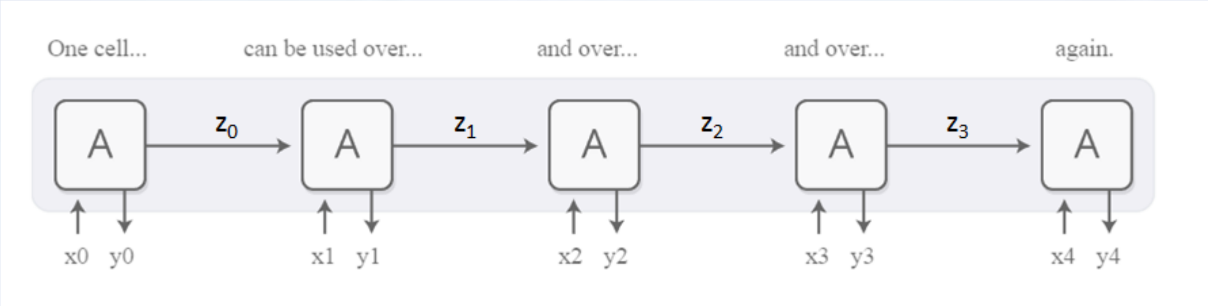
then looks at input x1, but this time with the additional input of z0, and outputs y1 and z1, and so on.

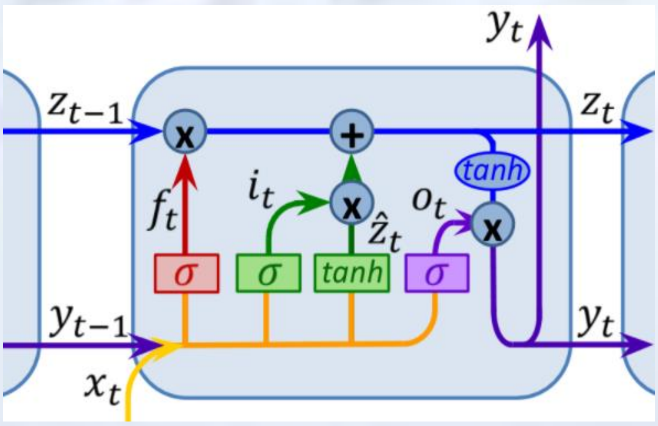


However, basic RNNs had difficulty in making use of information gained early in a long sequence, much later in that sequence, an ability often needed, for instance, in dealing with natural language with its long-distance dependencies. Most of the recent impressive successes of RNNs, in translation, voice recognition, image classification and more, have come from the special class of RNNs called Long Short-Term Memory RNNs (or LSTMs for short).

(B) LSTM:

LSTMs proposed in 1997 by **Sepp Hochreiter** and [Jürgen Schmidhuber]() are designed to remember information for long periods by default, with little or no change, so forgetting must be explicitly handled in this model, rather than assumed. Like in standard RNNs, LSTMs have the structure of repeating identical units of neural network.

There are numerous variants of LSTM units, each unique in some particular feature: we'll describe the basic typical structure of an LSTM unit. The basic difference with respect to an RNN unit is that, rather than a single simple layer (for instance, *tanh*), there are 4 layers designed to produce a clever interaction.



Key to the functioning of LSTMs is the hidden state. The blue line running along the top of the diagram, throughout the chain of (looped) units, only affected by controlled interactions. Hidden state z_{t-1} is received as input by the node from its predecessor. Hidden state z_t, passed on as output by the node, will be changed