Transformers and Attention

Time Series Modeling

Time series prediction 指的是预测

$$y_{1:T} = f_\theta(x_{1:T}) \tag{2}$$

其中 y_t 只能依赖于 $x_{1:t}$

RNN "latent state" Approach

RNN 的做法是在每一步维护一个 latent state h_t 来概括目前为止的所有信息。

优点：允许无限长的历史信息，扩展性好，对历史信息的表示很紧凑。

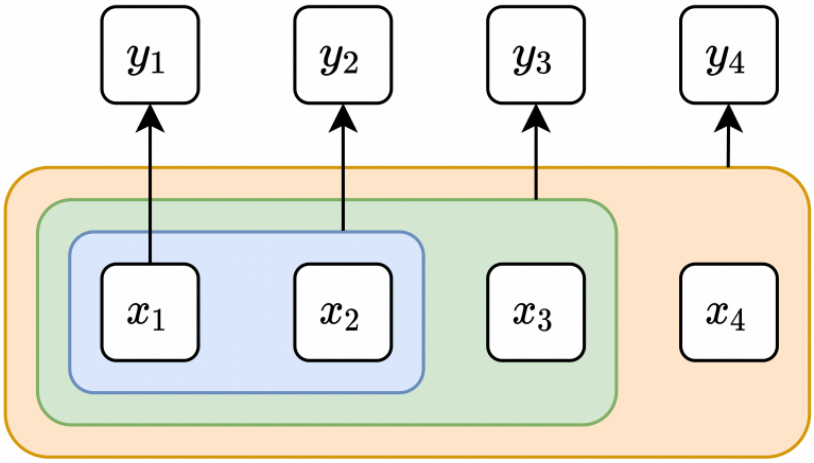
缺点：计算路径长，导致梯度爆炸/消减，难以训练。

The “direct prediction” Approach

Direct prediction 直接取前 t 个输入来产生预测结果 y_t, 即

$$y_t = f_\theta(x_{1:t}) \tag{3}$$

只需要一个能对不同数量的输入进行预测的函数即可。



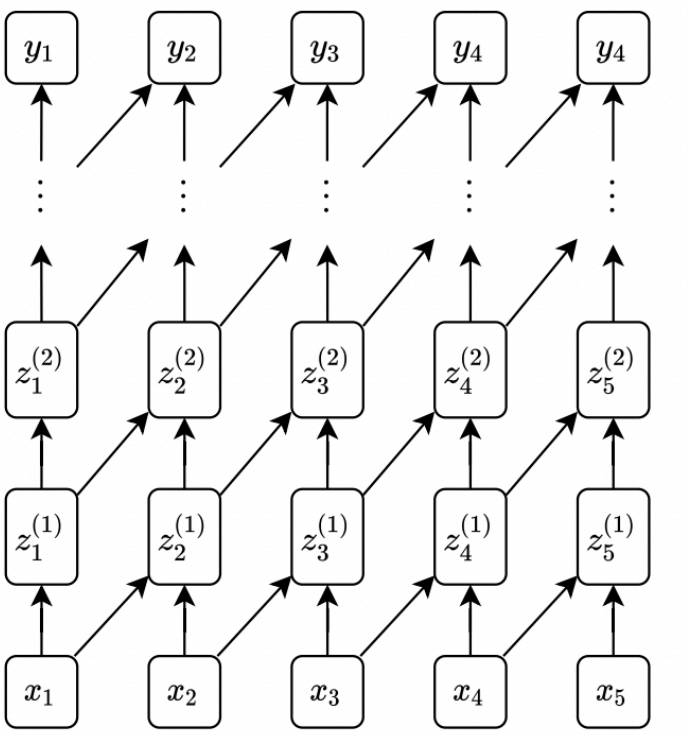
优点：计算路径短

缺点：没有紧凑的状态表示，实践中允许的历史信息量有限。

CNNs for Direct Prediction

核心思想：对卷积进行约束，$z_t^{(i+1)}$ 只能依赖于 $z_{t-k:t}^{(i)}$

这样的 CNN 称为 TCN (Temporal Convolutional Networks)



优点：简单

缺点：感受野 (receptive field) 受限，无法考虑到所有之前的输入，如果要考虑，就必须增加网络深度，导致参数量增加。

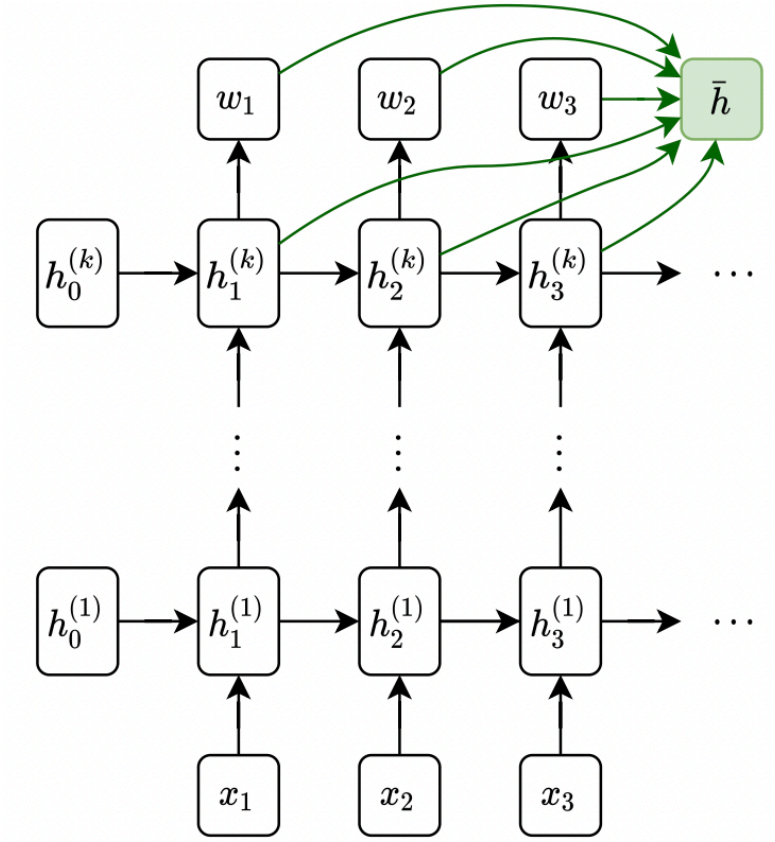
解决方法：

- 增大 kernel size: 但同时也会增加网络参数量
- Pooling layer: 不适合密集预测 (dense prediction)
- Dilated convolutions: 会跳过部分过去的状态/输入

Self-attention and Transformers

Attention

Attention 通常指将所有状态加权结合



$$
\begin{aligned}
z_t &= \theta^T h_t^{(k)} \\
w &= softmax(z) \\
\overline{h} &= \sum_{t=1}^{T} w_t h_t^{(k)}
\end{aligned}
\tag{4}
$$

核心思想：RNN 中过往输入的信息的传播路径比当前输入更长，由于梯度随路径衰减，过往输入与当前输入的占比不对等。因此 attention 考虑所有的 latent state, 避免不对等问题。

Self-attention

Self-attention 是 attention 机制的一个特例。

给定三个输入 $K, Q, V \in R^{T \times d}$, 定义 self-attention 操作如下：

$$
SelfAttention(K, Q, V) = softmax(\frac{KQ^T}{d^{1/2}})V \tag{5}
$$

其中 softmax 是对每行执行，$softmax(\frac{KQ^T}{d^{1/2}})$ 相当于一个 $T \times T$ 的权重矩阵，$K = Z^{(i)}W_K, Q = Z^{(i)}W_Q, V = Z^{(i)}W_V$

特点：

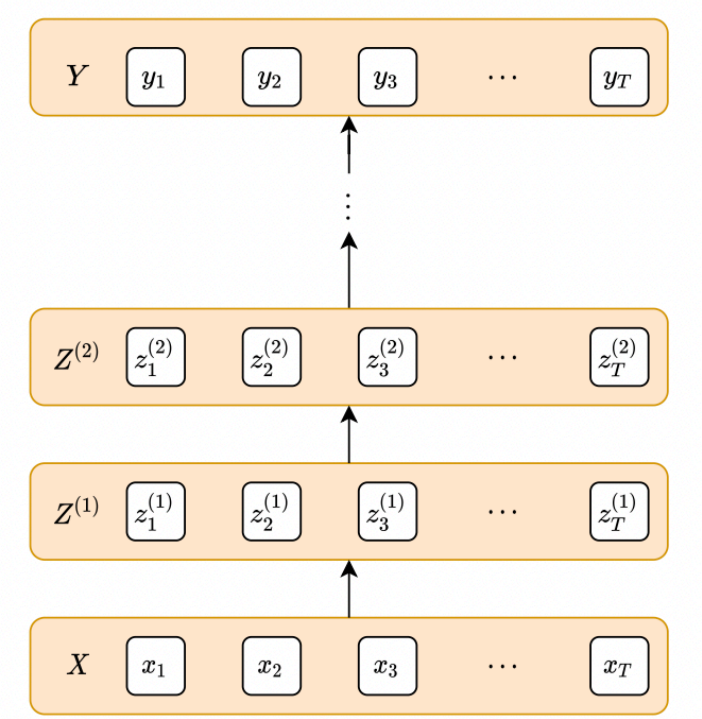
- Permutation invariant: 特征之间没有空间关系，可以任意变化
- 计算复杂度为 $O(T^2 + Td)$, 由于对 $T \times T$ 矩阵的非线性运算，难以降低

Transformers for Time Series

Transformer 用 attention 机制和 feedward 层来处理 time series:

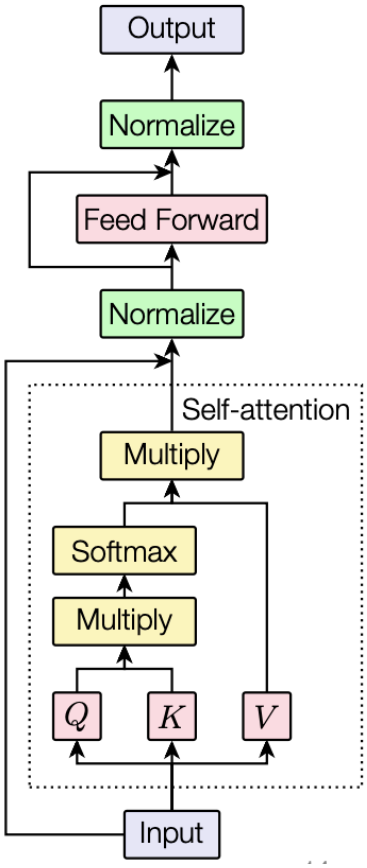
$$Z^{(i+1)} = Transformer(Z^{(i)}) \tag{6}$$

所有 time steps 是并行执行的，避免了 RNN 中所需的顺序处理。



Transformer Block

Transformer block 有如下结构：



$$
\begin{aligned}
\tilde{Z} &:= \text{SelfAttention}\left(Z^{(i)}W_K, Z^{(i)}W_Q, Z^{(i)}W_V\right) \\
&= \text{softmax}\left(\frac{Z^{(i)}W_K W_V^T {Z^{(i)}}^T}{d^{1/2}}\right) Z^{(i)}W_V \\
\tilde{Z} &:= \text{LayerNorm}\left(Z^{(i)} + \tilde{Z}\right) \\
Z^{(i+1)} &:= \text{LayerNorm}(\text{ReLU}(\tilde{Z}W) + \tilde{Z})
\end{aligned}
$$

首先经过 self-attention, 再经过 linear layer 和 ReLU.

优点：

- 感受野完整，能直接使用过去的数据
- 相比 CNN，计算两个位置之间的关联所需的操作次数不随距离增长，参数量不增加。

缺点：

- 输出都依赖于所有输入
- 没有对数据进行序列化，特征之间没有空间位置关系 (permutation invariant)

Masked Self-attention

核心思想：为了解决输出依赖于所有输入的问题，确保当前输出只依赖于目前为止的输入，可以将未来时间片的权重调整为 0.

$$\text{softmax}\left(\frac{KQ^T}{d^{1/2}} - M\right)V, \qquad M = \begin{bmatrix} & \infty \\ 0 & \end{bmatrix}$$

设为 ∞ 是因为 softmax 函数在负无穷时趋于 0.

虽然可以在技术上避免产生对未来输入的依赖，但直接 mask 掉更为简单。

Positional Encodings

核心思想：为了解决没有序列化的问题，可以将位置信息编码到输入中。

$$X \in \mathbb{R}^n = \begin{bmatrix} - & x_1^\top & - \\ - & x_2^\top & - \\ & \vdots & \\ - & x_T^\top & - \end{bmatrix} + \begin{bmatrix} \sin(\omega_1 \cdot 1) & \cdots & \sin(\omega_n \cdot 1) \\ \sin(\omega_1 \cdot 2) & \cdots & \sin(\omega_n \cdot 2) \\ \vdots & \ddots & \vdots \\ \sin(\omega_1 \cdot T) & \cdots & \sin(\omega_n \cdot T) \end{bmatrix}$$

其中 w_i 根据对数时间线 (logarithmic schedule) 决定。

实际中会编码到 X 的 d 维中。

Transformers beyond Time Series

Transformer 在图像领域也很强大，是目前的主流框架。

关键挑战在于：

- 如何表示数据以达到 $O(T^2)$ 的复杂度
- 如何编码位置信息
- 如何构建 mask 矩阵