对比维度 / 考点	MLP	CNN	RNN	LSTM	Transformer
更适合处理什么数据	固定长度向量、表格型输入。	图像、空间网格数据。	序列、文本、时序数据。	序列、文本、时序数据。	序列、文本；扩展到多模态（比如图像）时，只要能把数据映射成 token，也可以按类似思路处理。
输入怎么进入网络	整体向量一次送入。	局部窗口在空间上滑动进入。	逐 token 串行送入。	逐 token 串行送入。	整段 token 并行送入。
核心机制	全连接层 + 非线性激活。	卷积、局部连接、参数共享、池化。	隐藏状态递推。	RNN 基础上引入记忆状态和门控机制。	注意力机制 + 位置编码。
最核心的区分	整体展平成向量输入，无记忆。	显式利用图像局部空间结构，局部连接，无记忆。	隐藏状态用作记忆，按时间顺序一步一步传隐藏状态。	是改进的 **RNN**；仍按顺序传隐藏和记忆状态，加入门控机制更容易保留长期信息。	直接用注意力机制让 **token** 两两建立关系，不靠一步一步传。
是否天然适合变长序列	否。	否。	是。	是。	是。
是否有参数共享	没有。	有。卷积核在不同位置复用。	有。时间步共享 $U/W/V$；若含偏置，b, b_y 也共享。	有。和 RNN 一样在时间步共享参数。	有。层内线性映射参数共享于各 token 位置。
历史/上下文信息怎么保留	没有专门记忆机制。	不靠时间记忆，主要抓局部空间结构。	通过隐藏状态 S_t 传递历史信息。	通过隐藏状态和记忆状态共同保留信息。	不靠递推隐藏状态，而是直接建模 token 间关系。
是否容易并行	好。	好。	差。	差。	好。
最擅长什么	向量映射、分类/回归基础结构。	图像局部空间关联、层级特征提取。	顺序依赖、基础时序建模。	比 RNN 更强的长期信息保留。	全局关系、长距离关系建模、并行训练。
结构要求与局限	固定大小输入，图像展平成向量输入，忽略空间结构。	局部连接，滑动输入。	长程依赖难保留；BPTT 训练困难。	只是一定程度上改进了 RNN，更容易保留长期信息（太长的还是不行），但仍然是串行结构。	用注意力机制直接让 token 两两建立关系，更容易直接建模长距离依赖；注意力本身不关心顺序，所以输入要显式加入位置编码。
代表性公式	$y = \sigma(Wx + b)$；x 是输入向量，W 是权重矩阵，b 是偏置，σ 是逐元素激活函数。若题目说明没有偏置，则不写 b。	$H_{\text{out}} = \lfloor \frac{H+2P-K}{S} \rfloor + 1$，$W_{\text{out}} = \lfloor \frac{W+2P-K}{S} \rfloor + 1$；$H, W$ 是输入高和宽，P 是 padding，K 是卷积核大小，S 是 stride；C_{out} 由卷积核个数决定。	$S_t = f(Ux_t + WS_{t-1} + b)$，$o_t = g(VS_t + b_y)$；$x_t$ 是当前输入，S_{t-1} 是上一时刻隐藏状态，S_t 是当前隐藏状态，o_t 是输出，U, W, V 是权重矩阵，b, b_y 是偏置；若无偏置则去掉 b, b_y。	遗忘门、输入门、输出门；记忆状态 C_t。	Attention(Q, K, V) = softmax$(QK^T/\sqrt{d_k})V$；Q 是 Query，表示当前 token 要查什么；K 是 Key，表示可被匹配的标识；V 是 Value，表示被加权汇总的信息。若 $Q \in R^{m \times d_k}$，$K \in R^{n \times d_k}$，$V \in R^{n \times d_v}$，则输出为 $R^{m \times d_v}$；K 和 V 的数量必须相同，Q 和 K 的最后一维必须相同。位置编码表示顺序。

对比维度 / 考点	MLP	CNN	RNN	LSTM	Transformer
参数量会考什么	若含偏置：$d_{\text{out}}d_{\text{in}} + d_{\text{out}}$；若不含偏置：$d_{\text{out}}d_{\text{in}}$。	若含偏置：$K^2C_{\text{in}}C_{\text{out}} + C_{\text{out}}$；若不含偏置：$K^2C_{\text{in}}C_{\text{out}}$。注意不乘 H_{out} 和 W_{out}。	若含偏置：$d_h d_x + d_h^2 + d_y d_h + d_h + d_y$；对应 U, W, V, b, b_y。若不含偏置，去掉最后 $d_h + d_y$。其中 $U \in R^{d_h \times d_x}$，$W \in R^{d_h \times d_h}$，$V \in R^{d_y \times d_h}$。	这份对比表里不展开整套参数量，只记住比 RNN 结构更复杂。	这份对比表里不展开 Transformer 整层参数量，重点放在注意力公式和 Q/K/V 维度关系。
高频易错考点（判断 / 填空 / 混淆）	参数量；偏置是否计入要看题目；激活函数逐元素作用；如果层间全是线性，多层整体仍等价线性变换；不能说 MLP 天然适合图像和文本序列。	卷积核个数决定输出通道数；padding/stride 如何影响输出大小；池化通常没有可学习参数；padding 不会增加可学习参数；卷积偏置是否计入要看题目；池化不是必须层。	参数共享；当前状态依赖当前输入和上一时刻隐藏状态；能处理变长靠参数共享和重复使用同一单元，不是因为不同时间步使用不同权重。	遗忘门决定保留多少旧记忆；输入门决定写入多少新信息；输出门决定输出多少记忆信息；它是改进的 RNN，只是一定程度上缓解长期信息难保留的问题，并没有彻底解决 RNN 的问题。	位置编码表示顺序；Q 表示当前 token 要查什么，K 表示可被匹配的标识，V 是被汇总的信息；LayerNorm 逐 token 计算；causal mask 遮住未来位置；self-attention 中 Q/K/V 来自同一序列；K 和 V 数量必须相同，Q 和 K 最后一维必须相同。
最容易混的 4 组区别	1. **MLP vs CNN**：关键不是“层数多少”，而是有没有显式利用空间局部结构。MLP 整体看向量，CNN 用局部卷积窗口看图像。 2. **RNN vs LSTM**：二者都按时间步串行处理序列；LSTM 比 RNN 多了记忆状态和门控机制，所以更容易保留长期信息。 3. **LSTM vs Transformer**：LSTM 仍然是串行传历史；Transformer 不靠一步一步传，而是直接计算 token 之间的关系。 4. **CNN vs Transformer**：CNN 主要利用局部空间结构；Transformer 主要利用 token 之间的全局关联关系。				