Thought for 24 seconds思考24秒

(a)（一个）

In a fully connected neural network layer, each output neuron is connected to every input neuron. For each spatial position (i, j), the output feature map $y_{i,j,k}$ is computed as a weighted sum over all input channels at that position, plus a bias:

在完全连接的神经网络层中，每个输出神经元都连接到每个输入神经元。对于每个空间位置 (i, j)，输出特征图 $y_{i,j,k}$ 计算为该位置所有输入通道的加权和，加上偏差：

$$y_{i,j,k} = \sum_{l=1}^{C} w_{l,k}\, x_{i,j,l} + b_k$$

- $w_{l,k}$ are the weights connecting input channel l to output channel k.
  $w_{l,k}$ 是连接输入通道的权重 l 至输出通道 k 。
- b_k is the bias for output channel k.b_k 是输出通道的偏置 k 。

Number of learnable parameters:可学习参数的数量：

- Weights: $C \times D$ (since each of the D output channels connects to all C input channels).
  重量： $C \times D$ （因为每个 D 输出通道连接到所有 C 输入通道）。
- Biases: D (one bias per output channel).偏见： D （每个输出通道一个偏置）。

Total parameters: $C \times D + D$**总参数：** $C \times D + D$

(b)(二)

In a spatial convolutional neural network with a 3×3 filter size, each output feature at position (i, j) depends on a 3×3 neighborhood in each input channel. The output is computed as:

在空间卷积神经网络中 3×3 滤波器大小，每个输出特征在位置 (i, j) 取决于一个 3×3 每个输入通道中的邻域。输出计算如下：

$$y_{i,j,k} = \sum_{u=-1}^{1} \sum_{v=-1}^{1} \sum_{l=1}^{C} w_{u,v,l,k}\, x_{i+u,j+v,l} + b_k$$

- $w_{u,v,l,k}$ are the weights of the convolutional filters.$w_{u,v,l,k}$ 是卷积滤波器的权重。
- b_k is the bias for output channel k.b_k 是输出通道的偏置 k 。

Number of learnable parameters:可学习参数的数量：

- Weights per filter: $3 \times 3 \times C = 9C$每个过滤器的重量： $3 \times 3 \times C = 9C$
- Total weights: $9C \times D$ (since there are D filters).
  总重量： $9C \times D$ （因为有 D 过滤器）。
- Biases: D偏见： D

Total parameters: $9C \times D + D$**总参数：** $9C \times D + D$

(c)(三)

For a convolutional layer with a 1×1 filter size, the output at each position depends only on the input at the same position:

对于具有 1×1 过滤器大小，每个位置的输出仅取决于同一位置的输入：

$$y_{i,j,k} = \sum_{l=1}^{C} w_{l,k}\, x_{i,j,l} + b_k$$

- $w_{l,k}$ are the weights of the 1×1 filters.$w_{l,k}$ 是的权重 1×1 过滤器。
- b_k is the bias for output channel k.b_k 是输出通道的偏置 k 。

Number of learnable parameters:可学习参数的数量：

- Weights: $C \times D$重量： $C \times D$
- Biases: D偏见： D

Total parameters: $C \times D + D$**总参数：** $C \times D + D$

(d)(四)

Using a single index i for spatial positions ($i = 1, 2, \ldots, PQ$), the expression without biases becomes:使用单个索引 i 对于空间位置（ $i = 1, 2, \ldots, PQ$ ），没有偏差的表达式变为：

$$y_{i,k} = \sum_{l=1}^{C} w_{l,k}\, x_{i,l}$$

- Here, $x_{i,l}$ is the input at position i and channel l.这里， $x_{i,l}$ 是位置处的输入 i 和频道 l 。
- $y_{i,k}$ is the output at position i and channel k.$y_{i,k}$ 是位置处的输出 i 和频道 k 。
- $w_{l,k}$ are the weights connecting input channel l to output channel k.
  $w_{l,k}$ 是连接输入通道的权重 l 至输出通道 k 。

Number of learnable parameters: $C \times D$ (since biases are omitted)

可学习参数的数量： $C \times D$ （因为省略了偏差）

(e)（五）

Arranging the inputs and outputs into matrices:将输入和输出排列成矩阵：

- X is a $PQ \times C$ matrix where each row i corresponds to the input feature vector $x_{i,:}$ at spatial position i.
  X 是一个 $PQ \times C$ 矩阵，其中每行 i 对应于输入特征向量 $x_{i,:}$ 在空间位置 i 。
- Y is a $PQ \times D$ matrix where each row i corresponds to the output feature vector $y_{i,:}$ at spatial position i.
  Y 是一个 $PQ \times D$ 矩阵，其中每行 i 对应于输出特征向量 $y_{i,:}$ 在空间位置 i 。

The matrix equation becomes:矩阵方程变为：

$$Y = XW$$

- W is a $C \times D$ weight matrix.W 是一个 $C \times D$ 权重矩阵。

Conclusion about the relation between a convolutional neural network and a Transformer:关于卷积神经网络和 Transformer 之间关系的结论：

The operation $Y = XW$ shows that a convolutional neural network with 1×1 filters (and no biases) performs a linear transformation on the input features at each spatial position, identical across all positions. This is similar to the linear layers used in Transformers, where inputs are transformed via weight matrices. Both architectures apply matrix multiplications to process inputs, highlighting a fundamental similarity in how they handle data transformations.

操作 $Y = XW$ 表明卷积神经网络 1×1 过滤器（并且无偏差）对每个空间位置的输入特征执行线性变换，在所有位置上都是相同的。这类似于 Transformer 中使用的线性层，其中输入通过权重矩阵进行转换。两种架构都应用矩阵乘法来处理输入，突出了它们处理数据转换方式的基本相似性。

Final Answer:最终答案：

(a) $y_{i,j,k} = \sum_{l=1}^{C} w_{l,k} x_{i,j,l} + b_k$; Number of parameters: $C \times D + D$
（一个） $y_{i,j,k} = \sum_{l=1}^{C} w_{l,k} x_{i,j,l} + b_k$;参数数量： $C \times D + D$