Each encoder position can attend to every position in the preceding layer of the encoder.

3. The second location is at the self-attention layers of the decoder. Each decoder position can attend to every other position in the decoder preceding it and to itself.

The encoder is comprised of 6 identical layers, each made up of two sub-layers. The first is a multi-head self-attention layer and the second is a feed-forward network, positionwise fully connected. The output of each sub-layer is LayerNorm(x + Sublayer(x)), where Sublayer(x) is the function implemented by that sub-layer. Normalization is performed on the residual connections (see the green "Add & Norm" box in the figure at page 12).

The decoder, too, is comprised of 6 identical layers, each made up of three sub-layers: the two sub-layers of the encoder, plus a sub-layer performing multi-head attention over the encoder stack output. Each sub-layer is followed by a normalization layer.

For Transformer to take into account the order of the input sequence, even without any recurrence or convolution, it must somehow be provided with the relative or absolute position of the sequence tokens. This is the function of the "positional encodings" at the bottoms of the encoder and decoder stacks.

Google team reports that they trained Transformer on a wide variety of completely unrelated datasets - and it was subsequently successfully tested on all of them.

As an example, the problem of parsing English sentences has been under research for decades. It can be solved with sequence-to-sequence neural networks, but requires a lot of tuning. The authors report it took them only a few days to add the parsing data-set generator to Transformer and achieve very good results after a week's long training:

Parsing Model	F1 score (higher is better)
Transformer (T2T)	91.3
Dyer et al.	91.7
Zhu et al.	90.4
Socher et al.	90.4
Vinyals & Kaiser et al.	88.3

Transformer is written using TensorFlow tools. Most staples of deep learning are