It’s Raw! Audio Generation with State-Space Models

Докладчик: Спирин Иван
Рецензент-исследователь: Ткаченко Егор
Хакер: Виноградова Ульяна

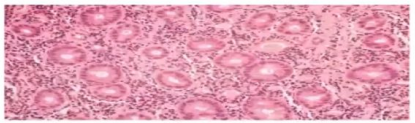
НИС МОП, 09.11.2022

Задача: моделирование длинных последовательностей

Music
Speech

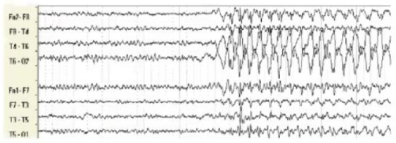
Audio

Images
Videos



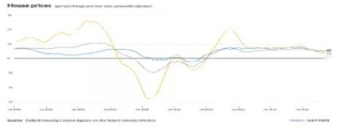
Spatio-temporal

EEG
EKG



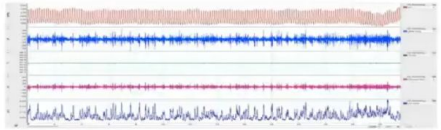
Biosignals

Stocks
Housing



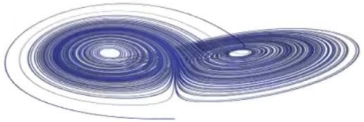
Market Data

Meteorology
Energy
Oil/gas
Spacecraft



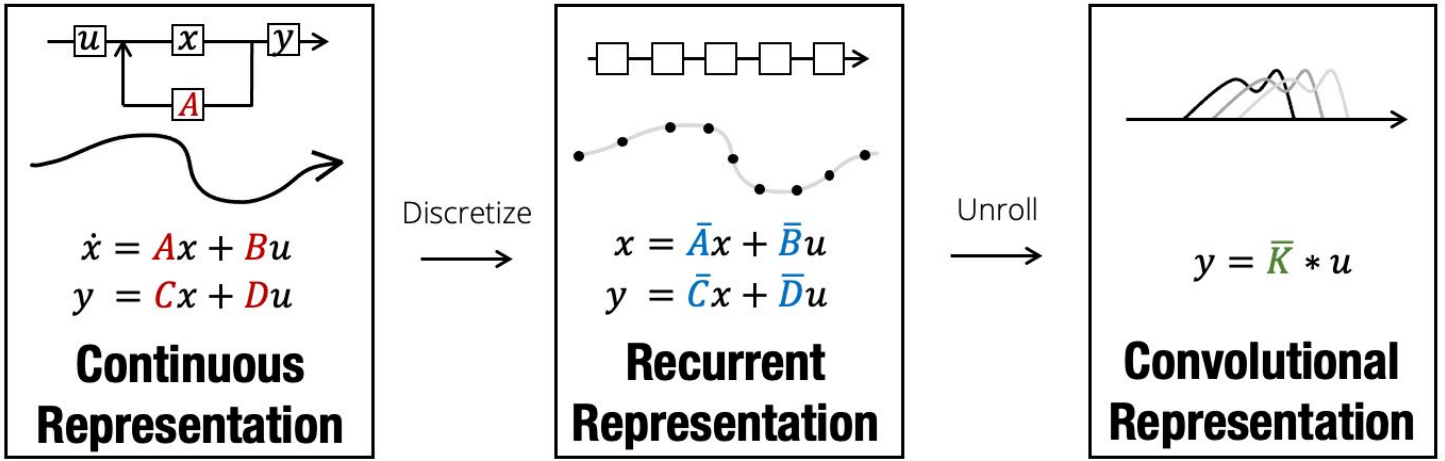
Telemetry Logs

CFD
PDEs

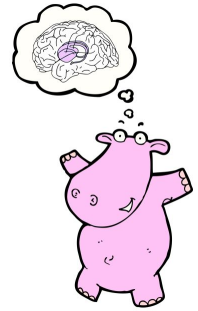


Dynamical Systems

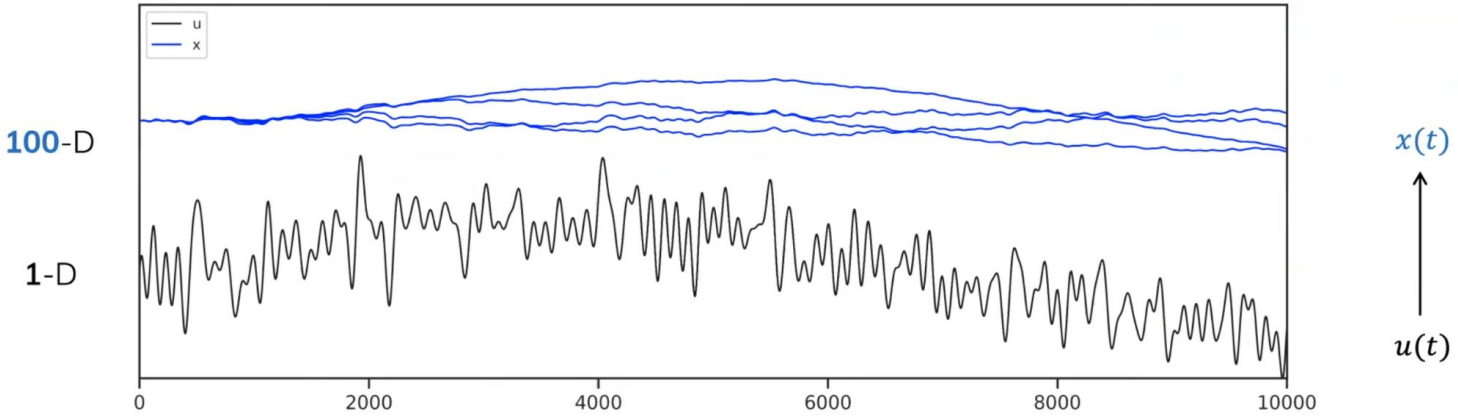
State Space Models



State Space Models



$$x' = \boldsymbol{A}x + \boldsymbol{B}u$$

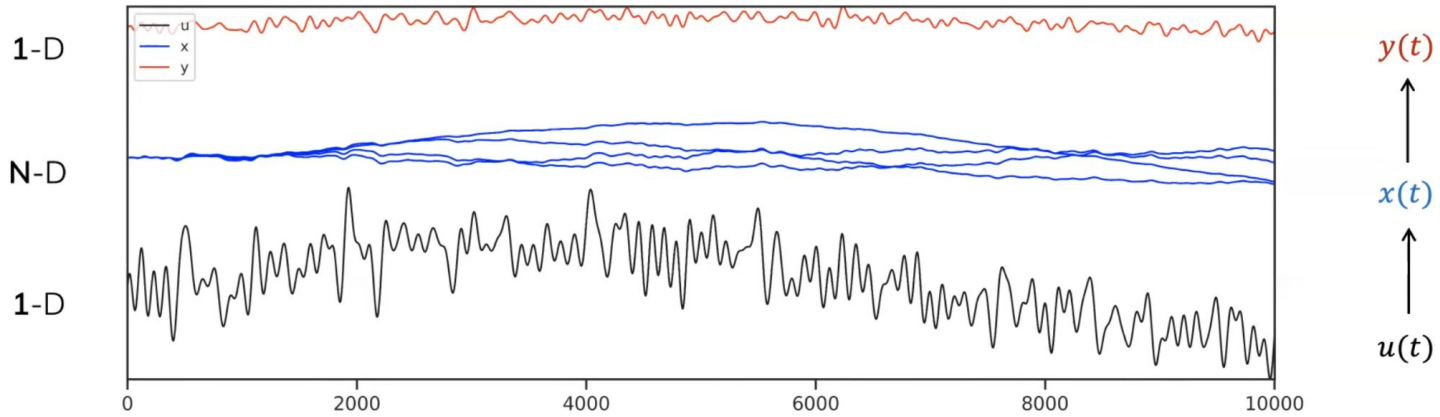


Оператор HiPPO отображает входной сигнал ***u*** в сигнал более высокой размерности ***x***

State Space Models

$$x' = \boldsymbol{A}x + \boldsymbol{B}u$$

$$y = \boldsymbol{C}x + \boldsymbol{D}u$$



SSM отображает x в одномерный выход y

State Space Models

Дискретизация:

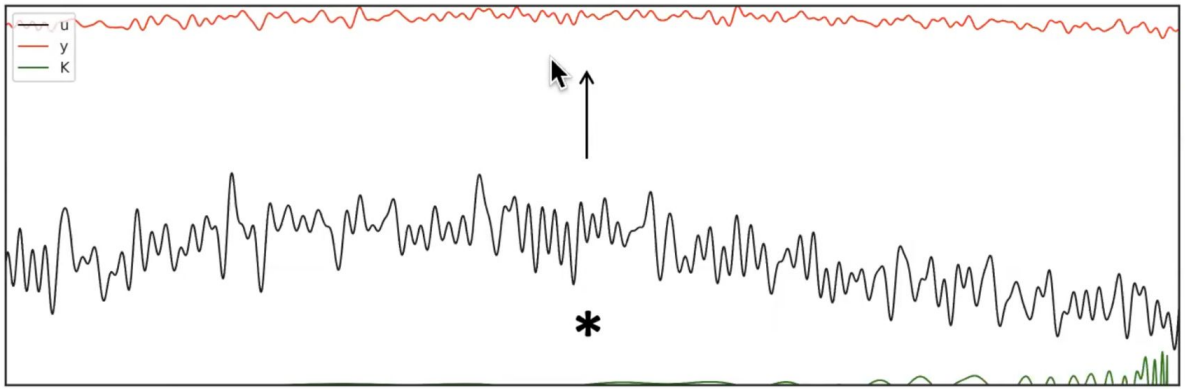
$$\overline{\boldsymbol{A}} = (\boldsymbol{I} - \Delta/2 \cdot \boldsymbol{A})^{-1}(\boldsymbol{I} + \Delta/2 \cdot \boldsymbol{A})$$
$$\overline{\boldsymbol{B}} = (\boldsymbol{I} - \Delta/2 \cdot \boldsymbol{A})^{-1}\Delta \boldsymbol{B}$$
$$\overline{\boldsymbol{C}} = \boldsymbol{C}$$

Итоговые формулы:

$$x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$
$$y_k = \overline{\boldsymbol{C}} x_k$$

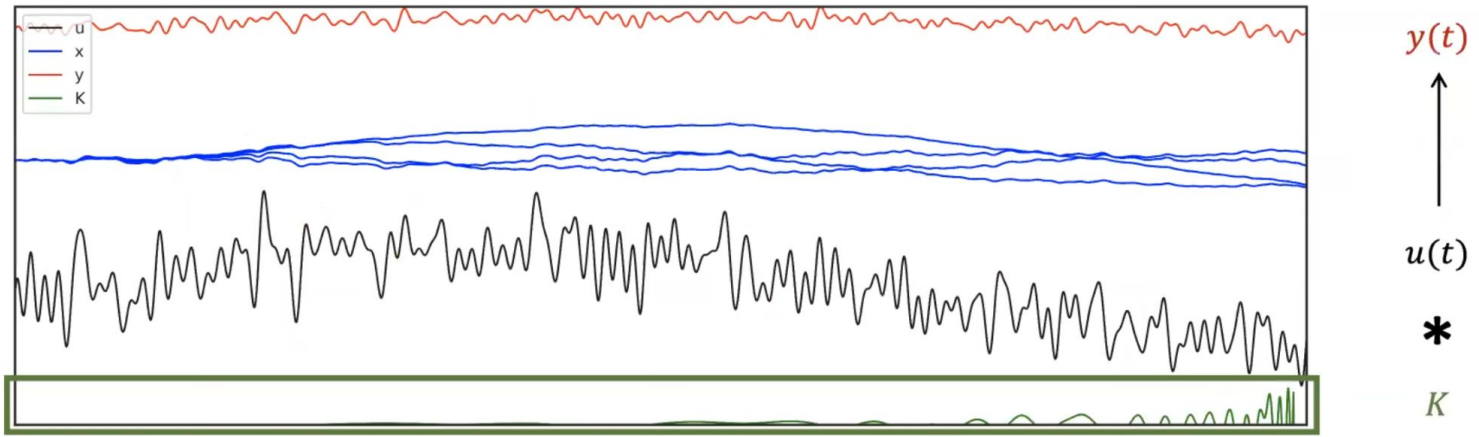
State Space Models

$$y = u * K$$



Можем смотреть на модель как на CNN с ядром K

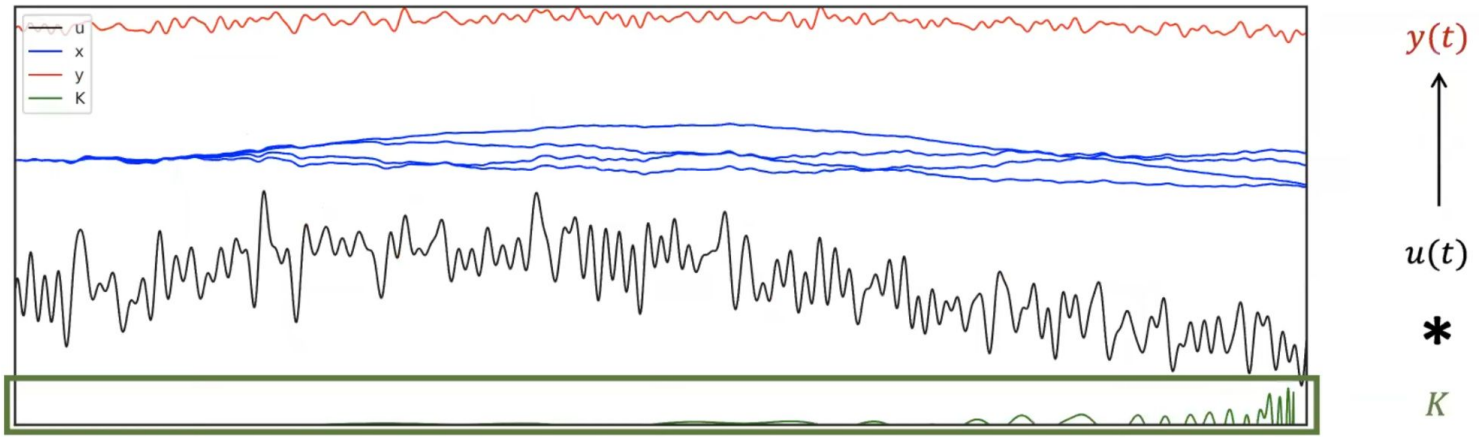
State Space Models



Сложность: $O(N^2 L)$

Память: $O(NL)$

Structured State Space (S4)



Сложность: $O(N^2 L) \rightarrow O(N + L)$

Память: $O(NL) \rightarrow O(N + L)$

Structured State Space (S4)

Параметризация: $A = \Lambda - PQ^*$

Algorithm 1 S4 Convolution Kernel (Sketch)

Input: S4 parameters $\boldsymbol{\Lambda}, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{B}, \boldsymbol{C} \in \mathbb{C}^N$ and step size Δ
Output: SSM convolution kernel $\overline{\boldsymbol{K}} = \mathcal{K}_L(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \overline{\boldsymbol{C}})$ for $\boldsymbol{A} = \boldsymbol{\Lambda} - \boldsymbol{P}\boldsymbol{Q}^*$ (equation (5))

1: $\widetilde{\boldsymbol{C}} \leftarrow \left(\boldsymbol{I} - \overline{\boldsymbol{A}}^L\right)^* \overline{\boldsymbol{C}}$ ▷ Truncate SSM generating function (SSMGF) to length L

2: $\begin{bmatrix} k_{00}(\omega) & k_{01}(\omega) \\ k_{10}(\omega) & k_{11}(\omega) \end{bmatrix} \leftarrow \left[\widetilde{\boldsymbol{C}}\ \boldsymbol{Q}\right]^* \left(\frac{2}{\Delta}\frac{1-\omega}{1+\omega} - \boldsymbol{\Lambda}\right)^{-1} [\boldsymbol{B}\ \boldsymbol{P}]$ ▷ Black-box Cauchy kernel

3: $\hat{\boldsymbol{K}}(\omega) \leftarrow \frac{2}{1+\omega}\left[k_{00}(\omega) - k_{01}(\omega)(1 + k_{11}(\omega))^{-1}k_{10}(\omega)\right]$ ▷ Woodbury Identity

4: $\hat{\boldsymbol{K}} = \{\hat{\boldsymbol{K}}(\omega) : \omega = \exp(2\pi i \frac{k}{L})\}$ ▷ Evaluate SSMGF at all roots of unity $\omega \in \Omega_L$

5: $\overline{\boldsymbol{K}} \leftarrow \mathsf{iFFT}(\hat{\boldsymbol{K}})$ ▷ Inverse Fourier Transform

Structured State Space (S4)

Результаты:

	Training Step (ms)			Memory Alloc. (MB)		
Dim.	128	256	512	128	256	512
LSSL	9.32	20.6	140.7	222.1	1685	13140
S4	4.77	3.07	4.75	5.3	12.6	33.5
Ratio	1.9×	6.7×	**29.6**×	42.0×	133×	**392**×

Параметризация улучшает эффективность: до 30 раз по времени, до 400 раз по памяти.

Structured State Space (S4)

Результаты:

	Convolution[3]	Recurrence	Attention	S4
Parameters	LH	$\boldsymbol{H^2}$	$\boldsymbol{H^2}$	$\boldsymbol{H^2}$
Training	$\boldsymbol{\tilde{L}H(B+H)}$	BLH^2	$B(L^2H+LH^2)$	$\boldsymbol{BH(\tilde{H}+\tilde{L})+B\tilde{L}H}$
Space	$\boldsymbol{BLH}$	$\boldsymbol{BLH}$	$B(L^2+HL)$	$\boldsymbol{BLH}$
Parallel	**Yes**	No	**Yes**	**Yes**
Inference	LH^2	$\boldsymbol{H^2}$	L^2H+H^2L	$\boldsymbol{H^2}$

Вычислительная сложность различных моделей.

L - длина последовательности, B - размер батча, H - размер скрытого пространства.

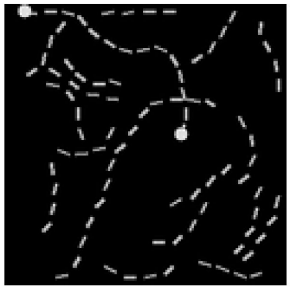
Structured State Space (S4)

Результаты:

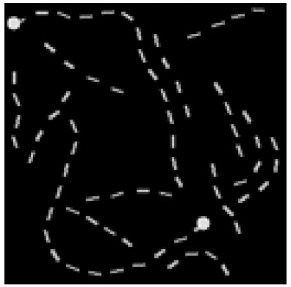
Model	ListOps	Text	Retrieval	Image	Pathfinder	Path-X	Avg
Transformer	36.37	64.27	57.46	42.44	71.40	✗	53.66
Reformer	37.27	56.10	53.40	38.07	68.50	✗	50.56
BigBird	36.05	64.02	59.29	40.83	74.87	✗	54.17
Linear Trans.	16.13	65.90	53.09	42.34	75.30	✗	50.46
Performer	18.01	65.40	53.82	42.77	77.05	✗	51.18
FNet	35.33	65.11	59.61	38.67	77.80	✗	54.42
Nyströmformer	37.15	65.52	79.56	41.58	70.94	✗	57.46
Luna-256	37.25	64.57	79.29	47.38	77.72	✗	59.37
S4	**59.60**	**86.82**	**90.90**	**88.65**	**94.20**	**96.35**	**86.09**

Long Range Arena: A Benchmark For Efficient Transformers

Примеры из Path-X

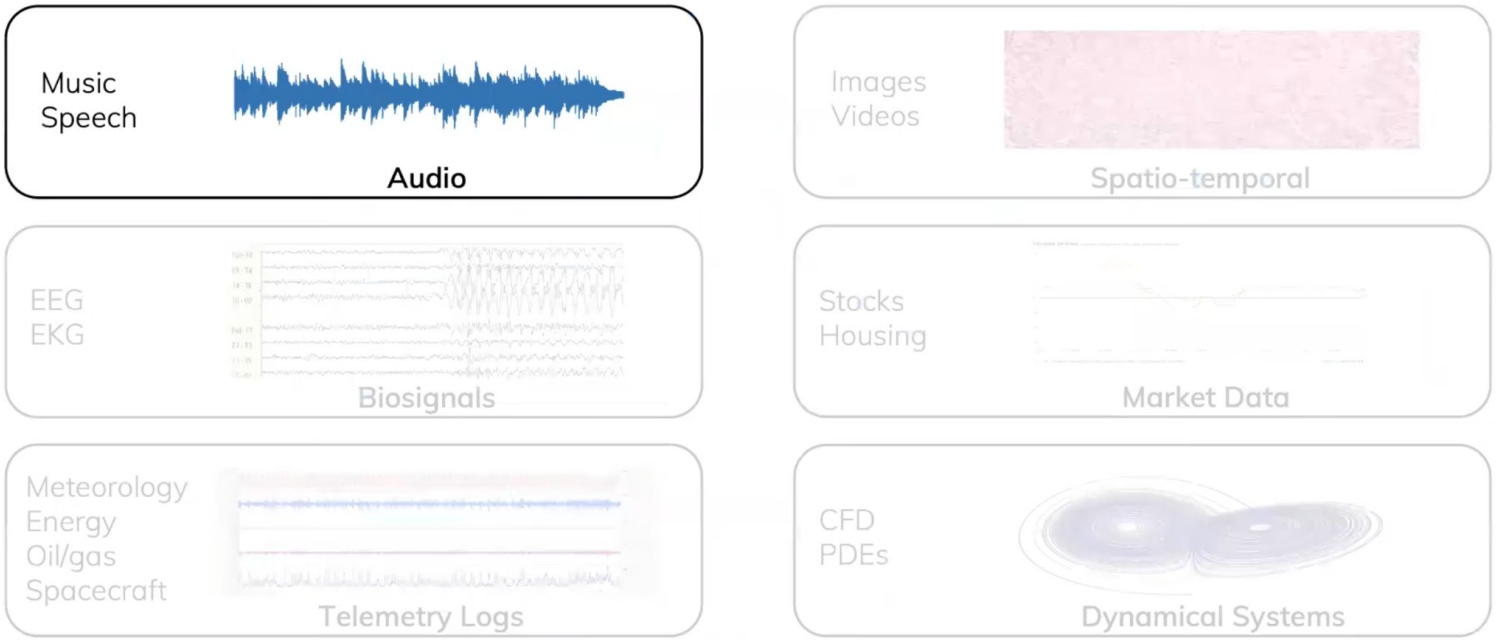


(a) A positive example.

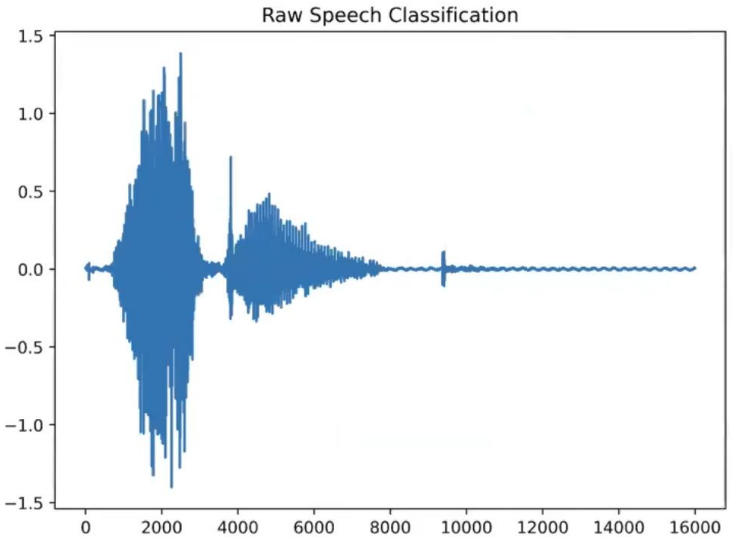


(b) A negative example.

Задачи аудио-моделирования



Задачи аудио-моделирования



1 секунда = 16000 значений

Длина:	161	16000	8000(test)
	MFCC	Raw	0.5×
Transformer	90.75	✗	✗
Performer	80.85	30.77	30.68
ODE-RNN	65.9	✗	✗
NRDE	89.8	16.49	15.12
ExpRNN	82.13	11.6	10.8
LipschitzRNN	88.38	✗	✗
CKConv	**95.3**	71.66	65.96
WaveGAN-D	✗	96.25	✗
LSSL	93.58	✗	✗
S4	93.96	**98.32**	**96.30**

Классификация SC10 с препроцессингом и без него

Задача аудио-генерации

Два вида моделей: *авторегрессионные* (AR) - генерируют выходные значения последовательно - и *неавторегрессионные* (non-AR), которые выдают весь выход за раз.

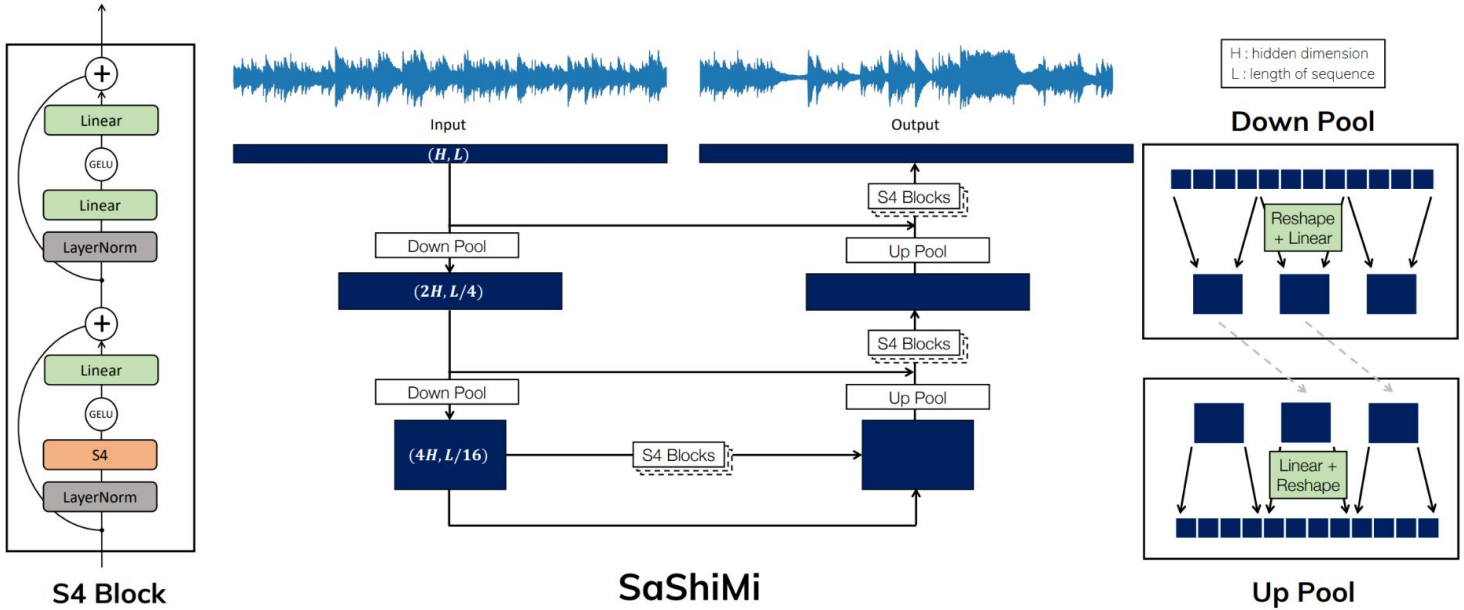
non-AR:

- Эффективнее на инференсе, лучше качество.
- DiffWave (Kong et al. 2021), WaveGAN (Donehue et al. 2019).

AR:

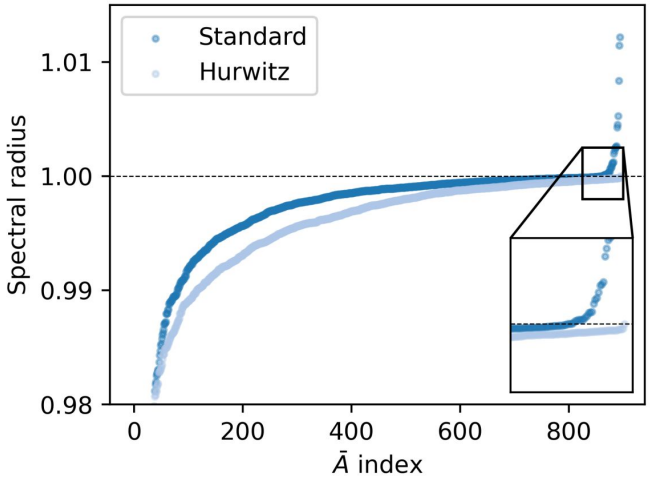
- Генерируют последовательность неограниченной длины.
- WaveNet (van den Oord et al. 2016), SampleRNN (Mehri et al. 2017).

SaShiMi model



SaShiMi model

- Эксперименты показали, что параметризация $A = \Lambda - PQ^*$ может вести к численной неустойчивости.
- Новая параметризация .
- Теперь собственные значения строго ограничены сверху единицей по модулю.



Learned	Frozen	NLL	Stable generation
–	$\Lambda + pq^*$	1.445	✓
$\Lambda + pq^*$	–	1.420	✗
$\Lambda - pp^*$	–	1.419	✓

SaShiMi model

- non-AR модификация - двунаправленный слой S4:

$$y = \mathsf{Linear}(\mathsf{Concat}(\mathsf{S4}(x), \mathsf{rev}(\mathsf{S4}(\mathsf{rev}(x)))))$$

- Эксперименты проводим в двух задачах:
    a. Генерация фортепианной музыки (Beethoven, YouTubeMix).
    b. Генерация 1-секундных звуковых дорожек с цифрами 0-9 (SC09).

Category	Dataset	Total Duration	Chunk Length	Sampling Rate	Quantization	Splits (train-val-test)
Music	Beethoven	10 hours	8s	16kHz	8-bit linear	Mehri et al. [28]
Music	YouTubeMix	4 hours	8s	16kHz	8-bit mu-law	$88\% - 6\% - 6\%$
Speech	SC09	5.3 hours	1s	16kHz	8-bit mu-law	Warden [42]

SaShiMi model

Результаты:

Beethoven

Model	Context	NLL	@200K steps	@10 hours
SampleRNN*	1024	1.076	–	–
WaveNet*	4092	1.464	–	–
SampleRNN†	1024	1.125	1.125	1.125
WaveNet†	4092	1.032	1.088	1.352
SaShiMi	**128000**	**0.946**	**1.007**	**1.095**

*Reported in Mehri et al. [28] †Our replication

YouTubeMix

Model	Test NLL	MOS (fidelity)	MOS (musicality)
SampleRNN	1.723	$\mathbf{2.98 \pm 0.08}$	1.82 ± 0.08
WaveNet	1.449	2.91 ± 0.08	2.71 ± 0.08
SaShiMi	**1.294**	2.84 ± 0.09	$\mathbf{3.11 \pm 0.09}$
Dataset	-	3.76 ± 0.08	4.59 ± 0.07

SaShiMi model

Результаты:

Двухслойная версия SaShiMi побеждает ведущие модели на датасете YouTubeMix, имея меньше параметров.

Model	NLL	Time/epoch	Throughput	Params
SampleRNN–2 tier	1.762	800s	112K samples/s	51.85M
SampleRNN–3 tier	1.723	850s	116K samples/s	35.03M
WaveNet–512	1.467	1000s	185K samples/s	2.67M
WaveNet–1024	1.449	1435s	182K samples/s	4.24M
SaShiMi–2 layers	1.446	205s	596K samples/s	1.29M
SaShiMi–4 layers	1.341	340s	316K samples/s	2.21M
SaShiMi–6 layers	1.315	675s	218K samples/s	3.13M
SaShiMi–8 layers	1.294	875s	129K samples/s	4.05M
Isotropic S4–4 layers	1.429	1900s	144K samples/s	2.83M
Isotropic S4–8 layers	1.524	3700s	72K samples/s	5.53M

SaShiMi model

Результаты:

SaShiMi - первая AR модель, которая может генерировать сэмплы высокого качества на датасете SC09. В сочетании с DiffWave получаем SOTA архитектуру.

Model	Params	NLL	FID ↓	IS ↑	mIS ↑	AM ↓	Human (κ) Agreement	MOS		
								Quality	Intelligibility	Diversity
SampleRNN	35.0M	2.042	8.96	1.71	3.02	1.76	0.321	1.18 ± 0.04	1.37 ± 0.02	2.26 ± 0.10
WaveNet	4.2M	1.925	5.08	2.27	5.80	1.47	0.408	1.59 ± 0.06	1.72 ± 0.03	2.70 ± 0.11
SaShiMi	4.1M	**1.891**	**1.99**	**4.12**	**24.57**	**0.90**	**0.832**	$\mathbf{3.29 \pm 0.07}$	$\mathbf{3.53 \pm 0.04}$	$\mathbf{3.26 \pm 0.09}$
WaveGAN	19.1M	-	2.03	4.90	36.10	0.80	0.840	2.98 ± 0.07	3.27 ± 0.04	3.25 ± 0.09
DiffWave	24.1M	-	1.92	5.26	51.21	0.68	0.917	4.03 ± 0.06	4.15 ± 0.03	$\mathbf{3.45 \pm 0.09}$
w/ SaShiMi	23.0M	-	**1.42**	**5.94**	**69.17**	**0.59**	**0.953**	$\mathbf{4.20 \pm 0.06}$	$\mathbf{4.33 \pm 0.03}$	3.28 ± 0.11
Train	-	-	0.00	8.56	292.5	0.16	0.921	4.04 ± 0.06	4.27 ± 0.03	3.59 ± 0.09
Test	-	-	0.02	8.33	257.6	0.19				

SaShiMi model

Результаты:

SaShiMi по всем метрикам лучше WaveNet на разных этапах обучения (в качестве слоя DiffWave). Потенциально можно улучшить все модели, использующие в себе WaveNet.

Architecture	Params	Training Steps	FID ↓	IS ↑	mIS ↑	AM ↓	NDB ↓	Human (κ) Agreement	MOS		
									Quality	Intelligibility	Diversity
SaShiMi	23.0M	800k	1.42	5.94	69.17	0.59	0.88	0.953	4.20 ± 0.06	4.33 ± 0.03	3.28 ± 0.11
WaveNet	24.1M	1000k	1.92	5.26	51.21	0.68	0.88	0.917	4.03 ± 0.06	4.15 ± 0.03	3.45 ± 0.09
SaShiMi	23.0M	500k	2.08	5.68	51.10	0.66	0.76	0.923	3.99 ± 0.06	4.13 ± 0.03	3.38 ± 0.10
WaveNet	24.1M	500k	2.25	4.68	34.55	0.80	0.90	0.848	3.53 ± 0.07	3.69 ± 0.03	3.30 ± 0.08
SaShiMi (uni.)	7.1M	500k	2.70	3.62	17.96	1.03	0.90	0.829	3.08 ± 0.07	3.29 ± 0.04	3.26 ± 0.08
SaShiMi	7.5M	500k	1.70	5.00	40.27	0.72	0.90	0.934	3.83 ± 0.07	4.00 ± 0.03	3.34 ± 0.09
WaveNet	6.8M	500k	4.53	2.80	9.02	1.30	0.94	0.446	1.85 ± 0.08	1.90 ± 0.03	3.03 ± 0.10

Источники

- https://arxiv.org/pdf/2110.13985.pdf
- https://arxiv.org/pdf/2111.00396.pdf
- https://arxiv.org/pdf/2011.04006.pdf
- https://arxiv.org/pdf/2202.09729v1.pdf
- https://srush.github.io/annotated-s4/
- https://www.youtube.com/watch?v=Za84WQ7uM68
- https://www.youtube.com/watch?v=luCBXCErkCs