NEURAL INFORMATION PROCESSING SYSTEMS

HIERARCHICALLY GATED RECURRENT NEURAL NETWORK FOR SEQUENCE MODELING

*Zhen Qi[1], *Songlin Yang[2], ✆Yiran Zhong[1]

[1]OpenNLPLab, Shanghai AI Lab, [2]MIT CSAIL

RNN Striking Back

NEURAL INFORMATION PROCESSING SYSTEMS

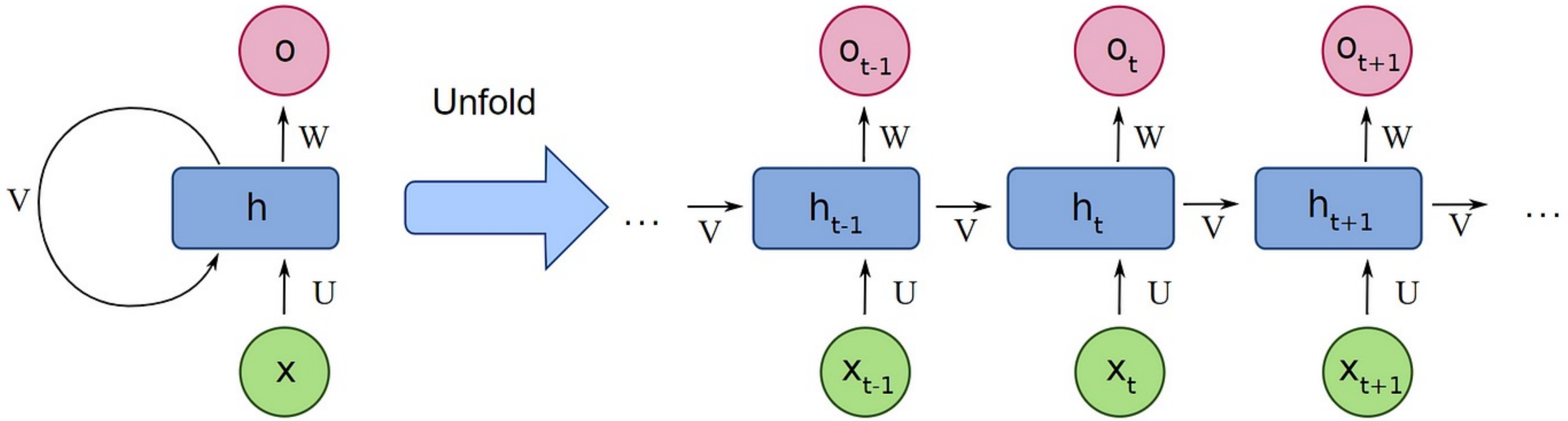
Advantages of RNNs

- Constant inference speed with computational complexity of O(dh)*.
- Linear training computational complexity of O(ndh), with respect to sequence length.
- Handle Variable-Length Sequences

Disadvantages of RNNs

- Lack of parallelism.
- Limited capability in modeling long-term dependencies.

Gradually replaced by Transformers in the deep learning era



*where d is the feature dimension, h is the hidden dimension, n is the sequence length.

RNN Striking Back

Our solution:

- Using element-wise linear recurrence (ELR) to facilitate parallelized training.
- Using Hierarchically Gated Recurrent Units (HGRU) to capture long-term dependencies

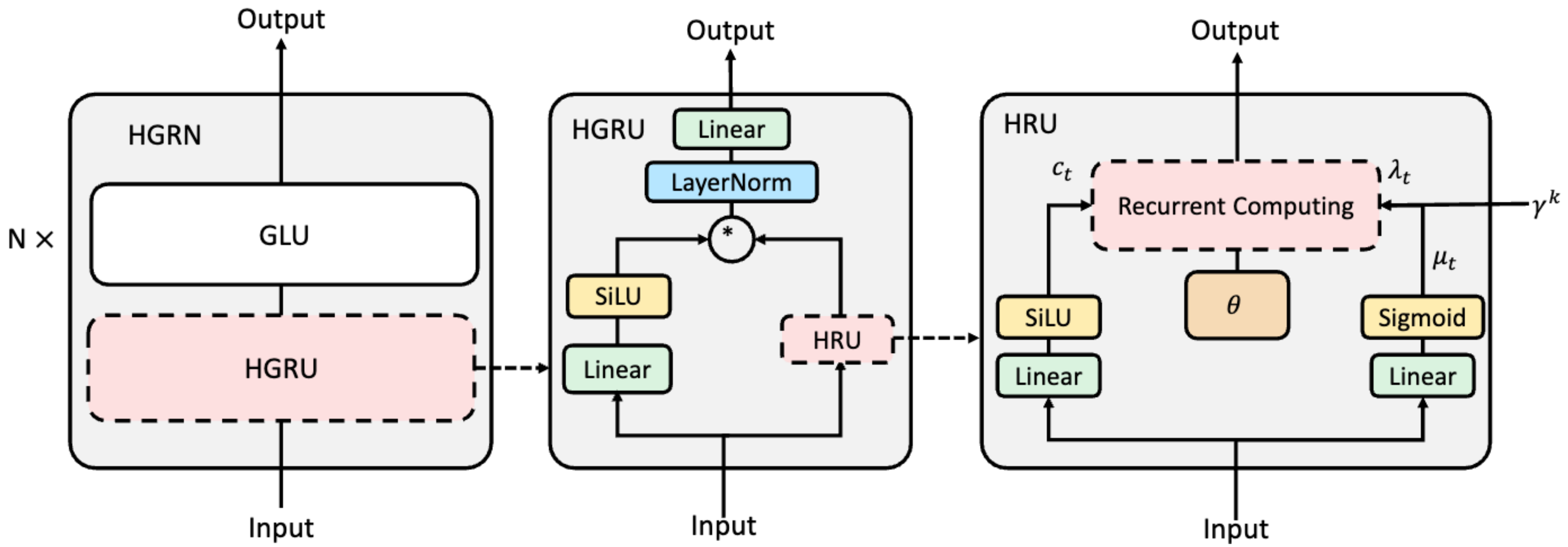


Figure 1: Illustration of the neural architecture. Each **HGRN** layer consists of a token mixer **HGRU** and a channel mixer GLU. **HGRU** employs linear recurrence in the complex domain: $\mathbf{h}_t = \lambda_t \odot \exp(i\theta) \odot \mathbf{h}_{t-1} + (1 - \lambda_t) \odot \mathbf{c}_t$. Here c_t is the input vector, θ is the rotation angle, μ_t is the output of the original forget gate, γ^k is the lower bound of the k-th layer, λ is the resulting data dependent decay rate: $\lambda_t = \gamma^k + (1 - \gamma^k) \odot \mu_t$.

HGRU Exploration

Let us start with a simple gated linear recurrent layer:

$$\mathbf{f}_t = \mathrm{Sigmoid}\left(\mathbf{x}_t \mathbf{W}_f + \mathbf{b}_f\right) \in \mathbb{R}^{1\times d},$$
$$\mathbf{i}_t = \mathrm{Sigmoid}\left(\mathbf{x}_t \mathbf{W}_i + \mathbf{b}_i\right) \in \mathbb{R}^{1\times d},$$
$$\mathbf{c}_t = \mathrm{SiLU}\left(\mathbf{x}_t \mathbf{W}_t + \mathbf{b}_z\right) \in \mathbb{R}^{1\times d},$$
$$\mathbf{h}_t = \mathbf{f}_t \odot \mathbf{h}_{t-1} + \mathbf{i}_t \odot \mathbf{c}_t \in \mathbb{R}^{1\times d},$$
$$\mathbf{h}_0 = \mathbf{0} \in \mathbb{R}^{1\times d},$$

where $\odot$ denotes the element-wise product. Following the terminology used in the RNN literature, we refer to $\mathbf{f}_t$ and $\mathbf{i}_t$ as the forget and input gates, respectively. It is worth noting that $\mathbf{f}_t$ and $\mathbf{i}_t$ depend only on $\mathbf{x}_t$ and not on $\mathbf{h}_{t-1}$.

Algorithm 1 Recurrent Computing

```
Input: c_t ∈ ℂ^{1×d}, μ_t, θ, γ^k ∈ ℝ^{1×d}, t = 1, ..., n, k = 1, ..., H.
Init: h = 0 ∈ ℂ^{1×d}, H ∈ ℂ^{n×d}.
for t = 1 to n do
    begin
    λ_t = γ^k + (1 − γ^k) ⊙ μ_t.
    h = λ_t exp(iθ) h + (1 − λ_t) c_t.
    [H]_t = h.
    end
return H.
```

HGRU Exploration

- **Complex-valued recurrence.**

$$\mathrm{Re}(\mathbf{c}_t) = \mathrm{SiLU}\left(\mathbf{x}_t \mathbf{W}_{cr} + \mathbf{b}_{cr}\right) \in \mathbb{R}^{1\times d},$$
$$\mathrm{Im}(\mathbf{c}_t) = \mathrm{SiLU}\left(\mathbf{x}_t \mathbf{W}_{ci} + \mathbf{b}_{ci}\right) \in \mathbb{R}^{1\times d}.$$

Regarding the forget gate values, we find it convenient to use the exponential representation of complex numbers and parameterize

- **Lower bound on forget gate values.**

we set a monotonically increasing lower bound on the forget gate values. It ensures that the forget gate values in the lower layers remain relatively small, enabling the necessary forgetting of past information for modeling short-term dependencies. In the uppermost layer, the forget gate values approach one, facilitating the effective modeling of long-term dependencies.

- **Tying input and forget gates.**

$$\mathbf{h}_t = \lambda_t \odot \exp(i\theta) \odot \mathbf{h}_{t-1} + (1-\lambda_t) \odot \mathbf{c}_t \in \mathbb{C}^{1\times d}.$$

- **Output gates and projection.**

$$\mathbf{g}_t = \mathrm{Sigmoid}(W_g \mathbf{x}_t + b_g) \in \mathbb{R}^{1\times 2d},$$
$$\mathbf{o}'_t = \mathrm{LayerNorm}(\mathbf{g}_t \odot [\mathrm{Re}(\mathbf{h}_t), \mathrm{Im}(\mathbf{h}_t)]) \in \mathbb{R}^{1\times 2d},$$
$$\mathbf{o}_t = \mathbf{o}'_t \mathbf{W}_o + \mathbf{b}_o \in \mathbb{R}^{1\times d}.$$

Token mixing perspective of HGRU

Expanding $\mathbf{h}_t = \lambda_t \odot \exp(i\theta) \odot \mathbf{h}_{t-1} + (1-\lambda_t) \odot \mathbf{c}_t \in \mathbb{C}^{1\times d}$ we have:

$$\mathbf{h}_t = \sum_{s=1}^{t} (1-\lambda_s) \left[\prod_{k=s+1}^{t} \lambda_k \exp(i\theta) \right] \mathbf{c}_s = \sum_{s=1}^{t} (1-\lambda_s) \left[\prod_{k=s+1}^{t} \lambda_k \right] \exp(i(t-s)\theta) \mathbf{c}_s$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \vdots \\ \mathbf{h}_n \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 1-\lambda_1 & 0 & \cdots & 0 \\ (1-\lambda_1)\lambda_2 \exp(i\theta) & 1-\lambda_2 & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ (1-\lambda_1)\left[\prod_{k=2}^{n} \lambda_k\right] \exp(i(n-1)\theta) & \ldots & \ldots & 1-\lambda_n \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \vdots \\ \mathbf{c}_n \end{bmatrix}$$

So the token mixing module can be formed as follows:

$$\mathbf{H} = \mathbf{A}\mathbf{C}.$$

Note that the token mixing matrix $\mathbf{A}$ can be decomposed into two parts $\mathbf{A} = \mathbf{\Lambda} \odot \mathbf{\Theta}$:

$$\mathbf{\Lambda} = \begin{bmatrix} 1-\lambda_1 & 0 & \cdots & 0 \\ (1-\lambda_1)\lambda_2 & 1-\lambda_2 & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ (1-\lambda_1)\left[\prod_{k=2}^{n} \lambda_k\right] & \ldots & \ldots & 1-\lambda_n \end{bmatrix}, \mathbf{\Theta} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \exp(i\theta) & 1 & & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \exp(i(n-1)\theta) & \ldots & \ldots & 1 \end{bmatrix}$$

where $\mathbf{\Lambda}$ can be seen as an attention matrix and $\mathbf{\Theta}$ as a RPE.

Results

NEURAL INFORMATION PROCESSING SYSTEMS

Language modeling

Table 1: **Results on Wikitext-103** (TNN[59]'s setting). ↓ means *lower is better*.

Model	PPL (val)↓	PPL (test)↓	Params (M)
Attn-based			
Transformer [81]	24.40	24.78	44.65
FLASH [10]	25.92	26.70	42.17
1+elu [35]	27.44	28.05	44.65
Performer [7]	62.50	63.16	44.65
cosFormer [62]	26.53	27.06	44.65
MLP-based			
Syn(D) [76]	31.31	32.43	46.75
Syn(R) [76]	33.68	34.78	44.65
gMLP[42]	28.08	29.13	47.83
RNN-based			
S4 [22]	38.34	39.66	45.69
DSS [26]	39.39	41.07	45.73
GSS [49]	29.61	30.74	43.84
RWKV [55]	24.31	25.07	46.23
LRU [53]	29.86	31.12	46.24
FFT-based			
TNN [59]	23.98	24.67	48.68
Ours			
HGRN	24.14	24.82	46.25

Table 2: **Results on Wikitext-103** (Hyena[57]'s setting). All models are in GPT-2 small size (125M). ↓ means *lower is better*

Model	PPL↓
Transformer	18.6
Hybrid H3	18.5
Performer	26.8
Reformer	25.6
AFT-conv	28.2
Linear Attention	25.6
Hyena	18.6
Hyena-slim	18.5
HGRN	18.6

Table 3: **Results on the Pile.** All the model size is 1b. The lower the better.

Model	PPL↓
Transformer	4.56
LRU	5.07
HGRN	4.14

Results

NEURAL INFORMATION PROCESSING SYSTEMS

Table 4: **Performance Comparison on Commonsense Reasoning.**. PS: parameter size (billion). T: tokens (billion). HS: HellaSwag. WG: WinoGrande.

Model	PS	T	BOOLQ	PIQA	HS	WG	ARC-e	ARC-c	OBQA	AVG
GPT-Neo	0.13	300	61.71	63.06	30.40	50.43	43.73	23.12	26.20	42.66
OPT	0.16	300	55.47	62.95	31.35	50.43	43.52	22.70	28.00	42.06
Pythia	0.16	300	55.08	61.32	30.16	51.93	43.18	23.12	26.80	41.66
RWKV	0.17	-	-	65.07	32.26	50.83	47.47	24.15	29.60	41.56
HGRN	0.15	100	59.91	65.02	33.33	50.20	46.68	23.81	28.60	43.94
OPT	0.35	300	57.74	64.58	36.69	52.49	44.02	23.89	28.20	43.94
Pythia	0.4	300	60.40	67.08	40.52	53.59	51.81	24.15	29.40	46.71
BLOOM	0.56	350	55.14	64.09	36.97	52.80	47.35	23.98	28.20	44.08
RWKV	0.43	-	-	67.52	40.90	51.14	52.86	25.17	32.40	45.00
HGRN	0.35	100	59.05	66.70	38.12	51.70	49.20	25.26	30.60	45.80
GPT-Neo	1.3	300	61.99	71.11	48.93	54.93	56.19	25.85	33.60	50.37
OPT	1.3	300	57.77	71.71	53.70	59.35	57.24	29.69	33.20	51.81
Pythia	1.4	300	60.73	70.67	47.18	53.51	56.99	26.88	31.40	49.62
BLOOM	1.1	350	59.08	67.14	42.98	54.93	51.47	25.68	29.40	47.24
RWKV	1.5	-	-	72.36	52.48	54.62	60.48	29.44	34.00	50.56
HGRN	1	100	58.69	70.89	48.02	51.62	55.64	27.90	31.60	49.19

Results

Table 5: **Performance Comparison on SuperGLUE.** PS: parameter size (billion). T: tokens (billion).

Model	PS	T	WSC	WIC	RTE	CB	MULTIRC	BOOLQ	COPA	AVG
GPT-Neo	0.13	300	36.54	50.00	54.87	41.07	0.84	61.71	64.00	44.15
OPT	0.16	300	36.54	50.00	49.82	21.43	1.36	55.47	66.00	40.09
Pythia	0.16	300	36.54	50.16	52.71	41.07	2.52	55.08	65.00	43.30
HGRN	0.15	100	38.46	51.10	56.68	42.86	1.47	59.91	65.00	45.07
OPT	0.35	300	36.54	50.00	51.99	46.43	1.36	57.74	72.00	45.15
Pythia	0.4	300	57.69	50.31	52.71	35.71	1.68	60.40	70.00	46.93
BLOOM	0.56	350	40.38	50.00	52.71	41.07	1.05	55.14	61.00	43.05
HGRN	0.35	100	38.46	50.16	52.71	51.79	1.99	59.05	73.00	46.74
GPT-Neo	1.3	300	36.54	50.00	60.29	44.64	1.99	61.99	69.00	46.35
OPT	1.3	300	37.50	51.10	51.99	41.07	3.15	57.77	79.00	45.94
Pythia	1.4	300	36.54	50.00	53.07	35.71	0.94	60.73	72.00	44.14
BLOOM	1.1	350	36.54	50.00	52.71	41.07	0.73	59.08	68.00	44.02
HGRN	1	100	40.38	50.78	53.43	42.86	3.04	58.69	70.00	45.60

Image modeling

Table 7: **Performances comparison of image classification on ImageNet-1k. HGRN** performs favorably than competing methods with similar parameter sizes.

	DeiT-Tiny		DeiT-Small	
Model	Top1 Acc	Param (M)	Top1 Acc	Parma (M)
Deit	72.20	5.7	79.90	22.0
TNN	72.29	6.4	79.20	23.4
HGRN	74.40	6.1	80.09	23.7

Results

NEURAL INFORMATION PROCESSING SYSTEMS

Sequence length extrapolation

Table 14: The extrapolation performance of competing methods. The best result is highlighted in **bold** and the second in underline.↓ means *lower is better*.

Seqlen	Transformer PPL↓	LS PPL↓	FLASH PPL↓	1+elu PPL↓	Performer PPL↓	cosFormer PPL↓	gMLP PPL↓	S4 PPL↓	DSS PPL↓	GSS PPL↓	ALiBi PPL↓	TNN PPL↓	LRU PPL↓	**HGRU** PPL↓
512	24.78	24.05	24.69	28.05	63.16	27.06	29.13	30.74	41.07	39.66	24.15	24.67	31.12	24.85
768	41.36	23.49	16950.45	47.35	159.74	32.90	1.34E+9	30.41	40.50	39.76	23.38	24.25	30.72	24.4
1024	62.35	23.21	174165.47	70.47	504.30	55.28	8.93E+12	30.24	40.22	39.91	22.98	24.05	30.5	24.16
1280	82.52	23.07	346502.88	91.88	1020.28	102.88	1.58E+15	30.15	40.03	40.82	22.74	23.91	30.38	24.03
1536	100.17	22.97	647788.12	111.56	1568.83	175.26	4.96E+16	30.08	39.94	41.04	22.57	23.83	30.3	23.94
1792	118.42	22.97	1719873.5	129.92	2138.50	267.65	5.67E+17	30.04	39.85	41.08	22.52	23.79	30.24	23.88
2048	133.44	22.99	6.25E+6	147.09	2693.89	368.02	3.59E+18	30.00	39.79	41.53	22.43	23.73	30.19	23.82
3072	188.95	23.25	4.17E+10	206.88	4945.82	820.77	2.19E+20	29.91	39.64	44.08	22.24	23.63	30.09	23.71
4096	246.06	23.83	2.67E+13	267.87	7170.91	1335.51	1.61E+21	29.88	39.59	48.27	22.17	23.58	30.04	23.66
5120	270.93	24.56	1.26E+15	299.31	8443.15	1735.50	5.08E+21	29.85	39.54	53.32	22.11	23.54	30.01	23.62
6144	311.65	25.45	1.58E+16	352.62	10234.07	2146.19	1.16E+22	29.83	39.51	57.73	22.08	23.53	29.99	23.6
7168	346.58	26.42	8.11E+16	389.02	11420.56	2494.79	1.98E+22	29.82	39.49	60.25	22.07	23.51	29.97	23.58
8192	372.18	27.11	3.40E+17	411.50	12557.09	2902.24	2.78E+22	29.82	39.49	63.36	22.05	23.51	29.97	23.58
9216	387.29	28.78	1.22E+18	453.27	14847.66	3028.72	3.93E+22	29.80	39.46	74.92	22.03	23.49	29.96	23.56
10240	395.94	30.13	4.03E+18	457.06	13623.83	3247.83	4.93E+22	29.79	39.45	81.87	22.02	23.48	29.94	23.55
11264	426.54	31.14	1.07E+19	504.19	14661.77	3341.91	5.70E+22	29.79	39.46	87.67	22.00	23.48	29.94	23.55
12288	463.50	33.21	2.52E+19	555.38	17959.85	3644.81	7.18E+22	29.79	39.44	92.11	22.00	23.48	29.94	23.55
13312	506.35	34.72	4.96E+19	584.01	20026.35	3851.70	8.04E+22	29.78	39.43	96.00	22.00	23.47	29.93	23.54
14336	486.86	36.05	1.28E+20	589.83	20971.31	3951.26	9.41E+22	29.78	39.43	101.47	21.99	23.46	29.92	23.53
Avg	261.36	26.71	1.16E+19	299.86	8684.79	1764.75	2.41E+22	29.97	39.75	60.26	**22.40**	<u>23.70</u>	30.17	23.80

NEURAL INFORMATION PROCESSING SYSTEMS

THANKS

Code is released at