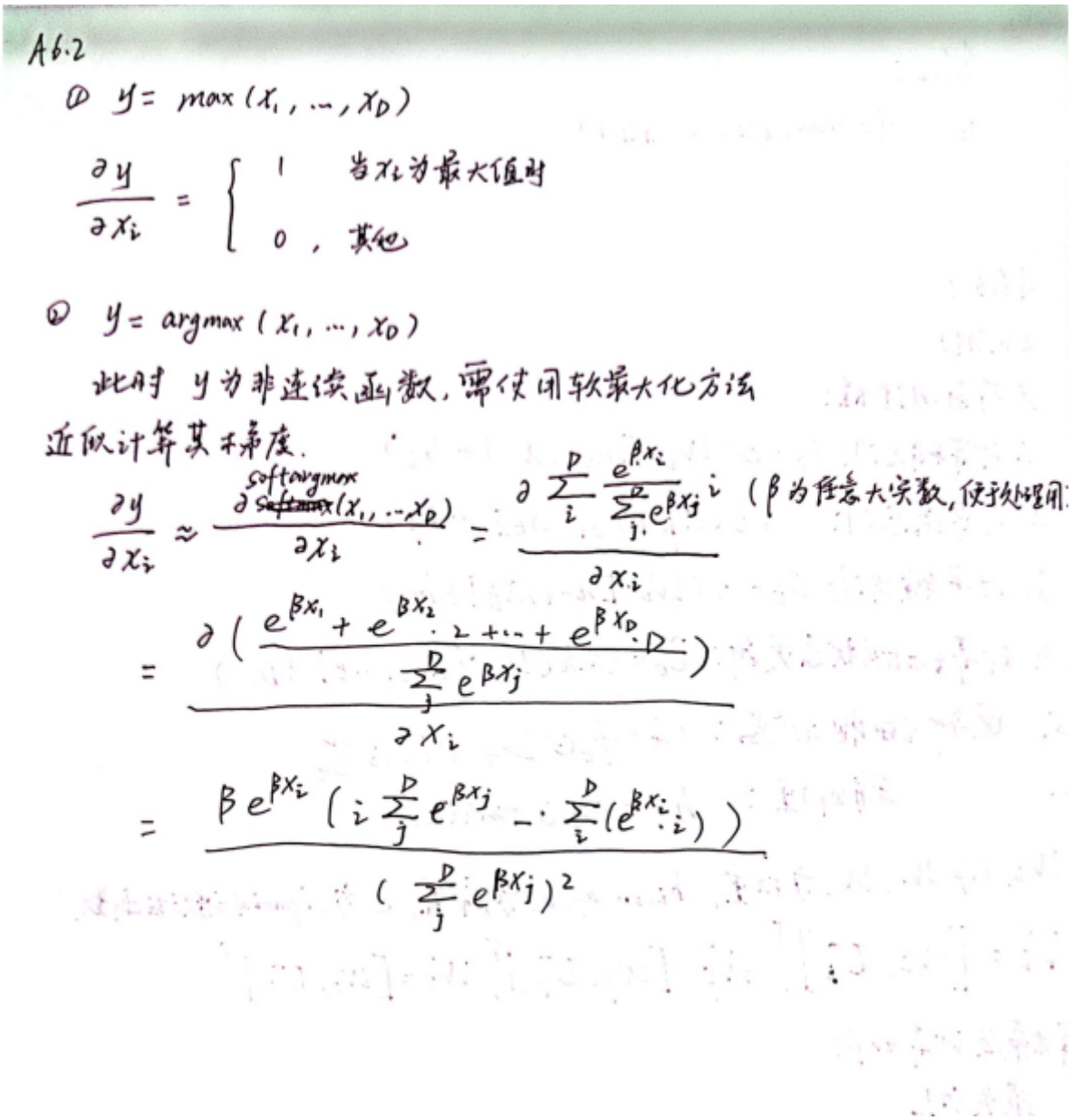
A6:分析与简答

A6.1 (5分) 分析卷积神经网络中用1×1的卷积核的作用。

在卷积神经网络（CNN）中，1×1的卷积核（也称为逐点卷积）扮演着多个重要的角色，包括但不限于：

1. **通道数的变换**：1×1卷积可以改变经过前面层处理后的特征图（feature map）的深度，即通道数。它可以用来增加或减少特征图的通道数，这在控制网络参数量和计算量方面非常有用。例如，在Inception网络结构中，1×1卷积被用来减少输入到更大卷积核（如3×3或5×5）的特征图的深度，从而减少计算复杂度。
2. **增加非线性**：虽然1×1卷积核自身不能捕获空间信息，但是通过在其后应用非线性激活函数（如ReLU），可以帮助网络学习复杂的函数映射。这一点对于增加网络的表达能力是有帮助的。
3. **计算效率**：1×1卷积核在计算上比较高效，特别是当用于降低通道数时，因为它们对每个位置仅应用一次卷积操作，减少了计算量。
4. **特征融合**：1×1的卷积可以在不同的通道之间进行信息融合。因为每个1×1卷积核会作用于前一层所有通道的同一空间位置，所以它能够将不同通道的信息结合起来，生成新的特征图。
5. **保持空间分辨率**：1×1卷积不改变特征图的空间尺寸（宽度和高度保持不变），这对于构建深层网络是有益的，因为它允许更多层进行操作而不丢失空间分辨率。
6. **实现跨通道参数化的交互和信息整合**：通过1×1卷积核的这种方式，网络能够创建跨通道的特征组合，有助于模型捕获输入数据中的不同表示。
7. **降低参数和计算成本**：在某些情况下，1×1卷积层可以替代全连接层，这可以显著降低参数的数量，因为它们只在每个位置进行线性组合，而不是在所有位置进行组合，这样可以有效减少模型的复杂性和过拟合的风险。

A6.2 (5分) 计算函数y = max(x(1),...,x(D))和函数y = argmax(x(1),...,x(D))的梯度。



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。

A6.3：

LSTM：

先有前向过程：

1. 计算输入门：$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

2. 计算遗忘门：$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$

3. 计算输出门：$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

4. 计算细胞状态更新：$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$

5. 更新细胞状态：$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$

6. 输出值：$h_t = o_t \odot \tanh(C_t)$

W_i, W_f, W_o, W_c 为权重，b_i, b_o, b_f, b_c 为偏置，σ 为 sigmoid 激活函数

$W_i = [w_i, U_i]^T$，$W_f = [w_f, U_f]^T$，$W_o = [w_o, U_o]^T$

有梯度计算如下：

损失为 L

$$\frac{\partial L}{\partial W_o} = \sum_{t=1}^{T} \frac{\partial L}{\partial h_t} \odot \tanh(C_t) \odot \sigma'(W_o \cdot [h_{t-1}, x_t + b_o) \cdot [h_{t-1}, x_t]^T$$

$$\frac{\partial L}{\partial W_i} = \sum_{t=1}^{T} \frac{\partial L}{\partial C_t} \odot \tilde{C}_t \odot \sigma'(W_i \cdot [h_{t-1}, x_t] + b_i) \cdot [h_{t-1}, x_t]^T$$

$$\frac{\partial L}{\partial W_f} = \sum_{t=1}^{T} \frac{\partial L}{\partial C_t} \odot C_{t-1} \odot \sigma'(W_f \cdot [h_{t-1}, x_t] + b_f) \cdot [h_{t-1}, x_t]^T$$

$$\frac{\partial L}{\partial W_c} = \sum_{t=1}^{T} \frac{\partial L}{\partial C_t} \odot i_t \odot \tanh'(W_c \cdot [h_{t-1}, x_t] + b_c) \cdot [h_{t-1}, x_t]^T$$

$$\frac{\partial L}{\partial b_o} = \sum_{t=1}^{T} \frac{\partial L}{\partial h_t} \odot \tanh(C_t) \odot \sigma'(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\frac{\partial L}{\partial b_i} = \sum_{t=1}^{T} \frac{\partial L}{\partial C_t} \odot \tilde{C}_t \odot \sigma'(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\frac{\partial L}{\partial b_f} = \sum_{t=1}^{T} \frac{\partial L}{\partial C_t} \odot C_{t-1} \odot \sigma'(W_f [h_{t-1}, x_t] + b_f)$$

$$\frac{\partial L}{\partial b_c} = \sum_{t=1}^{T} \frac{\partial L}{\partial C_t} \odot i_t \odot \sigma'(W_c [h_{t-1}, x_t] + b_c)$$

$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_t} + \frac{\partial L}{\partial C_t} \odot o_t \odot \tanh'(C_t) \cdot W_o^T$$

$$\frac{\partial L}{\partial h_{t-1}} = \frac{\partial L}{\partial h_{t-1}} + \frac{\partial L}{\partial C_t} \odot o_t \odot \tanh'(C_t) \cdot W_o^T$$

LSTM 如何避免梯度消失：

原始 RNN 中

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \frac{\partial L}{\partial y^{(T)}} \cdot \frac{\partial y^{(T)}}{\partial o^{(T)}} \cdot \frac{\partial o^{(T)}}{\partial h^{(T)}} \left(\prod_{k=t+1}^{T} \tanh'(z^{(k)}) W \right) \frac{\partial h^{(t)}}{\partial W}$$

其中 $\tanh'(z^{(k)}) = \mathrm{diag}(1 - (z^{(k)})^2) \leq 1$，若 W 特征值小于 1，梯度则会消失，W 特征值大于 1，梯度则爆炸。

而 LSTM 中

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \frac{\partial L}{\partial y^{(T)}} \cdot \frac{\partial y^{(T)}}{\partial o^{(T)}} \cdot \frac{\partial o^{(T)}}{\partial C^{(T)}} \prod_{k=t+1}^{T} \frac{\partial C^{(k)}}{\partial C^{(k-1)}} \cdot \frac{\partial C^{t}}{\partial W}$$

$\prod_{k=t+1}^{T} \frac{\partial C^{(k)}}{\partial C^{(k-1)}} = (f^{(k)} f^{(k+1)} \cdots f^{(T)}) + \text{other}$，每一步 $\frac{\partial C^{k}}{\partial C^{k-1}}$ 可自由选择在 $[0,1]$ 间，或大于 1，因为 f 可学习，故梯度在远距离不至于完全消失，能够解决梯度消失问题。

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。

当自注意力模型作为神经网络的一层使用时，它与卷积层和循环层在建模长距离依赖关系的效率和计算复杂度方面存在显著差异。

自注意力层：

优势：

- **长距离依赖**：自注意力机制能够直接计算序列中任意两点之间的依赖关系，无论这两点之间的距离有多远。这使得自注意力非常适合处理长距离依赖问题。
- **并行计算**：自注意力机制允许在处理序列时并行计算所有的依赖关系，这在现代多核硬件上特别有优势。

劣势：

- **计算复杂度**：自注意力层的计算复杂度随着输入序列长度的增加而呈平方增长，这在处理非常长的序列时可能会导致显著的计算成本。
- **空间复杂度**：除了计算复杂度，自注意力层还需要存储注意力矩阵，其空间复杂度也是平方级别的。

卷积层：

优势：

- **局部连接**：卷积层通过局部感受野捕捉空间或时序数据中的局部特征，这对于图像等数据非常有效。
- **参数共享**：卷积层的权重在整个输入域中共享，减少了模型的参数数量。

劣势：

- **固定的感受野**：卷积层的感受野大小是固定的，这意味着为了捕捉长距离的依赖关系，需要多层卷积堆叠，这可能会导致效率低下。
- **序列问题**：对于序列数据，尤其是长度可变的序列，卷积层不如自注意力机制直观。

循环层（如LSTM或GRU）：

优势：

- **序列数据**：循环层天然适合处理序列数据，特别是对时间依赖性建模。
- **可变长度处理**：它们可以处理不同长度的序列，而不需要输入修剪或填充。

劣势：

- **顺序计算**：循环层必须按顺序处理序列的每个元素，这限制了它们的并行能力。
- **长距离依赖**：虽然理论上循环层可以处理长时间的依赖，但在实践中，特别是在非常长的序列中，它们可能会遇到梯度消失或梯度爆炸的问题。

综合来看，自注意力模型在建模长距离依赖关系方面具有明显优势，尤其是它们可以直接计算任意两点之间的依赖，而不受限于固定的感受野或必须顺序处理的限制。然而，这种能力是以更高的计算和空间复杂度为代价的，特别是对于长序列数据。卷积层和循环层在某些情况下可能更加高效，尤其是在序列较短或者需要强调局部特征时。