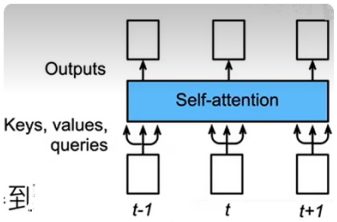
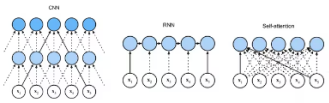
自注意力和位置编码

- 自注意力
  - 给定序列x1…xn，其中xi是长为d的向量
  - 自注意力池化层将xi当做key, value, query对序列抽取特征得到y1…yn
  - . $y_i = f(x_i, (x_1, x_1), \dots, (x_n, x_n))$



到

- 与CNN，RNN对比

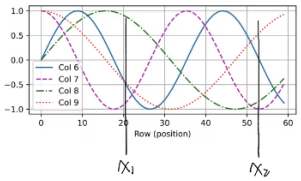


	CNN	RNN	自注意力
计算复杂度	O(knd^2)	O(nd^2)	O(n^2d)
并行度	O(n)	O(1)	O(n)
最长路径	O(n/k)	O(n)	O(1)

最长路径即CV中感受野

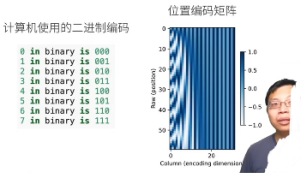
自注意力机制在求序列中计算会很慢

- 位置编码
  - 与CNN，RNN相比，自注意力机制没有记录位置信息
  - 位置编码将位置信息注入输入里
    - 假设长为n的序列是 $X \in R^{n \times d}$ 那么使用位置编码矩阵 $P \in R^{n \times d}$ 来计算输出X+P作为自编码输入
    - P的元素如下计算：
    - . $p_{i,2j} = \sin(i/10000^{2j/d})$，$p_{i,2j+1} = \cos(i/10000^{2j/d})$



(i, j)代表 P的第i行第j列

  - 样本x之间不会存在所有维度值全都相同的情况，x位置编码唯一



左图计算机的二进制编码可视作用一个三维特征来表示 0~7 的位置，每个维度在 0, 1 变化，最前面的变化慢

  - 位置编码和计算机的二进制编码相似，核心思想是对序列中的第i个样本，给定长为d的独一无二的位置信息，然后加入到数据中作为自编码输入，使得模型能够看到数据的位置信息。
- 相对位置信息

记 $\omega_j = 1/10000^{2j/d}$，那么

$$\begin{bmatrix} \cos(\delta\omega_j) & \sin(\delta\omega_j) \\ -\sin(\delta\omega_j) & \cos(\delta\omega_j) \end{bmatrix} \begin{bmatrix} p_{i,2j} \\ p_{i,2j+1} \end{bmatrix} = \begin{bmatrix} p_{i+\delta,2j} \\ p_{i+\delta,2j+1} \end{bmatrix}$$

更关注相对位置

投影矩阵跟 i 无关

  - 在一个序列中，假设一个词出现在另外一个词某个位置的时候，不管这对词出现在序列中的什么位置，都是可以通过一个同样的线性变换查找出来的
  - 这样编码的好处在于模型能够更加关注相对的位置信息，而不是关注一个词出现在一个句子中的绝对位置