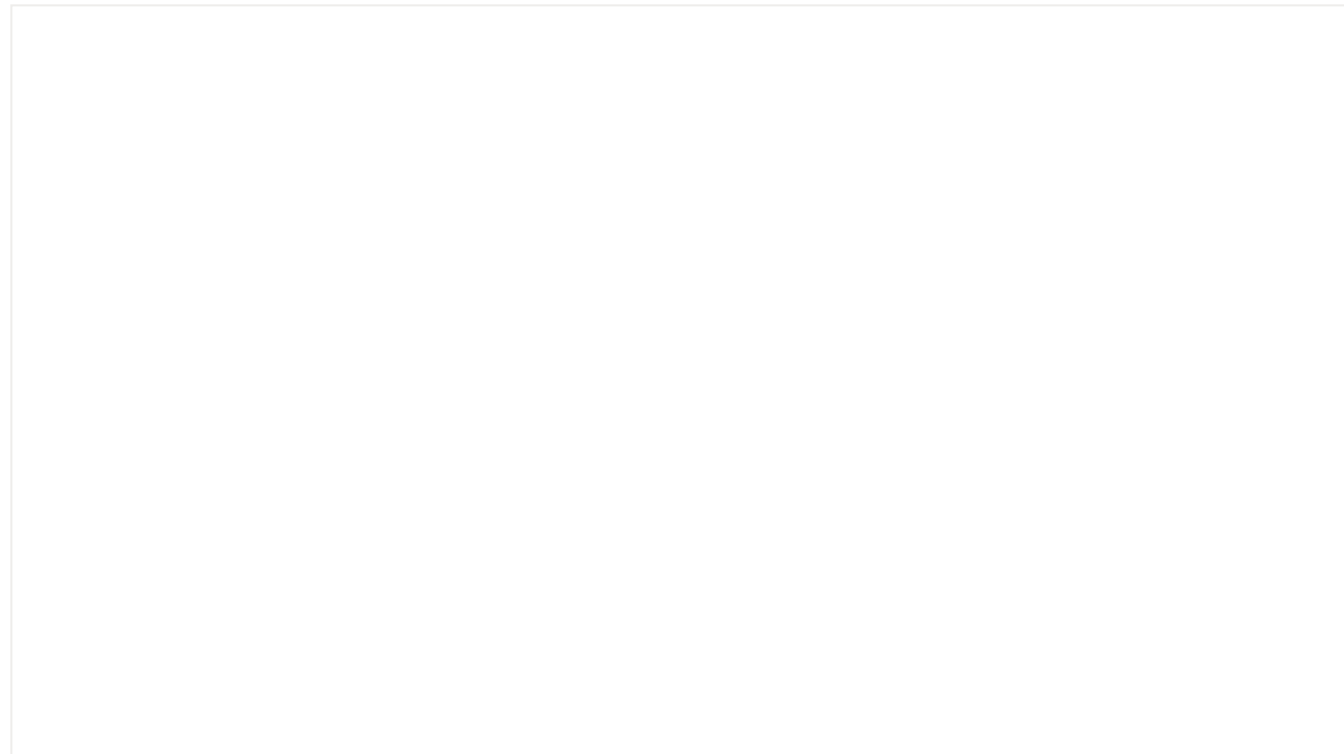
第一层胶囊试图预测第二层胶囊将输出什么。



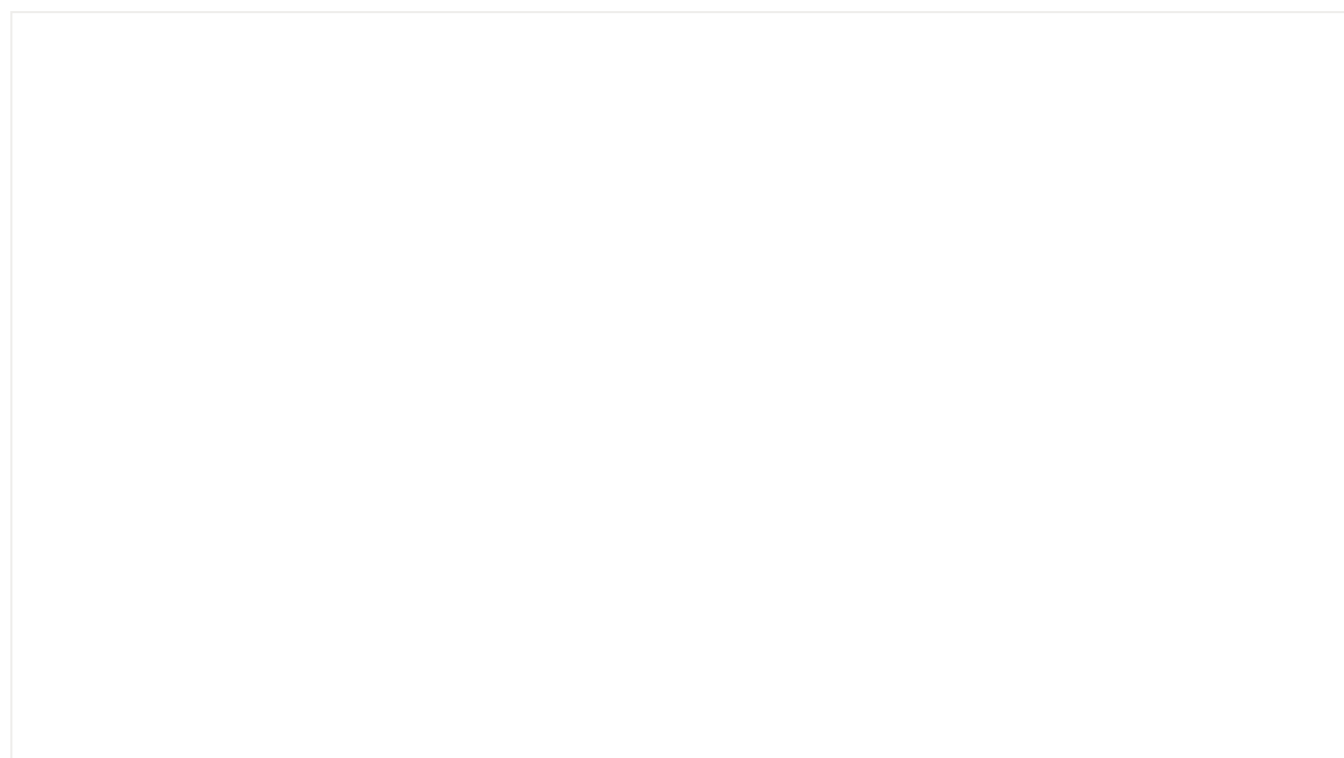
例如，让我们考虑检测矩形的胶囊。我会称之为矩形胶囊。



假设下一层只有两个胶囊，房子胶囊和船胶囊。由于矩形胶囊检测到一个旋转了 16° 的矩形，所以房子胶囊将检测到一个旋转了 16° 的房子，这是有道理的，船胶囊也会检测到旋转了 16° 的船。这与矩形的方向是一致的。



所以，为了做出这个预测，矩形胶囊所做的就是简单地计算一个变换矩阵 $W_{i,j}$ 与它自己的激活向量 u_i 的点积。在训练期间，网络将逐渐学习第一层和第二层中的每对胶囊的变换矩阵。换句话说，它将学习所有的部分 - 整体关系，例如墙和屋顶之间的角度，等等。



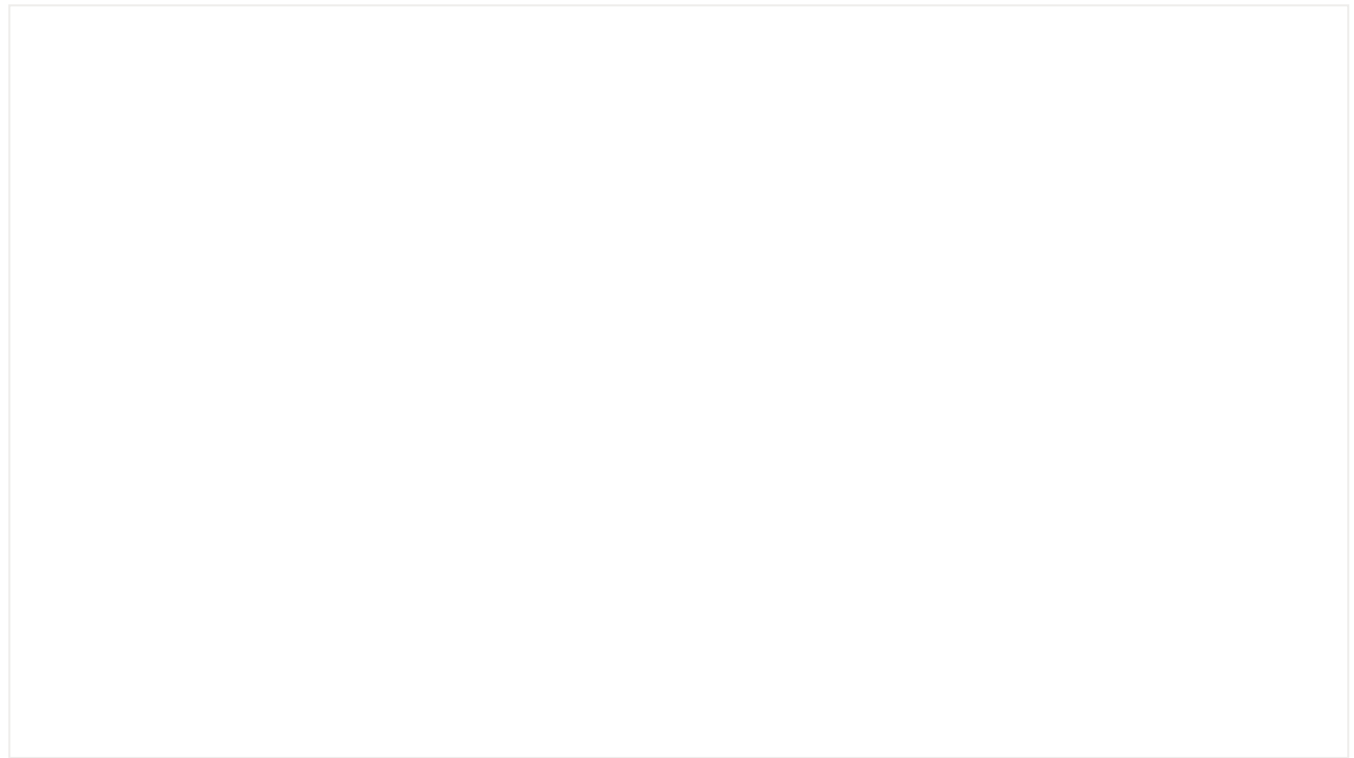
现在让我们看看三角形的胶囊是什么。



这一次，它更有趣了：给定三角形的旋转角度，它预测房子的胶囊会检测到一个倒置的房子，并且船胶囊会探测到一艘船旋转16°。这些位置与三角形的旋转角度是一致的。

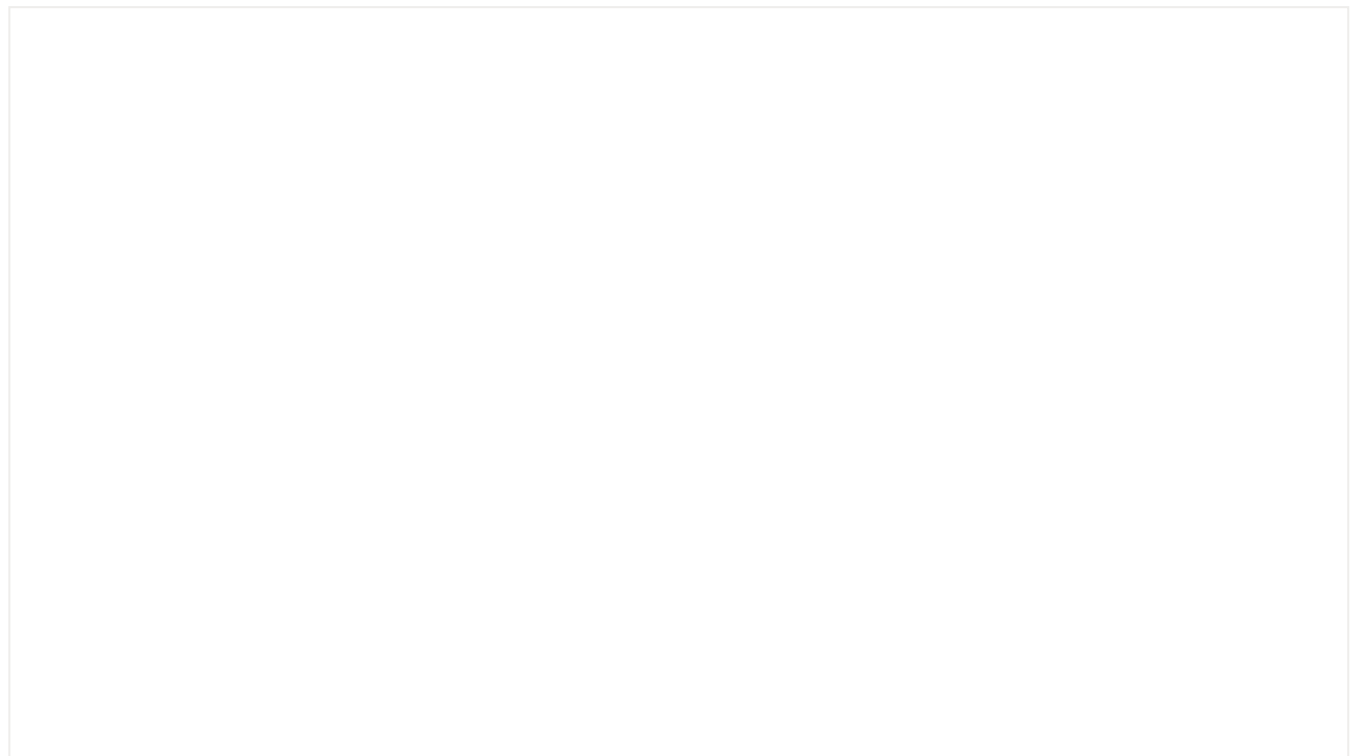


现在我们有一堆预测输出，也就是图上的四个，我们下一步用它们做什么呢？



正如你所看到的，矩形胶囊和三角胶囊在船胶囊的输出方面有着强烈的一致性。
换言之，矩形胶囊和三角胶囊都同意船会以什么样的形式输出来。

然而，矩形胶囊和三角胶囊他们俩完全不同意房子胶囊会产出什么，从图中可以看出房子的输出方向是一上一下的。。



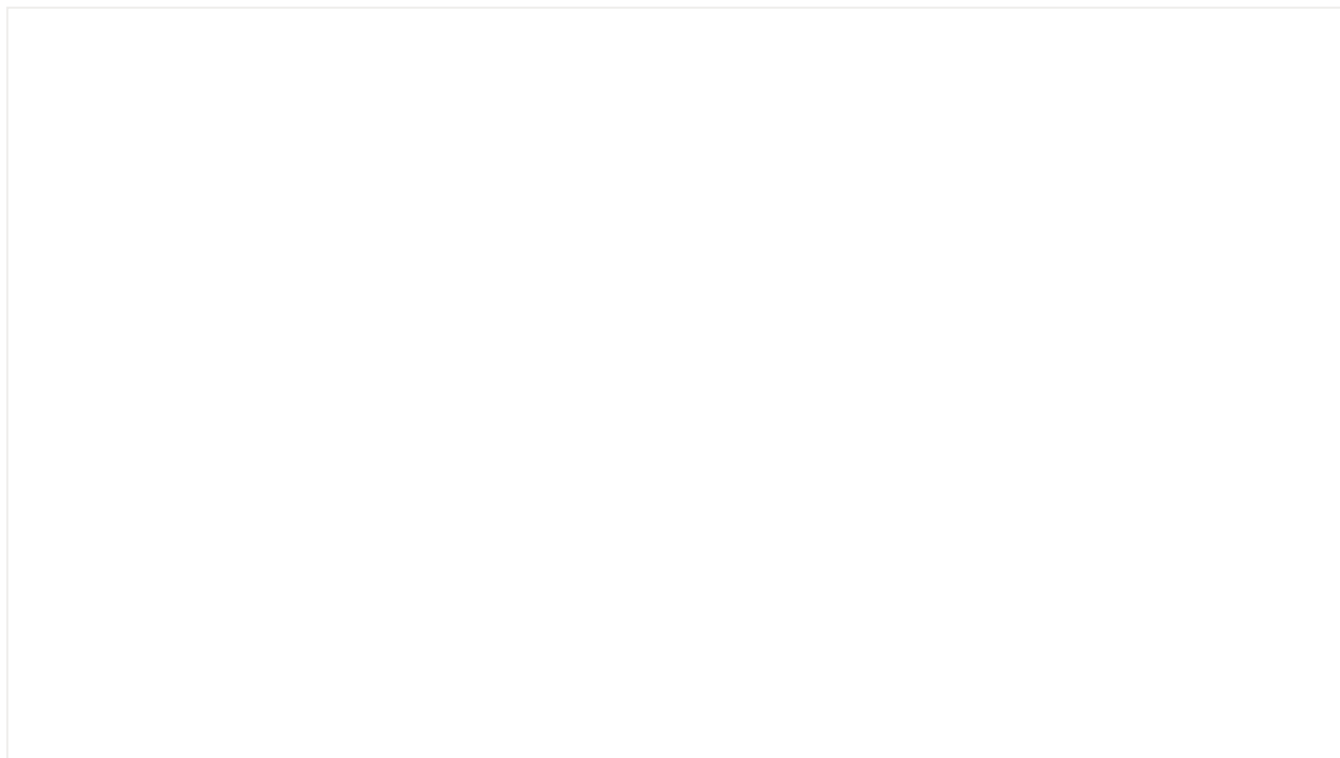
因此，可以很合理的假设矩形和三角形是船的一部分，而不是房子的一部分。
既然我们知道矩形和三角形是船的一部分，矩形胶囊和三角胶囊的输出也就是真的只关注船胶囊，所以就没有必要发送这些输出到任何其他胶囊，这样只会增加噪音。这叫做同意协议路由。它有几个好处：

首先，由于一个胶囊的输出仅路由到下一层的想对应的胶囊中，所以下一层的这些胶囊将得到更清晰的输入信号，同时也更能准确地确定物体的姿态。

第二，通过查看激活的路径，您可以轻松地查看部件的层次结构，并确切地知道哪个部分属于哪个对象（如矩形属于小船或者三角形属于船等等）。

最后，按同意协议路由帮助解析那些有重叠对象的拥挤场景（我们将在几个幻灯片中看到）。

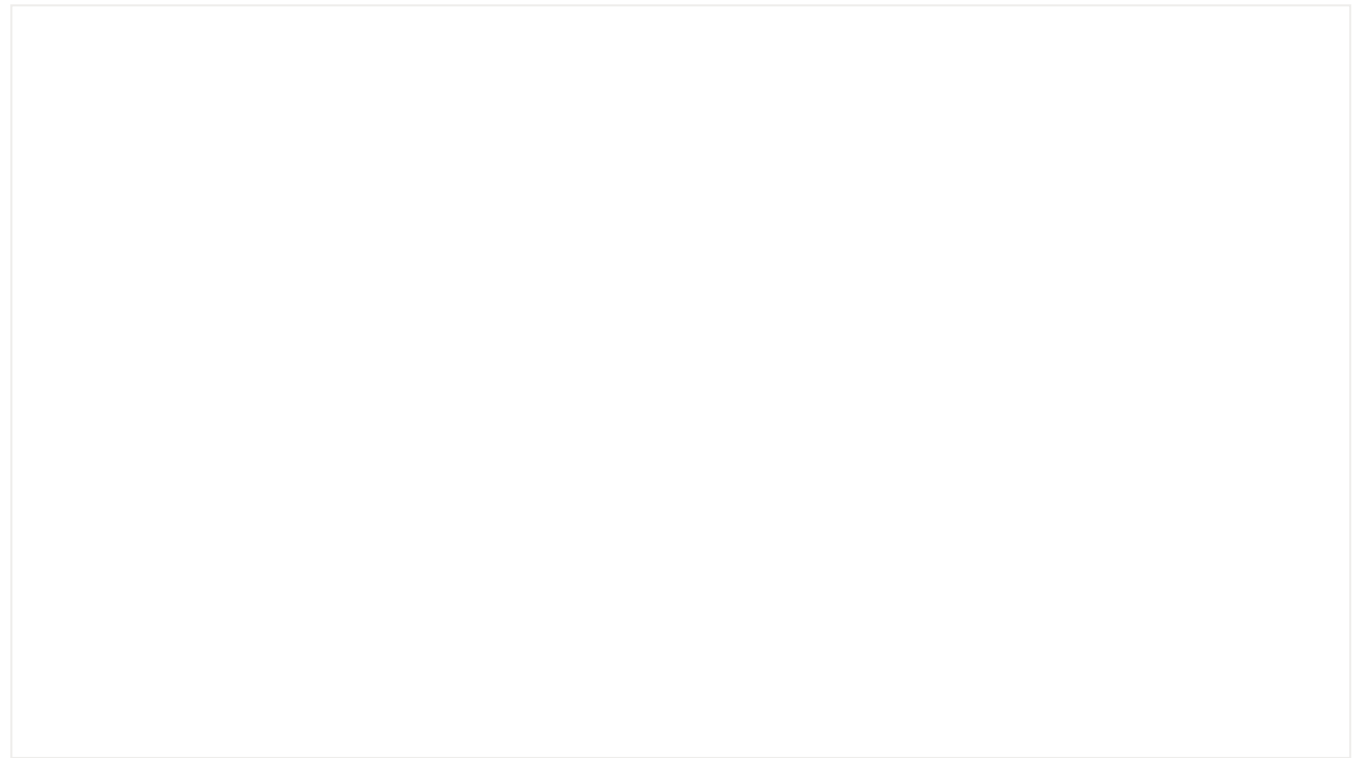
但是首先，让我们看看协议是如何在胶囊网络中实现的。



在这里，我把船的各种姿态都表示出来，正如低层次的胶囊可能会预测的那样。

例如，这些圆圈中的一个可能代表矩形胶囊对船的最可能姿势的看法，而另一个圆圈可能代表三角胶囊的想法，如果我们假设有许多其他低层的胶囊，然后我们可能就会有大量用于船胶囊的预测向量。

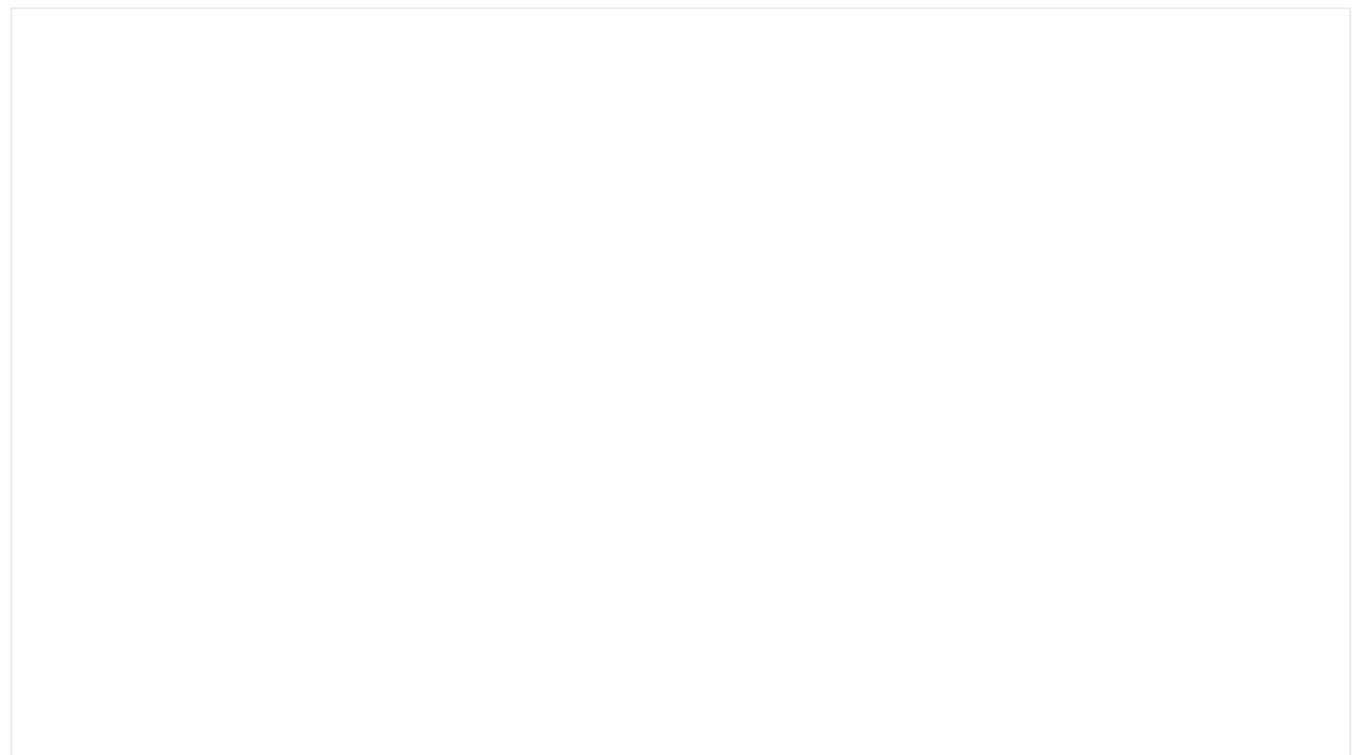
在这个例子中，有两个姿态参数：一个代表旋转角度，另一个代表船的大小。正如我前面提到的，姿态参数可以捕获许多不同类型的视觉特征，如倾斜、厚度或精确定位。



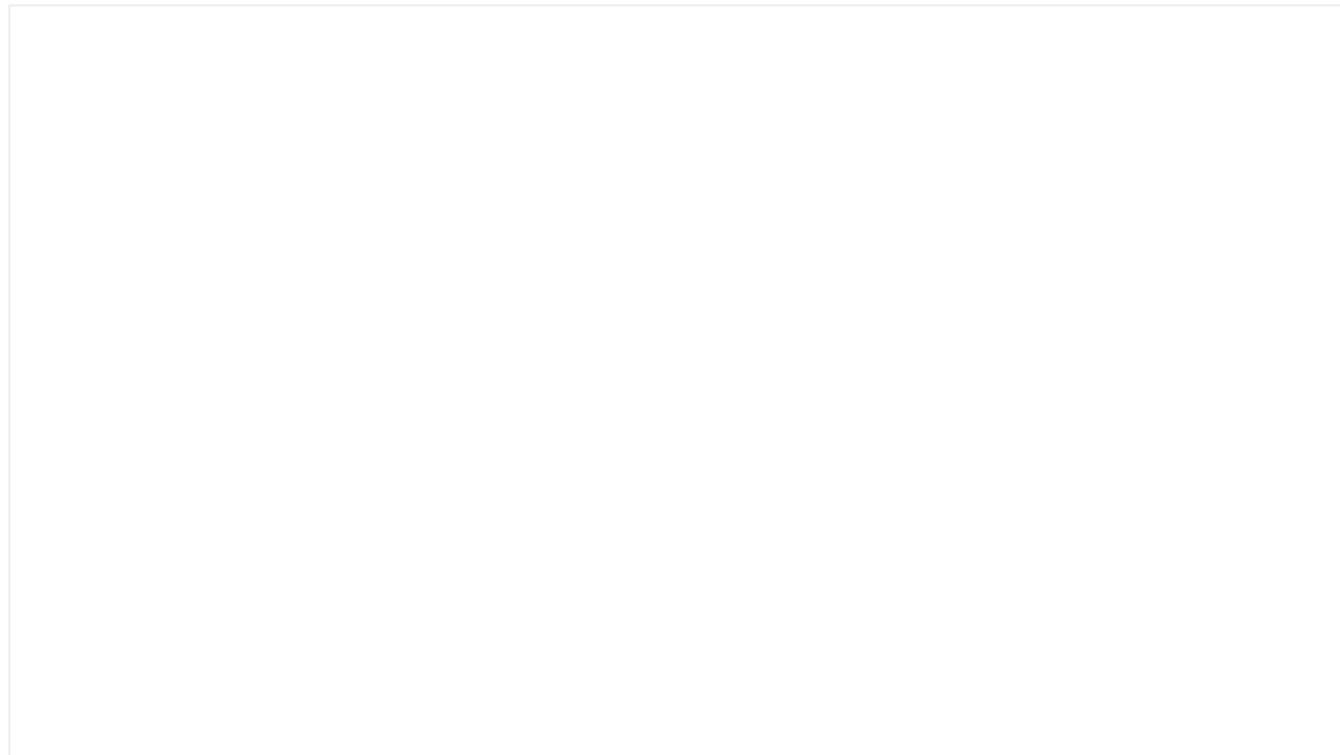
所以我们做的第一件事，就是计算所有这些预测的平均值。然后我们就得到了一个平均向量。下一步是度量每个预测向量与平均向量之间的距离。我在这里会用欧氏距离做演示，但胶囊网络实际使用点积。

随后我们要测量每个预测向量与平均预测向量的一致性程度。

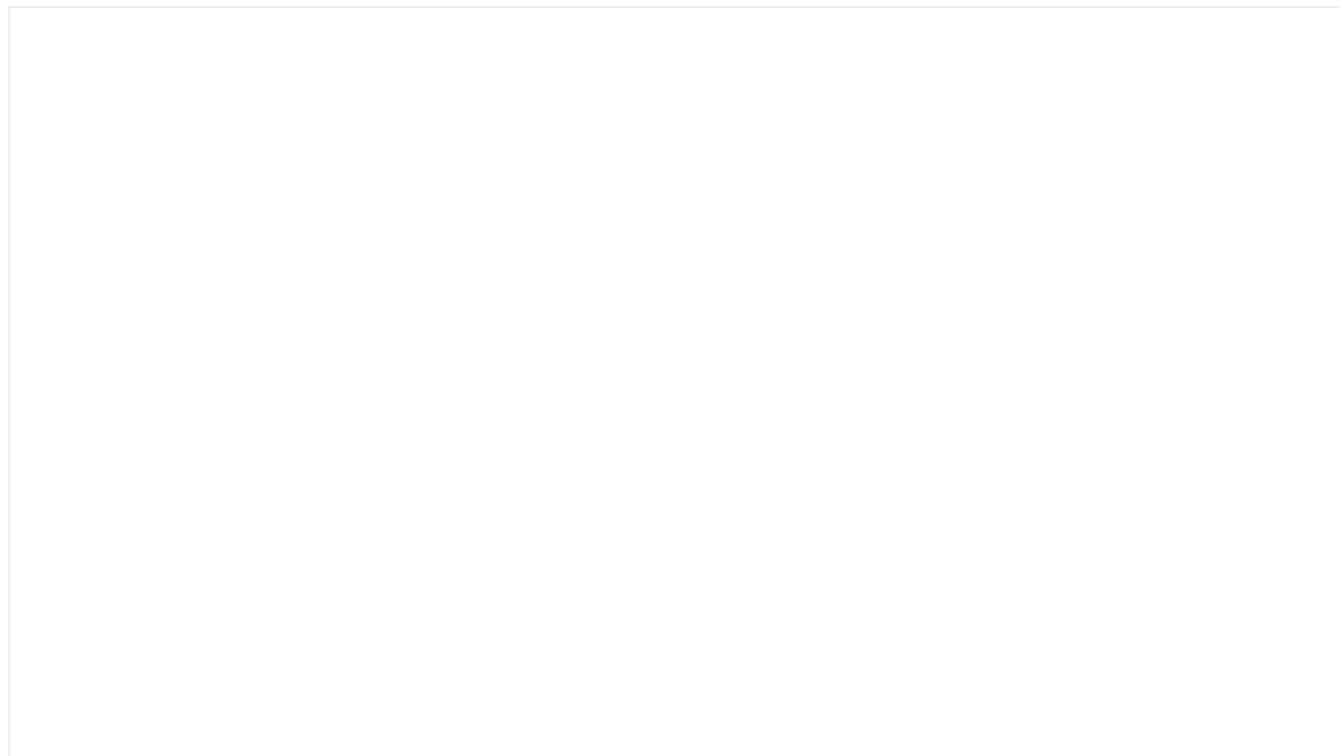
利用这个计算出的一致性程度值，我们可以相应地更新每个预测向量的权重。



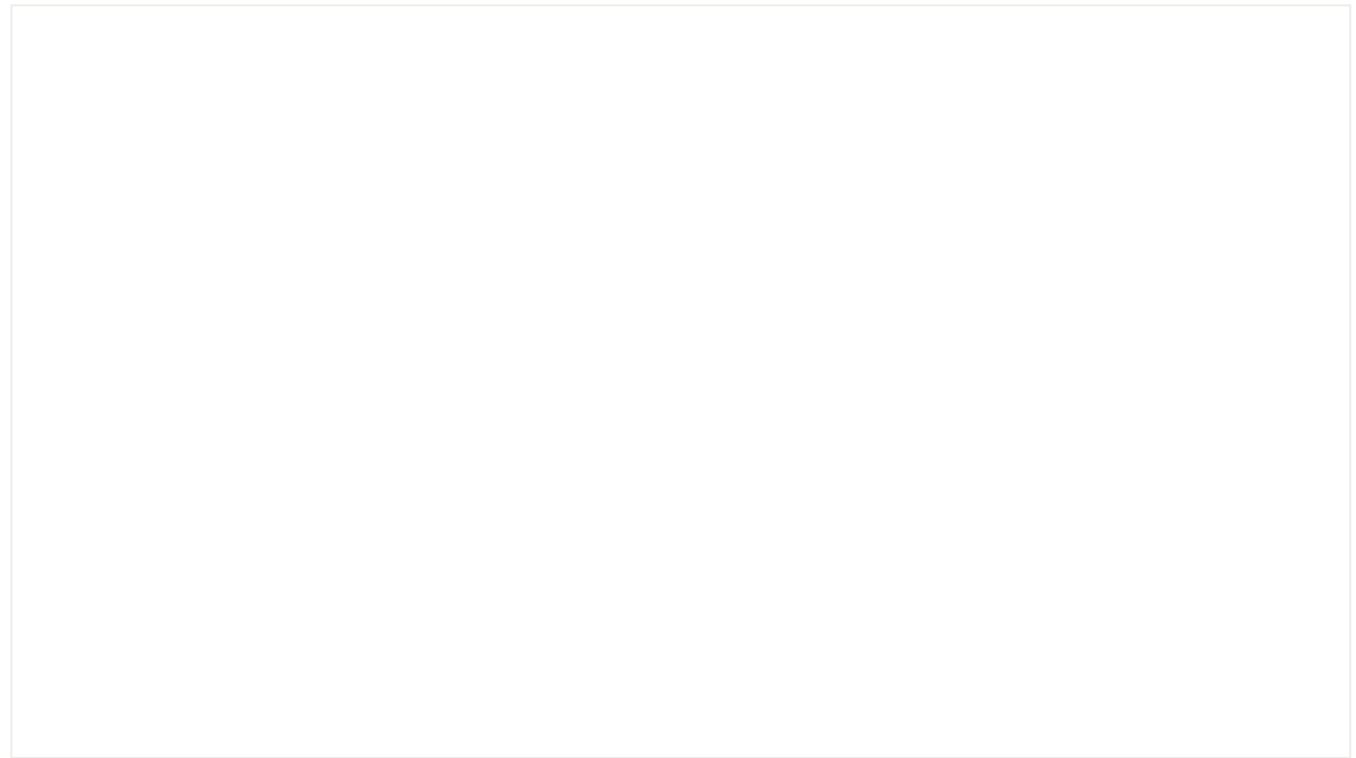
注意，远离平均值的预测向量现在有一个非常小的重量，在图中可以看到颜色比较浅。而最接近平均值的向量有更大的权重，我们用黑色来代表。



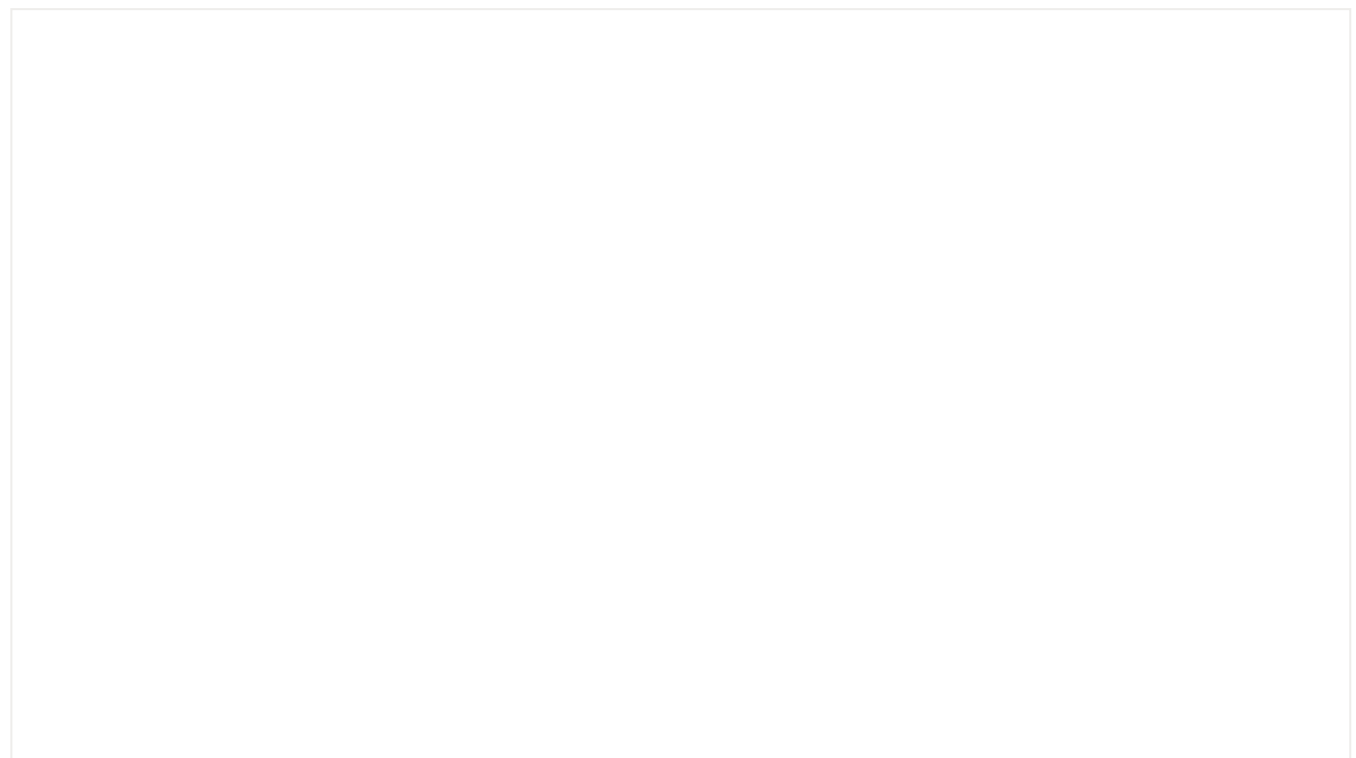
现在我们可以再一次计算均值（或者说，加权平均数），你会注意到跟上图相比它稍微向聚类的中心移动。



接下来，我们可以再次更新权重，现在聚类中的大部分向量变黑了，



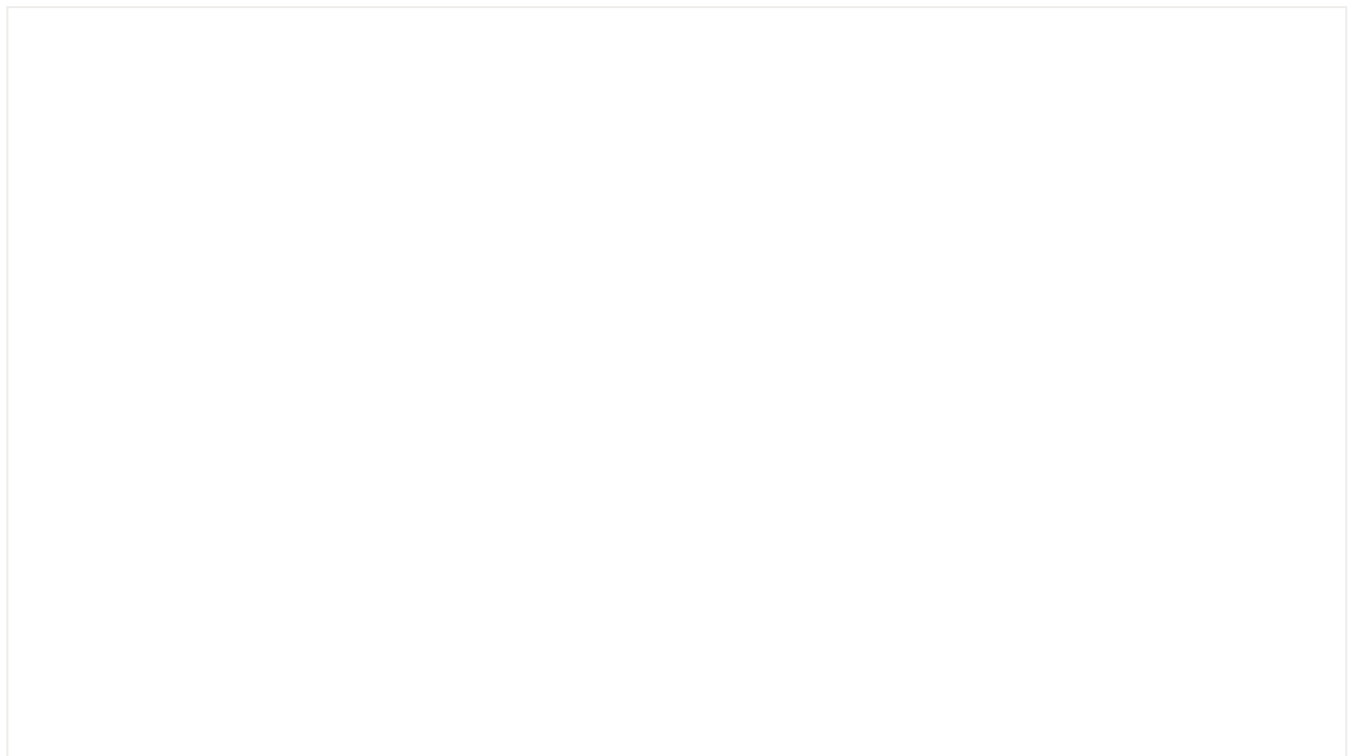
我们可以再次更新平均值。我们可以重复这个过程几次，在实践中，3到5次迭代通常是足够的。我想这可能提醒你，如果你知道k-均值聚类算法的话，就很容易明白这是我们如何找到这个所有向量都任何的聚类的。现在让我们看看整个算法在细节方面的工作原理。



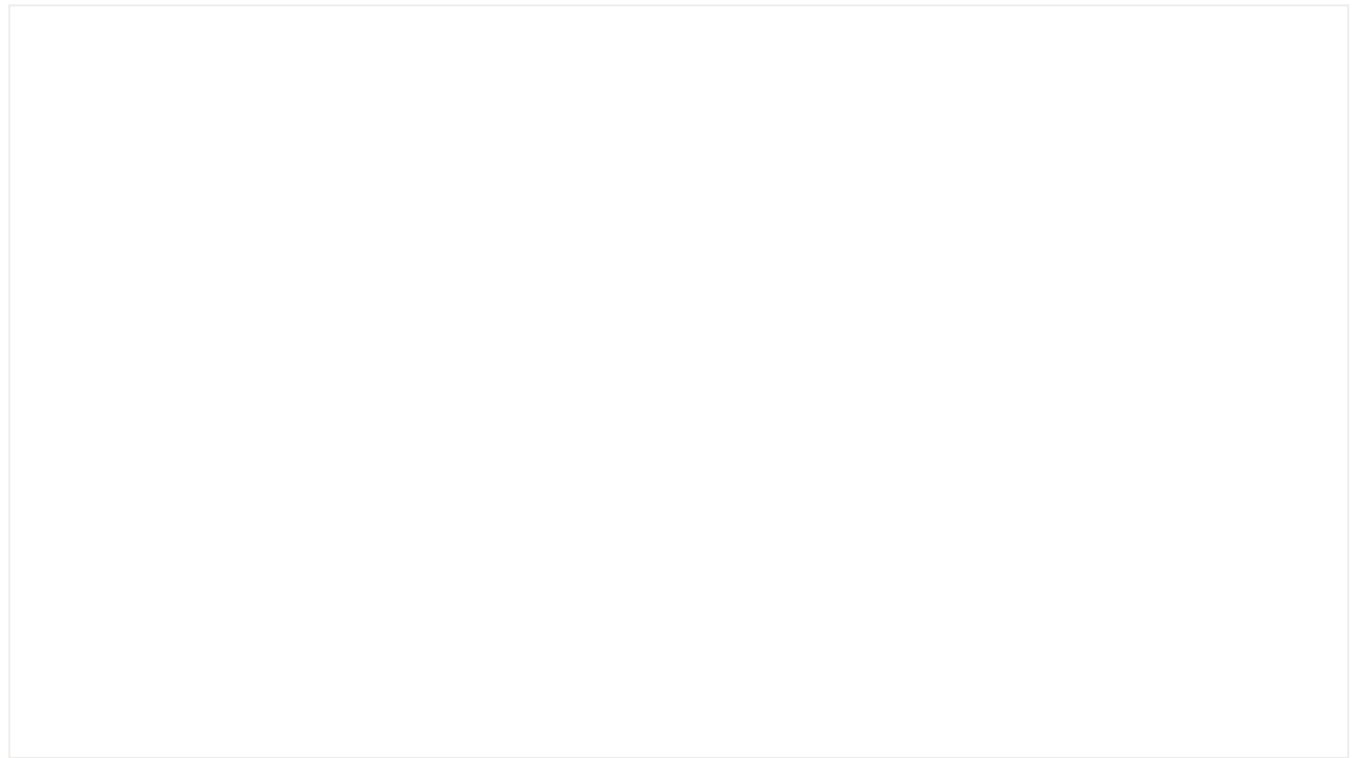
首先，对于每个预测的输出，我们首先设置原始路由权重 b_{ij} 等于0。



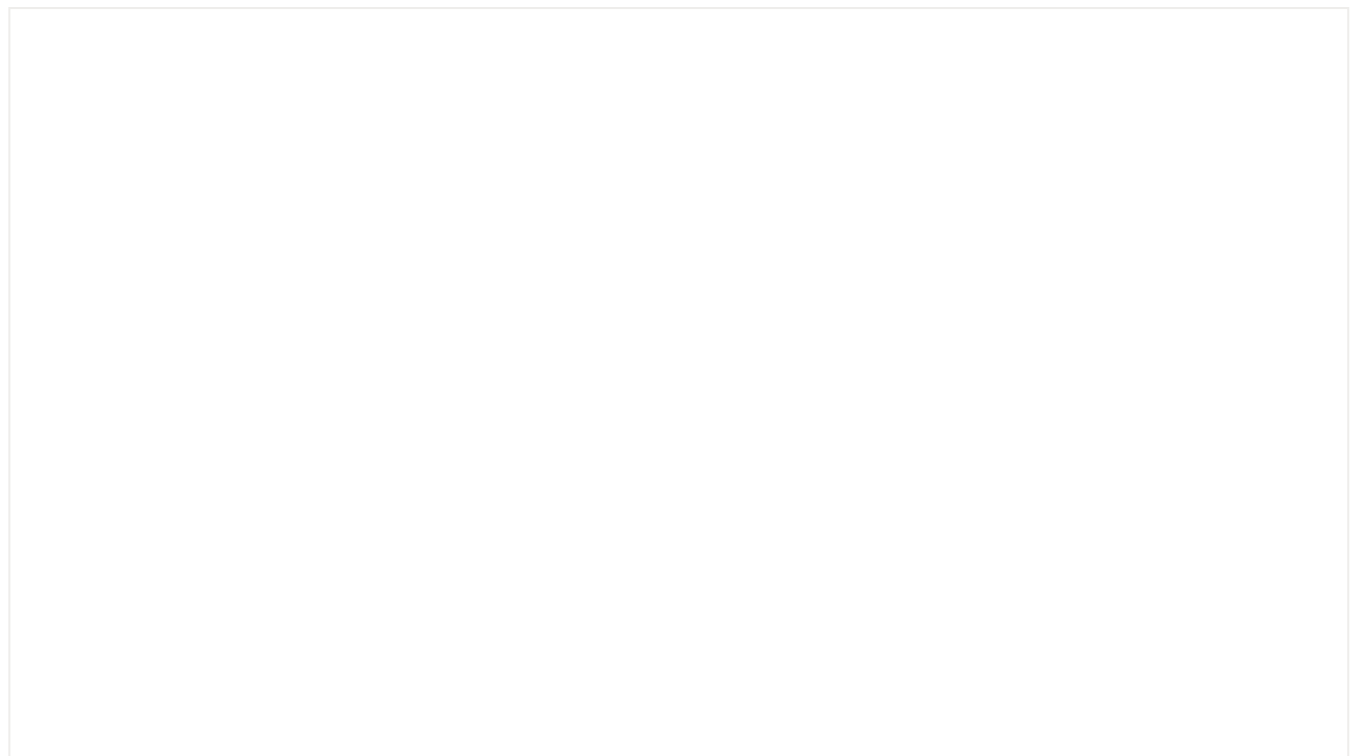
接下来对于每个基本的胶囊，我们将为应用softmax函数对他们的初始权重进行归一化。这样就得出每个预测输出的实际路由权重，在本例中是0.5。



接下来，我们计算下一层的每个胶囊的预测的加权和。



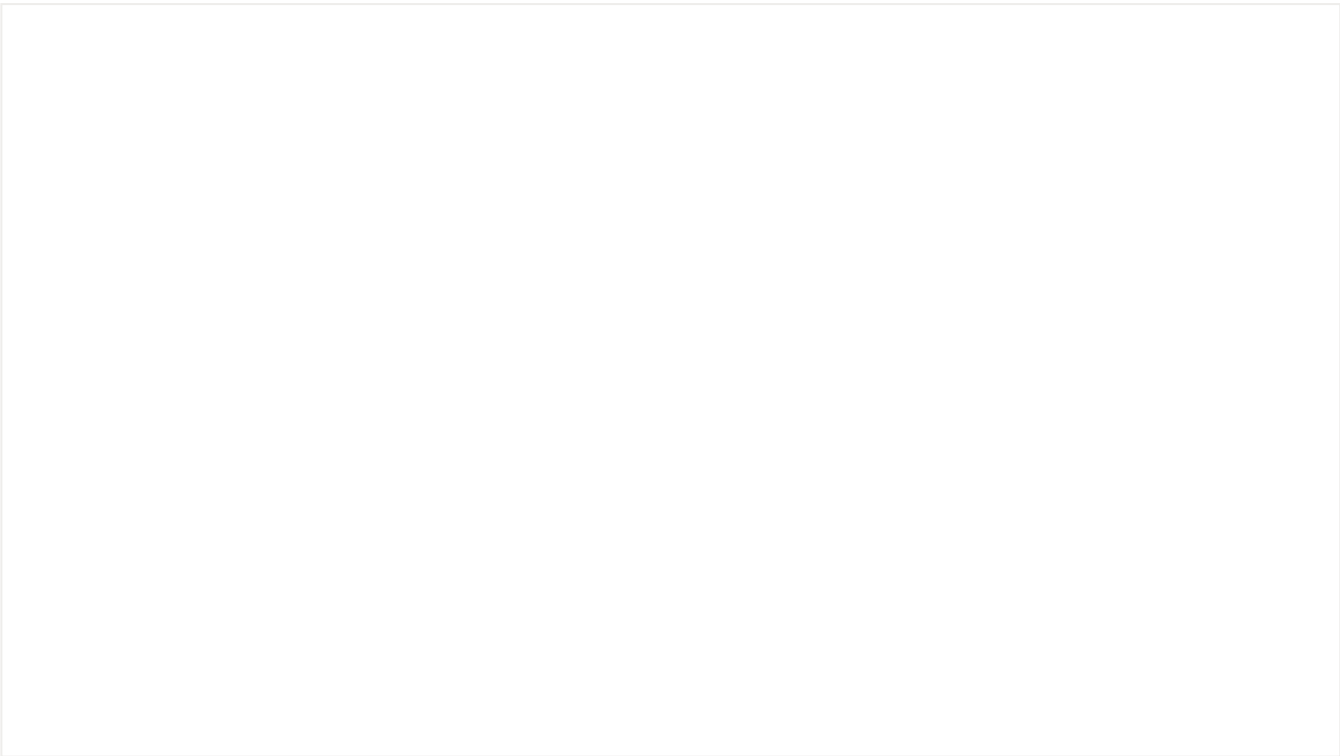
这可能会使向量长于1，所以通常会用到归一化函数。



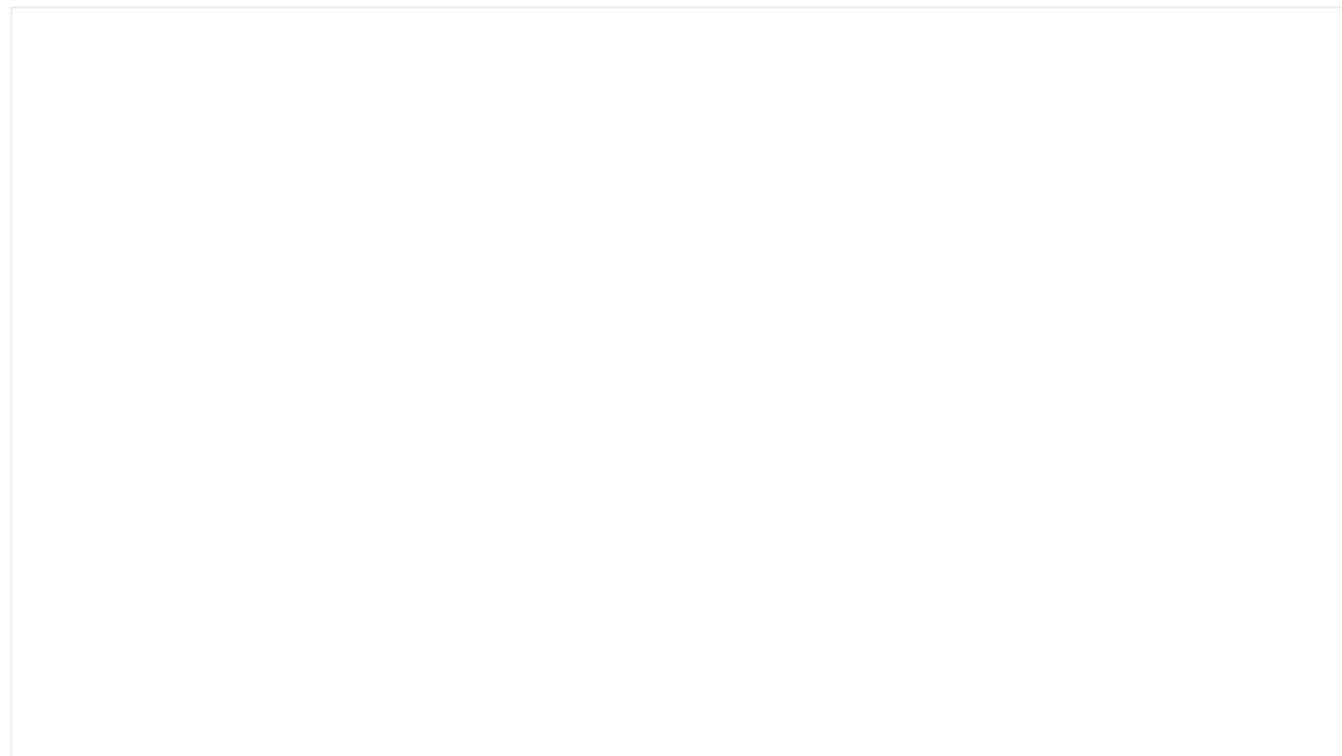
然后，我们现在有了房子胶囊和船舱的实际输出。但这不是最终的输出，这仅仅是第一轮的第一次迭代。



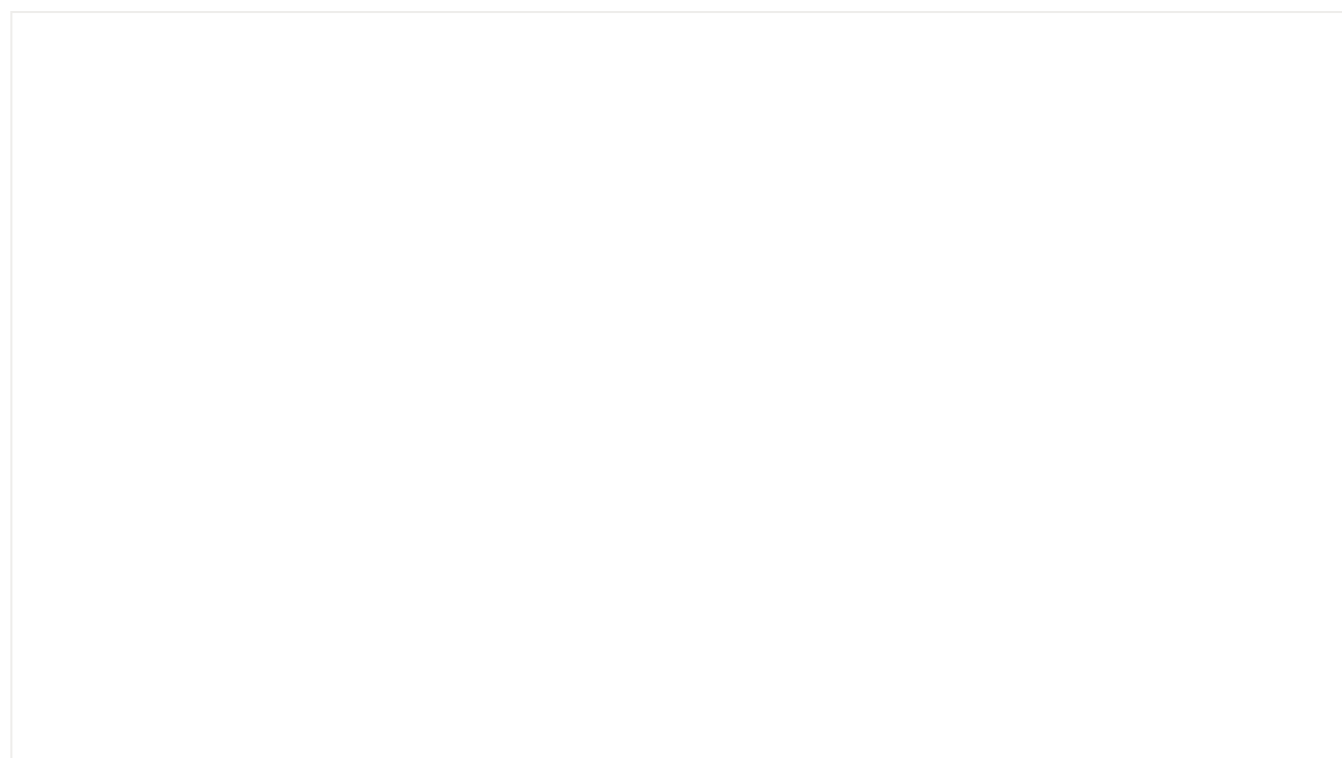
现在我们可以看到哪些预测是最准确的。例如，矩形胶囊对船舱的输出做出了很好的预测。看上去它真的很接近。



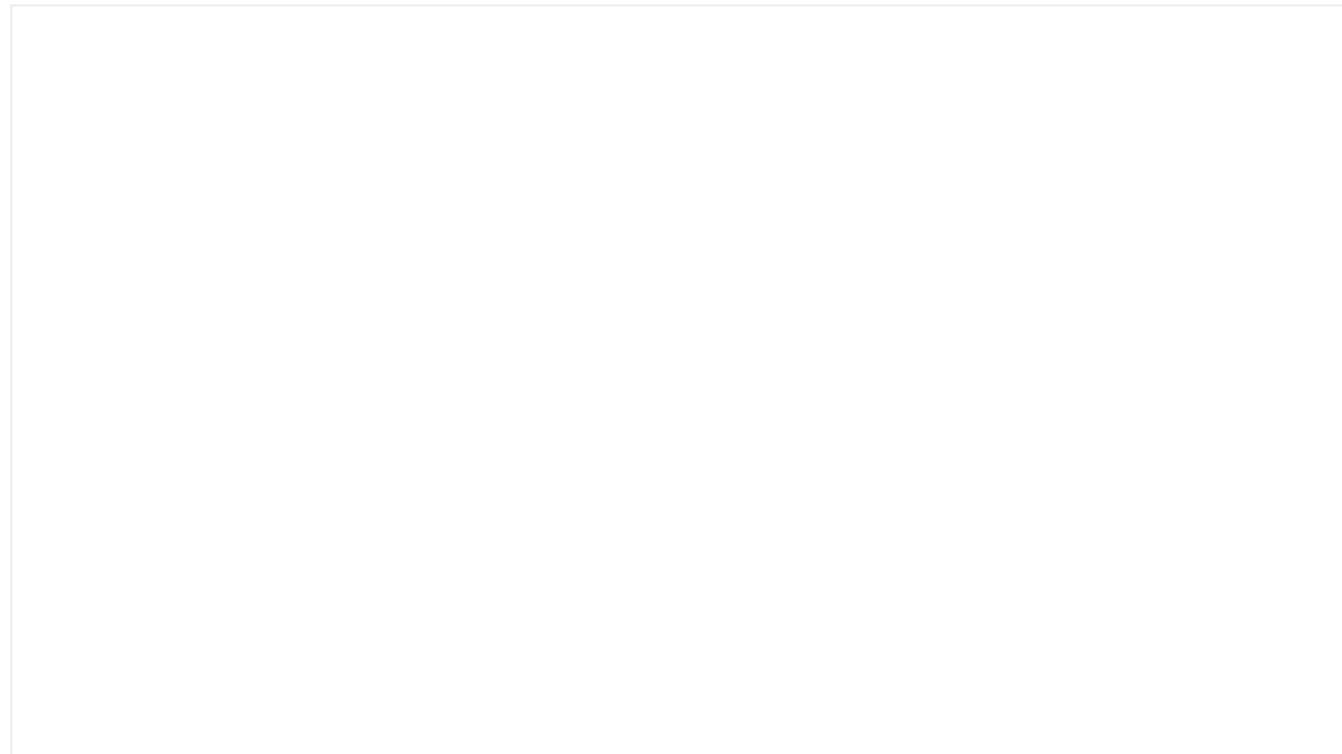
这是通过计算预测输出向量 \hat{u}_j 和实际乘积向量 v_j 的点积来估计的。这个点积操作被简单地添加到预测输出的原始路由权重 b_{ij} 中。所以这个特定的预测输出的权重增加了。



当预测的结果是一个强烈的同意时，这个点积也会很大，所以好的预测将有更高的权重。

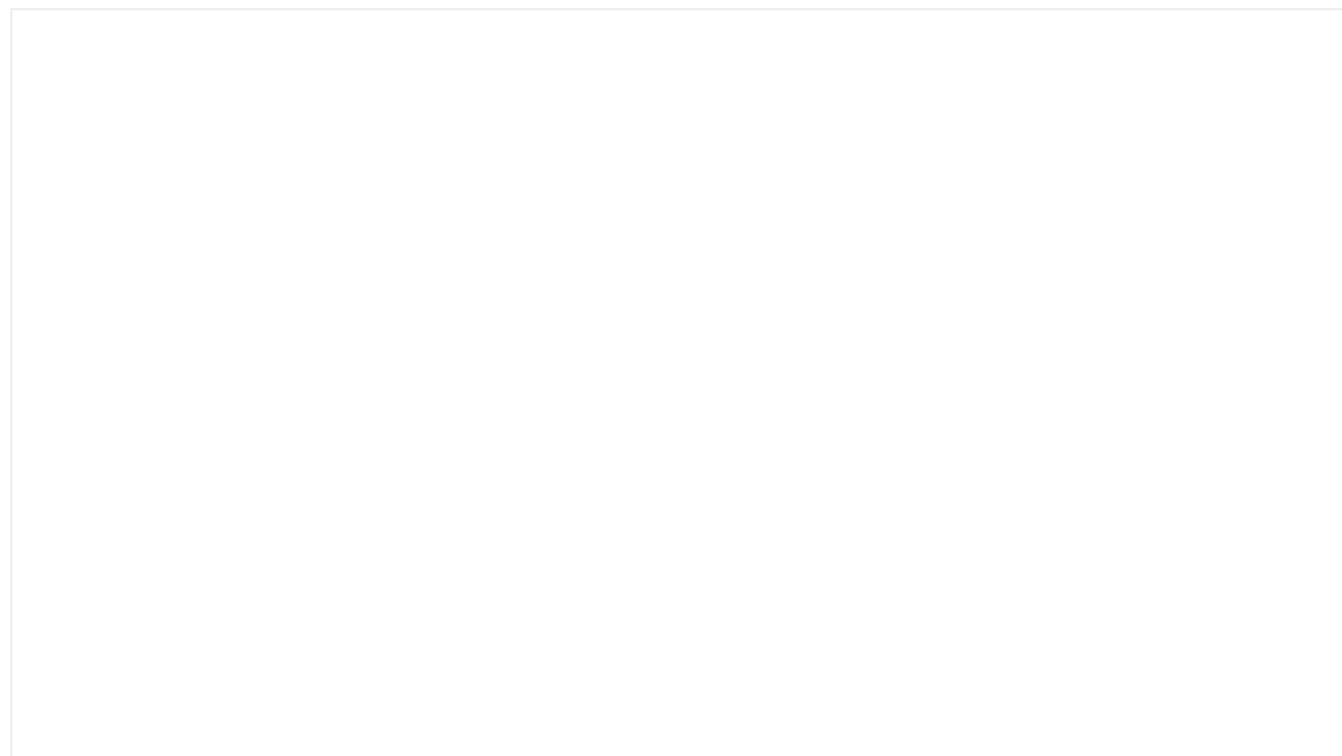


另一方面，长方形胶囊对房子胶囊的输出作出了相当糟糕的预测，所以这种情况下的点积将相当小，这个预测向量的原始路由权重不会增长太多。



接下来，我们再次更新路由权值计算的原始权重的softmax函数。

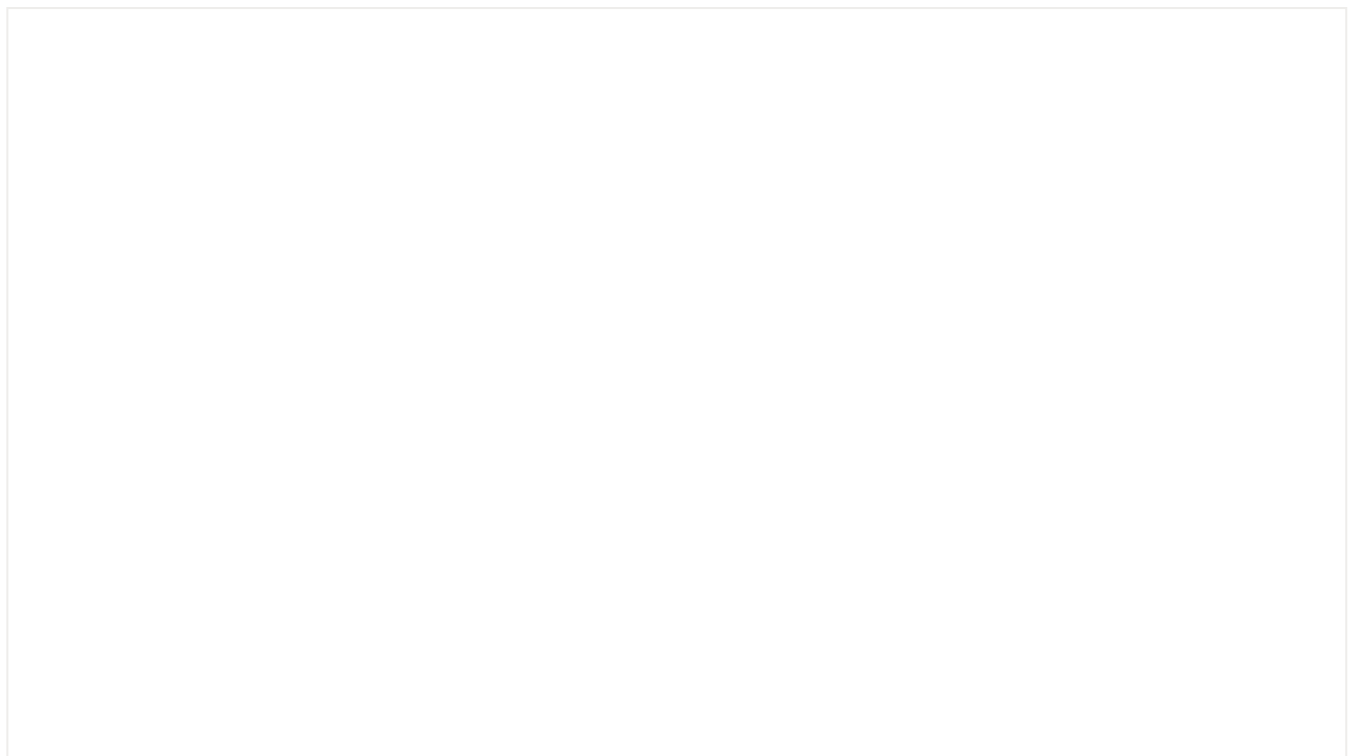
正如你所看到的，矩形胶囊对船胶囊的预测矢量从初始的0.5更新到现在的0.8，而对房子胶囊的预测矢量下降到0.2。所以它的大部分输出现在去了船胶囊，而不是房子胶囊。



我们重复以上的操作再次计算下一层胶囊预测的输出向量的加权和，也就是对下一层是房子胶囊和船胶囊的预测。此时，房子胶囊得到很少的输入，它的输出是一个很小的向量。



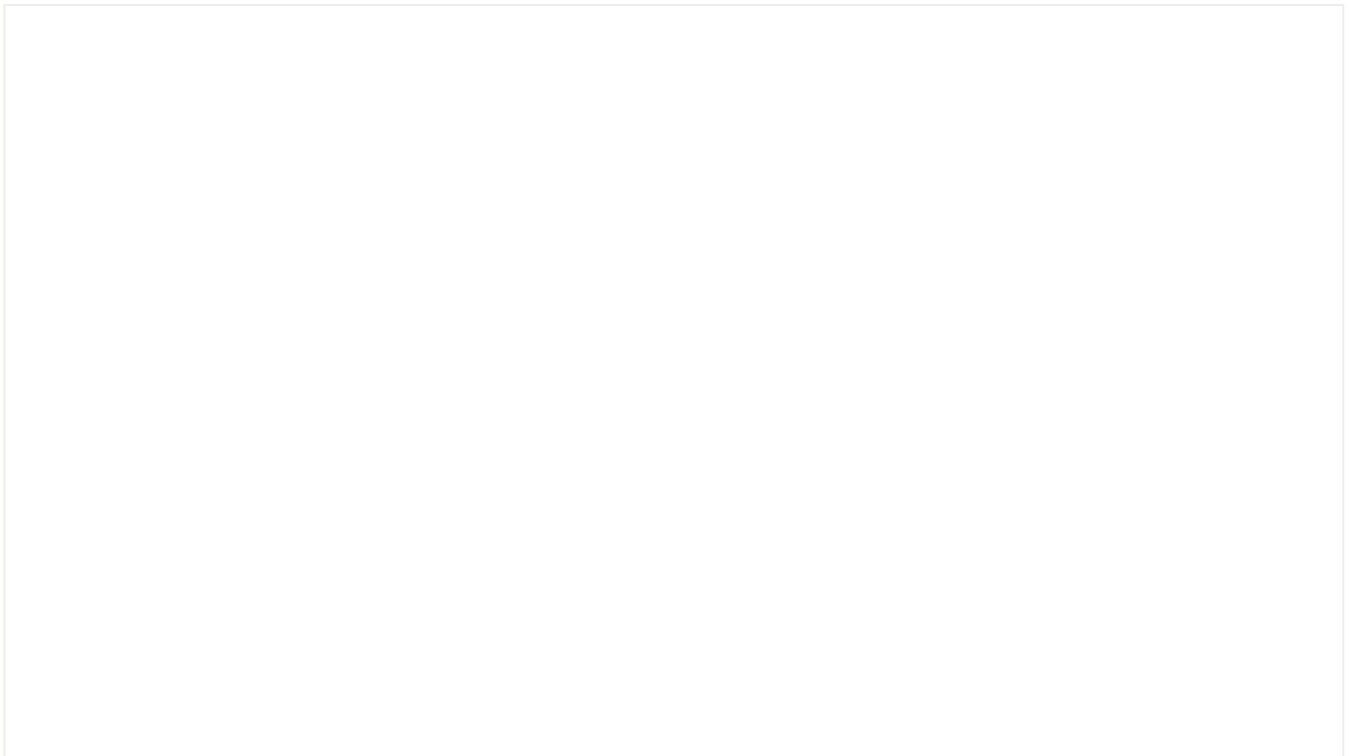
另一方面船胶囊得到很多输入，它的输出向量远远长于1，所以我们又把它压扁（归一化）了。



至此第二轮就结束了，正如你所看到的，在几次迭代中，我们已经可以排除房屋并且清楚地挑选出船。也许一两个回合之后，我们可以停下来，继续以同样的方式进入下一个胶囊层。

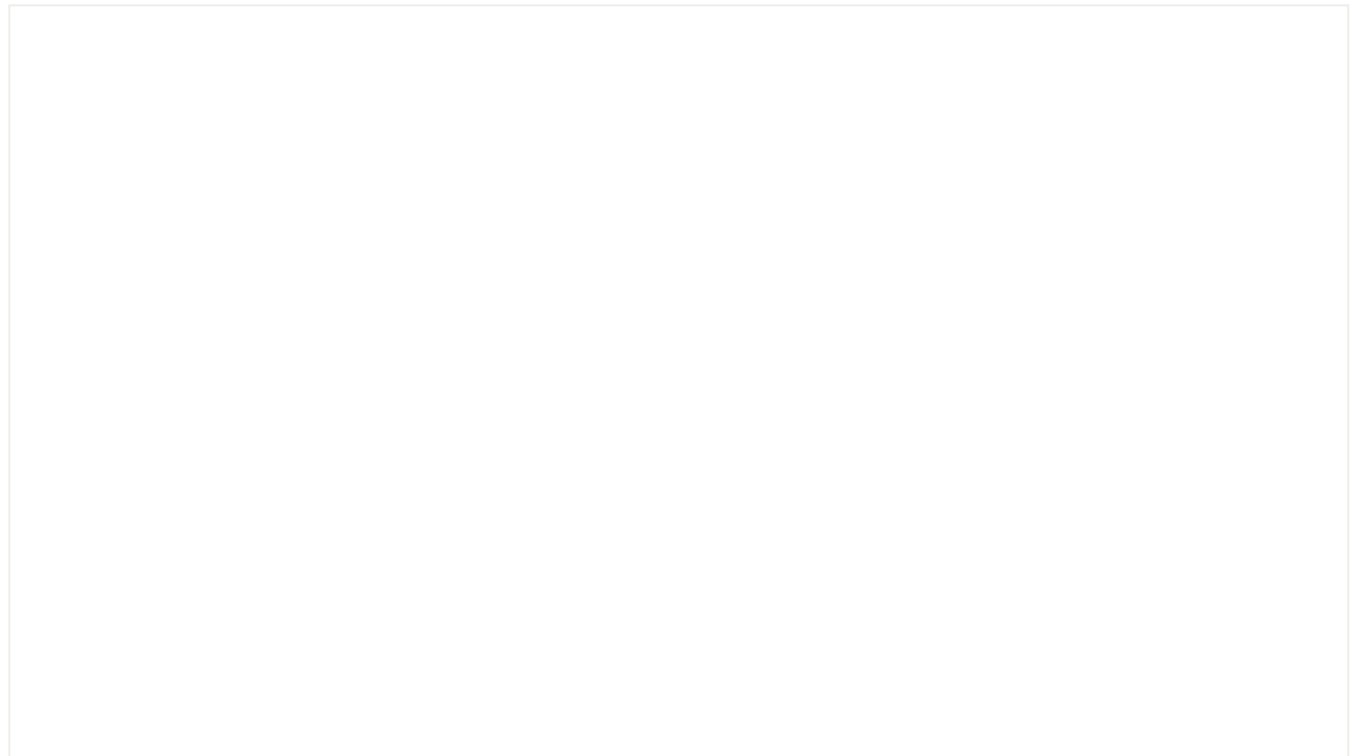


正如我前面提到的，通过协议来处理拥挤重叠的场景是非常有用的，如图中所示的场景。

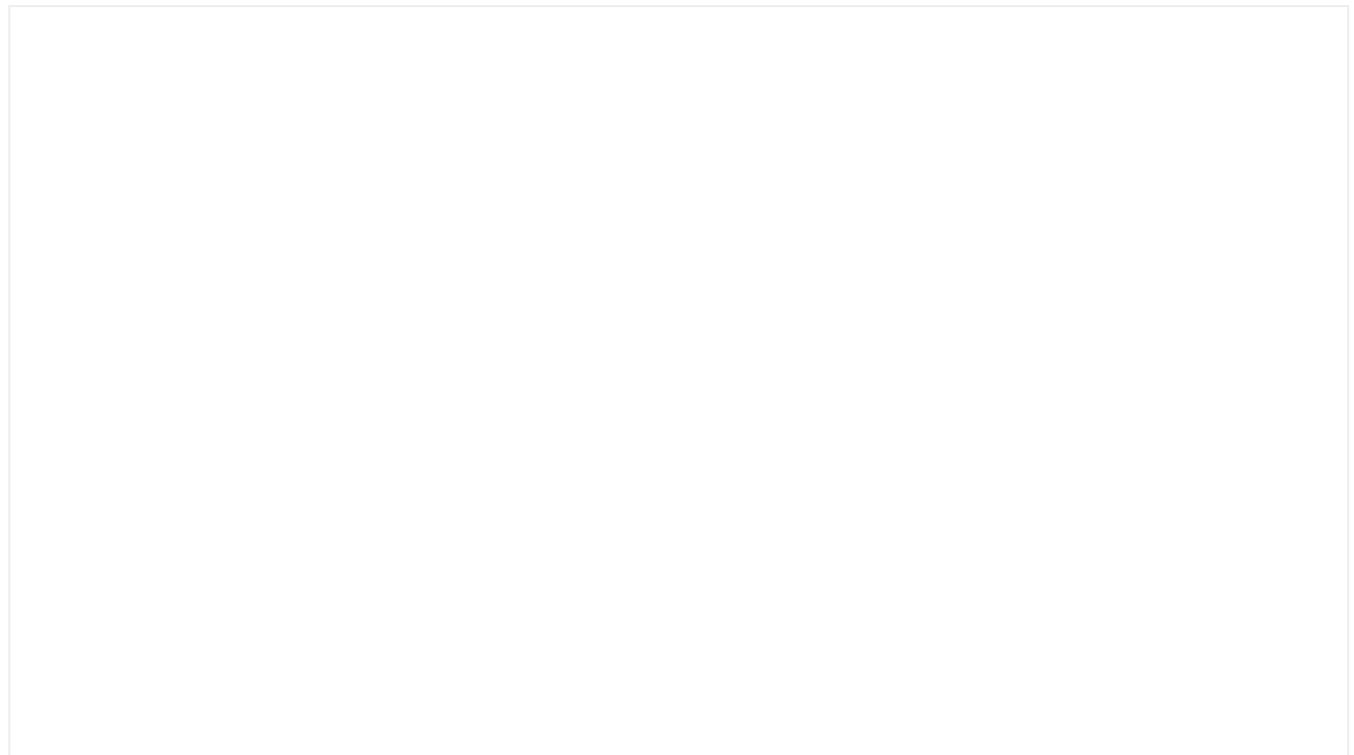


该图像的一种解释是（可以看到图像中有一点模糊），你可以在中间看到一个倒挂的房子。

在这种情况下，就没法解释底部矩形或顶部三角形，也没有办法解释它们到底属于哪个位置。



解释图像的最好方法是，在顶部有一个房子，底部有一艘船。并通过协议的路由倾向于选择这个解决方案，因为它使所有的胶囊都状态最佳，每一个都对下一层的胶囊进行完美的预测。这样就可以消除歧义了。



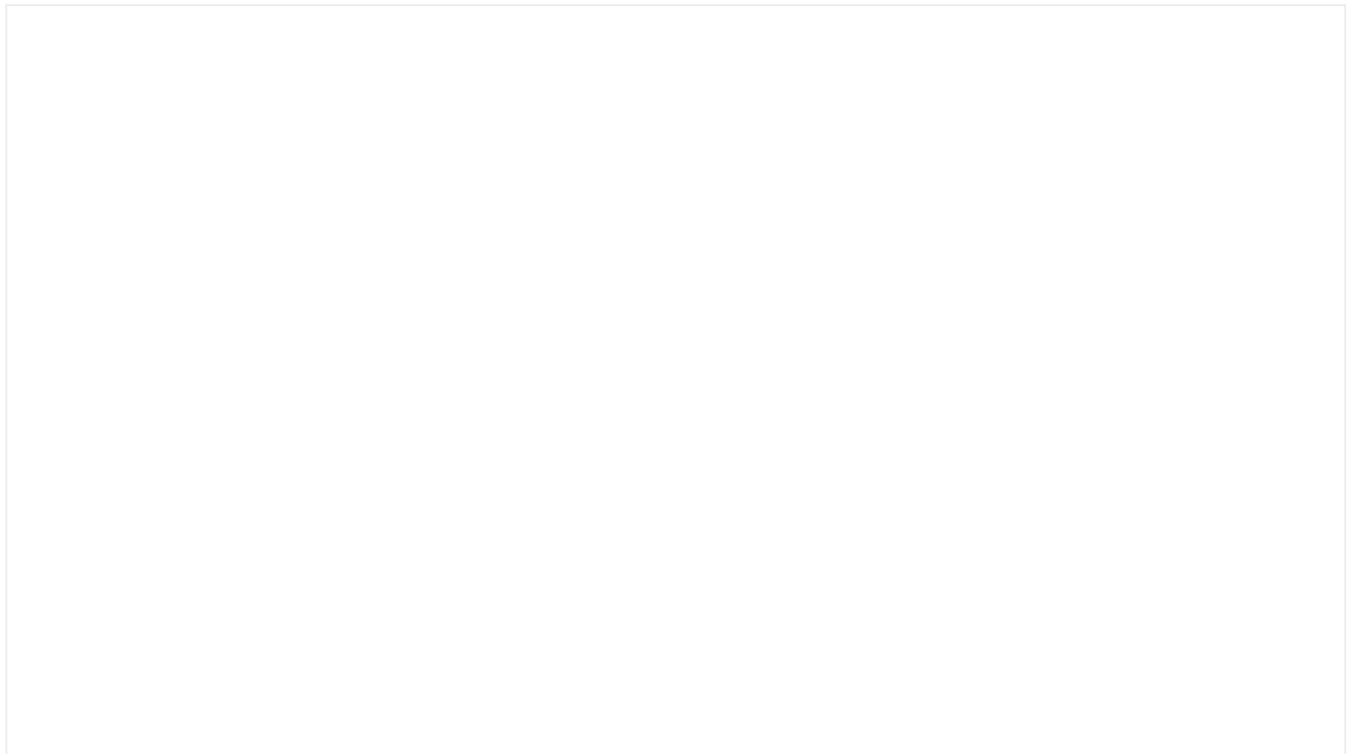
Okay，既然你知道胶囊网络是如何工作的，那么你可以用一个胶囊网络做什么？

首先，你可以创建一个好的图像分类器。只需要在最顶层为每一个类分配一个胶囊，这几乎就是这个网络的全部内容了。

你只需要再添加一个用来计算顶层激活向量长度的层，这一层给出了每一类的估计概率。然后和常规的分类神经网络一样，你可以通过最小化交叉熵损失来训练网络，这样你就可以完成了一个图像分类器。



然而，在论文中，他们使用了一个边缘（margin）损失，使得对图像进行多分类成为可能。

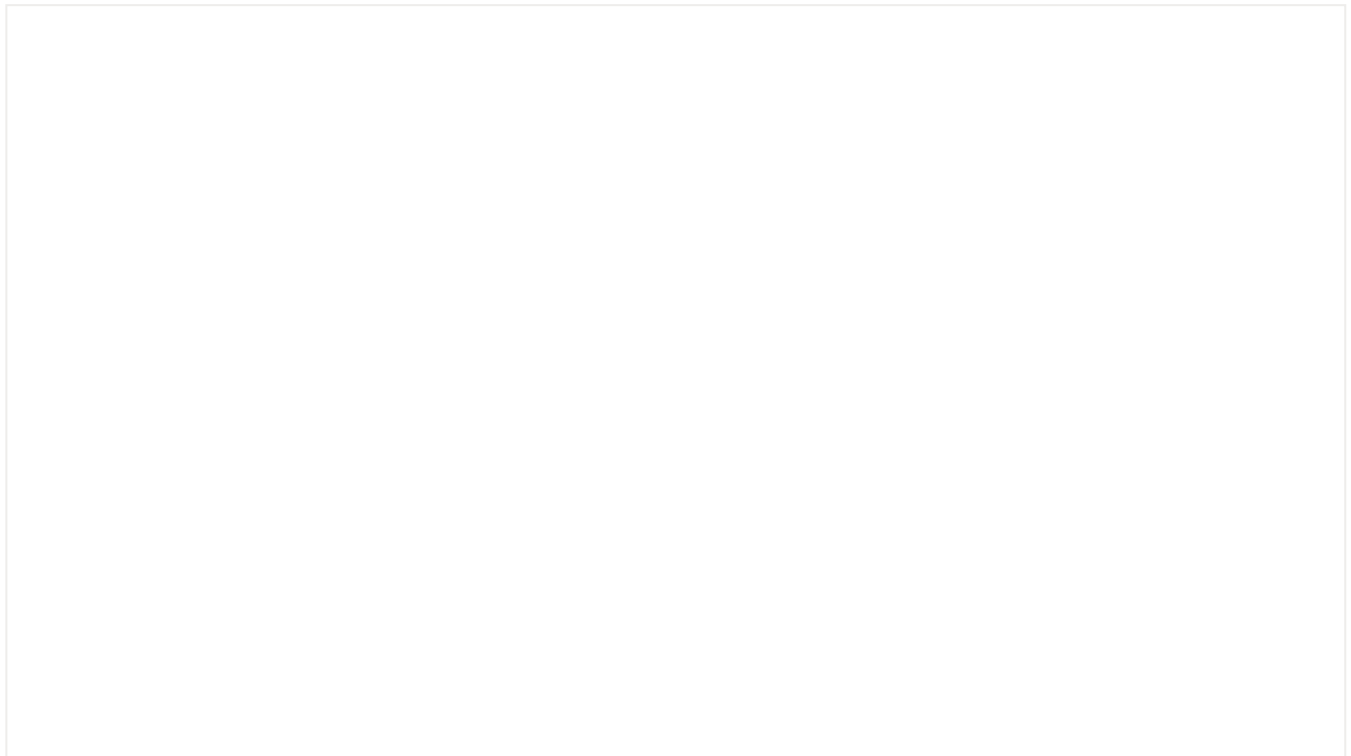


因此，简单来说，这个边缘损失就是下面这样的：如果图像中存在出现了第k类的对象，那么相应这个类的顶层胶囊应该输出一个长度至少为0.9的向量。这样才足够长到确信是这一类。

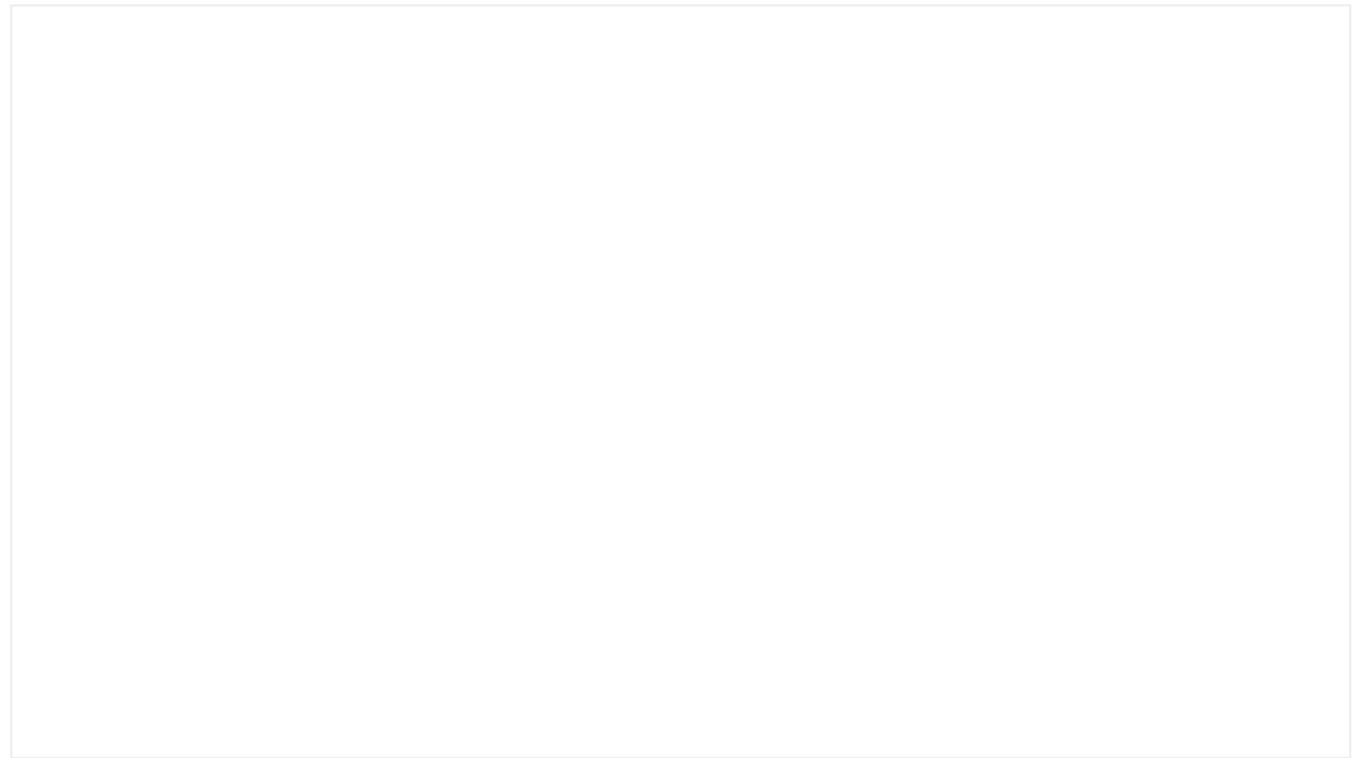
相反，如果图像中没有第k类的对象，则该胶囊将输出一个短向量，该向量的平方长度小于0.1。因此，总损失是所有类损失的总和。



在论文中，他们还在胶囊网络顶端添加了一个解码器网络。它只有3个全连接层，并且在输出层中有一个sigmoid激活函数。它通过最小化重建图像和输入图像之间的平方差，来重构输入图像。



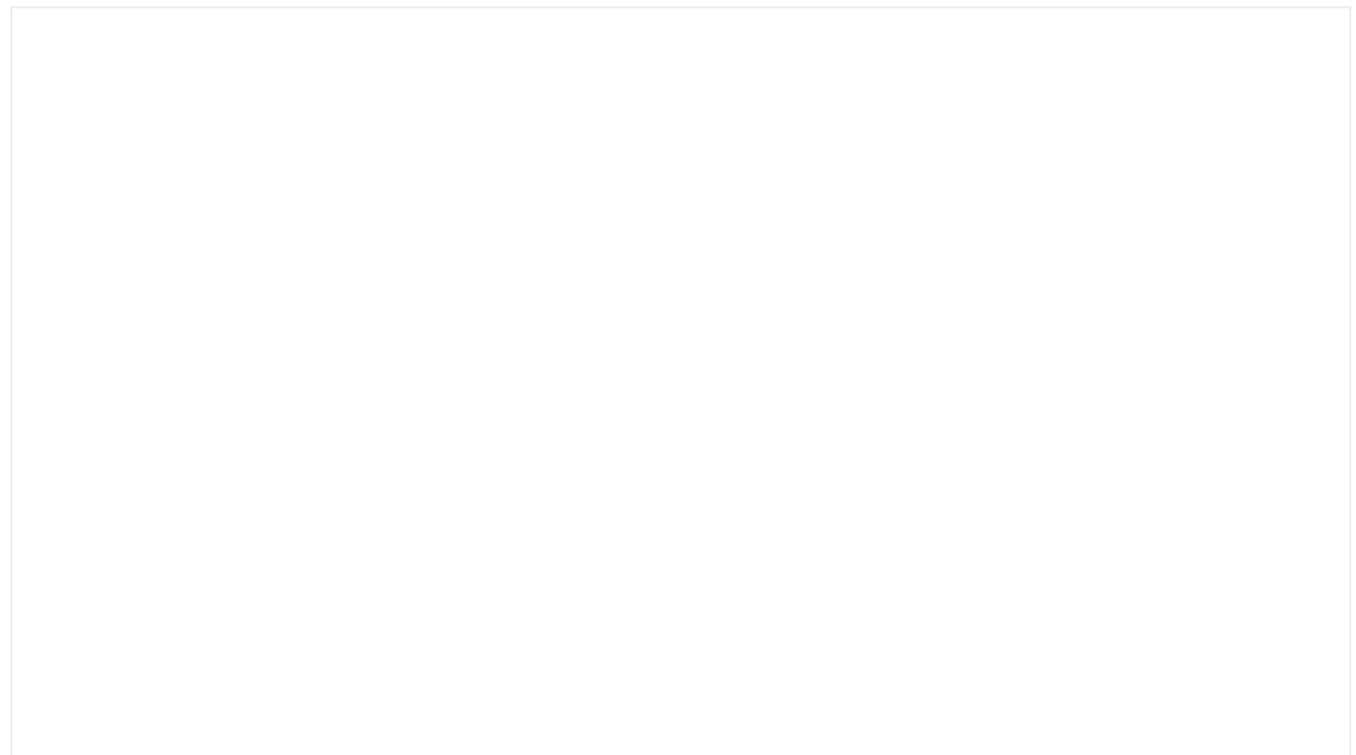
全部损失是我们先前讨论的边缘损失，加上重建损失（为确保边缘损失占主导地位，应大幅度减少重建损失）。应用这种重建损失的好处是，它迫使网络保存重建图像所需的所有信息，直至胶囊网络的顶层及其输出层。这种约束的行为有点像正则化：它减少了过度拟合的风险，有助于模型泛化到新的实例。



就这样，你知道一个胶囊网络是如何工作的，以及如何去训练它。接下来，让我们看看论文中展示的一些有趣的结果。

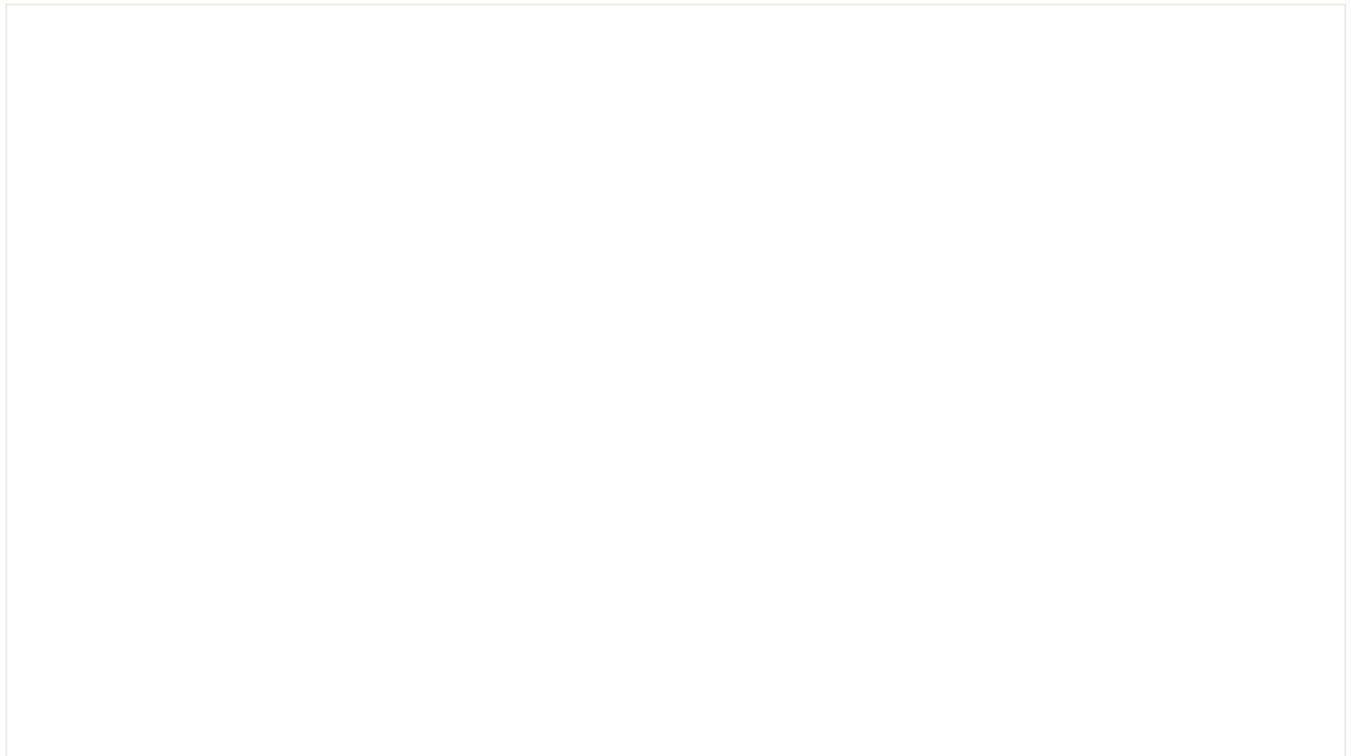
这是文中的图1，展示了对于MNIST数据集的完全胶囊网络。你可以看到前两个正则卷积层，其输出被重新构建和压缩，以获得主胶囊的激活向量。这些初级胶囊按照6 6的网格进行组织，在这个网格中每一个cell有32个初级胶囊，每个胶囊的主要输出8维向量。

因此，第一层胶囊全连接成10个输出胶囊，输出16维向量。这些向量的长度用来计算边缘损失。



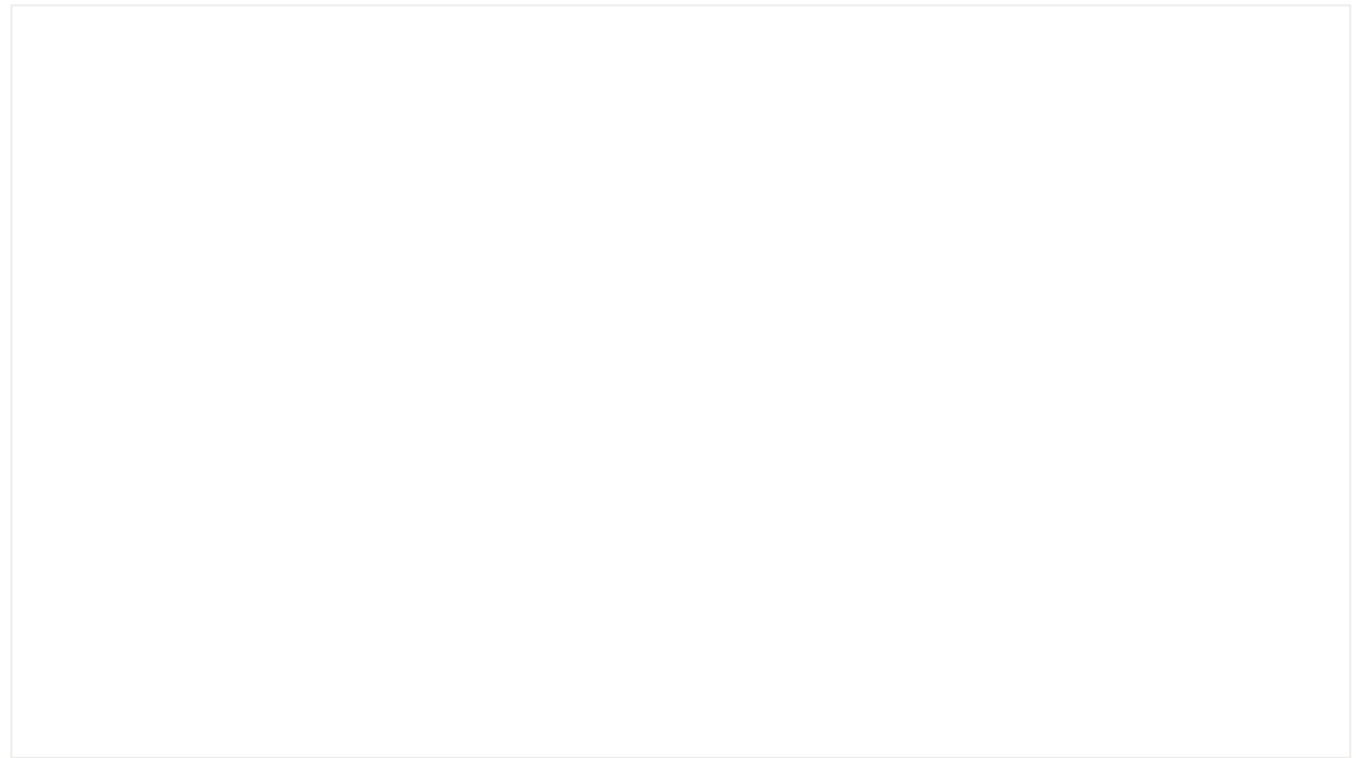
这是论文中的图2，展示了胶囊网络顶层的解码器。它是由两个全连接的ReLU层加上一个全连接的sigmoid层组成，该解码器输出784个数字，对应重构图像的像素个数（图像是 $28 \times 28 = 784$ 像素）。

重建图像与输入图像的平方差是重建损失。



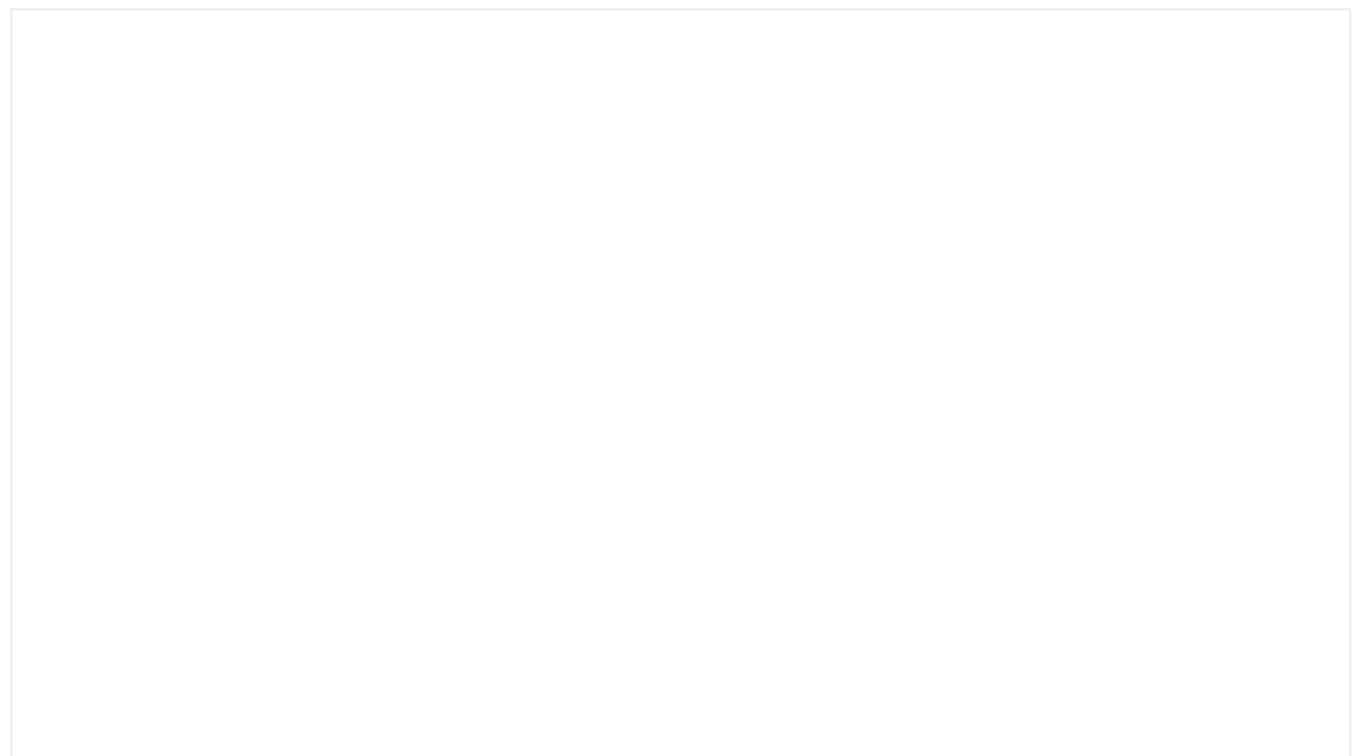
这是论文中的图4。胶囊网络的一个优点是激活向量通常是可解释的。例如，该图展示了当逐渐修改顶层胶囊输出的16个维度中的一个时，所得到的重建图像。你可以看到第一个维度似乎代表尺度和厚度，第四个维度表示局部倾斜，第五个维度表示数字的宽度加上轻微的平移得到确切的位置。

因此，可以很清楚大部分参数分别是表示什么的。



最后，让我们总结一下胶囊网络的利弊。胶囊网络已经达到对MNIST数据集的最佳精度。在CIFAR10数据集上的表现还可以继续提升，也是很值得期待的。胶囊网络需要较少的训练数据。它提供等变映射，这意味着位置和姿态信息得以保存。这在图像分割和目标检测领域是非常有前景的。

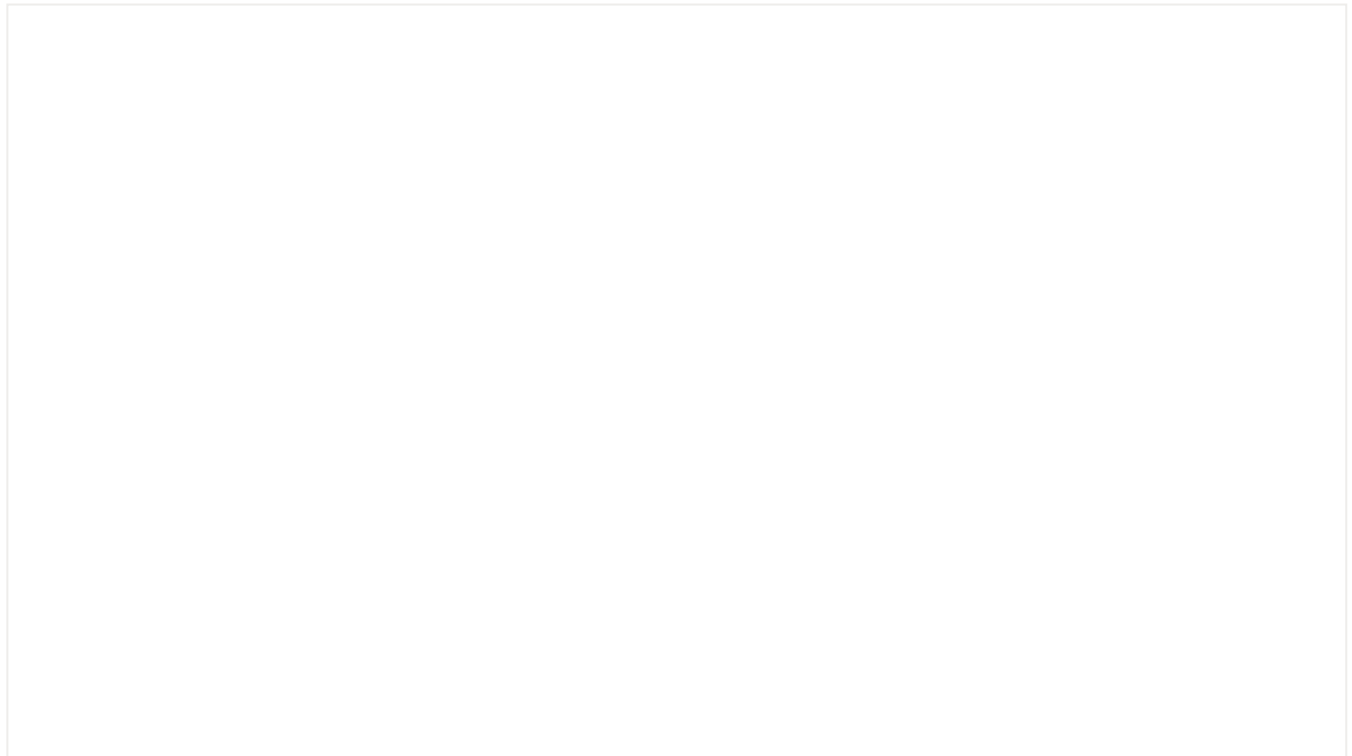
路由协议算法对于处理拥挤的场景具有很好的效果。路由树还映射目标的部分的层次结构，因此每个部分都分配给一个整体。它对旋转、平移和其他仿射变换有很强的健壮性。激活向可解释性也比较好。最后，这是Hinton大神的idea，前瞻性是毋庸置疑的。



然而，该网络有一些缺点：首先，如前面所提到在CIFAR10数据集上的准确性还不高。另外，现在还不清楚胶囊网络是否可以建模规模较大的图像，如ImageNet数据集，准确度是多少？胶囊网络也很慢，在

很大程度上是因为具有内部循环的路由协议算法。

最后，在给定的位置上只有一个给定类型的胶囊，因此如果一个胶囊网络彼此之间太接近，就不可能检测到同一类型的两个对象。这被称为胶囊拥挤，而且在人类的视觉中也能观察到。



我强烈建议你看一看胶囊网络实现代码，如这里列出的（链接将在下面的视频中描述）。花点时间，你应该可以理解代码的所有内容。

实施胶囊网络的主要困难是，它包含了路由协议算法形成的内回路。在Keras的代码实现和tensorflow实现可以比PyTorch麻烦一点，不过也是可以做到的。如果你没有特别的语言偏好，那么pytorch代码是最容易理解的。

NIPS 2017 Paper:

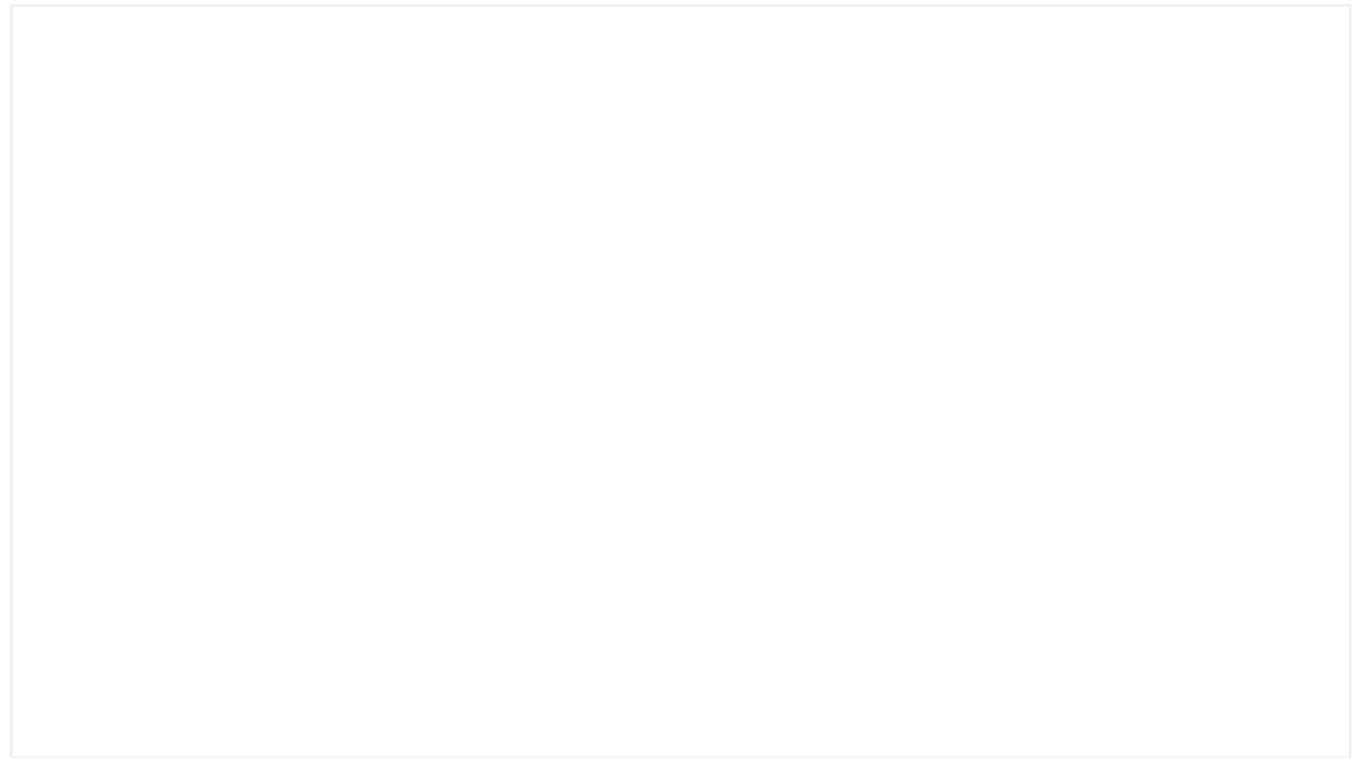
- * Dynamic Routing Between Capsules,
- * by Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton
- * <https://arxiv.org/abs/1710.09829>

The 2011 paper: * Transforming Autoencoders

- * by Geoffrey E. Hinton, Alex Krizhevsky and Sida D.Wang
- * <https://goo.gl/ARSWM6>

CapsNet implementations:

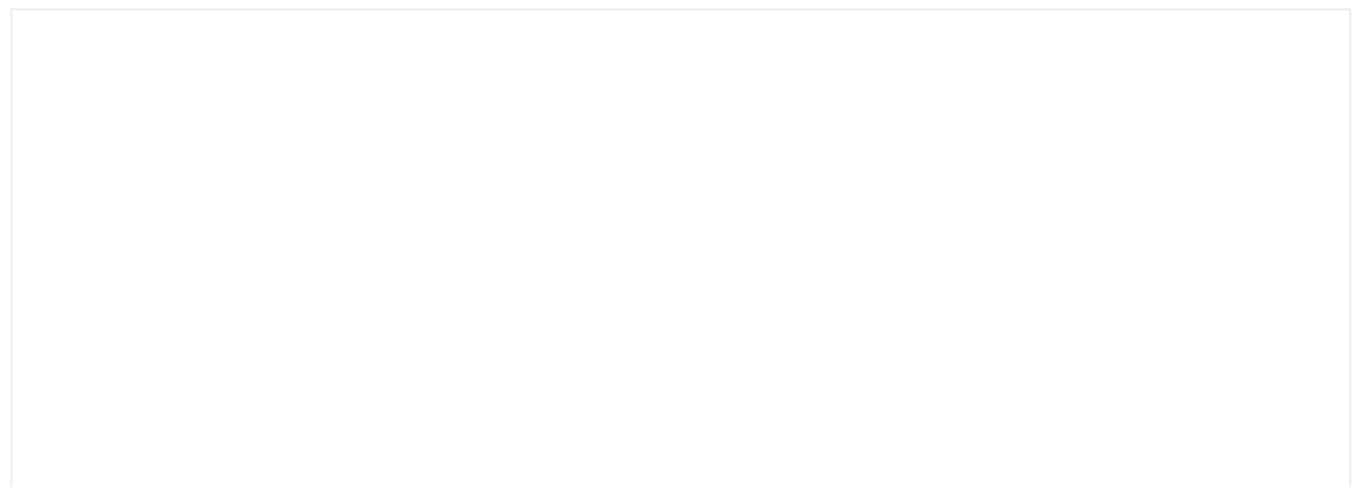
- * Keras w/ TensorFlow backend: <https://github.com/XifengGuo/CapsNet-keras>.
- * TensorFlow: <https://github.com/naturomics/CapsNet-Tensorflow>
- * PyTorch: <https://github.com/gram-ai/capsule-networks>



这就是我本节课讲的所有内容，希望你喜欢这个视频。如果你喜欢，请关注、分享、评论、订阅、blablabla。这是我的第一个真正的YouTube视频，如果你发现它有用，我可能会做更多。

如果你想了解更多关于机器学习、深度学习和深入的学习，你可能想读机器学习与我自己实现的scikit学习TensorFlow O'Reilly的书。它涵盖了非常多的话题，有很多的实例代码，你可以在我的GitHub账户中找到，在这里留下视频链接。今天就到这里，下次再见！

论文信息：Dynamic Routing Between Capsules



论文地址：<https://arxiv.org/pdf/1710.09829.pdf>

摘要：Capsule 是一组神经元，其活动向量（activity vector）表示特定实体类型的实例化参数，如对象或对象部分。我们使用活动向量的长度表征实体存在的概率，向量方向表示实例化参数。同一水平

的活跃 capsule 通过变换矩阵对更高级别的 capsule 的实例化参数进行预测。当多个预测相同时，更高级别的 capsule 变得活跃。我们展示了判别式训练的多层 capsule 系统在 MNIST 数据集上达到了最好的性能效果，比识别高度重叠数字的卷积网络的性能优越很多。为了达到这些结果，我们使用迭代的路由协议机制：较低级别的 capsule 偏向于将输出发送至高级别的 capsule，有了来自低级别 capsule 的预测，高级别 capsule 的活动向量具备较大的标量积。

本文经授权转载自专知：Quan_Zhuanzhi，点击“阅读原文”可查阅原文。



新智元

立即体验新智元小程序，一键直达AI大咖

 小程序

内容转载自公众号

专知 专知

了解更多 >

[阅读原文](#)