**Attention Model**

* **Introduction**

Before the attention model, we use an encoder-decoder network to work with sequential data, such as predicting time-series or translation. However, the model suffers two drawbacks: i) All sequences have different length and type of information stored and it is unlikely we can represent the entire information using only one memory block ii) Vanishing gradient. Attention model comes to be to resolve those drawbacks.

* **Attention Model**

Pay attention now, because we will get into the attention model now.

Instead of reading input sentence in their entirety and generating the output sentence all at once, the attention mechanism allows a network to focus on the most relevant parts of the input and produces output a bit at a time. A standard attention model consists of a pre-attention LSTM, an attention block and a post-attention LSTM. The function of each block is explained as follows:

**Pre-attention LSTM**

The pre-attention LTSM is nothing interesting, it is the same as an encoder LSTM, inputting the sequence and outputting the activation or the hidden state.

**Attention Block**

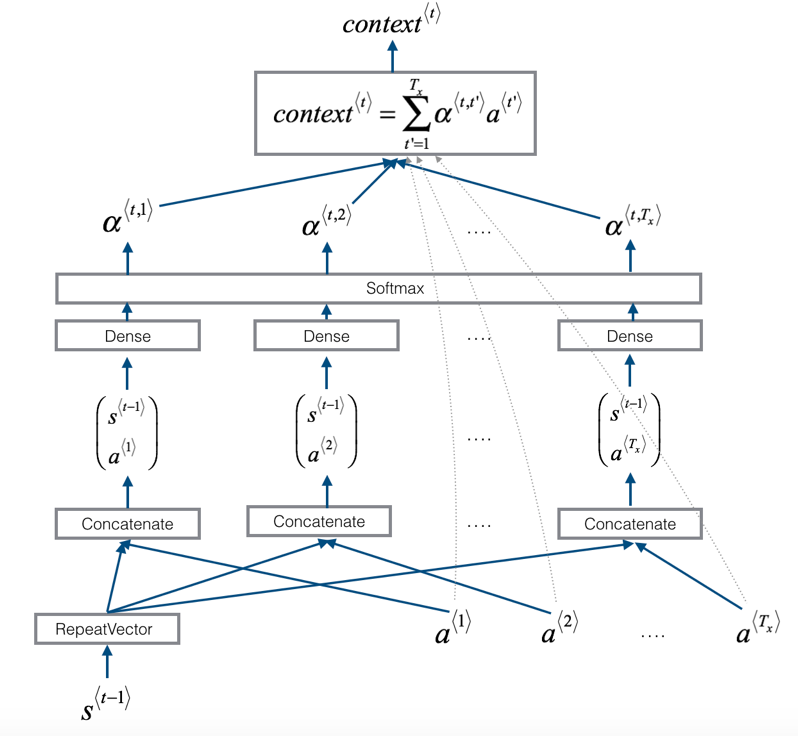
What is interesting is the attention block, each attention block inputs the activation of the input time-sequences we are interested in (in this diagram is the entire sequence and in reality, we only choose the relevant part). It outputs a context vector which, as the name suggests, provides context information for generating the output we are interested in.

The context vector is calculated by adding up the activations from pre-attention LSTM weighted by an **attention weight** , which tells us the amount of attention that the model pays to to generate

And then this context vector is fed into the post-attention LSTM to generate the output.

**Attention Weight**

The attention weight is dependent on the cell state of the post-attention LSTM and the activation of the pre-attention LSTM. What it means is that we want to determine the amount of attention to pay to each input using the input itself and the previous output we have generated.

****

**Figure 3. The Attention Block**

It turns out we don’t need to handcraft the attention weight on our own, we can learn it using a small neural network. The neural network combine and into a single scalar , which is then normalized using a softmax function to yield

|  |  |
| --- | --- |
|  |  |

**Continuous Model**

However, it turns out we cannot use the attention model directly into our data because all the classical RNN networks assumes that the data are sampled with constant interval. However, our astronomical data is not sampled on a regular interval. So, we must deal with it with care. There are several solutions to that

* Use a continuous model – GP, CARIMA
* Divide the data into several intervals / bins and synchronize the data format for each bin. But this approach usually leads to loss of information.
* Treat the time as an extra feature

The LSTM doesn’t have a continuous equivalence, so we treat the time as an extra feature using phased LSTM.

**Phased LSTM**

* **Classical LSTM**

The main difference to classical RNNs is the use of the gating functions which represents the input, forget and output gate at time respectively. is the cell activation vector whereas and represent the input feature vector and the hidden output vector respectively.

* **Phased LSTM**

The Phases LSTM model extends the LSTM model by adding a new time gate . The opening and closing of this gate are controlled by an independent rhythmic oscillation specified by three parameters.

* The first parameter controls the real-time period of the oscillation
* The second controls the ratio of the duration of the “open” phase to the full period.
* The third controls the phase shift of the oscillation to each Phases LSTM cell.

All the parameters can be learned during the training process. We propose here a successful linearized formulation of the time gate, with analogy to the rectified linear unit that propagates gradients as well

is an auxiliary variable which represent the phase inside the rhythmic cycle. The gate has three phases

* Phase one: “openness” of the gate rises from to
* Phase two: “openness” of the gate drops from t
* Phase three: the gate is closed and the previous cell state is maintained

The leak with rate is active in the closed phase and plays a similar role as the leak in a parametric “leak” rectified linear unit by propagating important gradient information even when the gate is closed.