## 1.6 Recurrent Neural Networks

The general idea of an RNN is that for a given input $x^{\langle t \rangle}$, we compute an activation $a^{\langle t \rangle}$ as a function of both $x^{\langle t \rangle}$ and a previous activation, $a^{\langle t-1 \rangle}$. The simplest form can be expressed as

$$a^{\langle t \rangle} = g(W_a[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_a) \ . \tag{14}$$

This simple formulation, when used over many iterations, is commonly plagued by the vanishing gradient problem, just as DNNs are. For instance, in the sentence *The cat, who just ate lunch at Popeyes, was full*, the RNN must memorize whether *cat* is singular or plural for the entire modifying clause, and it is hard to backpropagate that information through all the middle words.

We address this problem with the GRU cell (Gated Recurrent Unit). The intuition is to have some memory cell $c$, which keeps track of relevant information. In the above sentence, for example, a single feature in $c$ could represent whether the cat is singular or plural.

For every new time step, we compute a candidate new memory cell $\tilde{c}^{\langle t \rangle} = \tanh(w_c[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$ as a function of the previous memory cell and the current input. But how important is this new memory cell with respect to the old memory cell? We can give it a weight from 0 to 1 (in practice, either very close to zero or very close to one) computed using an *update gate* $\Gamma_u = \sigma(w_u[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$. Now, we finally arrive at our new memory cell $c^{\langle t \rangle} = \Gamma_u \odot \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) \odot c^{\langle t-1 \rangle}$.

It turns out that in addition to the update gate, which dictates how much of the new memory cell comes from the candidate memory cell, a *relevance gate*, $\Gamma_r$, which tell us the relevance of the previous memory cell $c^{\langle t-1 \rangle}$, also helps. We therefore arrive at the five equations for the GRU:

$$\tilde{c}^{\langle t \rangle} = \tanh(w_c[\Gamma_r \odot c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c) \tag{15}$$

$$\Gamma_u = \sigma(w_u[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u) \tag{16}$$

$$\Gamma_r = \sigma(w_r[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_r) \tag{17}$$

$$c^{\langle t \rangle} = \Gamma_u \odot \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) \odot c^{\langle t-1 \rangle} \tag{18}$$

$$a^{\langle t \rangle} = c^{\langle t \rangle} \tag{19}$$

For the LSTM cell, we use three gates instead of two. We remove the relevance gate and replace it with an *forget gate* and *output gate*. The forget gate $\Gamma_f$ replaces $(1 - \Gamma_u)$ from the GRU, and the output gate gives $a^{\langle t \rangle} = \Gamma_o \odot \tanh c^{\langle t \rangle}$.