3.3.2. Universal Transformers

논문 : https://arxiv.org/abs/1807.03819

2018년 7월 공개 (GPT 공개 1달 후)

ALBERT의 참고문헌으로 정리할 가치가 있어 기록

3줄 요약

1. RNN같은 모델은 순차적으로 계산하기 때문에 Feed Forward Network, CNN에 비해 연산,학습 속도가 느림. 그걸해결할려고 Transformer가 등장했으나 Task별로 파라미터를 학습해야됨!
2. 우리는 모델의 레이어에 대한 시퀀셜 처리로 파라미터 수를 감소시키겠다! -> Transformer의 Encoder,Decoder에 대한 파라미터 공유
3. 논문 투고 당시 SOTA + 레이어가 늘어나도 파라미터가 증가하지 않음!

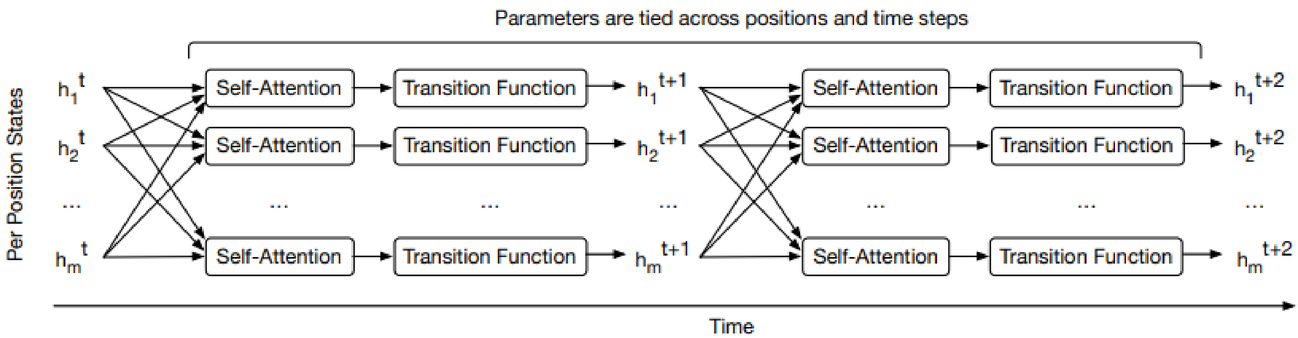

Figure 1: The Universal Transformer repeatedly refines a series of vector representations for each position of the sequence in parallel, by combining information from different positions using self-attention (see Eqn 2) and applying a recurrent transition function (see Eqn 4) across all time steps $1 \leq t \leq T$. We show this process over two recurrent time-steps. Arrows denote dependencies between operations. Initially, h^0 is initialized with the embedding for each symbol in the sequence. h_i^t represents the representation for input symbol $1 \leq i \leq m$ at recurrent time-step t. With dynamic halting, T is dynamically determined for each position (Section 2.2).

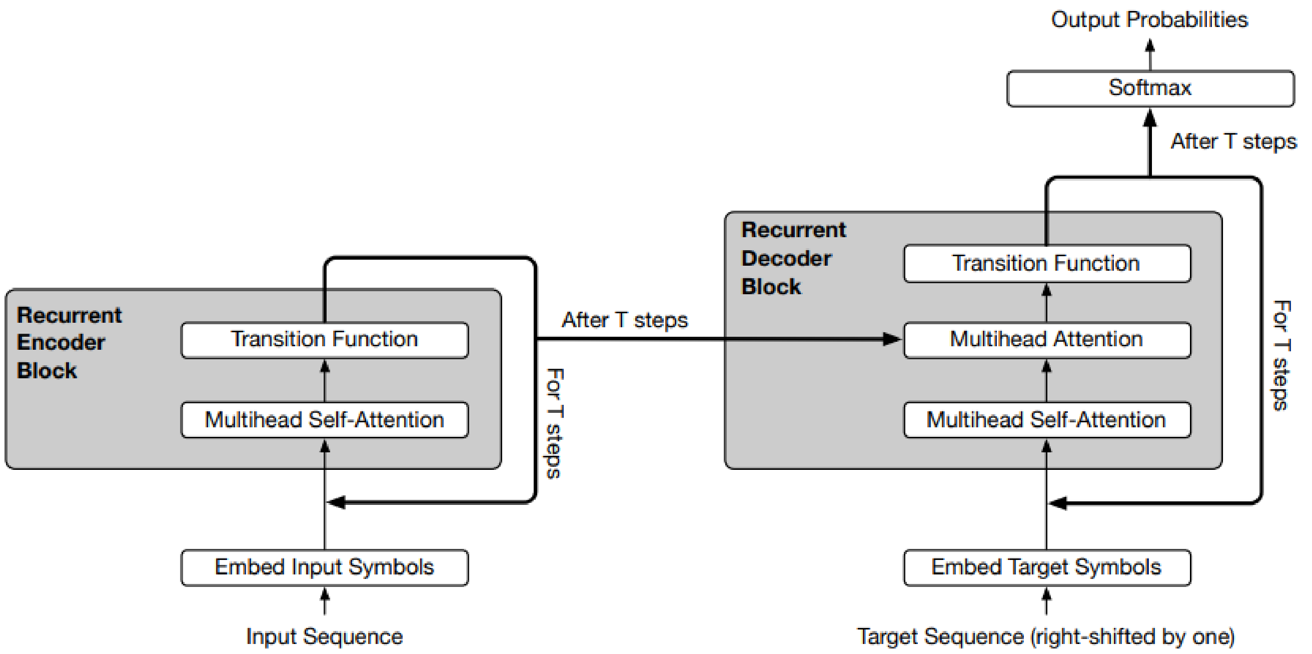

Figure 2: The recurrent blocks of the Universal Transformer encoder and decoder. This diagram omits position and time-step encodings as well as dropout, residual connections and layer normalization. A complete version can be found in Appendix A. The Universal Transformer with dynamic halting determines the number of steps T for each position individually using ACT (Graves, 2016).

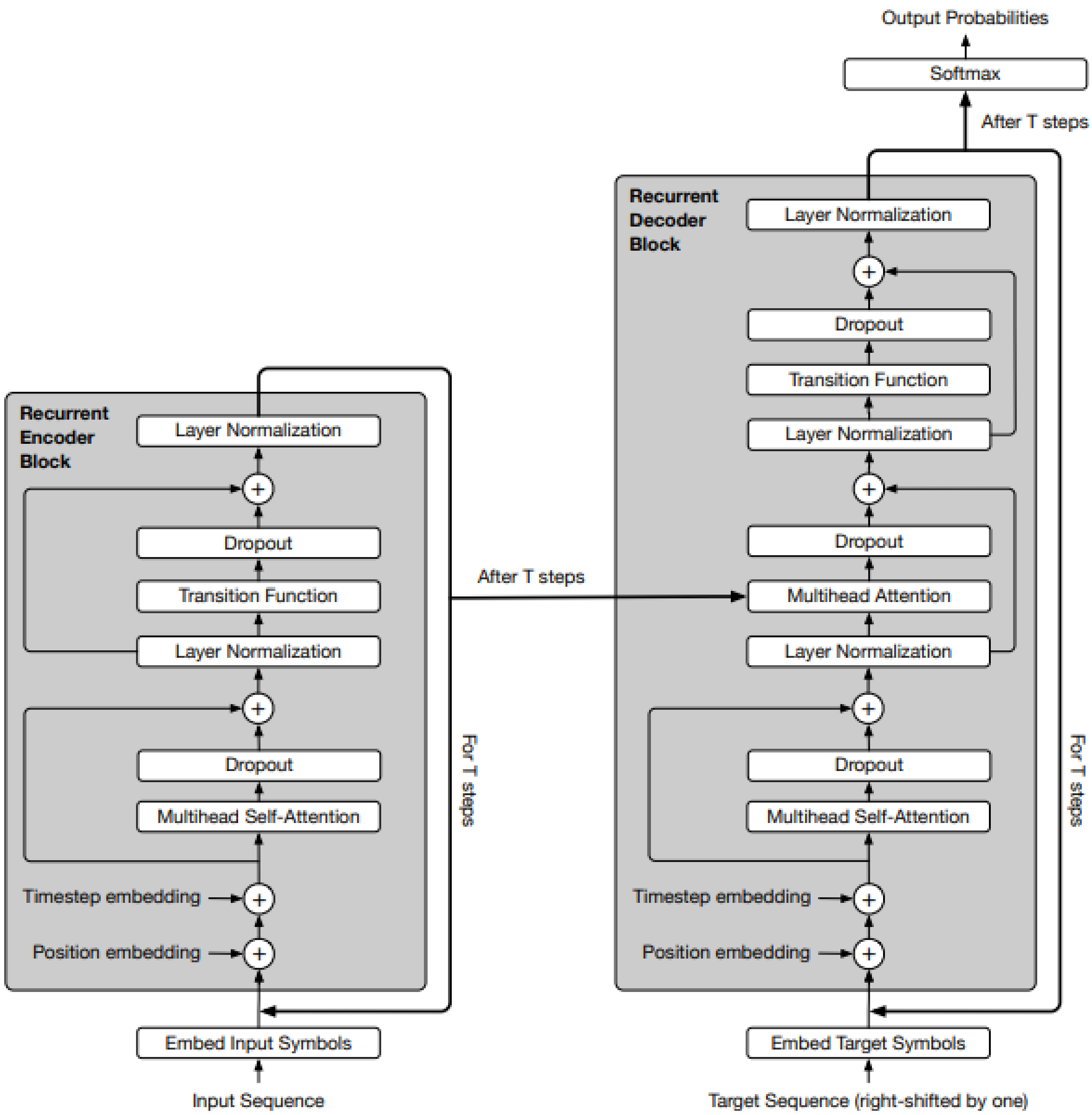

Figure 4: The Universal Transformer with position and step embeddings as well as dropout and layer normalization.