Multimodal ML Reading Group

Universal Transformer [1]

Yao-Hung Hubert Tsai

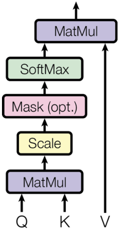
Machine Learning Department, Carnegie Mellon University

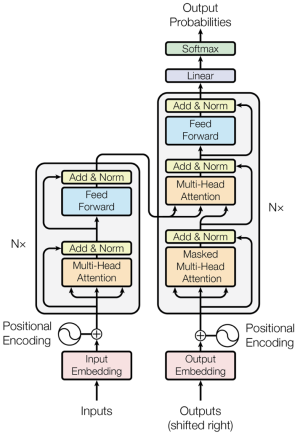
April 17, 2019

Transformer

- Designed for Sequence tasks.
- Core:
  - Scaled Dot-Product Attention

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$





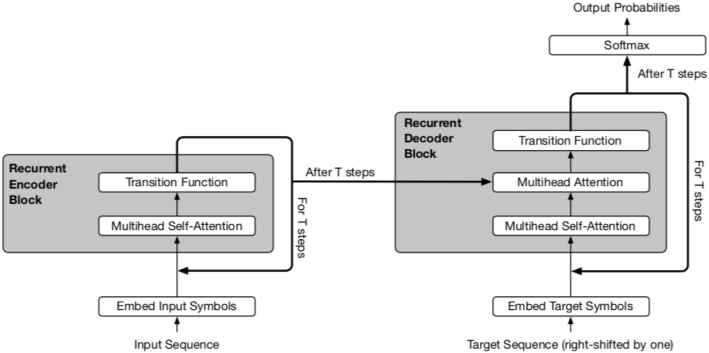
Transformer

- Concurrently process all inputs in a sequence.
  - Easy parallelization and faster training (cf. RNN).
  - Superb in handling long-term dependency.
- Fail to generalize in tasks that RNN succeeds.
  - Copying strings/ logical inference tasks.
  - Hypothesis: These tasks benefit from the recurrent inductive bias of RNN.
- Research Question
  - Can we integrate the recurrent inductive bias into Vanilla Transformer?

Universal Transformer

- High Level: Bring recurrent inductive bias into Transformer.
- Vanilla Transformer:
  - Fixed stack of distinct (attention) layers.
- Universal Transformer:
  - Dynamic stack of identical (attention) layers.

Universal Transformer



- T is determined by adaptive computation time (ACT) [2].

Experiment 1

Model	10K examples		1K examples	
	train single	train joint	train single	train joint
Previous best results:				
QRNet (Seo et al., 2016)	0.3 (0/20)	-	-	-
Sparse DNC (Rae et al., 2016)	-	2.9 (1/20)	-	-
GA+MAGE Dhingra et al. (2017)	-	-	8.7 (5/20)	-
MemN2N Sukhbaatar et al. (2015)	-	-	-	12.4 (11/20)
Our Results:				
Transformer (Vaswani et al., 2017)	15.2 (10/20)	22.1 (12/20)	21.8 (5/20)	26.8 (14/20)
Universal Transformer (this work)	0.23 (0/20)	0.47 (0/20)	5.31 (5/20)	8.50 (8/20)
UT w/ dynamic halting (this work)	**0.21 (0/20)**	**0.29 (0/20)**	**4.55 (3/20)**	**7.78 (5/20)**

Table 1: Average error and number of failed tasks ($> 5\%$ error) out of 20 (in parentheses; lower is better in both cases) on the bAbI dataset under the different training/evaluation setups. We indicate state-of-the-art where available for each, or '-' otherwise.

Experiment 2

Model	Number of attractors						
	0	1	2	3	4	5	Total
Previous best results (Yogatama et al., 2018):							
Best Stack-RNN	*0.994*	0.979	0.965	0.935	0.916	0.880	0.992
Best LSTM	0.993	0.972	0.950	0.922	0.900	0.842	0.991
Best Attention	**0.994**	**0.977**	0.959	0.929	0.907	0.842	**0.992**
Our results:							
Transformer	0.973	0.941	0.932	0.917	0.901	0.883	0.962
Universal Transformer	0.993	0.971	**0.969**	0.940	0.921	0.892	**0.992**
UT w/ ACT	**0.994**	0.969	0.967	**0.944**	**0.932**	**0.907**	**0.992**
Δ (UT w/ ACT - Best)	0	-0.008	0.002	0.009	0.016	0.027	-

Table 2: Accuracy on the subject-verb agreement number prediction task (higher is better).

Model	LM Perplexity & (Accuracy)			RC Accuracy		
	control	dev	test	control	dev	test
Neural Cache (Grave et al., 2016)	**129**	139	-	-	-	-
Dhingra et al. Dhingra et al. (2018)	-	-	-	-	-	0.5569
Transformer	142 (0.19)	5122 (0.0)	7321 (0.0)	0.4102	0.4401	0.3988
LSTM	138 (0.23)	4966 (0.0)	5174 (0.0)	0.1103	0.2316	0.2007
UT *base*, 6 steps (fixed)	131 (0.32)	279 (0.18)	319 (0.17)	**0.4801**	0.5422	0.5216
UT w/ dynamic halting	130 (0.32)	**134** (0.22)	**142** (0.19)	0.4603	**0.5831**	**0.5625**
UT *base*, 8 steps (fixed)	129(0.32)	192 (0.21)	202 (0.18)	-	-	-
UT *base*, 9 steps (fixed)	**129(0.33)**	214 (0.21)	239 (0.17)	-	-	-

Table 3: LAMBADA language modeling (LM) perplexity (lower better) with accuracy in parentheses (higher better), and Reading Comprehension (RC) accuracy results (higher better). '-' indicates no reported results in that setting.

Experiment 3

Model	Copy		Reverse		Addition	
	char-acc	seq-acc	char-acc	seq-acc	char-acc	seq-acc
LSTM	0.45	0.09	0.66	0.11	0.08	0.0
Transformer	0.53	0.03	0.13	0.06	0.07	0.0
Universal Transformer	0.91	0.35	0.96	0.46	0.34	0.02
Neural GPU*	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 4: Accuracy (higher better) on the algorithmic tasks. *Note that the Neural GPU was trained with a special curriculum to obtain the perfect result, while other models are trained without any curriculum.

	Copy		Double		Reverse	
Model	char-acc	seq-acc	char-acc	seq-acc	char-acc	seq-acc
LSTM	0.78	0.11	0.51	0.047	0.91	0.32
Transformer	0.98	0.63	0.94	0.55	0.81	0.26
Universal Transformer	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 5: Character-level (*char-acc*) and sequence-level accuracy (*seq-acc*) results on the Memorization LTE tasks, with maximum length of 55.

	Program		Control		Addition	
Model	char-acc	seq-acc	char-acc	seq-acc	char-acc	seq-acc
LSTM	0.53	0.12	0.68	0.21	0.83	0.11
Transformer	0.71	0.29	0.93	0.66	**1.0**	**1.0**
Universal Transformer	**0.89**	**0.63**	**1.0**	**1.0**	**1.0**	**1.0**

Table 6: Character-level (*char-acc*) and sequence-level accuracy (*seq-acc*) results on the Program Evaluation LTE tasks with maximum nesting of 2 and length of 5.

Experiment 4

Model	BLEU
Universal Transformer *small*	26.8
Transformer *base* (Vaswani et al., 2017)	28.0
Weighted Transformer *base* (Ahmed et al., 2017)	28.4
Universal Transformer *base*	**28.9**

Table 7: Machine translation results on the WMT14 En-De translation task trained on 8xP100 GPUs in comparable training setups. All *base* results have the same number of parameters.

Conclusion

- Universal Transformer (UT) introduces recurrent inductive bias into parallel-in-time computation models (Vanilla Transformers).
- Succeed in many tasks that Vanilla Transformers fail.

Truth

- Very unstable.
  - E.g., 5-layer fails, 6-layer works, and 7-layer fails again.
  - Not happen in identical-layer-RNN/ -TCN [3].
- Connection
  - Neural ODE [4].

$$x^T = f(x^{T-1})$$

  - Fixed-point representations for sequence (can be found in identical-layer-RNN/ -TCN). And the representations have analytical form, which equals to forwarding infinite-depth layers.

References

M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," *arXiv preprint arXiv:1807.03819*, 2018.

A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

S. Bai, J. Z. Kolter, and V. Koltun, "Trellis networks for sequence modeling," *arXiv preprint arXiv:1810.06682*, 2018.

T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems*, pp. 6571–6583, 2018.

The End