A theoretical framework for Attention

Erik de Godoy Perillo
Advisor: Profa. Dra. Esther Luna Colombini

State University of Campinas

August 1, 2019

1 Introduction

In this document, we briefly formulate a theoretical framework for the concept of Attention. This framework consists of two main parts:

- A *definition* of Attention in terms of its functionalities;
- A *model* of Attention.

Note that the first element aims at answering the question "*What* is Attention?" while the second aims at explaining *how* Attention emerges.

2 A definition of Attention

In this definition, we define a set of *entities of interest* and the phenomenon of Attention in terms of its *functionalities* and how it relates to the entities.

2.1 Why is our definition good?

We believe our definition encompasses what we generally (and intuitively) refer to as attention while being not too broad. Also, the definition is given in terms related to Computer Science so its functionality nicely translates to the domain, which is important since we (so far) intend to develop AI using computers as we know it today. We may not encompass every aspect of Attention and even be conflicting with other definitions. However, this is the set of postulates that we think is the most precise and useful and thus this is what we choose to use for future work to be based on.

2.2 Entities

Below is the list of entities - or "terms" - we use in this work, along with a brief discussion of the meaning we give to each term in the context of this work.

- **Data:** information, stimuli. It may be internal or external. Examples: visual information, audio, memories.
- **Program:** algorithm, sequence of computer (or mental) operations. Programs use data as input in order to carry out a sequence of operations that produces output data and/or actions.
- **Process:** the execution of a program on a specific data instance.
- **Computer:** the executor of processes, the brain.
- **Resource:** when not specified, we mean computational resources, e.g. CPU time.
- **Time:** the flow of time.
- **World:** the external environment.
- **Agent:** the actor in the world.
- **Actions:** the interaction of the agent with the world.
- **Goals:** the ends, objectives to be met.

2.3 What is Attention?

Data, *programs* and *processes* are virtually *infinite*. Computational *resources* and *actions* are finite. *Attention* is *the system of allocating resources to processes*. In other words, *attention* is the entity in *agents* that, given *context* and a set of *processes*, *allocates resources* to execute each of them in order to *produce outputs* in form of *data* and *actions* in a *correct sequential manner* and in *sensible time* in order to reach *goals*.

3 How Attention happens?

We propose a model for the phenomenon of Attention. Following the definition given in Section 2, in our model we assume that Attention takes place in the context of a "mind" that behaves like a computer that executes processes: it processes inputs via algorithms to produce an output in discrete steps.

We believe Attention can be modeled as a process that takes place *along the course of time*. At each time step in the process, *inputs* of different classes are used to produce a certain *output*. In a given *timeframe*, the sequence of processing steps produce the emergence of *focus*. In this section, we approach the entities and taxonomy related to the *end-to-end* process itself. In the next section, we specifically model processing at each time step so that the processes described in this section emerge.

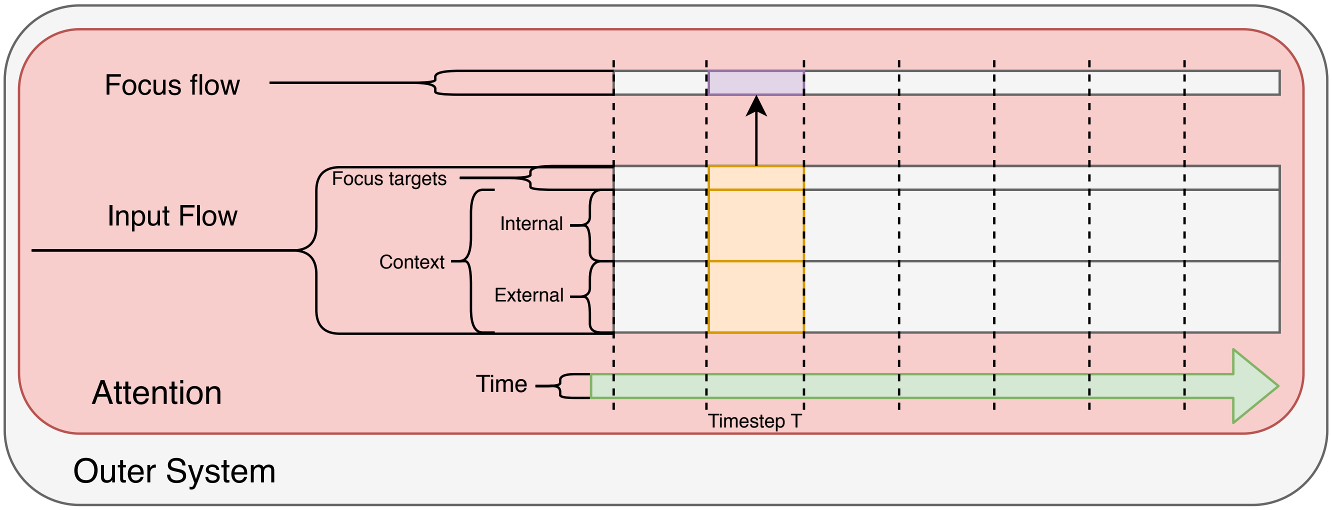

Figure 1: The process of Attention.

3.1 Input dimensions

The information to be used by the focus may be classified as:

- **Context**: Context for the focus to happen. It may be either *internal* or *external*.
- **Focus target**: Subjects of the focus.

3.2 Focus dimensions

3.2.1 Continuity

The continuity of focus can be either:

- **Soft**: Continuously spread accross the targets, "weighting".
- **Hard**: Discrete selection of one or more targets.

This division has been popular in Deep Learning research lately.

3.2.2 Flow along time

The flow of focus along time can be classified as:

- **Selective**: Selection of targets from the space of possible targets. The most common.
- **Oriented**: Changing the focus targets in a sequential manner so as to perform a task.
- **Sustained**: Keeping the focus on a set of targets along a period of time.
- **Divided**: Dividing focus accross a set of targets.

3.2.3 Focus Targets

The type of focus target may be:

- **Computing** resources: Selection of computation time for a given task.
- **Program**: Selection of a program among others to be executed.
- **Data**: Selection of data (mostly stimuli). The selection of data may be:
  - **Feature-based**: based on the features of a stimuli, such as color, orientation...
  - **Location-based**: based on the location of the stimuli, such as the coordinates of a pixel.
  - **Object-based**: based on a object (object may vary in the context of the task), such as a person in an image.

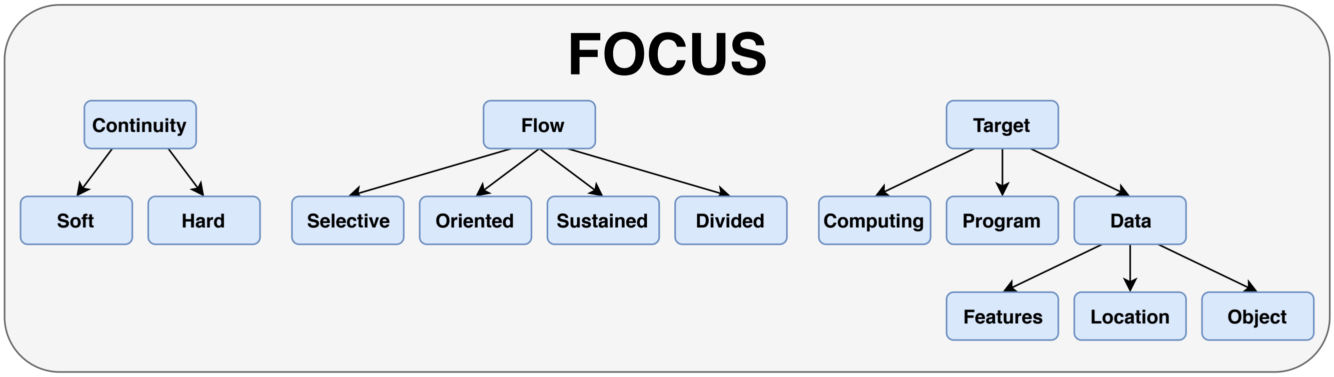

Figure 2: Focus attributes.

3.3 Examples

3.3.1 Image captioning with attention

The work on [1] proposes a recurrent neural network with attention for image captioning. While there are many abstract processing substeps in the process, the *end-to-end* effect is that of a focus with *soft* continuity, with *selective* and *orienting* flow, targetting *data* on a *Location-based* manner. Figure 3.3.1 illustrates the proposed model.

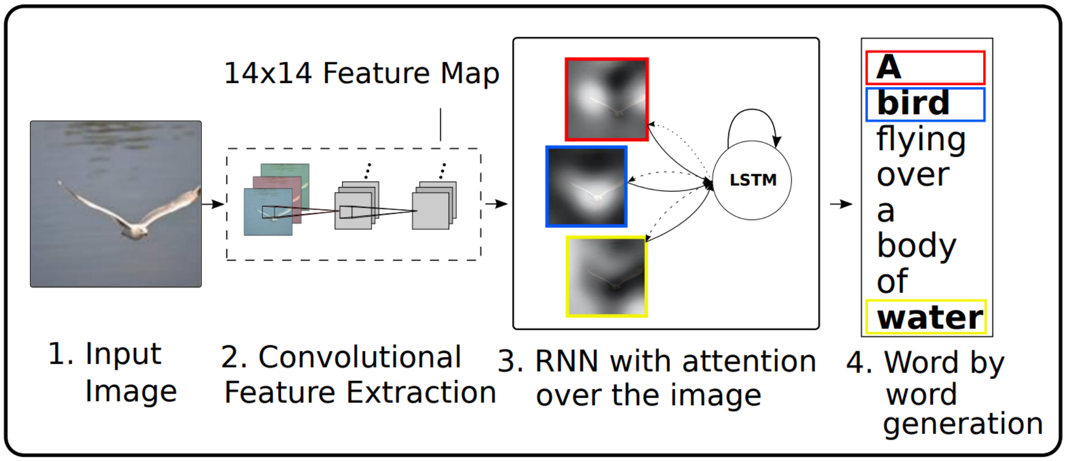
14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Figure 3: Diagram of natural language image description using Attention (from [1]).

4 Attentional Module

We propose that *Attention can emerge in any process* to be executed by such mind by means of *a series of components* — which we call *attentional modules*. These modules can alter data being processed and the execution flow of the algorithm and provide the functionalities of Attention.

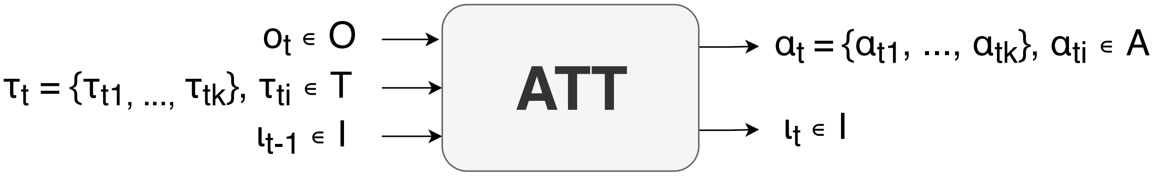
$o_t \in O$
$\tau_t = \{\tau_{t1}, ..., \tau_{tk}\}, \tau_{ti} \in T$
$\iota_{t-1} \in I$
ATT
$\alpha_t = \{\alpha_{t1}, ..., \alpha_{tk}\}, \alpha_{ti} \in A$
$\iota_t \in I$

Figure 4: Attentional module.

Figure 4 illustrates the attentional module. At each time step t, the module receives as *input*:

- Current *outer state* $o_t \in O$, where O is the *outer state set*.
- Group of *focus targets* $\tau_t = \{\tau_{t1}, \dots, \tau_{tk}\}, \tau_{ti} \in T$, where T is the *focus target set*.
- Past *inner state* $\iota_{t-1} \in I$, where I is the *inner state set*.

The module produces as *output* (as a function of both inputs):

- Current *inner state* $\iota_t \in I$.
- Current *focus output* $\alpha_t = \{\alpha_{t1}, \dots, \alpha_{tk}\}, \alpha_{ti} \in A$, where A is the *focus output set*.

4.1 Focus output

The focus output is the main element of the module: it can be used to allocate *finite resources* to a set of "candidate targets" by giving them an "importance score" which can be used in any arbitrary way in following steps – such as choosing the amount of computation to be dedicated to an element or which elements will be used as input to another step. Each element α_{tk} is respective to a target element τ_{tk}. Target elements ($\tau \in T$) may effectively be *programs* (tasks) or *data*.

4.1.1 Soft and Hard Attention

The focus output will generally be such that it acts as either *Soft* or *Hard Attention*:

- **Soft Attention:** $A = [0, 1]$, with $0 \leq \sum_{i=1}^{k} \alpha_{ti} \leq 1$
- **Hard Attention:** $A = \{0, 1\}$, with $0 \leq \sum_{i=1}^{k} \alpha_{ti} \leq M$ and $0 \leq M \leq |\tau_t|$

4.1.2 Using the output of the focus function

The focus function output may be used for the allocation of some resource in various ways, such as:

- Choosing the **amount** of **computation time** to be used at a certain step;
- Choosing a **subset** of **elements** to carry out further computations;
- **Weighting elements** to perform a certain computation.

4.2 Modules forming an attentional system

A system with Attention may contain more than one attentional modules – even in a recursive manner. Together, these modules always perform the function to allocate resources to processes.

Figure 4.2 shows the diagram of a possible system with attention. The module *TaskATT* uses hard attention to select a certain task k to be executed for some time at time step t. Among the computations of task k, there is the module *DataATT*, uses soft attention to allocate resources to a set of items. It is worth noting that time is relative to each attentional module: *TaskATT* has a temporal course over time steps t that is different from that of *DataATT*, which is over time steps t'. Also, their sets of inputs and outputs may differ.

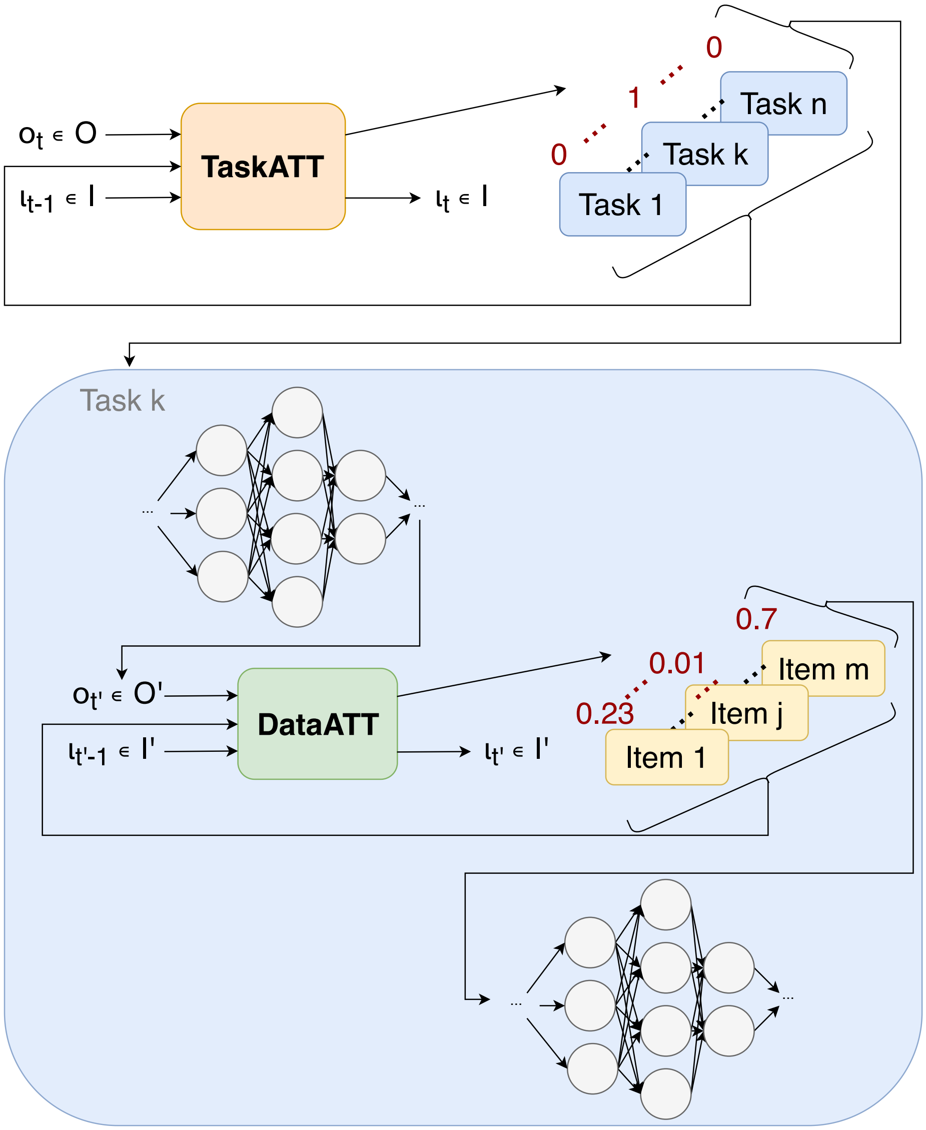

Figure 5: Example of a system that uses attention.

5 Validating the framework

In this section, we investigate some recent works on attention and see how they fit in our framework.

5.1 Image Caption Generation

The work [4] is among the first to propose using attention to image caption generation: the encoding of the input image is represented as a set of vectors – each respective to a certain spatial region of the image – and the attentional component gives weights to each vector at each step in order to produce another vector to be used in further computations.

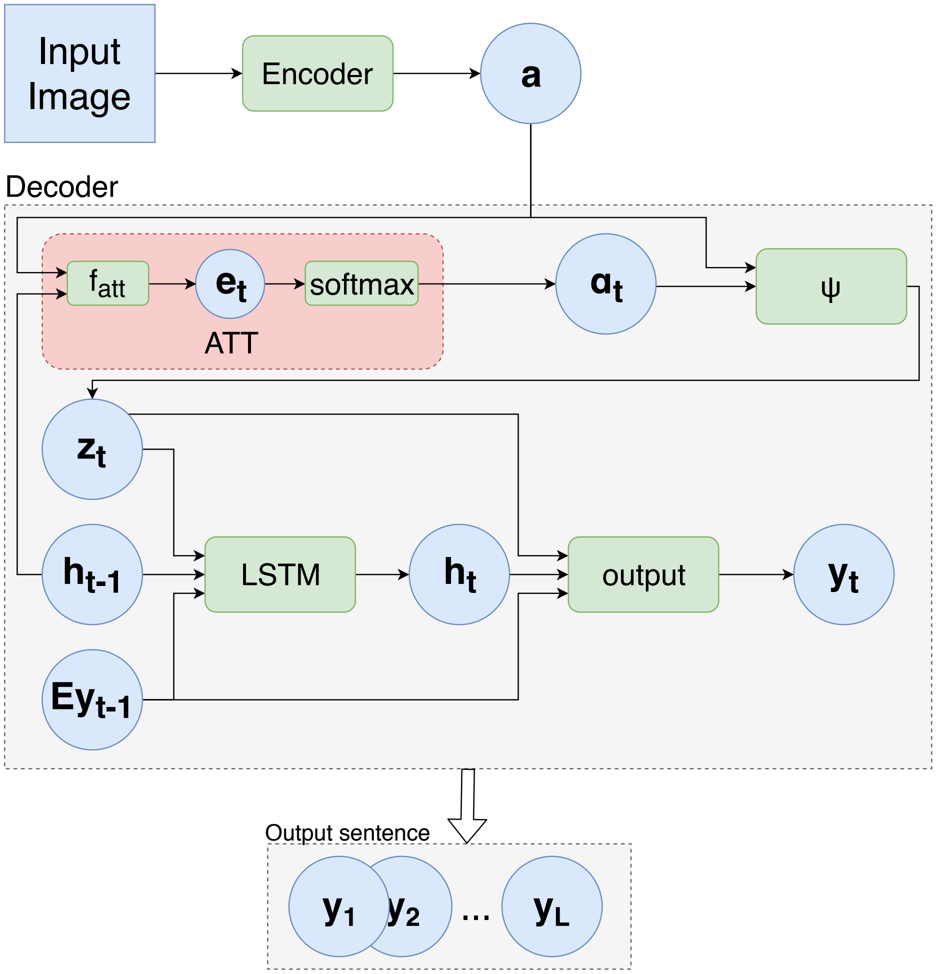

Figure 6: Model proposed for image captioning in [4] with attentional module.

Figure 5.1 illustrates the model proposed in the work. The steps that calculate the attention to each encoding vector can be "encapsulated" as an attentional module under our modeling: a, the input image encoding, is the *focus target input* τ_t, h_{t-1}, the hidden state of the model LSTM, is the *outer state input* o_t and α_t, the weights given to each encoding vector, is the *focus output*. In this case, $A = [0, 1]$. Note that, in this case, the *internal state* is empty.

5.2 Adaptive Computation Time

The work [2] proposes an RNN that can perform a variable number of computation "sub-steps" for each time step t'. The main idea is to calculate an amount $0 \leq p_{t',t} \leq 1$ to be "spend" for each computation sub-step t up until the moment the total spent reaches the "budget" of 1 (in which moment the computation is halted). The final value $y_{t'}$ is computed as an weighted average of the intermediate $y_{t',t}$ values and the weights are the values $p_{t',t}$.

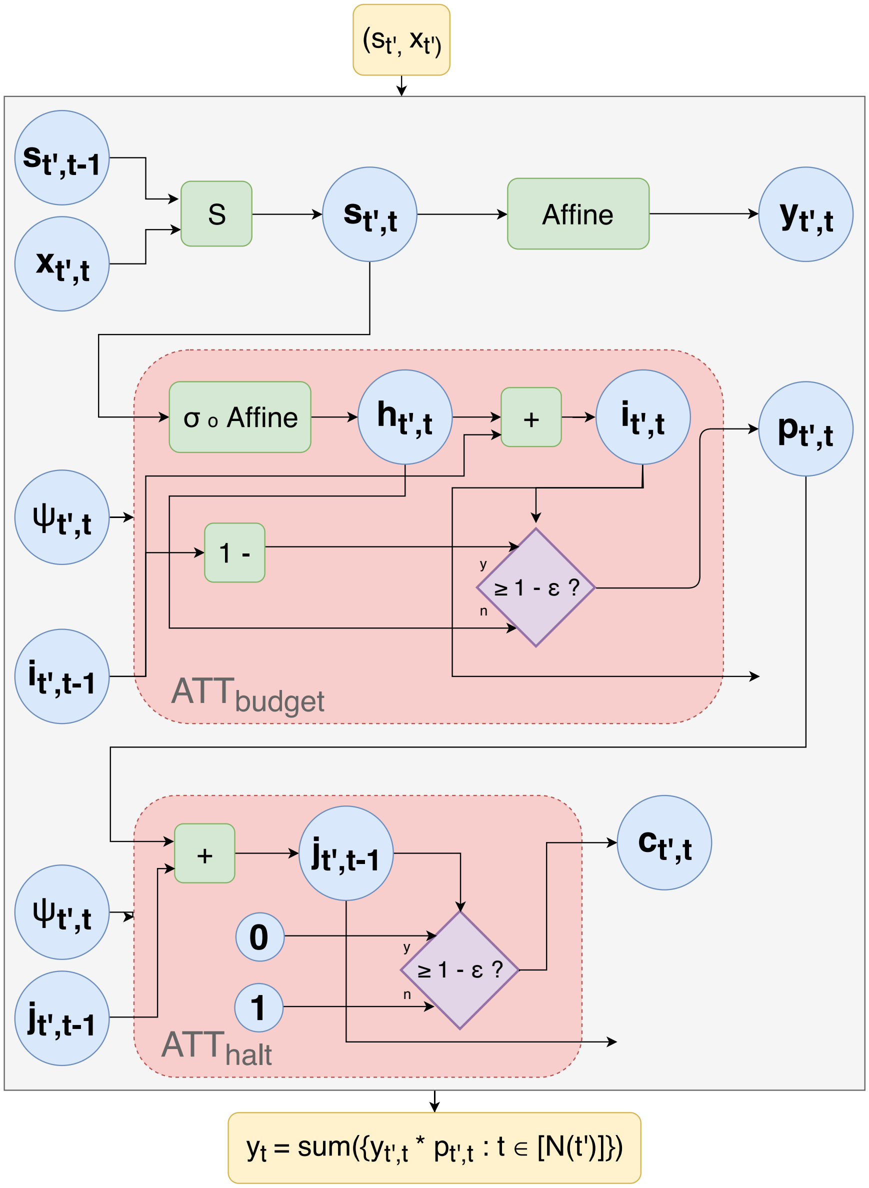

Figure 7: Model proposed for image captioning in [2] with attentional module.

Figure 5.2 illustrates the model proposed in the work. The proposed model can be thought as having two attention modules:

- ATT_{budget}, which computes the value $0 \leq p_{t',t} \leq 1$ to be spent at a given sub-step. In this analogy, $s_{t',t}$ – the state of the RNN cell – is the *outer state* o_t; ψ_t – a dummy element representing the current computation sub-step – is the *target* τ_t; and $i_{t',t}$ is the *inner state*. The *focus output* $p_{t',t}$, besides representing values to be consumed from the budget, can be thought of as an importance weight for the final output y_t, since the produced values are used to computed the weighted average.

- ATT_{halt}, which computes the value $c_{t',t} \in \{0, 1\}$, which is 1 if the cell should continue further sub-steps and 0 otherwise. In this analogy, $p_{t',t}$ is the *outer state* o_t; ψ_t – a dummy element representing the current computation sub-step – is the *target* τ_t; and $j_{t',t}$ is the *inner state*.

It is interesting to note that an effect that emerges from these two blocks is that *the model can allocate resources to processes* both by *choosing the data to use* (in the computation of each y_t weighted by a focus output) and *choosing the amount of computation time to use*.

5.3 Recurrent Attention Model of Visual Attention

The work [3] proposes a general recurrent model that uses visual attention at each step by selecting a "retina-like" representation of a portion of the input image to carry out further computations. At each time step t, the model uses the selected location l_{t-1} to extract a retina-like representation from input image. An arbitrary action a_t can be executed to possibly alter the environment.

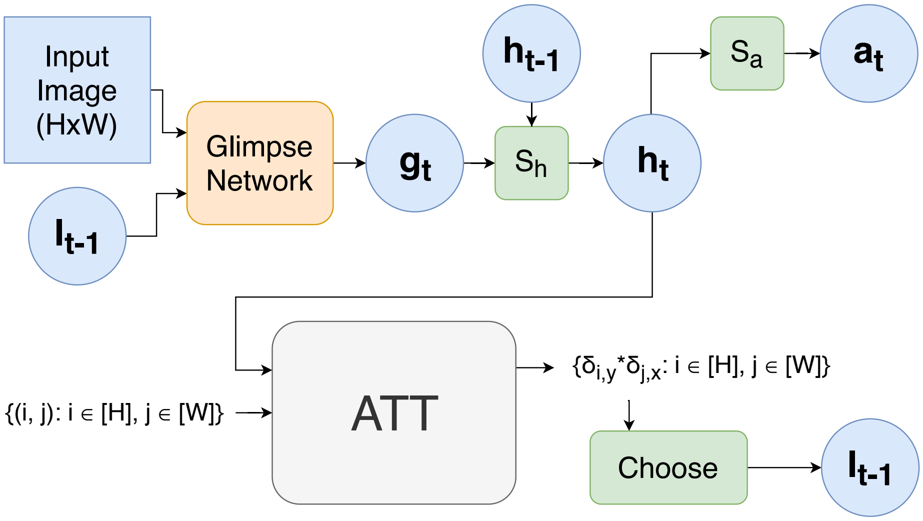

Figure 8: General recurrent architecture proposed in [3] with attentional module.

Figure 5.3 illustrates the model proposed in the work. In this representation, the hidden state of the RNN h_t is the *outer state* input o_t; The set of possible pixel coordinates $\{(i, j) : i \in [H], j \in [W]\}$ (with H, W as the height, width of the image) is the *focus targets* input τ_t; and the set $\{\delta_{i,y}\delta_{j,x} : i \in [H], j \in [W]\}$ is the *focus output*. Note that only the element $\delta_{i,y}\delta_{j,x}$ – which is respective to the chosen pixel coordinates (x, y) is equal to 1.

References

[1] KyungHyun Cho, Aaron C. Courville, and Yoshua Bengio. "Describing Multimedia Content using Attention-based Encoder-Decoder Networks". In: *CoRR* abs/1507.01053 (2015). arXiv: `1507.01053`. URL: `http://arxiv.org/abs/1507.01053`.

[2] Alex Graves. "Adaptive Computation Time for Recurrent Neural Networks". en. In: *arXiv:1603.08983 [cs]* (Mar. 2016). arXiv: 1603.08983. URL: `http://arxiv.org/abs/1603.08983` (visited on 09/11/2018).

[3] Volodymyr Mnih et al. "Recurrent Models of Visual Attention". In: *arXiv:1406.6247 [cs, stat]* (June 24, 2014). arXiv: `1406.6247`. URL: `http://arxiv.org/abs/1406.6247` (visited on 09/11/2018).

[4] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *CoRR* (2015). URL: `http://arxiv.org/abs/1502.03044`.