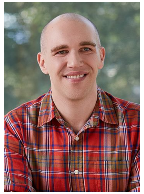
Reservoir Transformers

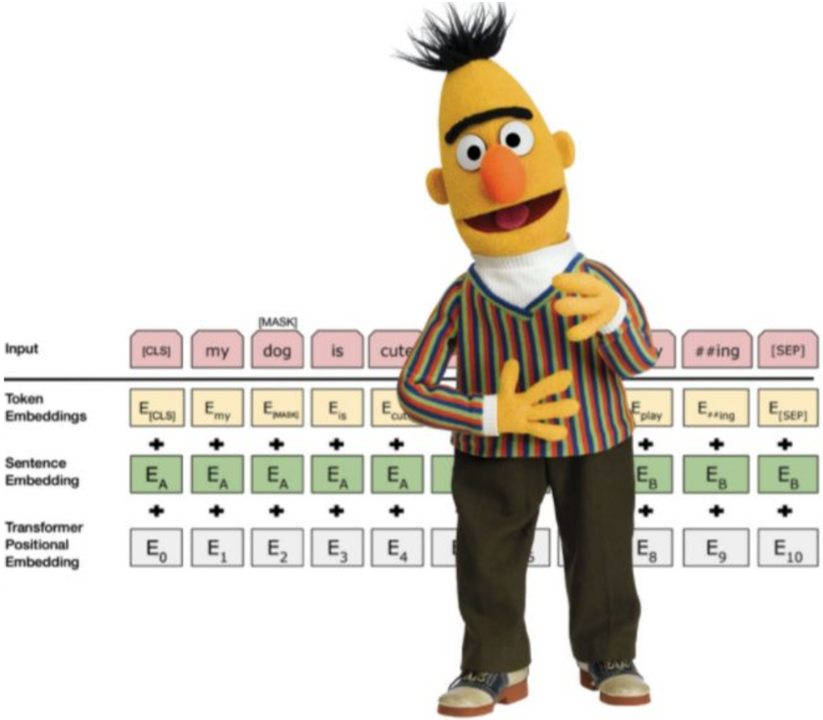
Sheng Shen[†], Alexei Baevski[‡], Ari S. Morcos[‡], Kurt Keutzer[†], Michael Auli[‡], Douwe Kiela[‡]

[†]UC Berkeley, [‡]Facebook AI

Transformer

- **NLP has been undergoing a revolution**



FACEBOOK AI

Transformer Limitations

- **How do we make Transformer more training efficient?**

Devlin et al. (2019). We pretrain our model using 1024 V100 GPUs for approximately one day.

Green AI

$4,600,000: The full cost of training GPT-3

Consumption	CO_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Artificial intelligence / Machine learning

Training a single AI model can emit as much carbon as five cars in their lifetimes

FACEBOOK AI

Transformer Limitations

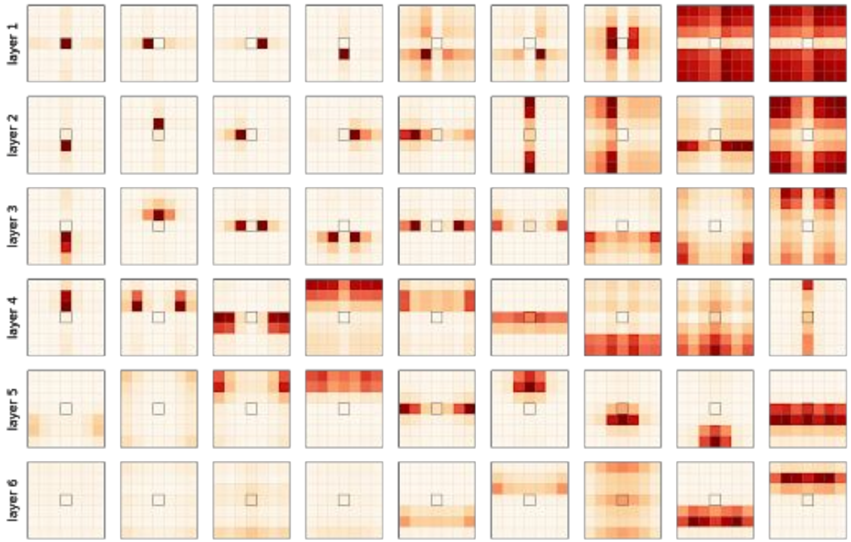
BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

- How do we make Transformer more training efficient?
- What do Transformer learn?

A Primer in BERTology: What We Know About How BERT Works

Anna Rogers, Olga Kovaleva and Anna Rumshisky

Posted Online 2020
https://doi.org/10.1162/tacl_a_00349



FACEBOOK AI

Reservoir Transformers

- In this work, we explore **a simple old idea** to make transformers **more training efficient**, and **understand what they learn**.

$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{\partial J}{\partial \theta_t}; \frac{\partial J}{\partial \theta_t} = \frac{\partial J}{\partial L_n} \frac{\partial L_n}{\partial L_{n-1}} \cdots \frac{\partial L_0}{\partial x}$$

Normal Transformer Layer

$$\theta^L_{t+1} \leftarrow \theta^L_t - \eta \frac{\partial J}{\partial \theta^L_t} \qquad \frac{\partial J}{\partial \theta_t^{L_3}}$$

Normal Transformer Layer

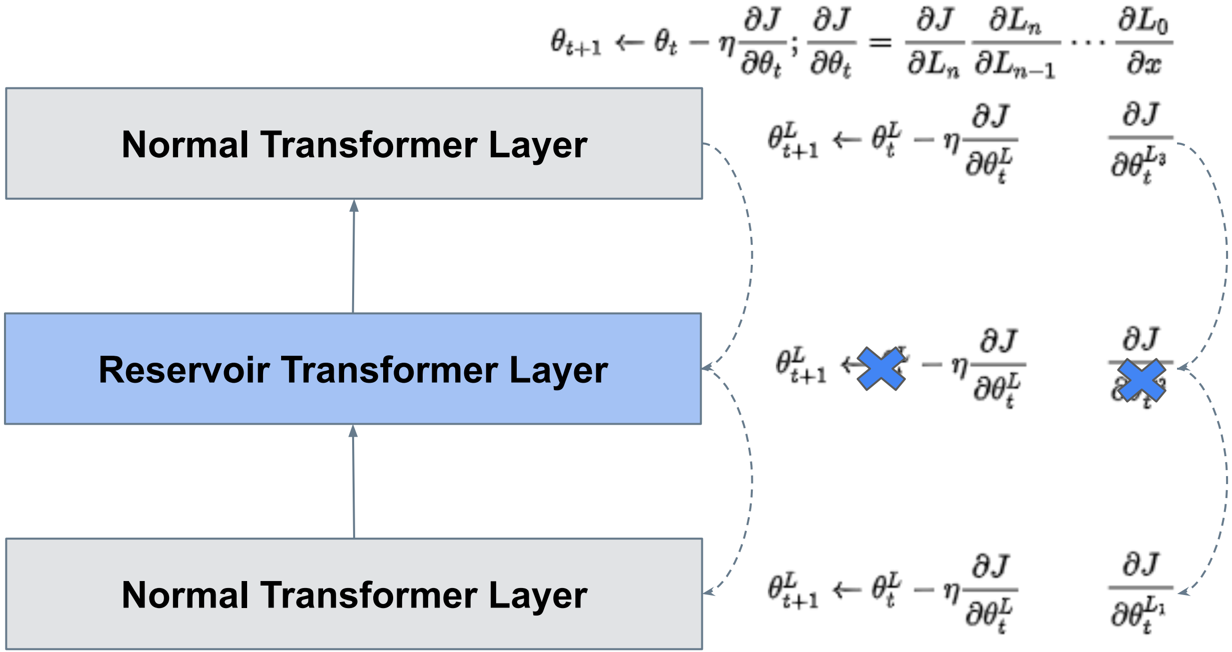
$$\theta^L_{t+1} \leftarrow \theta^L_t - \eta \frac{\partial J}{\partial \theta^L_t} \qquad \frac{\partial J}{\partial \theta_t^{L_2}}$$

Normal Transformer Layer

$$\theta^L_{t+1} \leftarrow \theta^L_t - \eta \frac{\partial J}{\partial \theta^L_t} \qquad \frac{\partial J}{\partial \theta_t^{L_1}}$$

Reservoir Transformers

- In this work, we explore **a simple old idea** (**random reservoir layer**).



FACEBOOK AI

Connect to Random LSTM

- A random LSTM on top of pretrained word embeddings is a surprisingly good sentence encoder.

Random

Model	Dim	MR	CR	MPQA	SUBJ	SST2	TREC	SICK-R	SICK-E	MRPC	STSB
BOE	300	77.3(.2)	78.6(.3)	87.6(.1)	91.3(.1)	80.0(.5)	81.5(.8)	80.2(.1)	78.7(.1)	72.9(.3)	70.5(.1)
BOREP	4096	77.4(.4)	79.5(.2)	88.3(.2)	91.9(.2)	81.8(.4)	**88.8(.3)**	85.5(.1)	82.7(.7)	73.9(.4)	68.5(.6)
RandLSTM	4096	77.2(.3)	78.7(.5)	87.9(.1)	91.9(.2)	81.5(.3)	86.5(1.1)	85.5(.1)	81.8(.5)	**74.1(.5)**	72.4(.5)
ESN	4096	**78.1(.3)**	**80.0(.6)**	**88.5(.2)**	**92.6(.1)**	**83.0(.5)**	87.9(1.0)	**86.1(.1)**	**83.1(.4)**	73.4(.4)	**74.4(.3)**

Random Wide sentence Encoder perform great on sentence classification task

FACEBOOK AI

Wieting, J. and Kiela, No Training Required: Exploring Random Encoders for Sentence Classification. ICLR 2019.

Connect to Reservoir Computing & Random Feature

- Fixed random (RNN) reservoir with "readout" function trained on top.

The "echo state" approach to analysing and training recurrent neural networks – with an Erratum note[1]

Herbert Jaeger
Fraunhofer Institute for Autonomous Intelligent Systems

January 26, 2010

Random Features for Large-Scale Kernel Machines

Ali Rahimi and Ben Recht

Reservoir computing approaches to recurrent neural network training

Mantas Lukoševičius, Herbert Jaeger

Evaluation

- Goal:
  - Can we speed up transformers by freezing-and-keeping-fixed “reservoir layers”?
  - Can we do so without sacrificing (too much) task performance?

- Metrics:
  - Area under the convergence curve (f is the network, g is the eval metric, t is wall-clock):

$$\int_{t=0}^{\hat{T}} \sum_{x,y \in \mathcal{D}} g_t(f(x), y)$$

  - Train time until max validation performance
  - Test set generalization
  - Number of trainable parameters
  - Probing Task Performance

Tasks

- Tasks (small, medium, big; from scratch and pretrained):
  - Machine Translation:
    - IWSLT (small)
    - WMT (medium)
  - Language Modelling:
    - Enwiki8 (big)
  - Masked Language Modelling:
    - RoBERTa pretraining (big)
    - Finetuning:
      - Sentiment: SST-2 (small)
      - Natural Language Inference: MultiNLI (medium)

Experiments

- Comparison of different reservoir layers:
  - Regular Transformer (FFN + MultiHeadSelfAttn)

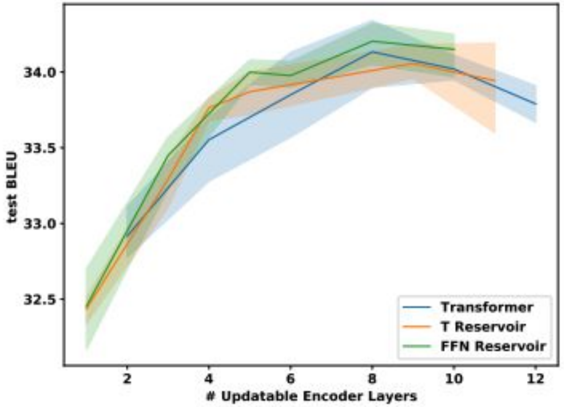
$$H = \mathrm{MultiHeadSelfAttn}(\mathrm{LayerNorm}(X)) + X$$
$$L = \mathrm{FFN}(\mathrm{LayerNorm}(H)) + H$$

  - Transformer reservoir
  - FFN reservoir, no attention
  - (CNN and BiGRU in the appendix)

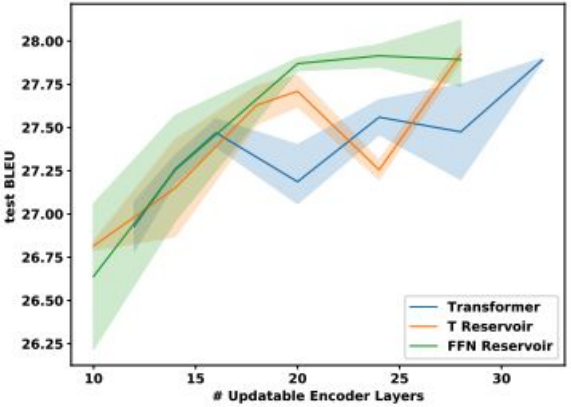
- Different freezing strategies (we found **interleaving trainable and reservoir layers** to work the best)

Results (Machine Translation & Language Modeling)

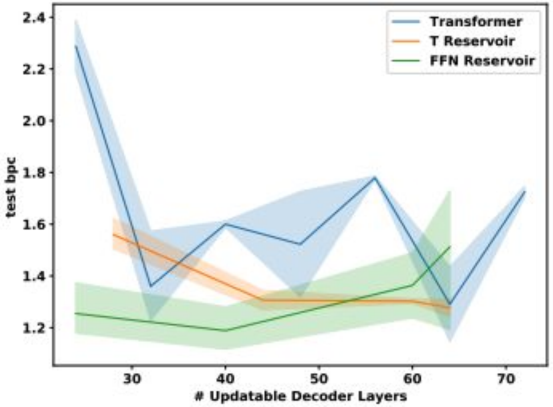
- **Comparable or better final performance** of T Reservoir & FFN Reservoir



IWSLT
(test BLEU)↑



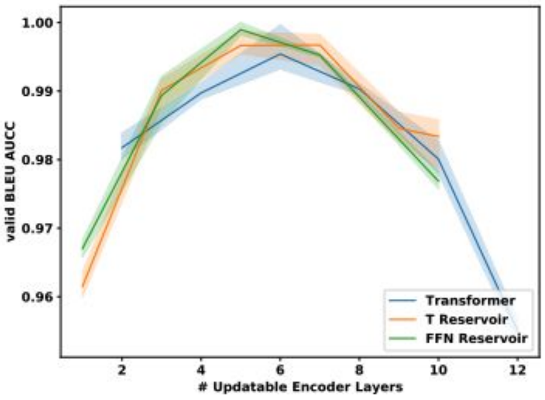
WMT
(test BLEU)↑



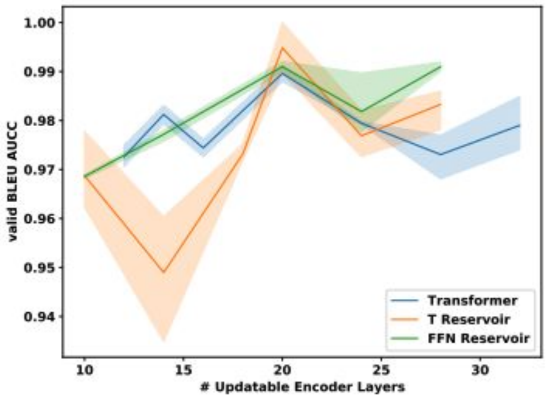
enwik8
(test bpc)↓

Results (Machine Translation & Language Modeling)

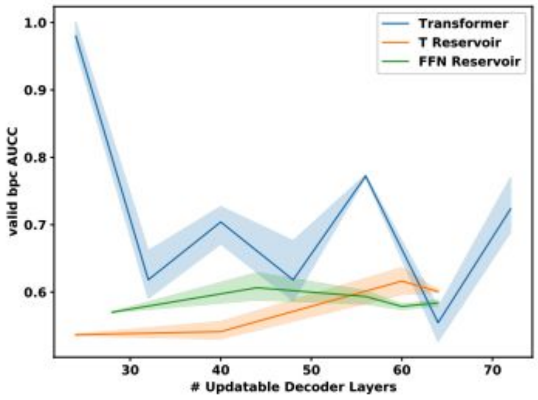
- T Reservoir & FFN Reservoir are **more training efficient** (better AUCC).



IWSLT
(valid BLEU AUCC)↑



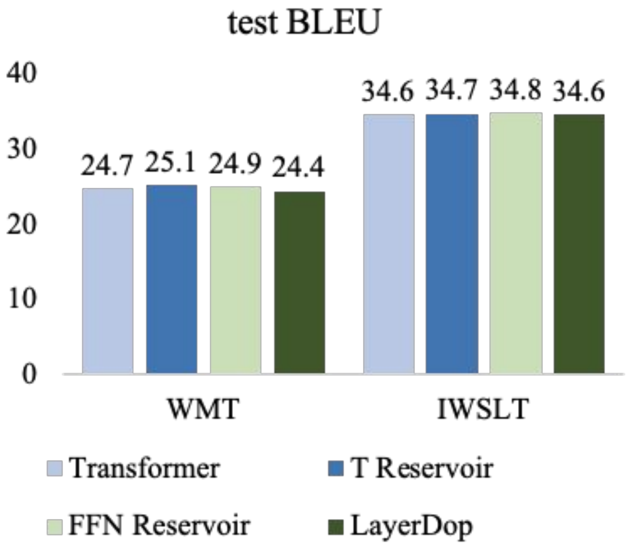
WMT
(valid BLEU AUCC)↑

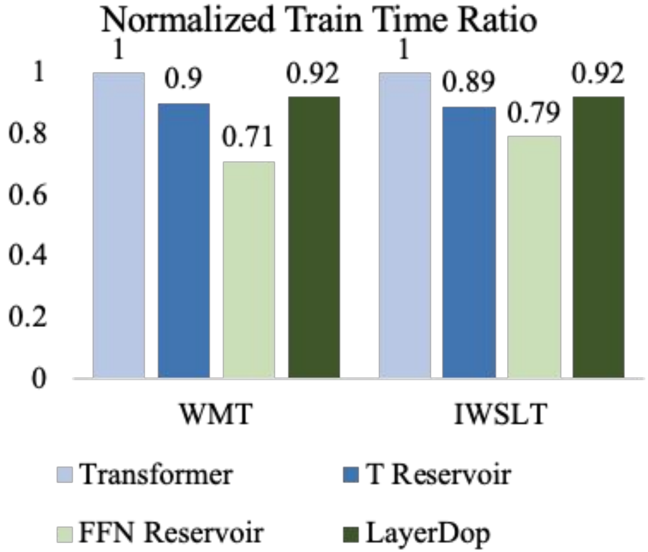


enwik8
(valid bpc AUCC)↓

Results (Machine Translation & Language Modeling)

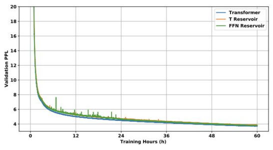
- **21%/30% training saving** with no performance drop with FFN Reservoir.

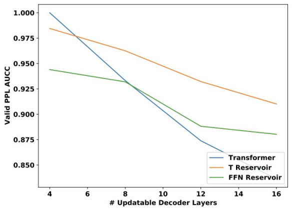




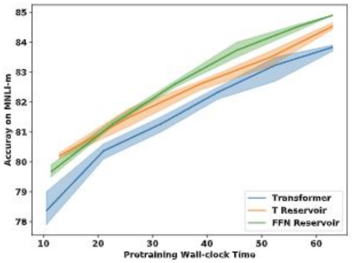
Results (Maksed Language Modeling)

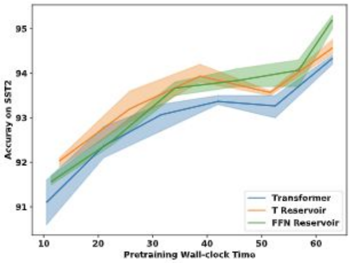
- Pretraining PPL shows no big difference:



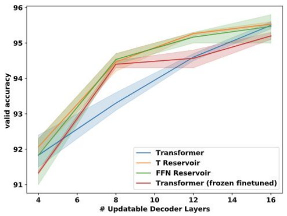


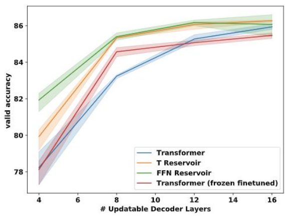
- Up to 25% time savings during pretraining for better performance:





- Downstream performance is better overall:





- Reservoir layers score better on **probing tasks**

Model	Layer	SentLen (Surface)	TreeDepth (Syntactic)	TopConst (Syntactic)	BShift (Syntactic)	Tense (Semantic)	SubjNum (Semantic)	ObjNum (Semantic)	SOMO (Semantic)	CoordInv (Semantic)
Transformer	1	84.56 ± 0.54	32.30 ± 0.41	54.40 ± 0.33	49.99 ± 0.01	80.98 ± 0.32	76.26 ± 0.09	50.01 ± 0.19	76.38 ± 0.61	54.33 ± 0.47
	2	87.22 ± 0.07	33.63 ± 0.57	58.38 ± 0.20	50.12 ± 0.17	82.84 ± 0.68	78.65 ± 0.19	51.47 ± 0.53	78.00 ± 1.12	54.66 ± 0.55
	3	84.25 ± 0.16	32.60 ± 0.17	54.41 ± 0.10	50.02 ± 0.01	81.72 ± 0.59	77.00 ± 0.13	51.32 ± 0.64	76.57 ± 1.13	54.13 ± 0.51
	4	87.37 ± 0.20	32.59 ± 0.29	50.06 ± 0.21	69.76 ± 0.26	81.63 ± 1.17	76.47 ± 0.09	52.41 ± 1.49	76.15 ± 0.84	52.62 ± 1.34
	5	84.61 ± 0.24	31.14 ± 0.48	44.76 ± 0.38	74.82 ± 0.11	80.16 ± 0.19	73.66 ± 0.16	52.95 ± 1.77	72.90 ± 0.21	51.26 ± 1.14
	6	82.56 ± 0.25	30.31 ± 0.40	39.30 ± 0.40	78.80 ± 0.38	81.88 ± 0.47	75.30 ± 0.07	56.21 ± 1.26	74.37 ± 0.16	51.44 ± 1.04
	7	70.85 ± 0.13	26.65 ± 0.72	40.70 ± 0.13	78.98 ± 0.32	85.11 ± 0.31	72.03 ± 0.46	58.15 ± 0.46	68.71 ± 0.91	55.39 ± 0.27
	8	66.23 ± 1.33	23.46 ± 0.44	25.19 ± 1.02	77.42 ± 0.27	80.35 ± 0.45	67.55 ± 0.99	54.94 ± 2.04	63.69 ± 2.32	50.58 ± 0.83
	9	71.17 ± 0.29	31.21 ± 0.31	58.42 ± 0.29	85.55 ± 0.44	86.77 ± 0.19	80.30 ± 0.08	64.36 ± 1.20	81.68 ± 0.45	66.90 ± 0.49
	10	73.19 ± 0.50	27.74 ± 0.53	41.01 ± 0.22	83.56 ± 0.96	86.13 ± 0.35	83.04 ± 0.04	62.01 ± 0.59	79.73 ± 0.21	62.60 ± 1.04
	11	71.37 ± 0.42	30.22 ± 0.28	48.58 ± 0.35	84.40 ± 0.44	87.28 ± 0.59	82.34 ± 0.15	61.10 ± 0.14	80.00 ± 0.40	64.44 ± 0.38
	12	71.66 ± 0.12	33.43 ± 0.18	64.38 ± 0.20	87.38 ± 0.02	88.41 ± 0.09	84.46 ± 0.25	63.01 ± 0.05	81.80 ± 0.27	65.72 ± 0.16
T Reservoir	1	87.75 ± 0.10	31.60 ± 0.21	50.38 ± 0.23	50.00 ± 0.00	80.40 ± 0.18	76.47 ± 0.20	50.53 ± 0.14	73.48 ± 0.15	53.55 ± 0.70
	2	81.28 ± 0.23	34.20 ± 0.41	61.41 ± 0.42	60.64 ± 0.65	81.50 ± 0.77	76.33 ± 0.08	50.73 ± 0.34	74.28 ± 0.67	56.82 ± 0.10
	3	**89.28 ± 0.09**	**36.42 ± 0.11**	**67.36 ± 0.45**	**75.64 ± 0.52**	**85.42 ± 0.18**	**80.53 ± 0.02**	**52.50 ± 1.80**	**78.47 ± 1.81**	**57.16 ± 0.27**
	4	74.31 ± 0.32	32.42 ± 0.83	55.19 ± 0.33	73.41 ± 0.00	79.56 ± 0.00	75.15 ± 0.08	53.68 ± 0.66	75.02 ± 0.19	56.89 ± 0.08
	5	**88.03 ± 0.22**	**38.34 ± 0.64**	**68.65 ± 0.29**	**82.25 ± 0.12**	**86.80 ± 0.02**	**82.27 ± 0.33**	**57.95 ± 0.24**	**80.82 ± 0.91**	**58.05 ± 0.10**
	6	74.55 ± 0.37	33.13 ± 0.29	52.70 ± 0.81	79.21 ± 0.13	85.70 ± 0.36	77.43 ± 0.03	57.26 ± 0.19	75.38 ± 0.66	51.95 ± 1.30
	7	**85.82 ± 0.37**	**37.63 ± 0.13**	**70.43 ± 0.05**	**84.12 ± 0.35**	**86.88 ± 0.07**	**82.86 ± 0.30**	**61.17 ± 0.21**	**80.79 ± 0.17**	**61.83 ± 0.95**
	8	71.69 ± 0.71	30.32 ± 0.01	48.44 ± 0.30	79.12 ± 0.12	84.75 ± 0.09	79.23 ± 0.11	59.53 ± 0.16	76.80 ± 0.41	57.34 ± 0.14
	9	**85.86 ± 0.12**	**37.89 ± 0.03**	**69.53 ± 0.37**	**85.55 ± 0.12**	**87.98 ± 0.22**	**84.13 ± 0.01**	**63.06 ± 0.01**	**82.55 ± 0.31**	**66.07 ± 0.05**
	10	69.22 ± 0.23	25.58 ± 0.35	29.20 ± 0.58	78.57 ± 0.09	85.02 ± 0.03	75.68 ± 0.16	57.55 ± 1.57	74.70 ± 0.02	55.02 ± 0.64
	11	65.70 ± 0.05	30.57 ± 0.03	47.56 ± 0.02	81.20 ± 0.00	86.78 ± 0.02	83.73 ± 0.05	60.38 ± 0.17	80.59 ± 0.15	62.50 ± 0.11
	12	70.61 ± 0.18	34.45 ± 0.20	64.19 ± 0.10	84.53 ± 0.03	87.48 ± 0.16	84.86 ± 0.14	62.75 ± 0.14	82.08 ± 0.03	64.73 ± 0.06

Table 7: RoBERTa Probing Results. The line in bold text are the the frozen layers in the T Reservoir.

FACEBOOK AI

Why does it work?

- Summary:
  - **Train faster** for shorter, with **fewer parameters**, get **better results**

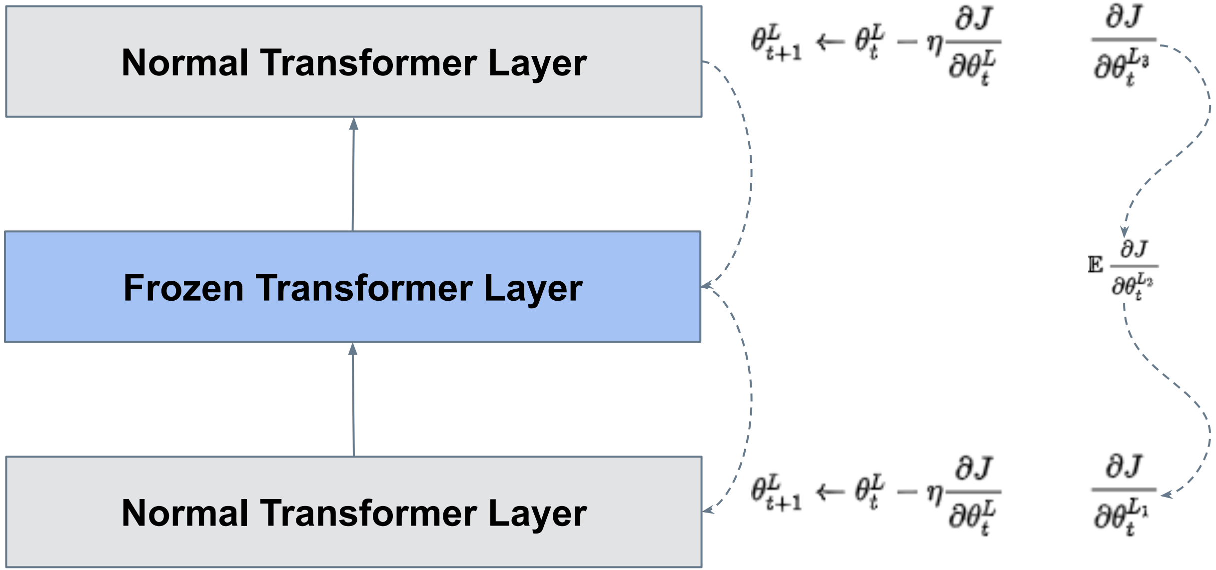
- Findings:
  - Reservoir layers improve efficiency
  - Appears to generalize better
    - Why?
      - Noise helps generalization
      - Detrimental noise can be removed by subsequent layers
      - “Cheap additional parameters”

- More details in the paper (ACL 2021).

Backskip (more efficient Reservoir Transformer)

- Estimate gradients for reservoir layers to skip **activation gradient computation**.

$$\theta_{t+1} \leftarrow \theta_t - \eta \frac{\partial J}{\partial \theta_t};\ \frac{\partial J}{\partial \theta_t} = \frac{\partial J}{\partial L_n} \frac{\partial L_n}{\partial L_{n-1}} \cdots \frac{\partial L_0}{\partial x}$$



Results

- Even **better training saving** with no performance drop.

Model	**Max BLEU**	**AUCC**	**Train time**
Transformer	34.59 ± 0.11	114.57 ± 0.08	142.28 ± 1.87
T Reservoir	34.80 ± 0.07	115.26 ± 0.26	134.49 ± 1.70
Backskip Reservoir	34.75 ± 0.05	115.99 ± 0.23	119.54 ± 1.78

Table 3: Validation max BLEU, AUCC at 4h and wall-clock time per epoch (averaged over multiple runs, in seconds) on IWSLT comparing backskipping with regular and reservoir transformers.