移动端AI之模型压缩

分享人： 曾 冠 奇

2017/4/7



分享提纲

1 移动端AI的背景

2 模型压缩的理论

3 模型压缩实战

提纲挈领

本次的分享主要以介绍别人的工作为主。自己的工作由于某些原因暂时不介绍。主线是感性认识到理性认识，理性认识到具体实践。由浅入深，深入浅出。主要讲思想，线上一个小时讲公式我估计大家印象也不深，公式还是需要自己手工推理的。

多个案例讲述移动端AI的情况和背景（感性认识，浅层）

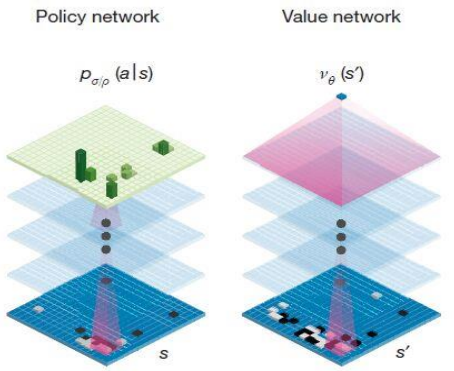
前：2016年2月15日，斯坦福韩松的论文：压缩网络

中：2016年5月17日，神 Yoshua Bengio的二值化网络（事实上，Yoshua Bengio已经在这一块发了多篇论文，这次选的是最有代表性的）

后：2016年8月2日Mohammad Rastegari的XNOR-Net。由于这个研究他们成立了独角兽公司，能把事情做到极致，解决痛点问题就能成为独角兽。

深入理论后，浅出指导实践。
通过实践上手，感受模型压缩的魅力。
这次选择了基于tensorflow的量化方法。

1.1 移动端AI背景与意义



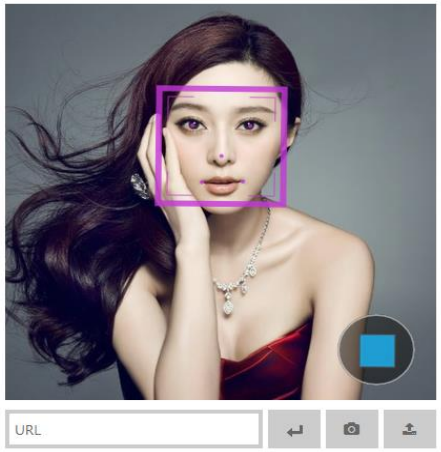
AlphaGo

?

深度学习的普遍应用

语音识别

手机上的人脸检测

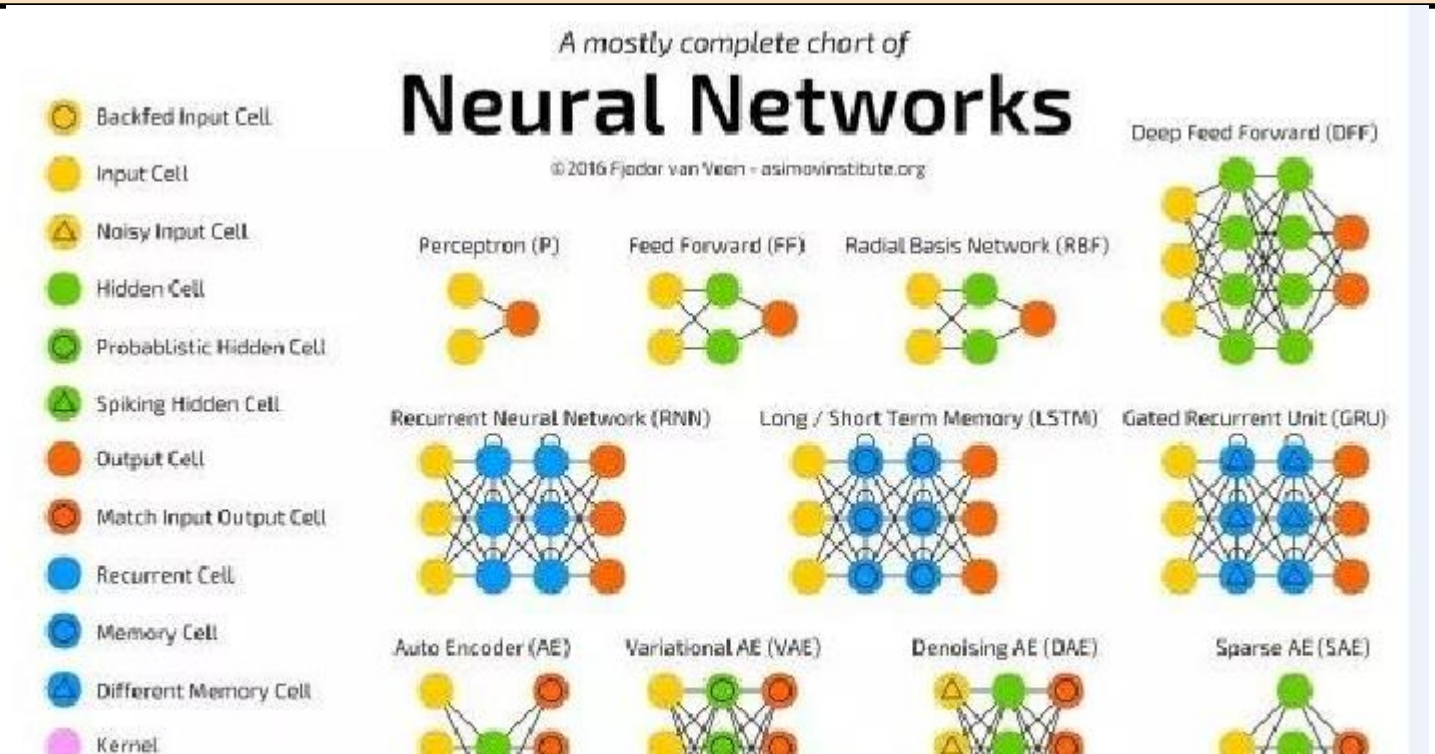


军队机器人

1.2 几个案例

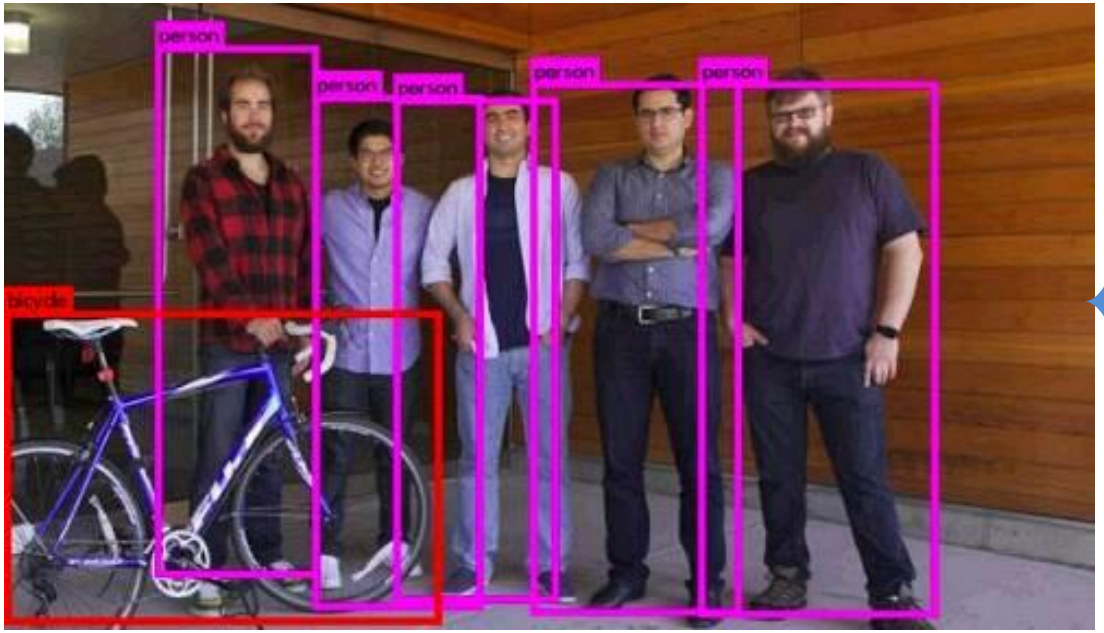
1、余凯创办地平线机器人公司，剑指深度学习嵌入式芯片

2、苹果开放深度学习开发架构Medal。只要你懂Swift，只需要一台iPhone 7手机，你就能开发基于深度学习的应用！



1.2几个案例

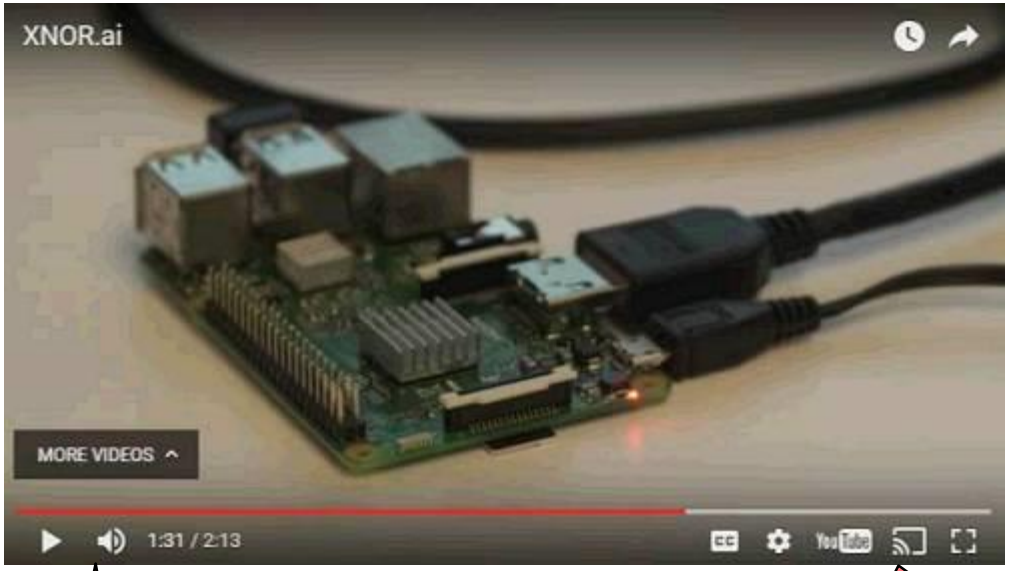
2017年2月2日，位于美国西雅图的 AI 创业公司 XNOR.AI 宣布获得来自Madrona Venture Group和艾伦人工智能研究所（Allen Institute for Artificial Intelligence）等机构的260万美元的种子融资。XNOR.AI 利用二值化神经网络等技术对深度学习网络进行压缩，致力于开发有效地在移动端或嵌入式设备上运行的深度学习算法。



XNOR.AI团队CEO Ali Farhadi是华盛顿大学计算机系教授，同时也是艾伦人工智能研究所的计算机视觉方向的负责人。是非常惊艳的实时物体检测框架YOLO的主要贡献者

XNOR.AI的CTO Mohammad Rastegari是艾伦人工智能研究所研究科学家，也在计算机视觉领域有接近十年的研究经历。2016 年 3 月，Mohammad Rastegari 等人在ECCV论文（XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks）中首次提出了 XNOR-Net 的概念

1.3一些应用

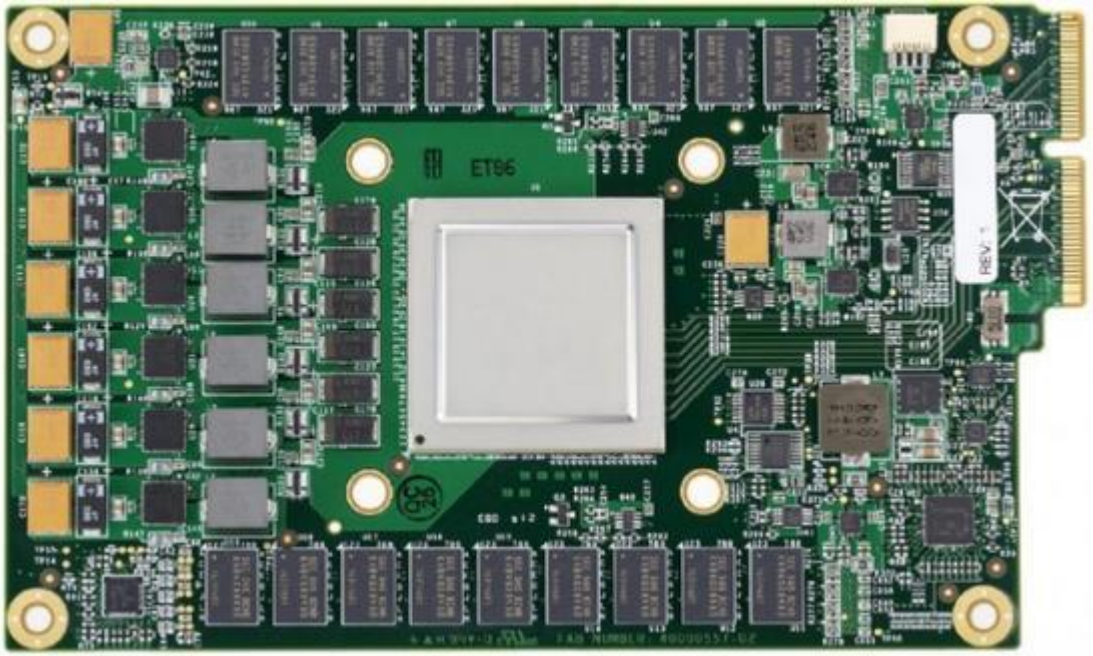


树莓派

手机端的微软识花

一个广阔的前景是智能家居。更广阔的前景还有智能机器人

1.4关于TPU（Tensor processing unit）



主要用于前向传播

谷歌的专用机器学习芯片TPU处理速度要比GPU和CPU快15-30倍（和TPU对比的是英特尔Haswell CPU以及Nvidia Tesla K80 GPU），而在能效上，TPU更是提升了30到80倍

tensor在翻滚，从软件到硬件，谷歌正在给人工智能提供解决方案。这块芯片不大，主要用在前向传播的时候，这个是嵌入式最需要的
Tensor processing unit

分享提纲

1 移动端AI的背景

2 模型压缩的理论

3 模型压缩实战

模型压缩的理论（三篇论文）

1. DEEP COMPRESSION: COMPRESSING DEEP NEURAL NETWORKS WITH PRUNING, TRAINED QUANTIZATION AND HUFFMAN CODING

2. Binarized Neural Networks: Training Neural Networks withWeights and Activations Constrained to +1 or 1

3. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks

2.1 模型深度压缩（一张图看懂世界）

剪枝：剪掉一些权值低的连接，可以减少10倍权值数目

量化：对权值进行编码，相当于对权值进行聚类，用聚类的中心代替这一个类别的权值

哈弗曼编码：聚类得到的中心长度不一致，进行哈弗曼编码可有效减少存储空间

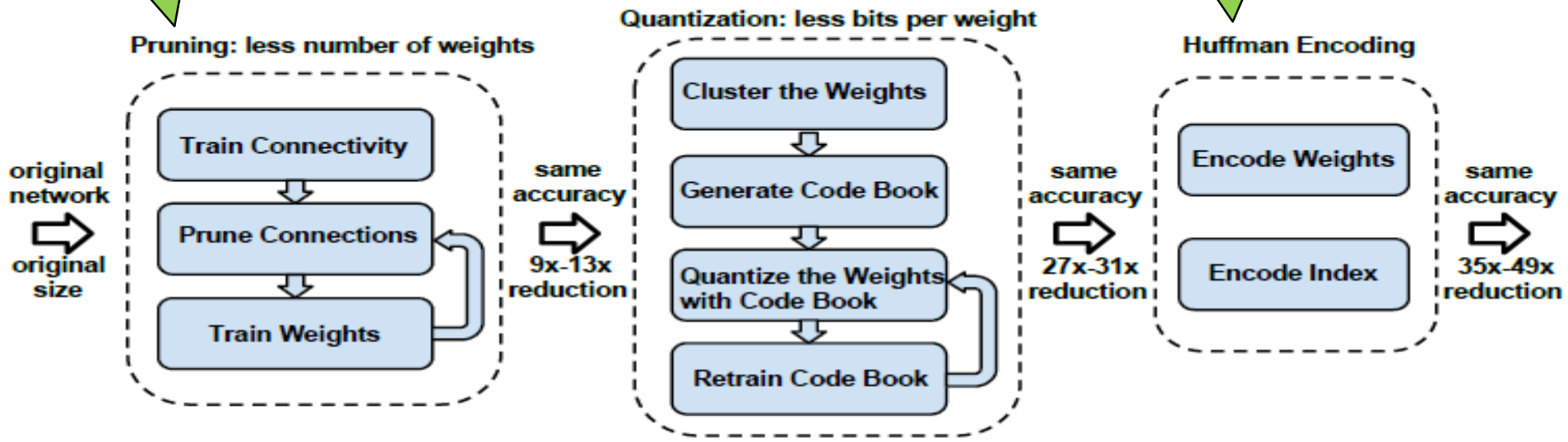


Figure 1: The three stage compression pipeline: pruning, quantization and Huffman coding. Pruning reduces the number of weights by 10×, while quantization further improves the compression rate: between 27× and 31×. Huffman coding gives more compression: between 35× and 49×. The compression rate already included the meta-data for sparse representation. The compression scheme doesn't incur any accuracy loss.

2.1 模型深度压缩

三个方面

剪枝：
权值小的，如接近0的。0.1，-0.23等全部剪掉，需要遍历整个权值空间
O(n)

量化：
权值1.8、2.1、2、2.2全部用2代替。这个处理起来没有一定的定论。看你自己怎么操作了。1.6是用1.5还是用2呢？

哈弗曼编码：
这个就是数据结构里的。

点评：
1、编码实现较复杂
2、是对训练好的模型进行的处理，而不是在训练过程中的处理，因而有精度损失。
3、还是陷入权值具体的值中，压缩比有限
4、只有压缩，没有加速。

模型压缩的理论（三篇论文）

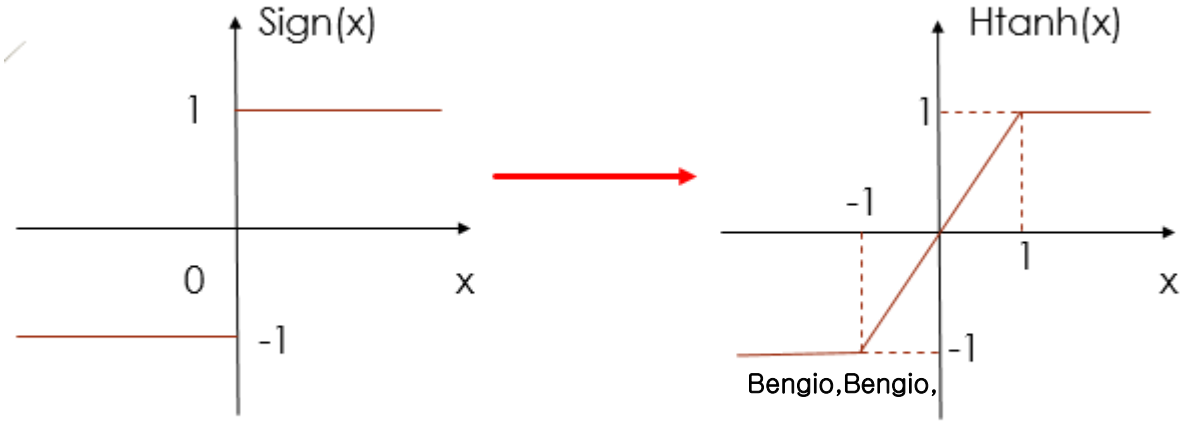
1. **DEEP COMPRESSION: COMPRESSING DEEP NEURAL NETWORKS WITH PRUNING, TRAINED QUANTIZATION AND HUFFMAN CODING**

2. **Binarized Neural Networks: Training Neural Networks withWeights and Activations Constrained to +1 or 1**

3. **XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks**

2.2 二值化网络（1）

$$q = \mathrm{Sign}(r)$$



$$g_r = g_q 1_{|r|\leq 1}.$$

Htanh

- 对weights和activations进行二值化。如上图左，Binarization function很简单，就是一个符号函数。但符号函数不好进行梯度的反向传播，因此就把它近似成了右边的Htanh(x)的函数，这样在[-1,1]区间内导数就等于1。

2.2 二值化网络（2）

Ensure: updated weights W^{t+1}, updated BatchNorm parameters θ^{t+1} and updated learning rate η^{t+1}.

{1. Computing the parameters' gradient:}

{1.1. Forward propagation:}

for $k = 1$ to L **do**

  $W_k^b \leftarrow \text{Binarize}(W_k)$

  $s_k \leftarrow a_{k-1}^b W_k^b$

  $a_k \leftarrow \text{BatchNorm}(s_k, \theta_k)$

  if $k < L$ **then**

    $a_k^b \leftarrow \text{Binarize}(a_k)$

  end if

end for

- 首先权重Wk 经过二值化，然后与上层二值化后的激活值abk−1相乘，再进项BatchNormalization得到这一层的激活值ak，由于BatchNorm的参数θk不是二值的，因此ak也不是二值的，我们需要再对它做二值化得到二值化后的激活值abk。

2.2 二值化网络（3）

{1.2. Backward propagation:}
{Please note that the gradients are not binary.}
Compute $g_{a_L} = \frac{\partial C}{\partial a_L}$ knowing a_L and a^*
for $k = L$ to 1 **do**
  if $k < L$ **then**
    $g_{a_k} \leftarrow g_{a_k^b} \circ 1_{|a_k|\leq 1}$
  end if
  $(g_{s_k}, g_{\theta_k}) \leftarrow \text{BackBatchNorm}(g_{a_k}, s_k, \theta_k)$
  $g_{a_{k-1}^b} \leftarrow g_{s_k} W_k^b$
  $g_{W_k^b} \leftarrow g_{s_k}^\top a_{k-1}^b$
end for

Algorithm 4 Running a BNN. L is the number of layers.

Require: a vector of 8-bit inputs a_0, the binary weights W^b and the BatchNorm parameters θ.
Ensure: the MLP output a_L.
  {1. First layer:}
  $a_1 \leftarrow 0$
  for $n = 1$ to 8 **do**
    $a_1 \leftarrow a_1 + 2^{n-1} \times \text{XnorDotProduct}(\mathrm{a_0^n}, \mathrm{W_1^b})$
  end for
  $a_1^b \leftarrow \text{Sign}(\text{BatchNorm}(a_1, \theta_1))$
  {2. Remaining hidden layers:}
  for $k = 2$ to $L - 1$ **do**
    $a_k \leftarrow \text{XnorDotProduct}(a_{k-1}^b, W_k^b)$
    $a_k^b \leftarrow \text{Sign}(\text{BatchNorm}(a_k, \theta_k))$
  end for
  {3. Output layer:}
  $a_L \leftarrow \text{XnorDotProduct}(a_{L-1}^b, W_L^b)$
  $a_L \leftarrow \text{BatchNorm}(a_L, \theta_L)$

- 反向传播过程如下，权重和激活值的更新并不是二值的，因为这样做的话误差会很大。
- 输入层的特征是没有进行二值化的，输入的图像像素值分布在[0,255]之间，可以用8比特来表示，这样就能将输入的实值像素值变成二值化的编码了。整体BNN的流程如左下，将乘法运算都变成了XNOR运算
- 点评：简单粗暴，精度较低，可以进行模型加速

模型压缩的理论（三篇论文）

1. **DEEP COMPRESSION: COMPRESSING DEEP NEURAL NETWORKS WITH PRUNING, TRAINED QUANTIZATION AND HUFFMAN CODING**

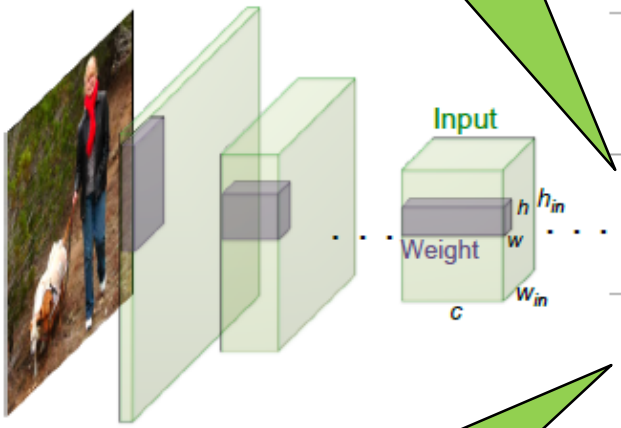
2. **Binarized Neural Networks: Training Neural Networks withWeights and Activations Constrained to +1 or 1**

3. **XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks**

3.1 位运算网络（一张图看懂世界）

只将权值二值化，输入的图像卷积后的特征不变化

二值化的操作简单粗暴，大于0的为1，小于0的为-1



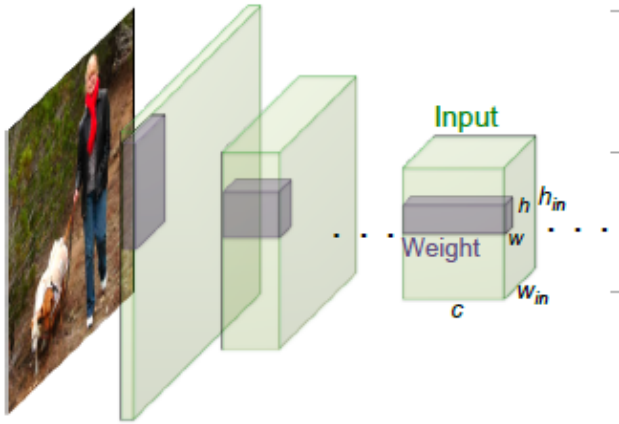
	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Real-Value Weights 0.12 -1.2 ... 0.41 / -0.2 0.5 ... 0.68	+ , − , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Binary Weights 1 -1 ... 1 / -1 1 ... 1	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input **(XNOR-Net)**	Binary Inputs 1 -1 ... -1 / -1 1 ... 1; Binary Weights 1 -1 ... 1 / -1 1 ... 1	XNOR , bitcount	~32x	~58x	%44.2

权值和输入的图像卷积后的特征一起二值化，从而卷积操作可以位运算进行，运算速度提升58倍

3.1 位运算网络（互动环节与课后思考题）

家庭作业：这样简单粗暴的二值化，它是怎么实现反向传播的梯度下降的？

为什么二值化权值的网络准确率没什么变化，而一起二值化图像像素卷积特征精度降低10个百分点？



	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs: 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Real-Value Weights: 0.12 -1.2 ... 0.41 / -0.2 0.5 ... 0.68	+ , − , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs: 0.11 -0.21 ... -0.34 / -0.25 0.61 ... 0.52; Binary Weights: 1 -1 ... 1 / -1 1 ... 1	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input **(XNOR-Net)**	Binary Inputs: 1 -1 ... -1 / -1 1 ... 1; Binary Weights: 1 -1 ... 1 / -1 1 ... 1	XNOR , bitcount	~32x	~58x	%44.2

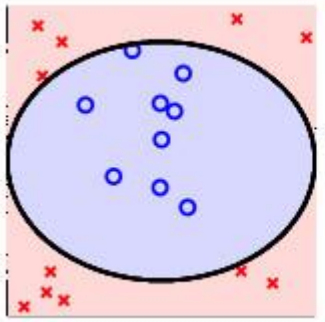
4.3 深度思维（思辨、总结、提高、悟道)

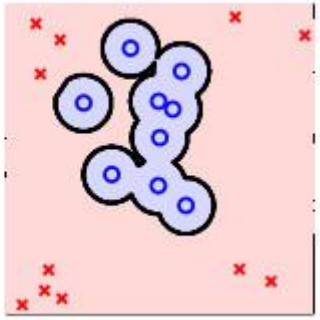
- **问题--来自于总结：**
- **模型压缩是不是伪科学？这个方向怎么样？**

模型压缩说白了就是把一些权重（矩阵）、数据（矩阵）简单化表示，咱1是1，2是2，这样简单化表示能丰富表达模型结构和样本空间吗？

1和2全部用1来替代，是不是一种正则化？

机器学习中有一个原理叫奥卡姆剃刀原理。说的就是简单的模型比复杂的模型鲁棒性好





点评：

万法归宗，大道从简。我们从浅层学习，逐渐发展到深度学习，提高了不少精度，然后对深度学习模型进行压缩，又进行了一次螺旋式上升，提高应用的范围。

由浅入深，深入浅出，九浅一深？

分享提纲

1 移动端AI的背景

2 模型压缩的理论

3 模型压缩实战

3.1 参见网页演示

https://petewarden.com/2016/05/03/how-to-quantize-neural-networks-with-tensorflow/
教程的地址。这个可以作为模型压缩的入门

https://github.com/tensorflow/tensorflow/tree/master/tensorflow/tools/quantization
代码路径
照着教程做就可以了。tensorflow官方网站上也有教程，我一直没有找到

6 欢迎交流

欢迎关注微信公众号，交流心得分享知识，共建AI生态系统。
小福利：关注下方微信公众号，回复metal。获取苹果深度学习架构metal开发文档和demo。
最后我的ppt参考了不少别人的文章，在此表示感谢。

谢谢！