SimVP: Simpler yet Better Video Prediction,

Z. Gao, C. Tan, L. Wu and S. Z. Li,

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

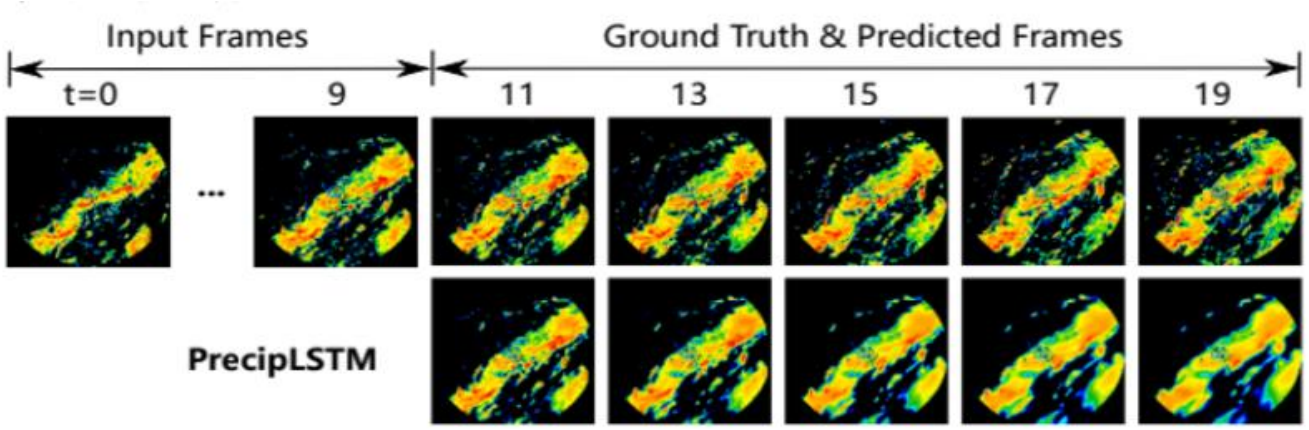
Citations： 235

王少丰

2024-12-2

时空序列预测

- **时空序列预测**是一个跨越**空间**和**时间**两个维度的预测问题，通常涉及预测某些时空数据，如交通流量、气象数据、环境监测数据等。在这一任务中，难点是在捕捉**时间序列**的变化的同时，还需要处理**空间维度**的依赖关系。



背景

归纳偏置（Inductive Bias） 是指模型在学习过程中所依赖的先验假设或假定，这些假设帮助模型更有效地学习数据的结构。不同的神经网络架构具有不同的归纳偏置，导致它们在处理不同类型数据时表现出不同的优势和局限性

- CNN
  - 局部性（Locality）：图像中的某些特征是局部的、重复出现的
  - 平移不变性（Translation Invariance）:假设特征在图像中是平移不变的
- RNN
  - 时间顺序性（Temporal Ordering）：RNN 假设数据具有时间或顺序依赖，即当前时刻的数据不仅与当前输入相关，还与先前时刻的数据有依赖关系。
  - 时间不变性（Time Invariance）：每个时间步上都使用相同的计算方式和参数。
- ViT
  - 每个局部区域（即图像块）都可能与其他局部区域之间存在全局依赖关系

背景

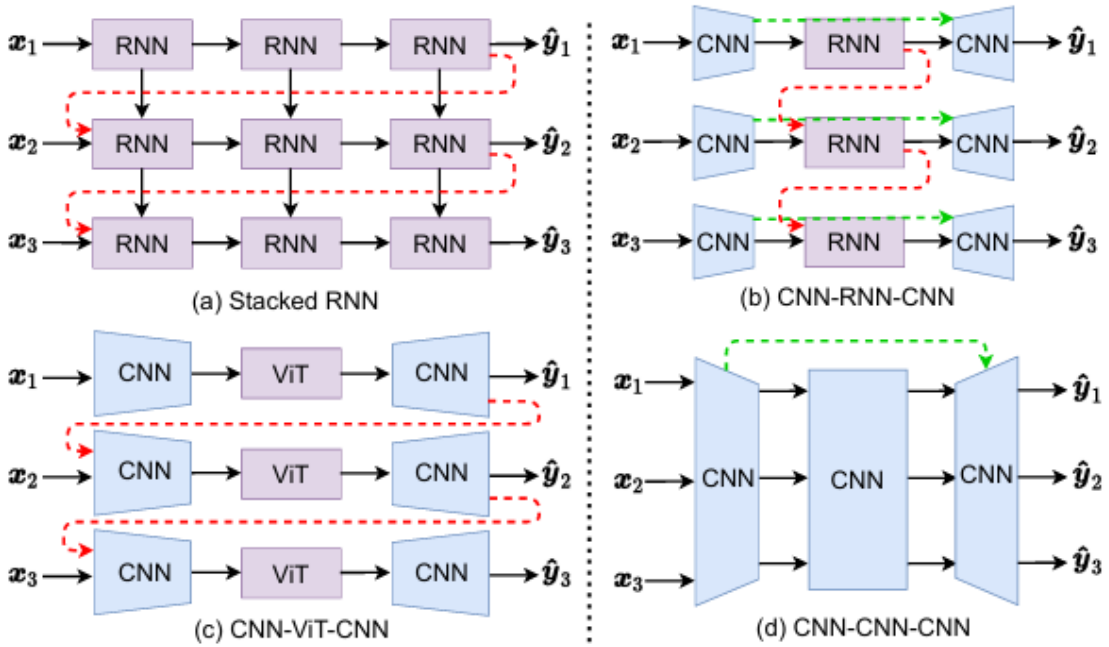
- 自回归模型（**Non-Autoregressive Models, NAR**）
  - **特点：**自回归模型在每个时间步的预测过程中都需要依赖之前的输出
  - **优点：**能够捕捉序列中元素之间的依赖关系，尤其是在长期依赖建模上表现优秀。
  - **缺点：**推理速度较慢，由于每个时间步都需要依赖前一步的输出，因此生成过程是递归的。
- 非自回归模型（**Autoregressive Models, AR**）
  - **特点：**不依赖于前一步的生成内容，因此可以并行生成整个序列或序列的部分内容
  - **优点：**生成速度更快，由于生成过程是并行的，非自回归模型能够在很短的时间内生成整个输出序列。
  - **缺点：**生成质量可能较差，难以捕捉长距离依赖

背景

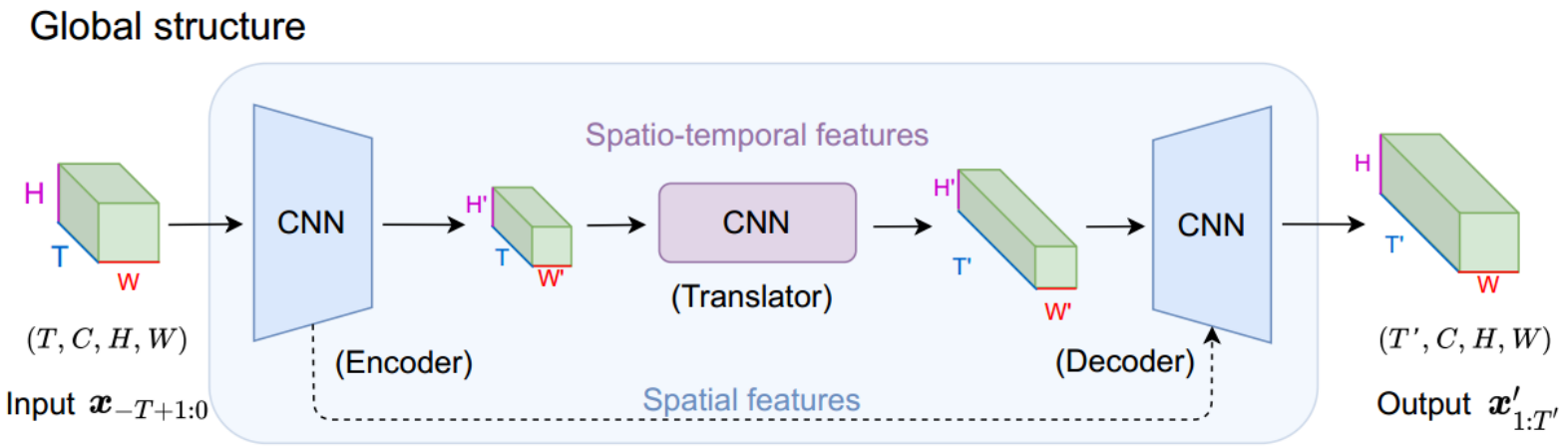
在这之前，时空序列预测的研究大多是用CNN+RNN的自回归模型，比如经典的ConvLSTM，PredRNN+。

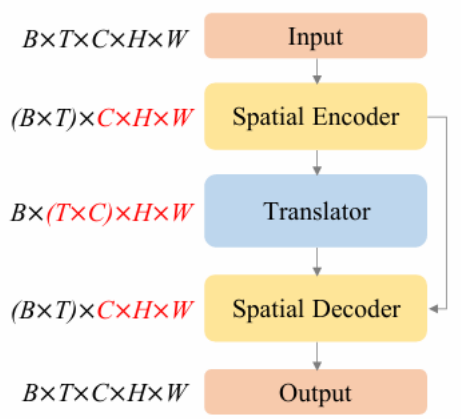
但是这些CNN+RNN类模型，在性能上取得了显著的提升，但是模型的复杂性不断增加，同时对于它们在良好表现中的必要性仍然缺乏深入的理解。

因此，文章提出了一个问题：是否可以开发出一个更简单的模型，以提供更好的理解和性能？



模型架构



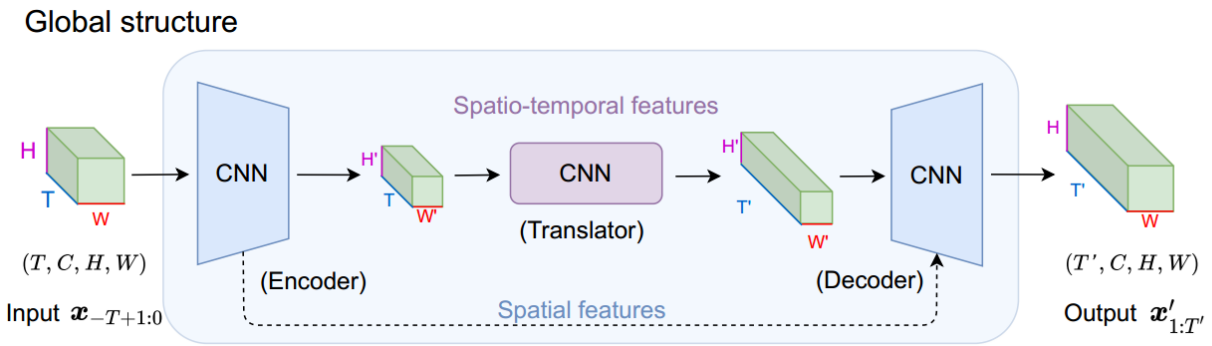


(a) The detailed flowchart of SimVP.

整体架构，由Encoder，Translator和Decoder组成，其中

- Encoder用于提取空间特征
- Translator学习时间演化
- Decoder集成时空信息来预测未来帧

模型架构



Encoder部分堆叠Ns个$ConvNormReLU$块（$Conv2d + LayerNorm + LeakyReLU$）来提取空间特征。

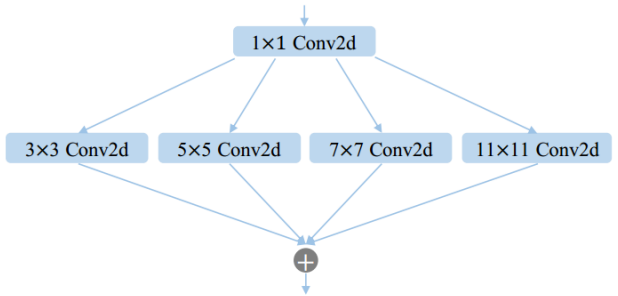
$$z_i = \sigma(\mathrm{LayerNorm}(\mathrm{Conv2d}(z_{i-1}))), 1 \leq i \leq N_s$$

Decoder部分堆叠Ns个$unConvNormReLU$块（$ConvTranspose2d + GroupNorm + LeakyReLU$）重建真实帧。

$$z_k = \sigma(\mathrm{GroupNorm}(\mathrm{unConv2d}(z_{k-1})))$$

Translator部分堆叠Nt个$Inception$模块来学习时间的演化

$$z_j = \mathrm{Inception}(z_{j-1}), N_s < j \leq N_s + N_t$$



(a) Inception temporal module

表现/消耗

Performance comparision of various methods on Moving MNIST

Method	Conference	MSE	SSIM	Github
ConvLSTM [71]	(NIPS 2015)	103.3	0.707	PyTorch
PredRNN [67]	(NIPS 2017)	56.8	0.867	PyTorch
PredRNN-V2 [68]	(Arxiv 2021)	48.4	0.891	PyTorch
CausalLSTM [65]	(ICML 2018)	46.5	0.898	Tensorflow
MIM [69]	(CVPR 2019)	44.2	0.910	Tensorflow
E3D-LSTM [66]	(ICLR 2018)	41.3	0.920	Tensorflow
PhyDNet [18]	(CVPR 2020)	24.4	0.947	PyTorch
CrevNet [73]	(ICLR 2020)	22.3	0.949	PyTorch
SimVP	–	23.8	0.948	PyTorch

Computation comparison on Moving MNIST, including memory overhead, per-frame FLOPs, and total training time.

Method	Memory	FLOPs	Training time
ConvLSTM	1043MB	107.4G	–
PredRNN	1666 MB	192.9 G	–
CausalLSTM	2017 MB	106.8 G	–
E3D-LSTM	2695 MB	381.3 G	–
CrevNet *	224 MB	1.652 G	$\approx 10d$ (300k iters)
PhyDNet *	200 MB	1.633 G	$\approx 10d$ (2k epochs)
SimVP *	412 MB	1.676 G	$\approx 2d$ (2k epochs)

消融实验

		model 1	model 2	model 3	model 4	model 5	model 6	model 7	model 8	model 9	SimVP
Archi	S-UNet	–	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T-UNet	✓	–	✓	✓	✓	✓	✓	✓	✓	✓
	#Groups	4	4	1	4	4	4	4	4	4	4
	Norm	G	G	G	B	G	G	G	G	G	G
Kernel	(3) $+c_t$	–	–	–	–	✓	–	–	–	–	–
	(5)$+c_t$	–	–	–	–	–	✓	–	–	–	–
	(7)$+c_t$	–	–	–	–	–	–	✓	–	–	–
	(11)$+c_t$	–	–	–	–	–	–	–	✓	–	–
	(11)$+2c_t$	–	–	–	–	–	–	–	–	✓	–
	(3,5,7,11)$+c_t$	✓	✓	✓	✓	–	–	–	–	–	✓
MSE	Moving MNIST	41.7	41.5	44.8	41.0	58.9	51.1	49.1	46.3	44.8	41.7
	TrafficBJ (×100)	43.5	41.5	44.5	44.3	43.5	44.0	43.3	42.3	42.3	42.0
	Human3.6 (/10)	32.4	33.4	33.0	34.8	37.3	34.0	32.9	33.4	32.1	32.0
	Summary	↓	↓	↓↓↓	↓	↓↓↓	↓↓↓	↓↓↓	↓↓	↓	–

Q1：哪种架构设计在提高性能方面起着关键作用?

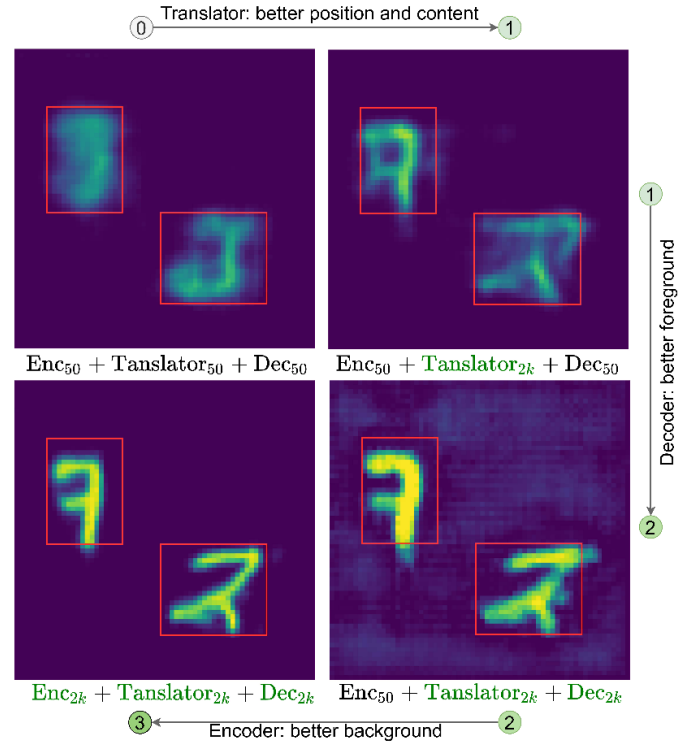
Q2：Conv kernel如何影响性能?

Q3: Enc、Translator和Dec分别扮演什么角色?

A1:group convolution >group normalization ≈ S-UNet ≈ T-UNet.

A2：随着kernel大小的增加，可以看到显著的性能增益。SimVP选择多尺度核[3,5,7,11]。

A3:Translator重点放在预测对象的位置和内容上，Decoder负责优化前景对象的形状，Encoder通过空间 U-Net 连接的方式来消除背景误差。



实验和消融实验

- 实验：在四个数据集上与 HeteroFL，FjORD 进行 TOP-1 ACC 对比
- 消融实验：验证了 r_1 （基向量的维度）和 r_2 （基向量的个数）的影响