A6.1 (5分) 分析卷积神经网络中用1×1的卷积核的作用

解答：

1. 特征图通道变换：1x1卷积可以改变特征图的通道数。它可以用来增加或减少

特征图的深度。例如，如果输入特征图有256个通道，而我们希望输出特征图只有64个通道，那么可以使用64个1x1的卷积核来实现。

1. 计算参数的减少：相比于使用大的卷积核（如3x3或5x5），1x1卷积核可以大大减少模型的参数数量。这有助于构建更轻量级的网络，减少计算资源消耗和过拟合的风险。
2. 非线性增强：虽然1x1卷积本身是线性操作，但在卷积层之后通常会接一个非线性激活函数（如ReLU）。这样，即使是小尺寸的卷积核也能增加网络的非线性表达能力。
3. 跨通道信息整合：1x1卷积可以实现不同通道之间的信息整合。在传统的卷积操作中，每个卷积核只作用于单个通道，而1x1卷积可以将不同通道的信息混合在一起，从而实现特征融合。
4. 网络深度的增加：在不显著增加计算负担的前提下，使用1x1卷积可以增加网络的深度，有助于学习更复杂的特征表示。
5. 维度压缩和特征选择：通过调整1x1卷积核的数量，可以对输入特征图进行维度压缩，这在某种程度上起到了特征选择的作用，有助于提升网络的泛化能力。

A6.2 (5分) 计算函数和函数的梯度。

解答：

1. 函数 的梯度：

这个函数返回所有输入中的最大值。在梯度的计算中，只有最大值对应的输入会对输出有影响，因此该输入的梯度为1，其余输入的梯度为0。

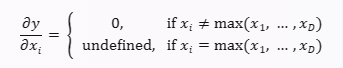


1. 函数y=argmax(x1,…,xD) 的梯度：

y = argmax(x1,⋯,xD)表示的是使函数y = max(x1,⋯,xD)取得最大值的x的索引，也就是说，y是一个整数，它的值是x中最大元素的位置。

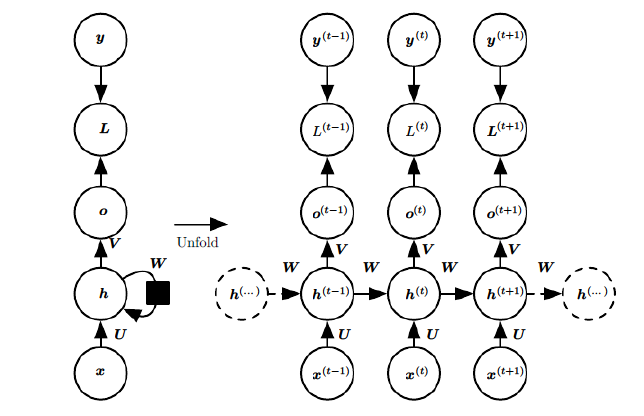
由于y是一个离散的函数，它的值只能在{1, 2, …, D}中取，因此，当x中有多个元素达到最大值时，y是不连续的，也就是说，y在这些点上是不可导的，梯度是未定义的。例如，如果x = (1, 2, 2, 1)，那么y可以是2或者3，这取决于argmax的实现方式，但无论如何，y在x的这个值上都是不可导的。

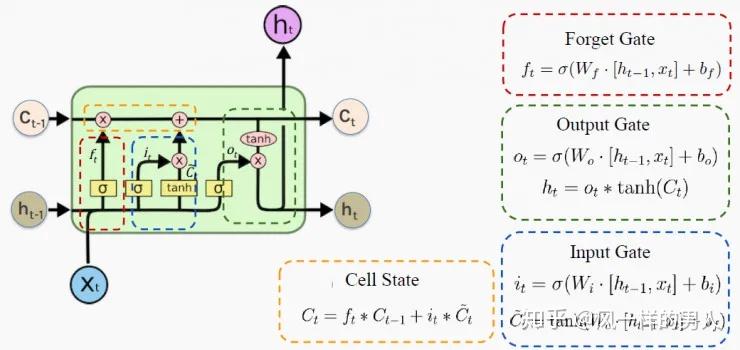
当x中只有一个元素达到最大值时，y是一个常数，它的值是这个元素的位置，因此，y在x的这个值上是可导的，梯度是零向量。例如，如果x = (1, 2, 3, 1)，那么y = 3，这是一个常数，所以y对x的任何分量的偏导数都是0。



A6.3 (5分) 推导LSTM网络中参数的梯度，并分析其避免梯度消失的效果。

解答：





LSTM网络的反向传播过程需要计算损失函数L对各个参数的梯度

LSTM网络由四个门控单元组成：输入门、遗忘门、输出门和单元状态。每个门控单元都有一个权重矩阵和一个偏置向量，分别记为 W 和 b 。LSTM网络的参数包括：

输入门的权重矩阵 和偏置向量

遗忘门的权重矩阵 和偏置向量

输出门的权重矩阵 和偏置向量

单元状态的权重矩阵 和偏置向量

对于隐藏层的参数，需要考虑每个门控单元的影响，得到：

输入门参数的梯度:

输入门权重梯度：

输入门偏置梯度：

遗忘门参数的梯度:

遗忘门权重梯度：

遗忘门偏置梯度：​​

输出门参数的梯度：

输出门权重梯度：

输出门偏置梯度：​​

单元状态参数的梯度:

单元状态权重梯度：

单元状态偏置梯度：

LSTM网络避免梯度消失的效果主要基于其独特的结构设计，这包括：

门控机制：

遗忘门（Forget Gate）：允许网络“忘记”不必要的信息，这有助于调节梯度流，并且防止梯度在反向传播时由于累乘效应而过小。

输入门（Input Gate）和输出门（Output Gate）：它们控制新信息的输入和当前单元状态的输出。这种控制能力意味着网络可以保护和传递重要的梯度，不让它们在多个时间步中消失。

常量误差轭（Constant Error Carousel, CEC）：

LSTM的核心是它的单元状态​，它的设计允许误差以不变的形式在单元之间循环。在理想情况下，这意味着梯度可以在很长的序列中传播而不衰减，这对于学习长期依赖至关重要。

线性依赖：

在更新单元状态时，部分更新是线性的。线性操作的梯度是常数，这有助于减少在反向传播时的梯度衰减。

激活函数的选择：

LSTM使用sigmoid和tanh函数，它们的导数在输入接近0时较大，在输入远离0时接近0。这意味着在门控信号较强时，梯度可以较好地流动，而在信号较弱时减少其影响，这有助于稳定梯度。

A6.4 (5分) 当将自注意力模型作为神经网络的一层使用时，分析它和卷积层以及循环层在建模长距离依赖关系的效率和计算复杂度方面的差异

解答：

1.长距离依赖关系的效率：指的是神经网络层能否有效地捕捉输入序列中不同位置之间的相关性，尤其是距离较远的位置。

自注意力模型是全连接的，它可以直接计算输入序列中任意两个位置之间的相似度，并根据相似度分配不同的权重，从而实现全局的信息整合。

卷积层是局部连接的，它通过一个固定大小的卷积核从局部感受野中提取特征，如果要捕捉长距离的依赖关系，就需要增加网络的深度或者扩大卷积核的大小，这会增加计算量和参数量。

循环层是顺序连接的，它通过一个隐藏状态来保存历史信息，并根据当前输入来更新隐藏状态，如果要捕捉长距离的依赖关系，就需要增加网络的宽度或者使用门控机制，这也会增加计算量和参数量。

因此，从效率的角度来看，自注意力模型可以更好地处理长距离依赖关系。

2.计算复杂度的差异：指的是神经网络层在前向传播和反向传播时所需的浮点运算数（FLOPs）和参数数量。

自注意力模型的计算复杂度主要取决于输入序列的长度和维度，以及注意力头的数量。如果输入序列的长度为n，维度为d，注意力头的数量为h，那么自注意力模型的FLOPs为O(n^2 d + n h d^2)，参数数量为O(h d2)。

卷积层的计算复杂度主要取决于输入特征图的尺寸和通道数，以及卷积核的大小和步长。如果输入特征图的尺寸为n\*n，通道数为c，卷积核的大小为k\*k，步长为s，那么卷积层的FLOPs为O(n2 c k^2 / s^2)，参数数量为O(c k^2)。

循环层的计算复杂度主要取决于输入序列的长度和维度，以及隐藏状态的维度。如果输入序列的长度为n，维度为d，隐藏状态的维度为h，那么循环层的FLOPs为O(n (d h + h^2))，参数数量为O(d h + h^2)。

因此，从计算复杂度的角度来看，自注意力模型和卷积层的FLOPs都随着输入序列的长度的平方增长，而循环层的FLOPs只随着输入序列的长度的线性增长，但是自注意力模型和卷积层的参数数量都随着输入序列的维度的平方增长，而循环层的参数数量只随着输入序列的维度的线性增长。