Deep Learning Workshop - Music Generation via S4

Gal Bezalel

Tel-Aviv University

10 November 2024

Agenda

Agenda

Motivation

- Can we use SSM to generate high-fidelity music?
- Can we do it with a small-scale model ($\ll 10^9$ params)?
- Can it be productized?

Agenda

Preliminaries - S4[1]

- A state-space model:

$$\frac{dx}{dt} = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

- Discretizing dt allows us to apply the model on sequential data $\rightarrow$ dt is a learnable parameter!
- Moreover, we can unroll the (discrete) model and create a convolutional kernel (a huge filter).
- A should be HiPPO matrix, modeling Legendre polynomial coefficients $\rightarrow$ good for long sequences, but compute intensive.
  - Instead, using Diagonal + Low Rank (DPLR: $A = \lambda + pq^*$) factorization speeds up computation.
- More neat math in the Annotated S4.

[1] Albert Gu, Karan Goel, and Christopher Ré. *Efficiently Modeling Long Sequences with Structured State Spaces*. 2022. arXiv: 2111.00396 [cs.LG]. URL: https://arxiv.org/abs/2111.00396.

Perliminaries - SaShiMi[2]

- S4 block - a S4 unit with 2-layer FF with GELU activation, layer norm and skip connection.
- Multiscale architecture:
  - Repeated S4 blocks with varying hidden dimension ($H = 2^k$).
  - Input is passed to the S4 block + down-sampled (pooling) for encoding, up-sampled for decoding.

[2] Karan Goel et al. *It's Raw! Audio Generation with State-Space Models*. 2022. arXiv: 2202.09729 [cs.SD]. URL: https://arxiv.org/abs/2202.09729.

Agenda

Dataset

- Original dataset used in the article[3]: YouTubeMix
  - 4 hours of classical piano music.
  - A previous attempt to replicate and improve NLL
- **Our dataset: YouTubeBigBand**[4]
  - 2 hours of jazz trio.
  - A more complex waveform (multiple instruments, percussion).
  - Improvisation is inherent.
- Preprocessing (both):
  - Resampled at 16khz
  - 1min chunks

[3] DeepSound. *SampleRNN*. https://github.com/deepsound-project/samplernn-pytorch. 2017. URL: https://huggingface.co/datasets/krandiash/youtubemix.

[4] Gal Bezalel. *YouTubeBigBand*. https://huggingface.co/datasets/galbezalel/youtube_bigband. 2024.

Experiment 1 - Complete Training of 8 Layers SaShiMi Model

- Basically, replicate the original experiment, but with our Big Band dataset
- 19M params
- 1000 epochs, no regularization
- Time: 4 days on a single A100 (in practice, over a week using spot GCP instance, Colab)

Experiment 2 - Complete Training of *Ablated* 2 Layers SaShiMi Model

- As in the original article, validate the assumption that a smaller model can achieve similar results to larger model (and reduce costs).
- 1.5M params
- 1000 epochs (in the original article: 500 epochs), no regularization
- Time: 50 hours on a single A100 (in practice, 2 days using spot GCP instance, Colab)

Agenda

Motivation

Preliminaries

Experiments

Results & Demo

Conclusions

Metrics

Test metric	YouTubeMix - 8 layers	YouTubeBigBand - 8 layers	YouTubeBigBand - 2 layers
final/test/accuracy	0.4203284681	0.2766689062	0.274307102
final/test/accuracy@10	0.9719890952	0.8476241231	0.8452669382
final/test/accuracy@3	0.8351296782	0.5846688747	0.5805157423
final/test/accuracy@5	0.9241486192	0.7164889574	0.7128702998
final/test/bpb	2.063964605	3.149125099	3.16355896
final/test/loss	1.430631161	2.182806969	2.192811012
final/val/accuracy	0.4274106026	0.200661391	0.1991965473
final/val/accuracy@3	0.8423588276	0.4834408164	0.4809091985
final/val/accuracy@5	0.9283464551	0.6362654567	0.6337321401
final/val/bpb	2.029698849	3.614607096	3.624292135
final/val/loss	1.40688026	2.505454779	2.512167692

Generation examples - Demo

- ★ We will listen to a few (cherry-picked...) generated examples
- Generation is unbounded - can be conditioned (on a prefix of the dataset, up to $\sim$8s in our experiment) or not.
- Default generation hyperparams are: Temperature $= 1$, Top-P: 1
  - Traditional Temp. values (0.2-0.5) yielded samples with long, silent / noisy parts.
  - To preserve some consistency, we set Temp. $= 0.8$.

Agenda

Have we met our expectations?

- Can we use SSM to generate high-fidelity music? Potentially, yes.
- Can we do it with a small-scale model ($\ll 10^9$ params)? Potentially, yes.
- Can it be productized? No.
  - Currently, prompts are only taken from validation split
  - Time: Takes 10 minutes to generate 10s samples using ablated model, 30 minutes using full models (still follows logarithmic scale though!)

Lessons learned

- Audio generation is costly.
- *Good* audio generation is difficult (noted also by the original authors).
- In practice:
  - The ablated version is *indeed* on-par with the deeper model.
  - High temperature works great in musical domains (creativity?)
- The good stuff:
  - Relatively cheap model with a promise
  - Experience with preprocessing audio
  - Experience with tools: Hydra, GCP

Action Items

- Continue training, improve metrics - WIP
- Experiment with regularization - it's possible to add dropout and weight decay, will be tested on ablated version
- "Productize" - find a way to use prompts from completely new data
  - Condition on a short (few seconds), single prompt + concat the output
  - Could probably be engineered (create config, etc.)
- Many more things that might not be covered...
  - Audio signal: different sampling rate, quantization, chunk length.
  - Training: different LRs, fine-tuning, transfer learning.
  - Inference: grid search for Temp., Top-P.