**韩淑婷周报**

### 十八、第十八次2019年4月19日

## 论文方面

Verifying Binarized Neural Networks by Angluin-Style Learning

论文章概览：

研究的问题：verify the behavior of binarized neural networks on some input region验证BNN在某些输入区域上的行为

提出的算法：Angluin-style learning：把给定区域上的神经网络编译为OBDD

为何使用OBDD：OBDD便于我们进行一系列验证查询操作，包括计数、计算反例概率、识别反例共同特征等

实验结果：验证BNN在识别手写数字图片上的结果

背景：

为了提高神经网络的行为保证，人们需要了解和验证基础神经网络的行为。

神经网络的复杂结构使手动推断其行为变得不切实际。这引发了很多自动验证神经网络属性的研究。

我们提出的方法基于知识编译，适用于具有离散输入输出的神经网络。

BNN在运行时具有二进制权重和激活。在训练时，二进制权重和激活值用于计算梯度参数。在前向传递过程中，BNN大大减少了内存的大小和访问量，并用按位运算代替了大多数算术运算，显著提高了功率和效率。

训练BNN时，将实数变量二值化的函数通常有两种，确定二值化和随机二值化。

确定二值化:

随机二值化:

是Hard sigmoid函数虽然随机二值化看起来更合理，但硬件在实现时比较困难，因此我们大多数时候直接使用确定二值化的sign函数。

BNN鲁棒性：

对于一个特定输入x，x的领域内的输入输出结果与x相同。举个例子，图片分类问题，我们期望如果只改动几个像素，狗的图像仍保持分类为狗。由于调整图像的方式数与修改像素数呈指数关系，因此枚举验证鲁棒性难以实现。

二进制神经网络是前馈神经网络，其中权重和激活使用{−1,1}进行二进制化。 一个BNN由内部块和一个输出块组成。 内部块由三层组成：线性变换（LIN），批处理归一化（BN）和二值化（BIN）。

LIN层具有参数a（权重）和b（偏差）。 给定输入x，此层返回⟨a，x⟩+ b。

BN层具有参数μ（平均值），σ（标准偏差），α（权重）和γ（偏差）。 给定输入y，该层返回α（y-μ）/σ +γ。

BIN层返回其输入的符号（1或-1）。

输出模块由LIN层和ARGMAX层组成。 ARGMAX层选择激活程度最高的输出类。 Narodytska等人给出了有关这些块和层及其确切定义的更多详细信息。 [26]。 为了方便起见，我们考虑输出为0或1的BNN

BNN->CNF:

Verifying Properties of Binarized Deep Neural Networks

OBDD->CNF：

Tseitin, G.: On the complexity of derivation in propositional calculus. Studies in constructive mathematics and mathematical logic pp. 115–125 (1968)

当BNN和OBDD共享相同的输入x时，我们可以用公式φ=α∧β∧（O∨R）∧（¬O∨¬R）来检查它们的不等式[26]。 然后，只要存在x的某个实例使得（O∧R）∨（O∧R）（即BNN和OBDD不相同），则φ是可满足的。

## 神经网络

神经元（Neuron）：就像形成我们大脑基本元素的神经元一样，神经元形成神经网络的基本结构。想象一下，当我们得到新信息时我们该怎么做。当我们获取信息时，我们一般会处理它，然后生成一个输出。类似地，在神经网络里，神经元接收输入，处理它并产生输出，而这个输出被发送到其他神经元用于进一步处理，或者作为最终输出进行输出。

权重（Weights）：当输入进入神经元时，它会乘以一个权重。例如，如果一个神经元有两个输入，则每个输入将具有分配给它的一个关联权重。我们随机初始化权重，并在模型训练过程中更新这些权重。训练后的神经网络对其输入赋予较高的权重，这是它认为与不那么重要的输入相比更为重要的输入。为零的权重则表示特定的特征是微不足道的。让我们假设输入为a，并且与其相关联的权重为W1，那么在通过节点之后，输入变为a\*W1。

偏差（Bias）：除了权重之外，另一个被应用于输入的线性分量被称为偏差。它被加到权重与输入相乘的结果中。基本上添加偏差的目的是来改变权重与输入相乘所得结果的范围的。添加偏差后，结果将看起来像a\*W1+偏差。这是输入变换的最终线性分量。

激活函数（Activation Function）：一旦将线性分量应用于输入，将会需要应用一个非线性函数。这通过将激活函数应用于线性组合来完成。激活函数将输入信号转换为输出信号。应用激活函数后的输出看起来像f(a\*W1 + b)，其中f()就是激活函数。

在下图中，我们将“n”个输入给定为X1到Xn而与其相应的权重为Wk1到Wkn。我们有一个给定值为bk的偏差。权重首先乘以与其对应的输入，然后与偏差加在一起。而这个值叫做u。U =ΣW\* X+ b，激活函数被应用于u，即f(u)，并且我们会从神经元接收最终输出，如 yk = f(u)。

正向传播（Forward Propagation）：正向传播是指输入通过隐藏层到输出层的运动。在正向传播中，信息沿着一个单一方向前进。输入层将输入提供给隐藏层，然后生成输出。这过程中是没有反向运动的。

反向传播（Backpropagation）：当我们定义神经网络时，我们为我们的节点分配随机权重和偏差值。一旦我们收到单次迭代的输出，我们就可以计算出网络的错误。然后将该错误与成本函数的梯度一起反馈给网络以更新网络的权重。最后更新这些权重，以便减少后续迭代中的错误。使用成本函数的梯度的权重的更新被称为反向传播。