Comparison of S4, LRU, and benchmark models on Keyword spotting problem

Stanislav Efimov
Ramazan Fazylov
Varvara Furik
Boris Mikheev
Alexander Sharshavin

Skoltech

21.12.2023

The problem under consideration and its relevance

The keyword spotting problem is highly relevant in various domains and industries due to the increasing reliance on voice-controlled and natural language processing (NLP) technologies. It's important to be fast and high quality.

- One of the main issues is the processing speed of streaming audio.
- Another problem is dealing with audio data - thousands of samples in one second. Not all models can handle this type of data correctly.

The existing solutions and their disadvantages

To solve this task use different approaches

- RNN: does not remember long-range dependencies well, takes a long time to training, and has the problem of a vanishing or exploding gradient.
- CNN:
  - 1-D convlutions: kernel must be large, poor quality
  - 2-D convolutions: expect spectrogram - additional calculations
- Transformers: expect spectrogram - additional calculations, long time processing, large memory usage (computational and memory costs for attention layers scale quadratically $O(L^2)$ with the sequence length L).

Data

- We use SpeechCommands Dataset.
- This is a set of 105,829 one-second audio files, each containing a single spoken English word or background noise. The task is to detect preregistered keywords by classifying utterances into a predefined set of words.

Structured State Space for Sequence Modeling - idea

- Motivation:
  Transformers struggle to scale for the very long data sequences, but there exists a classical control theory approach: SSM.
- Challenges:
  Construct a stable disretased SSM algorithm that will scale linearly in memory and computational cost w.r.t. L
- Solution:
  HiPPO discretization with smart diagonalization, inversion and convolution filter.

Structured State Space for Sequence Modeling - method

- Parametrize data in the following way

$$\begin{cases} x'(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t), \end{cases}$$

with u(t) - input, x(t) - hidden state, y(t) - output

- Discretize data with step Δ and rewrite system

$\bar{A} = (I - \Delta/2 \cdot A)^{-1}(I + \Delta/2 \cdot A)$
$\bar{B} = (I - \Delta/2 \cdot A)^{-1}\Delta B$
$\bar{C} = C$
$x_k = \bar{A}x_{k-1} + \bar{B}u_k$
$y_k = \bar{C}x_k$

- Initialize A with Hippo matrix and represent as a sum of normal and low-rank, than diagonal and low-rank, where

$$A_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2}, if\ n > k \\ n+1, if\ n = k \\ 0, if\ n < k, \end{cases}$$

Linear Recurrent Units - idea

- Motivation:
  RNNs scale linarly with the sequence length, therefore they are potentially faster then transformers in inference
- Challenges:
  Reproduce performance of SSM models like S4 with deep RNN on the dataset with long-squence data.
- Solution:
  Use RNNs with with linear recurrences, add smart initialization and diagonalization of recurrence matrix.

Linear Recurrent Units - method

- Linear recurrences:

$$\begin{cases} x_k = Ax_{k-1} + Bu_k \\ y_k = Cx_k + Du_k \end{cases}$$

- Complex diagonal recurrent matrices

$$A = P\Lambda P^{-1},\ P \in \mathbb{C}^{N\times N}, \Lambda = (\lambda_1, .., \lambda_n) \in \mathbb{C}^{N\times N}$$

- Stable exponential parametrization

$$\Lambda = (\exp(-\nu + i\theta)),\ \nu, \theta \in \mathbb{R}^N$$

$$\lambda_j := \exp(-\exp(\nu_j^{\log}) + i\theta)$$

- Normalization

$$x_k = \Lambda x_{k-1} + \exp(\gamma^{\log}) \odot (Bu_k)$$

Results comparison

Model	S4	Transformer	LRU	CNN	LSTM	DNN
Accuracy	96.92%	96.85%	**97.23%**	95.17%	78.75%	62.59%

- S4 outperformes CNN, LSTM, DNN as baseline models, since this models are infamous for thier modest performance on long sequnce data.
- LRU performes the best out of all models. Since LRU inherites some SSM principles from S4 and combines them with solid numerical stabilization, it was reasonable to expect that for the very long sequnces (like in our dataset) it will outperform S4, and this exactly the result we obtained.

Contribution

- S4 model: Ramazan Fazylov, Alexander Sharshavin
- LRU model: Varvara Furik, Boris Mikheev
- Presentation: Stanislav Efimov, Varvara Furik, Alexander Sharshavin

Sources

1. Our beautiful github https://github.com/shallex/NLA_23_project
2. LRU https://arxiv.org/pdf/2303.06349.pdf
3. S4 https://arxiv.org/pdf/2111.00396.pdf
4. S4 https://srush.github.io/annotated-s4
5. LRU https://github.com/NicolasZucchet/minimal-LRU
6. Transformers https://github.com/wdjose/keyword-transformer
7. Dataset https://huggingface.co/datasets/speech_commands