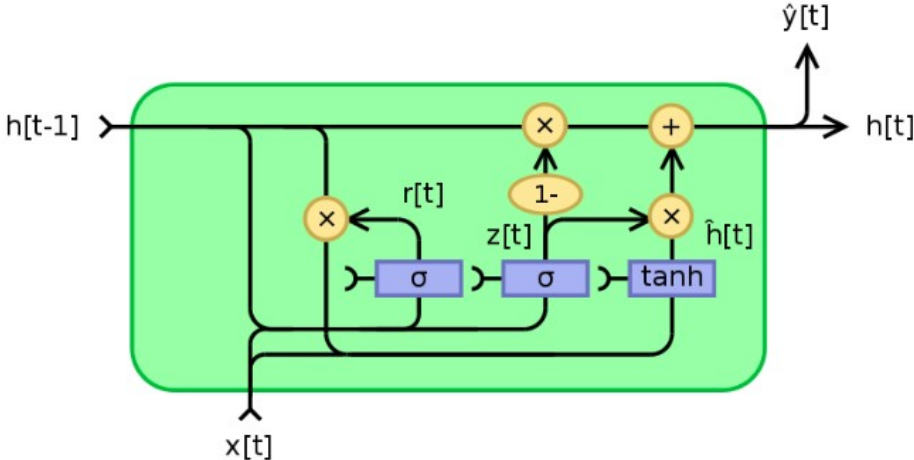


Word	Description
Gated Recurrent Unit (GRU)	<p>The GRU is a variant of the LSTM (Long Short Term Memory) and was introduced by K. Cho. It retains the LSTM's resistance to the vanishing gradient problem, but because of its simpler internal structure it is faster to train.</p> <p>Instead of the input, forget, and output gates in the LSTM cell, the GRU cell has only two gates, an <b>update gate</b> <math>z</math>, and a <b>reset gate</b> <math>r</math>. The update gate defines how much previous memory to keep, and the reset gate defines how to combine the new input with the previous memory.</p>  <p>The diagram illustrates the internal structure of a Gated Recurrent Unit (GRU). It takes two inputs: the previous hidden state <math>h[t-1]</math> and the current input <math>x[t]</math>. The reset gate <math>r[t]</math> is calculated by applying a sigmoid function <math>\sigma</math> to a combination of <math>x[t]</math> and <math>h[t-1]</math>. The update gate <math>z[t]</math> is calculated by applying a sigmoid function <math>\sigma</math> to another combination of <math>x[t]</math> and <math>h[t-1]</math>. The reset gate <math>r[t]</math> is used to scale the input <math>x[t]</math> (indicated by a multiplication node <math>\times</math>). The scaled input is then passed through a <math>\tanh</math> activation function. The output of the <math>\tanh</math> function is multiplied by <math>1 - z[t]</math> (indicated by a multiplication node <math>\times</math>). The previous hidden state <math>h[t-1]</math> is multiplied by <math>z[t]</math> (indicated by a multiplication node <math>\times</math>). The results of these two multiplications are then added together (indicated by a plus node <math>+</math>) to produce the new hidden state <math>h[t]</math>. The output <math>h[t]</math> is also used to calculate the predicted output <math>\hat{y}[t]</math>.</p> <p>Gated Recurrent Unit, fully gated version</p>