* **Recurrent NNs, LSTM** and **Gated Recurrent NNs** have been established as state-of-the-art approaches in sequence modeling and transduction problems such as language modeling and machine translation.
* *Def.* **Transduction problem** is a type of ML task where the goal is to transform or convert one structured object into another.

**Model Architecture**

* Most competitive neural sequence transduction models have **encoder-decoder** structure.
* The **encoder** maps an input sequence of symbol representations to a sequence of continuous representations .
* Given **z,** the decoder then generates an output sequence of symbols one element at a time.
* The model is *auto-regressive,* consuming the previously generated symbols as additional input when generating the text.

A diagram of a process

Description automatically generated

*Def.* **Attention function** maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a computability function of the query with the corresponding key.

A math equation with a square and a equal sign

Description automatically generated**Scaled Dot-Product Attention** is computed on a set of queries simultaneously, packed together into a matrix *Q.* The keys and values are also packed together into matrices *K* and *V.* The matrix of outputs is computed like this: ,

where is the dimension of keys vector.



The dot product is scaled by in order to prevent it from growing large in magnitude (and the *softmax* function from the regions where it has extremely small gradients).

**Multi-head attention** allows the model to jointly attend to information from different representation subspaces at different positions.A close-up of black text

Description automatically generated

A diagram of a multi-head attention

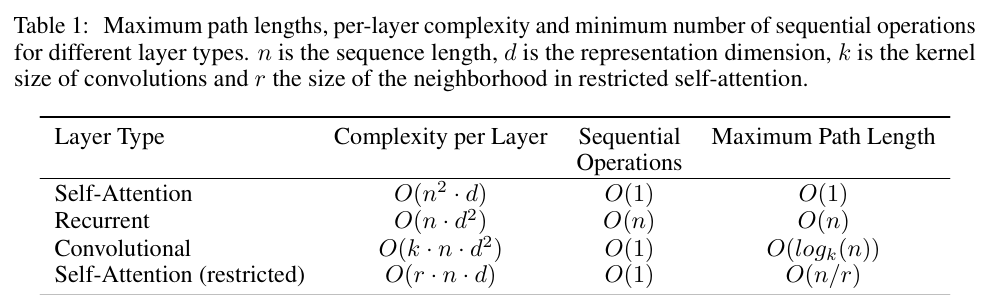
Description automatically generated

The **Transformer** uses *multi-head attention* in three different ways:

1. In “encoder-decoder attention” layers, the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder.
2. The encoder contains self-attention layers. In a self-attention layer all of the keys, values and queries come from the same place, in this case, the output of the previous layer in the encoder.
3. Self-attention layers in the decoder allow each position in the decoder to attend to all positions in the decoder up to and including their position. We need to prevent leftward information flow in the decoder to preserve the auto-regressive property.

A black text on a white background

Description automatically generatedEach of the layers in the encoder and decoder contains a fully connected *feed-forward network*, which is applied to each position separately and identically. This consists of two linear transformations with a ReLU activation in between.



We must inject some information about the relative or absolute position of the tokens in the sequence, in order for the model to make use of the order of the sequence, since our model contains no recurrence and no convolution.

A close-up of a number

Description automatically generated



*Self-attention layers* are faster than *recurrent layers* when the sequence length *n* is small than the representation dimensionality *d,* which is most often the case with sentence representations used by state-of-the-art models in machine translations, such as *word-piece* and *byte-pair* representations.

*Def.* **FLOP** – a single floating-point operation. For instance, adding two floating-point numbers counts as one FLOP. When discussing the complexity or computational demands of a model, **FLOPs** refers to the total number of floating-point operations required to execute a particular task or run a neural network.