

# Attention is all you need

VLC =  $\iiint$  Vision  
Learning *vLaC*  
Control

A little attention, please?

Jonathon Hare

Vision, Learning and Control  
University of Southampton

## Static attention

$$\hat{\mathbf{X}} = \text{softmax}(\mathbf{W})\mathbf{X}$$

or, *factorised*,

$$\hat{\mathbf{X}} = \text{softmax}(\mathbf{W}_1\mathbf{W}_2)\mathbf{X}$$

$$\hat{\mathbf{X}} = f(\mathbf{Z}, \theta)\mathbf{X}$$

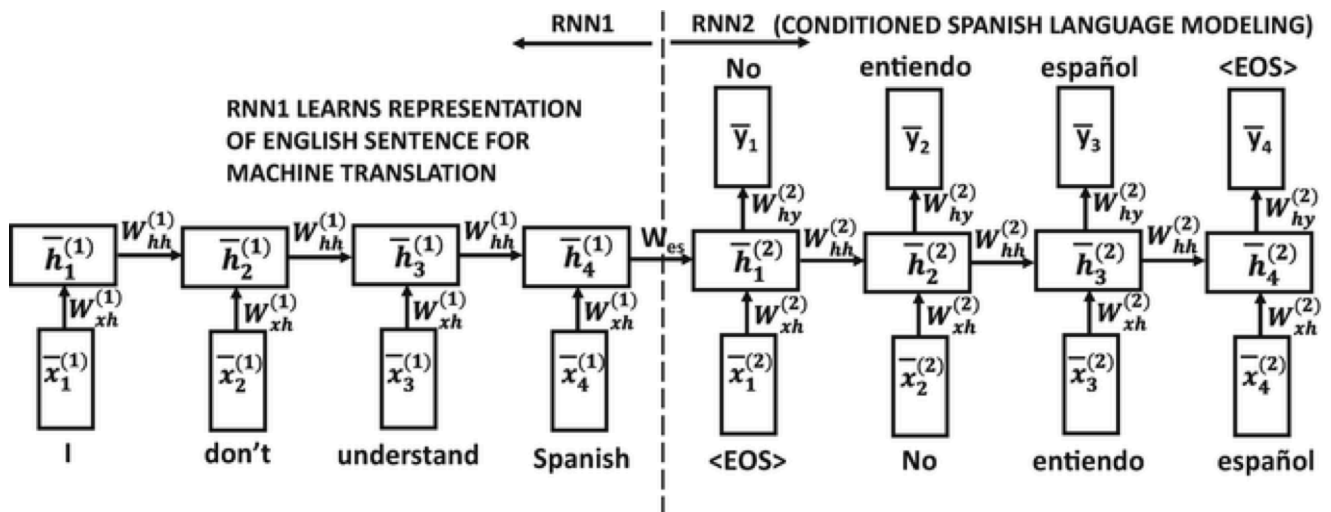
or, *factorised*,

$$\hat{\mathbf{X}} = f(\mathbf{Z}_f, \theta_f)g(\mathbf{Z}_g, \theta_g)\mathbf{X}$$

## (Dynamic) Attention vs Self-attention

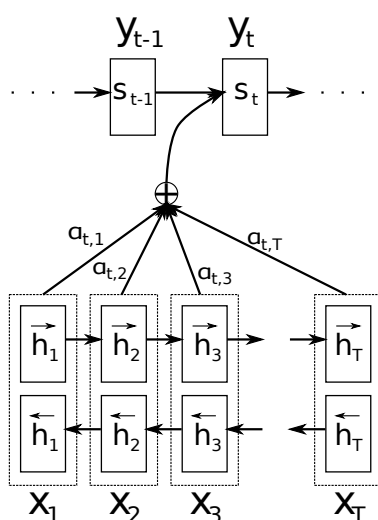
- In regular attention, the weights applied to  $\mathbf{X}$  are computed using some additional auxiliary input (e.g.  $\mathbf{Z}$ )
- Self-attention is only computed as a function of  $\mathbf{X}$  (equivalently  $\mathbf{Z} = \mathbf{X}$ )

# Dynamic Attention Example - Seq2Seq models



[https://link.springer.com/chapter/10.1007/978-3-319-73531-3\\_10](https://link.springer.com/chapter/10.1007/978-3-319-73531-3_10)

# Dynamic Attention Example - Seq2Seq models



$$\alpha_t = \text{softmax}([\text{score}(s_{t-1}, h_1), \dots, \text{score}(s_{t-1}, h_T)]^T)$$

$$\text{score}(s, h) = v^T \tanh(W[s; h])$$

$$c = \alpha_t^T H \text{ where } H = [h_1, h_2, \dots, h_T]^T$$

commonly known as “Additive Attention”, even though it is based on concatenation!

Bahdanau, D., Cho, K. and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. ICLR 2015.

# Hard Attention vs Soft-attention

- Soft-attention: use the softmax to smoothly attend mostly to one thing (but capture a bit of everything else)
- Hard attention: you specifically only attend to one thing: tricks (e.g. policy gradients or ST operator) from last lecture required to learn

## Aside: Relaxation of a map/hashtable/dictionary

## Scaled dot-product attention

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

- In the previous Seq2Seq example we could replace additive attention with scaled dot-product attention with something like  $\mathbf{Q} = f(\mathbf{s}_{t-1})$ ,  $\mathbf{K} = g(\mathbf{H})$  and  $\mathbf{V} = j(\mathbf{H})$ .
- The scaling  $1/\sqrt{d_k}$  is just to improve learning (larger  $d_k$  implies larger dot products, which pushes further towards the flatter bit of the softmax, and thus smaller gradients.)

## Scaled dot-product self-attention

$$\text{SelfAttention}(\mathbf{X}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

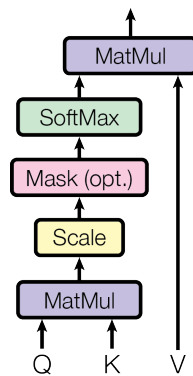
$$\mathbf{Q} = \mathbf{W}_q \mathbf{X}$$

$$\mathbf{K} = \mathbf{W}_k \mathbf{X}$$

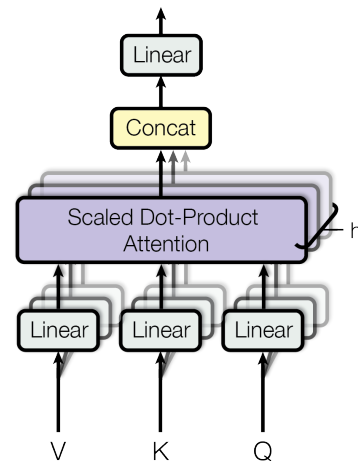
$$\mathbf{V} = \mathbf{W}_v \mathbf{X}$$

# Multi-head Attention

Scaled Dot-Product Attention



Multi-Head Attention



$$\text{MultiHead}(Q, K, V) = [\text{head}_1; \dots; \text{head}_n] W^O$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30.

# The Transformer

