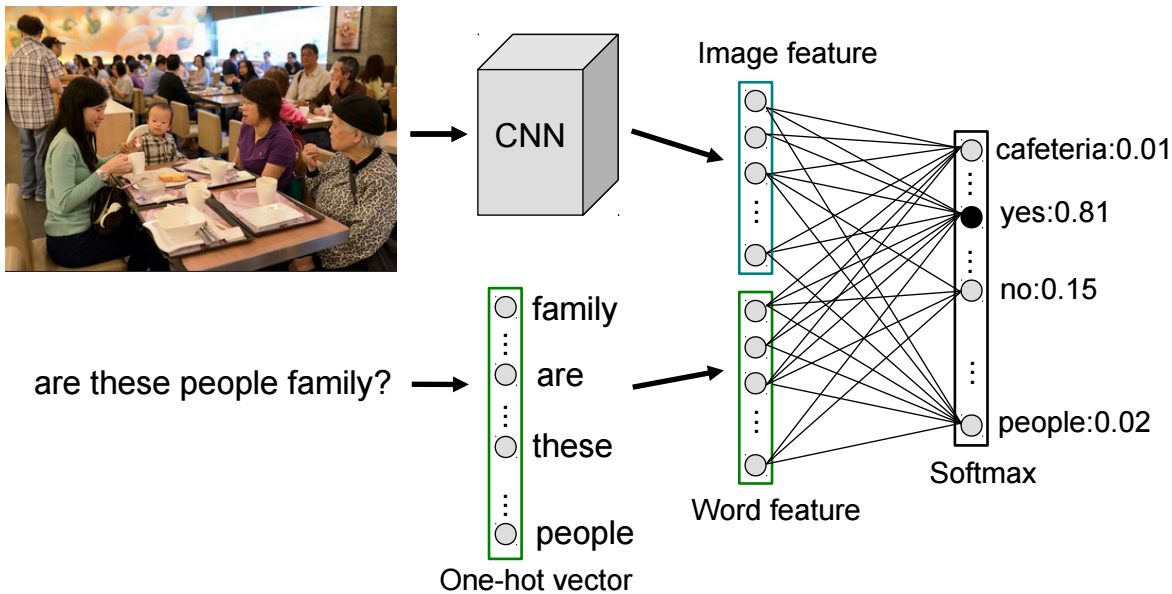
VQA任务中，机器根据给定的一张图和相关问题，回答该问题。 该任务由[Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In NIPS, 2014.]提出。和第六章中传统视觉工作不同的是，VQA需要同时识别多种物体并根据对问题的理解找到对应答案，这对机器的图像理解能力和文本理解能力要求都很高。

传统vqa

**1. 最简单的VQA结构：**

[B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering.]



图像采用CNN（pretrained on ImageNet）, 问题文本采用Bag of words(BOW) 表示，两个特征连接后与输出层全连接，然后过一个softmax直接分类，最后分类的结果（即都是单个词）就是答案。简单暴力有效，调参后在VQA上达到55.88%的准确率。这其中当然可以有一些变种，比如问题文本特征可以用LSTM替代BOW，[Mengye Ren, Ryan Kiros, and Richard S. Zemel. Exploring models and data for image question answering. In NIPS, 2015.]和[Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In ICCV, 2015.]就是这么干的。

**2. 图像加了关注点：**

把整幅图的分类特征扔进vqa真的合适么？比如问题“那个穿绿衣服的人头上的帽子是什么颜色的？”需要定位到的是帽子，但图中这个人占的面积很大，所以图像训练数据中该图分类可能是“人”，得到的特征也就是和人相关的特征了。于是人们想，那我应该让问题来指导对图像的关注点，借助于[Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In ICML, 2015.]中提的图像中加soft attention的概念，有些文章也在vqa中加起了图像attention，即question-guided attention。[Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In CVPR, 2016.]受Memory Network（在第七章讲过）启发，用两个连续attention模块进行多步attention推断。[Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In ICML, 2016.]在动态attention网络中加入多步attention推断。[Kevin J Shih, Saurabh Singh, and Derek Hoiem. Where to look: Focus regions for visual question answering. In CVPR, 2016.]在object proposal上单步找attention。

**3. 图像&文本一起加关注点：**

**3. 让问题决定看几步**

[Training Recurrent Answering Units with Joint Loss Minimization for VQA]中说，每个问题的难度都不同，比如“苹果是什么颜色的？”和“桌上的苹果是什么颜色的”需要查找的步骤就不同，那我能不能通过一个RNN让机器自己决定要走几步呢？ 于是文中给思考过程加了一个RNN，RNN中每个timestep都留一个自带attention的答案单元，这些单元共享模型参数，最后的loss是所有timestep的loss的和。由于在分析每步给出的answer时发现，步数越多越容易在简单问题上过拟合，所以只要第一个timestep给出的答案单元做对了，就停止学习（lr设为0）。