Universidade Estadual de Campinas
Instituto de Computação

Attention as a leverage for Deep Learning

Erik de Godoy Perillo

Supervisor: Prof.a. Dr.a. Esther Luna Colombini

Research Project
23 de setembro de 2018

Abstract

Attention is fundamental for intelligent beings. It is necessary for filtering the significant volumes of stimuli we constantly receive and for applying the adequate mental resources to perform tasks. Deep Learning is currently broadly applied to Artificial Intelligence. The use of Attention in Deep Learning has been increasingly frequent, resulting many times in better results. In this context, this work proposes the study and elaboration of approaches to use Attention in Deep Learning for more power and efficiency to solve problems in Artificial Intelligence. We aim at obtaining a framework generically applicable in broad problem classes such as Computer Vision, Natural Language Processing, Program Composition and others.

Introduction

We continually receive high volumes of multimodal stimuli from both external sources – such as visual, auditive signals – and internal sources – proprioception, memories et cetera. It would be very inefficient or even impossible to process all the information with the same intensity once a significant portion of it is irrelevant for the task executed at the moment and considering that we have limited cognitive capacity. When we read, our vision does not focus on all words equally, but instead on a small subset of the text at a time. When we are addressing a given subject (in a ''train of thought"), it tends to mediate the focus in the memory search process, essentially retrieving memories that are useful whereas many other irrelevant memories are not used. It often happens that something conspicuous – such as a bird abruptly appearing in front of us or a sudden sound – quickly draws our focus, ''stealing" it from what was previously being focused. The abilities to filter and select stimuli that are relevant for a task, to keep the focus for an extended period and to adequately direct mental processes is fundamental to human beings and other sophisticated forms of life. We name this set of abilities ''Attention" [11].

Attention can potentially play an essential role in Artificial Intelligence (AI). The pursue of intelligent machines is an old effort in Computer Science [40] and is still very relevant today due to the potential to radically benefit society. Although there have been significant advancements in the field of AI, it is broadly accepted that machines still cannot perform certain complex tasks nearly as efficiently as humans or some animals and the path to achieving more intelligence is still unclear, with many different proposals [27]. Part of the problem comes from the difficulty to properly define ''intelligence" itself, but surveys of the works on the subject [26] suggest that a reasonably accepted concept is the ability to perform elaborate tasks in complex and dynamic environments in order to achieve a wide variety of goals. From the narrow to the broader aspects of intelligence, the functionalities of Attention are of great importance – and it increases as the level of intelligence considered increases [21].

A considerable amount of advancements in AI in recent years comes from the popularization of Deep Learning (DL) [25]. As we will discuss in the following sections, the

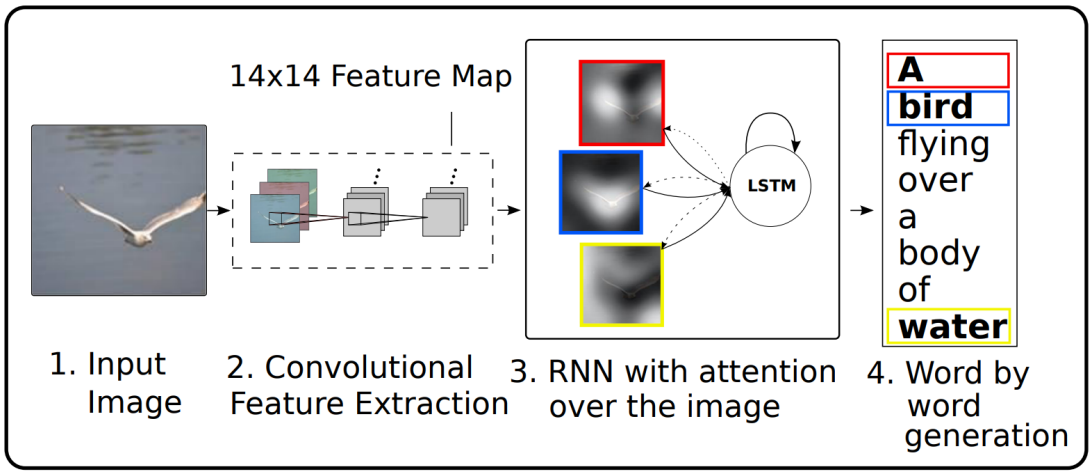

Figura 1.1: Diagram of natural language image description using Attention (from [9]).

technique mostly consists of artificial neural networks architectured in a hierarchical manner. DL showed to be effective in a variety of tasks in Computer Vision [23][20], audio processing [41] and Natural Language Processing (NLP) [42], mainly due to its ability to learn what features should be extracted (rather than relying on hand-crafted features). Along with the transposition from classic models to DL approaches, an increasingly high number of works on the field have been using concepts related to Attention in combination with DL to achieve better results. One example is image captioning (figure 1.1) where the task consists of giving a natural language description of a given image. The work presented in [9] shows that the task benefits from sequentially focusing on different parts of the image in a sequence, through the use of an attentional component in the model. Other examples – which will be discussed in-depth in following sections – include linguistic translation [3], audio recognition [7] and neural computation [19]. These are evidence that concepts of Attention have indeed been useful for the field.

Motivation and Objectives

In spite of the recent adoption of Attention by a variety of Deep Learning models and the significant improvements it has shown, it is conjectured that there are still many other tasks that are still not very well explored. Current works also tend to focus more on the filtering functionality of Attention, but there are other aspects – such as the allocation of mental resources in the course of time – that can be of potential benefit (we further discuss the taxonomy of Attention in following sections). Furthermore, we note that Attention

models currently being used are very specific to each problem in question. Some works propose a higher level of generalization [29], but we believe it is possible to go further. Therefore, the specific objectives of this work are:

- To perform an extensive literature review on the use of Attention in modern Deep Learning;
- To identify theoretical aspects of Attention itself from areas such as psychology and neuroscience;
- To establish general aspects of Attention to be applied to Deep Learning;
- To identify specific problems in different classes (robotics, vision, natural language, program composition) with improvement potential by the use of Attention;
- To propose and implement one or more solutions based on the findings of the work in order to validate the ideas and evaluate them in an application.

The main contribution of the work proposed is related to the first four items: we wish to *establish a theoretical framework of Attention as a series of components and its applicabilities to Deep Learning.* Recent works show that the effort on establishing more general concepts and frameworks for Deep Learning design have been broadly useful. Examples include the ideas of *Curriculum Learning* [4] and *Generative Adversarial Networks* [17].

Background

Attention

The interest in the concept of Attention exists since a long time ago. Throughout the years, Attention has been studied from various perspectives [11] such as philosophy, psychology, and neurology. There are multiple definitions of the concept. In the next items, we discuss some concrete aspects related to Attention.

A definition

We can define Attention as *the act of applying mental resources to selected stimuli following an allocation policy specific to a particular goal.* This rather broad definition captures well the main concepts related to Attention: in a world with virtually infinite *stimuli* to select from the environment, agents with otherwise *finite processing resources* (but with a variety of options of *mental processes* to perform) must choose what their actions will be (and in which stimuli) in a *correct sequential manner* and in *sensible time.* As mentioned before, other works may define Attention in a different manner that is perhaps even conflicting with ours but these are the terms that we choose our work to be based on – noting that they reasonably capture common concepts of interest by us and other works. [21]

Functionalities of Attention

Attention can be manifested in different manners depending on the goal. The most notable functionalities shown in intelligent beings are:

- **To select stimuli** such as looking at only a relevent portion of an image – to efficiently use resources on relevant information.
- **To sustain focus** on a specific semantic element for a period of time in order to complete a task.
- **To guide processing** in a sequential manner that is relevant for a task.

(a) (b)

Figura 2.1: Example of visual saliency. b) is the saliency map where higher intensity pixels represent regions that are more salient to humans than original image a).

- **To orient resources** to new important stimuli – such as an abrupt noise comming from somewhere – or even in alternating the focus to multiple tasks at the same time.

Bottom-up and Top-down Attention

Focus may emerge in two fundamentally different manners [11] [14]. In bottom-up Attention, the act of focusing is involuntarily started and guided by (usually) external and conspicuous stimuli, such as a shattering glass that tends to make us immediately turn our heads towards where the noise came from. Another example is visual saliency (figure 2.1): a glowing red ball suddenly appearing in your field of vision will probably grab your focus. In top-down Attention, focus is voluntarily guided by cognition and goals. If we are talking to someone in a crowded party, for example, we focus on what the specific person is saying – ignoring other people's words – in order to maintain the conversation.

Soft and Hard Attention

In recent years, there has been an useful distinction between soft and hard Attention [44]. Soft Attention regards defining a continuous distribution of importance across all elements of information for some task. In the example of visual saliency, one can determine a saliency map M to a given image I where each pixel will have a value in $[0, 1]$ regarding its saliency. Hard Attention regards determining a discrete subset of important information elements. Using again the problem of visual saliency as an example, one

might want to determine a specific location (i, j) of the image to be used as center of a small patch of the image that is the most relevant to be further processed.

Deep Learning

Deep Learning (DL) is a trend in modern AI [25]. Although DL started being broadly adopted around 12 years ago, some of its concepts date to much earlier than that [25]: foundations of artificial neural networks were already discussed in the 1950s, backpropagation was introduced in the 1970s and many other key concepts that are popular mostly in the last decade or less were introduced more than 30 years ago. Many fields of AI witnessed a major shift in paradigm in the last years: models applying DL concepts now achieve state-of-the-art results in different problems regarding Computer Vision, audio processing, NLP, neural computation among others [16]. DL used used both in supervised and unsupervised learning [25].

One of the key concepts of DL is that of hierarchy of features [25]: A deep sequence of layers apply non-linear transformations to the data in such a way that many models learn to extract features of hierarchical levels of abstraction. For this reason, DL is also regarded as Representation Learning. This characteristic enables such models to learn latent structure in intrinsically unstructured data such as images, text and audio signals. Another advantage is that of transfer learning: models that are primarily trained for a given task can be used and adapted for another task while using at least part of the representations learned. We discuss some concepts related to DL in following items.

Artificial Neural Networks

Artificial Neural Networks (ANNs) are usually adopted to prediction learning problems by means of learning a non-linear function aproximation. The ideas used in ANNs date to more than 50 years ago [36] and many of them are inspired from observed mechanisms of the human brain. Most of DL models are a variation of one of the families of ANNs that will be briefly discussed here.

One of the most basic examples is that of Multi Layer Perceprons (MLPs). The main caracteristic of this model is the use of hidden layers and neurons are a linear combination of previous layers followed by a non-linear activation. Each layer l_k (with n neurons) is

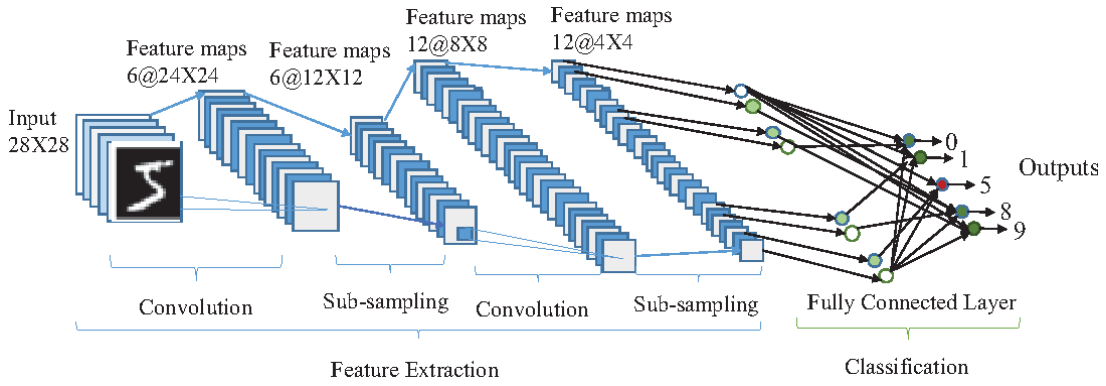

Figura 2.2: Diagram of a convolutional neural network. Learned filters extract features in an increasingly hierarchical manner.

connected to the previous layers l_{k-1} (with m neurons) and the neuron l_k^i, $1 \leq i \leq n$ is given value:

$$l_k^i = h\left(\sum_{j=1}^{m} l_{k-1}^j w_k^j + b_k^j\right)$$

Commonly used activation functions are the sigmoid, hyperbolic tangent and the Rectified Linear Unit (ReLU):

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

ReLU is a currently broadly adopted due to its high efficiency and training speed [30].

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are widely used in Computer Vision tasks such as image classification, localization and semantic segmentation. CNNs use the fact that images tend to have correlated pixels and use convolution filters in an hierarchical manner (figure 2.2) to learn features in increasing abstraction. For a certain layer, the i-th feature map m_i is, given filter weights W_i, bias b_i and nonlinearity function $h(x)$, obtained as:

$$m_i = h\left(W_i * x + b_i\right)$$

with $*$ as the convolution operation.

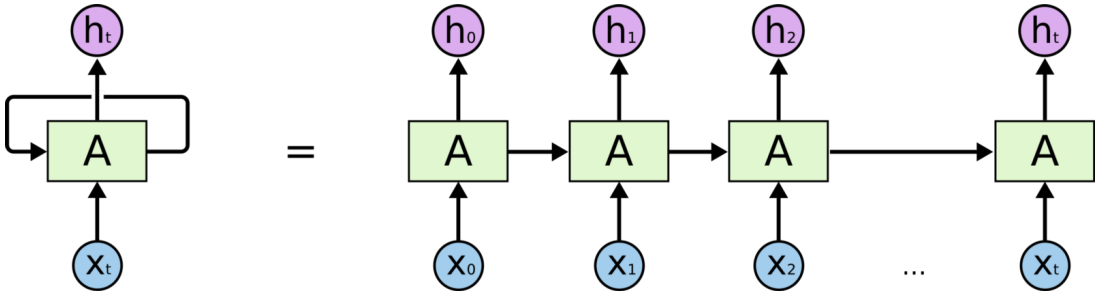

Figura 2.3: Diagram of a recurrent neural network. Time steps map previous outputs and current input to another time step.

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are characterized by a recursive architecture that uses the input of the current step and the output of the previous step to compute the predictions. The hidden state h_t at time step t, given input x_t, weight matrix W, previous state h_{t-1}, hidden-state-to-hidden-state matrix U and non-linearity $f(x)$ is given by:

$$h_t = f\left(Wx_t + Uh_{t-1}\right)$$

These architectures are widely used in NLP tasks such as machine translation [46]. Some variations over the original basic architecture such as LSTMs are also broadly adopted.

Modern architectures

Modern work on Deep Learning propose models of greater complexity than cited above. Some present completely novice architectures, but most of works tend to be based on a combination of basic neural networks architectures. The final results, however, may consist of completely new contributions to new tasks. Some examples will be discussed in chapter 3.

Learning process

The act of learning the appropriate weights of a given model is usually obtained by the minimization of a differentiable loss function that is based on the cost function $L(y, \hat{y})$ that characterizes the error between the true value y and the predicted value $\hat{y}$. Backpropagation [24] plays an important role in DL because it's used to adjust the

weights θ of models that have a differentiable cost function. A typical training process is composed of a forward-propagation step which computes the predictions over a set of input samples and a backpropagation step which computes the loss function and adjusts the weights of the model. In DL, common adjustment methods includes Gradient Descent (GD) [37] or variations which, for a given minibatch, adjusts weights according to:

$$\theta_{i+1} = \theta_i - \alpha \frac{\partial J}{\partial \theta}$$

where α is the learning rate.

Related Work

The topic of integrating Attention concepts into Deep Learning has been increasingly frequent in the community [44]. Augmenting the capabilities of neural network architectures with Attention has shown promising results in problems from a variety of fields in which Deep Learning is currently being applied to, such as Computer Vision, Natural Language Processing and differentiable programming in general. In this section, we address some recent works. We highlight how Attention was used by the authors and how it affected the performance of the proposed models on evaluation tasks.

Attention-based Encoder-Decoder Networks

Encoder-decoder networks are a general framework used generally for mapping from input to outputs that both are of highly-dimensional (often unstructured) data, having being successfully used for tasks such as machine translation [10]. One drawback of such architecture is that the encoded feature vector is of fixed size and structure – regardless of the input – and not necessarily preserves spatial/temporal structure from the input. The work in [8] proposes the usage of an attentional module in between encoder and decoder. The proposed model's encoder produces feature vectors that have a explicit spatio-temporal structure (*context set*) of the input and the attentional module uses a relevance evaluation method to select a subset of the outputs – either by soft Attention or hard Attention. This allows the encoder-decoder for more flexibility to select the components of the input that are of more relevance. The authors implemented and evaluated the method for several applications:

- *Image Caption Generation*: The goal of the task is to provide a natural language description of an input image. The proposed model uses a CNN as encoder and RNN as decoder – with the attentional model in between. The model was ranked third in *MS COCO Captioning Challenge* and provided highly interpretable results regarding the importance of the regions of the image to each component of the sentence (see figure 1.1).

- *Neural Machine Translation*: The authors proposed a RNN architecture augmented

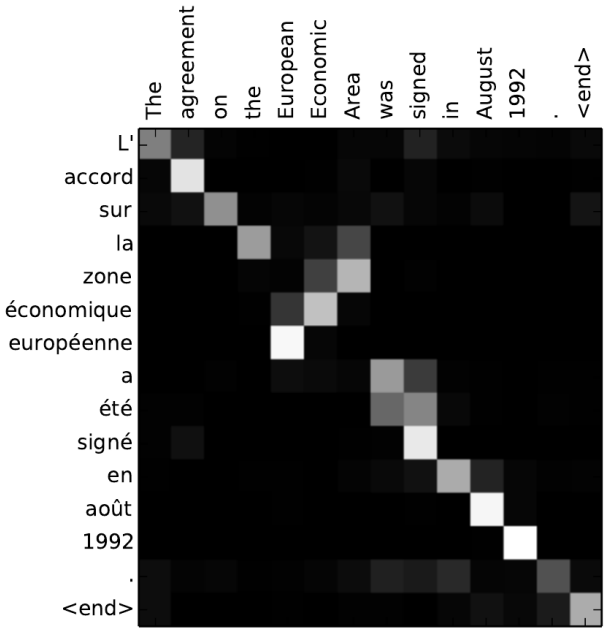

Figura 3.1: Visualization of Attention weights of the neural machine translation model based on Encoder-Decoder networks. Figure from [8].

with the Attention module, which provided relative improvement of roughly 60% when compared to the same model without Attention. The model also performs better than state of the art in some languages. It was also possible to obtain a weight matrix that maps the importance of input to output words since the context set provides structural information of the input (see figure 3.1).

- *Neural Speech Recognition*: The goal of the task is to translate audio to text sentences by using fully neural networks. The proposed model uses RNNs between the Attention module and the model achieved state-of-the art results in the TIMIT corpus [15] and the outputs provide Attention weights from the input signal to produced phonemes.

Overall, the proposed technique – besides achieving state of the art results – produces a semantic mapping from the input space to the output space even when they are of different nature – without explicitly being supervised to produce this mapping.

Adaptive computation time for RNNs

Most of current works use Attention as a mechanism for filtering. The authors of the work in [18] propose a RNN augmented with an Attention module that allows for dynamic

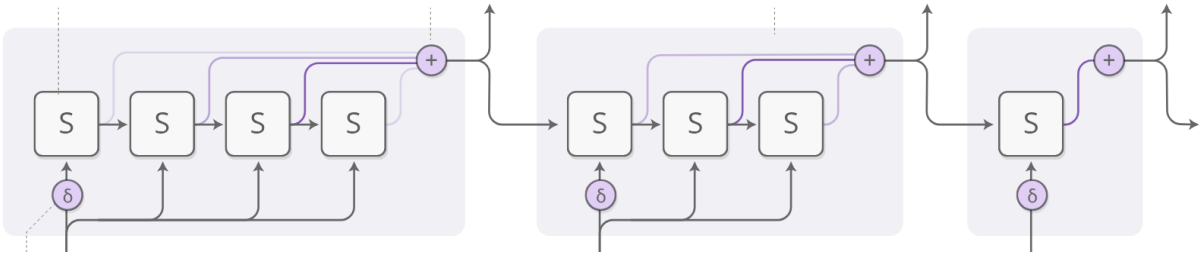

Figura 3.2: Diagram of the adaptive RNN. Each time step can have the amount of computations varied by an Attention distribution. Figure from [32].

inference of number of computation steps for each time step. It uses soft Attention to determine when to stop (see figure 3.2). The ability to allocate computational resources is an important functionality of Attention. The authors show that the mechanism allowed for the model to achieve considerably superior results in tasks such as adding and sorting (when compared to a model without adaptive computation time) because the model was enabled to dynamically perform more or less operations depending on the nature of each step. The authors also tested the model in the problem of character prediction on the Hutter prize Wikipedia dataset [22], in which the model yielded insights into the structure of the input data.

Neural Turing Machines (NTMs)

Neural Turing Machines [19] are one of the first attempts on building models that can learn to formulate programs based on DL architectures with continuous cost functions – and thus trainable via gradient descent. The proposed model is composed of a RNN connected to an external memory bank – which can be read/modified by the use of read/write heads in the model. The Attention mechamism is the component that allows for the read and write operations to be differentiable. On every read/write step, there is an Attention distribution – which is updated each step via content-based and location-based methods – that operates on vectors in the memory locations in a continuous manner. The authors show that the model is able to learn simple algorithms such as sorting and copying sequences.

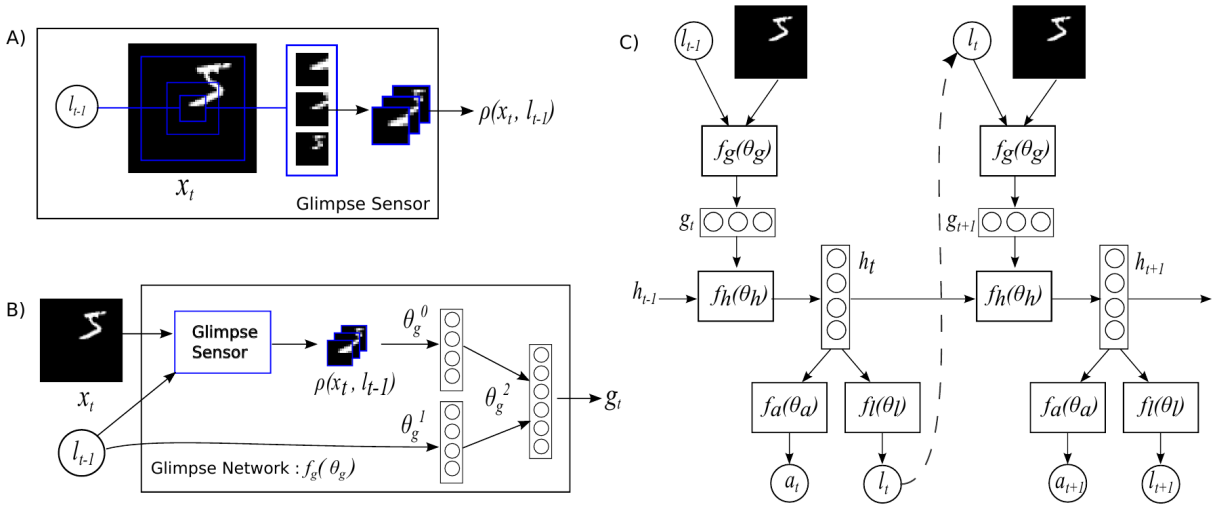

Figura 3.3: Overall architecture of the recurrent attentive model. **A)** is the *glimpse sensor* that extracts patches of different resolutions from image according to the location being attended. **B)** is the *glimpse module*, which combines information from previous attended locations and glimpses to encode a hidden state. **C)** is the RNN architecture augmented with the glimpse module. Figure from [28].

Recurrent Attention Model (RAM)

The work [28] considers a commonly known problem in Computer Vision: it is usually expensive to perform processing on images and widely used current models such as CNNs tend to require computational resources proportional to the number of pixels in the image. The work proposes a *Recurrent Attention Model (RAM)*, a recurrent neural network augmented by an attentional component regarded as *Glimpse Module* that is trained via Reinforcement Learning. The Glimpse Module enables the network to select a point in the image from which it extracts ''glimpses" – patches of the image at different resolutions but with the same dimensions. These glimpes and the selected location are encoded and given as input to produce the new hidden state of the core RNN architecture (see figure 3.3). The dimensionality of the glimpes is much smaller than that of the image and furthermore does not depend on the dimensions of the input image. The authors evaluate the model for classification tasks in the MNIST [13] dataset and variations in which the input images are filled more background pixels (resulting in a larger image) and clutter. The proposed model outperforms a convolutional neural network baseline. Furthermore, the attentional module in the model enables it to perform the same amount of computation regardless of the input size of the image and to focus sequentially only at the relevant parts of the image, which reduces the adversary effect of clutter.

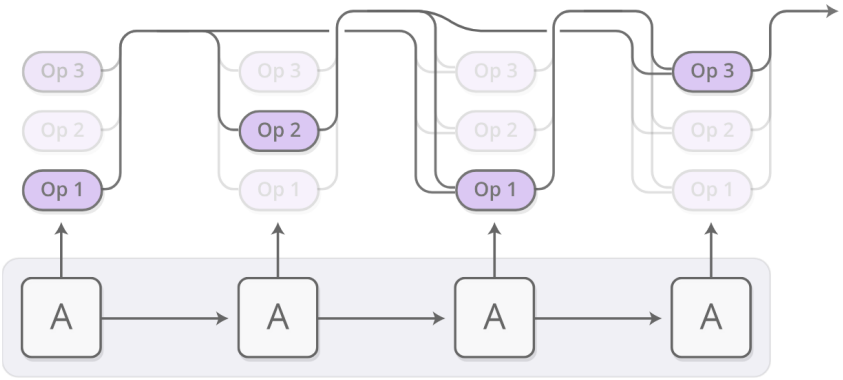

Figura 3.4: Sequence of operations being carried out by the Neural Programmer model. Attention distributes the weights to each operation. Figure from [32].

Neural Programmer

Deep Learning techniques have been useful for perception tasks in the last years, but tasks that involve complex logic and reasoning are still a major challenge. Authors in [31] propose a model that learns to induce programs by composing basic logic operations into more complex ones in sequence (figure 3.4). The model is differentiable and thus trainable via gradient descent because the authors use an Attention distribution at each step to select the operations to be used. The authors of the work evaluate the model on a synthetic table-comprehension dataset. The model achieved nearly perfect accuracy and yielded superior performance compared to LSTMs.

The Transformer architecture

Sequence models such as RNNs and more specifically LSTMs are broadly used for sequence modeling tasks such as machine translation. The work proposed in [43] aims at overcoming some challenges inerent to such sequential models, such as performance and obstacles to apply parallelization to the process. Complications also arise when the content of the inputs are long (such as long sentences in text) and recurrent models present difficulties in establishing relationships among words. The authors present the *Transformer*, an encoder-decoder feed-forward architecture with Attention as a key element. The input sentences are embedded and positional encoding is applied. Then, each layer of the transformer and the decoder employ either *scaled dot-product Attention* or *multi-head Attention*, which allows for contextual mapping and representation of long-relationships.

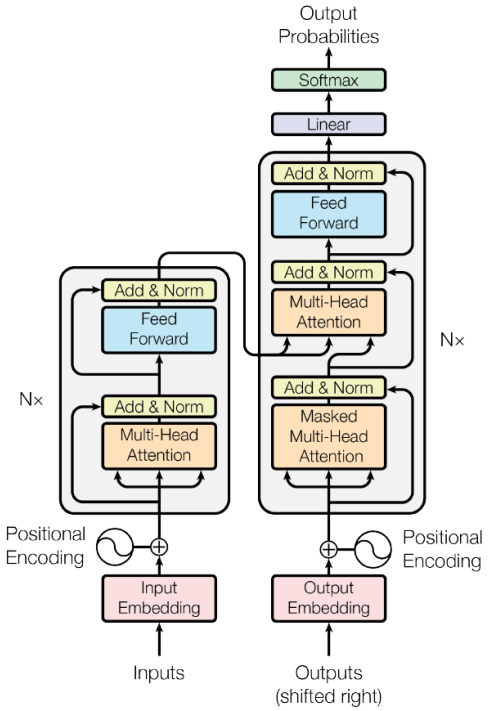

Figura 3.5: The transformer architecture. Figure from [43].

The proposed model achieved state of the art results on *WMT 2014 English-to-German* and *WMT 2014 English-to-French* translation tasks.

Methodology

Activities

The work can be summarized in three main activities (or **phases**):

- **1. Literature Review**: an extensive survey on current uses of Attention in modern Deep Learning.
- **2. Proposal of an Attention framework for Deep Learning**: devision of a set components of Attention currently used/to be used in Deep Learning design from previous activity and further survey.
- **3. Implementation of Attention in Deep Learning**: proposal of one or more models with components of Attention and evaluations on a set of tasks.

The activities to be executed are more specifically described as follows:

- **A1.1 - Theoretical definition of Attention and its components:** Using a variety of previous works [21][11], we establish a theoretical framework of Attention – extending what was discussed in section 2.1 – from which we base all future work. It is worth noting that this theoretical framework is not necessarily the same as the framework we propose to produce specifically for Deep Learning in phase 2.
- **A1.2 - Elaboration of survey:** Exploration of selected work under the point of view of the theoretical framework established in **A1.1**. For each work, we identify the main components of Attention the authors use, the consequences for the performance in the application domain and elaborate a critial evaluation.
- **A1.3 - Survey article writing:** Writing of an article with results of phase 2 to be sent to an appropriate journal.
- **A2.1 - Establishment of Attention components for specific Deep Learning domains:** From the theoretical framework obtained in **A1.1** and the exploration of current uses and results in **A1.2**, we devise sets of useful components of Attention

for specific main problem domains in which Deep Learning is broadly used, such as image classification, text-to-speech, language translation, image segmentation.

- **A2.2 - Establishment of Attention framework for Deep Learning:** From the theoretical framework obtained in **A1.1**, exploration of current uses and results in **A1.2** and results from **A2.1**, we elaborate a set of components of Attention under a single framework to be applied to more general areas of use of Deep Learning, such as Computer Vision, Sequence Processing, Program Composition.

- **A3.1 - Arrangement of experiments:** From the framework obtained in phase 2, we select a set of problem domains (such as text-to-speech), Deep Learning models to use, components of Attention to implement and metrics to evaluate the task. The activity aims at selecting all main devised components from phase 2 in order to evaluate the real consequences of their adoption against what was predicted.

- **A3.2 - Execution of experiments:** We implement and execute the planned experiments following a pre-defined protocol that pays special attention to reproducibility.

- **A3.3 - Evaluation of experimental results:** We evaluate the results using established metrics for each experiment, elaborating discussions that include interesting aspects of the results in general and comparisons between the theoretical predictions and the concrete outcomes.

- **A3.4 - Experiments article writing:** Writing of an article with results of phase 3 to be sent to an appropriate conference.

Other activities to be done related to the masters program are:

- **A0.1 - Course's requirement fulfillment.**

- **A0.2 - Qualification Exam.**

- **A0.3 - Masters dissertation.**

- **A0.4 - Defense of masters dissertation.**

Schedule

Activity	2018			2019											
	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
A0.1	*	*	*												
A1.1	*														
A1.2		*	*	*	*										
A1.3						*									
A0.2						*									
A2.1							*								
A2.2							*								
A3.1								*							
A3.2								*	*	*	*				
A3.3												*			
A3.4												*			
A0.3												*	*	*	*
A0.4															*

Preliminary Work

In order to perform an initial exploration on applying Attention concepts to Deep Learning in order to approach a problem, we investigate the application of visual saliency to image classification.

Visual saliency

Visual saliency (introduced in section 2.1.3) is a well-documented phenomenon in humans [45][39]. The phenomenon is normally classified as part of the *selective, bottom up* components of Attention [14]. We tend to guide our visual focus first to elements that are conspicuous (in the context of the scene). Figure 2.1 exemplifies the phenomenon. The problem of using computers to predict where humans look consists of, given image I, generate a *saliency map* S with pixels values in $[0, 1]$ whereas higher intensity pixels represent locations that are more likely to be salient to humans. There are datasets with human eye-fixation data [1] that enable training of Deep Learning models to approach the problem.

In a previous undergraduate research work, the authors of this document proposed a convolutional neural network architecture specific for visual saliency prediction [34] (see

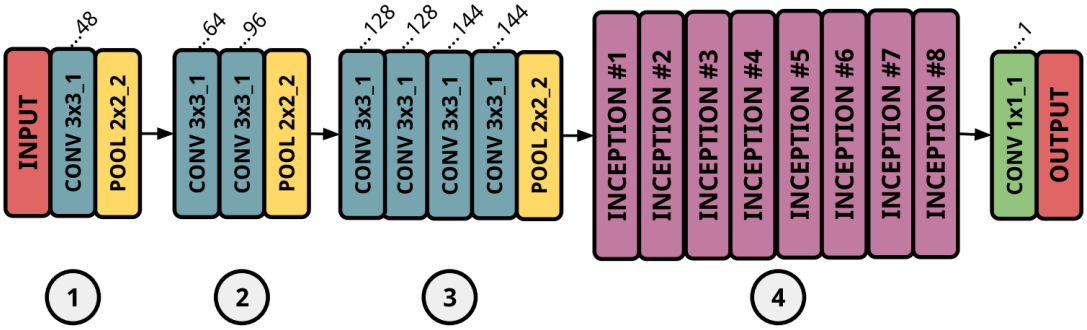

Figura 5.1: Illustration of the convolutional neural network architecture proposed in [34].

figure 5.1). The model was evaluated on the MIT300 benchmark [5] and it achieved performance comparable to the state-of-the art yet with a parameter reduction of 3/4 compared to similar models. The student was awarded ''best undergraduate research project" in WTD2017 [2] for the work.

Image Classification

Image classification is a classical problem in Computer Vision. It consists of, given image I, predicting one or more classes c_i from the set C to which the image belongs. ImageNet [12] consists of a dataset for image classification with of over 14 million images and also an image classification challenge. Part of the recent popularization of Deep Learning comes from the performance gains that Convolutional Neural Networks have yielded in the competition in past years [23][38].

Experiment: Visual Saliency for Image Classification

Many state-of-the art models for image classification support as input images with either fixed dimensions or a rather limited range of dimensions. For large images or images with too much clutter, there must be a pre-processing step to adjust the image for the network. A common practice is to resize the image. Another approach is to crop a certain region of the image. In such case, central cropping is usually applied, but it sometimes might discard parts that are relevant for the task. In the proposed experiment, we use visual saliency to guide the selection of the area of the image to be cropped.

Methodology

Visual saliency model

We use a visual saliency convolutional neural network model similar to the proposed in [34], but with a unet-like [35] architecture. This model was proposed in a following work [33] of the authors of this document and consists of even less parameters yet yields similar performance.

Classification model

For classification, we use Inception V3 [38] with pre-trained weights. The model accepts as input RGB images of width and height of, respectively, 299 and 299 pixels.

Dataset

47039 images from the ImageNet 2012 challenge validation set were used. There are images with a variety of dimensions.

Cropping strategies

Three cropping strategies are used:

1. **Central cropping**: a patch of dimensions 299x299 is cropped at the center of the image.

2. **Random cropping**: a random point i, j is selected with uniform probability and the patch is cropped around the point.

3. **Salient cropping**: the visual saliency network is used to compute the salient map S of the image and the point i, j is selected based on the saliency map.

Selecting the point from the saliency map

Given the saliency map S of the image, a normalized version of the squared map is computed:

$$S_n = S^2 / \sum_{i,j} \left(S^2\right)_{i,j}$$

The points i (row) and j (column) of the image I – with dimensions W, H – are then selected using an weighted average of the coordinates and the values of the normalized saliency map at the respective pixel:

$$i_{sal} = \left\lfloor \sum_{i=1}^{H} \sum_{j=1}^{W} i \left(S_n\right)_{i,j} \right\rfloor, j_{sal} = \left\lfloor \sum_{i=1}^{H} \sum_{j=1}^{W} j \left(S_n\right)_{i,j} \right\rfloor$$

Evaluation

For cropping strategy used, the 47039 generated images were used as input for the classifier. Then the average top-1 accuracy is computed for each configuration.

Implementation

The saliency network was implemented in *Tensorflow 1.10.1*. The classifier network was used from *keras 2.2.2* with *Tensorflow* backend. Experiment was carried out on a machine with *Ubuntu 16.04 LTS* and kernel *Linux 4.15.0-30-generic*. Inference was conducted on a GPU *NVIDIA GTX1080*. The code for the experiment is available at `https://github.com/erikperillo/sal-classif`.

Results

Table 5.1 shows the average top-1 error for each configuration on ImageNet validation set. Salient cropping showed a slight (but considerable) improvement in comparison to random cropping. Besides, it is worth noting that a well-known phenomenon named *center bias* [6] in likely to exist in the dataset: most photographs are more likely to be taken with the main object of interest at the center, but it may not be true in other settings such as more cluttered images or photographs not taken by humans. See figures 5.2, 5.3, 5.4 and 5.5 for examples of the saliency maps and cropped patches obtained.

Tabela 5.1: Cropping strategies and top-1 accuracy on ImageNet validation set.

Strategy	top-1 accuracy
Center cropping	0.766
Random cropping	0.729
Salient cropping	**0.769**

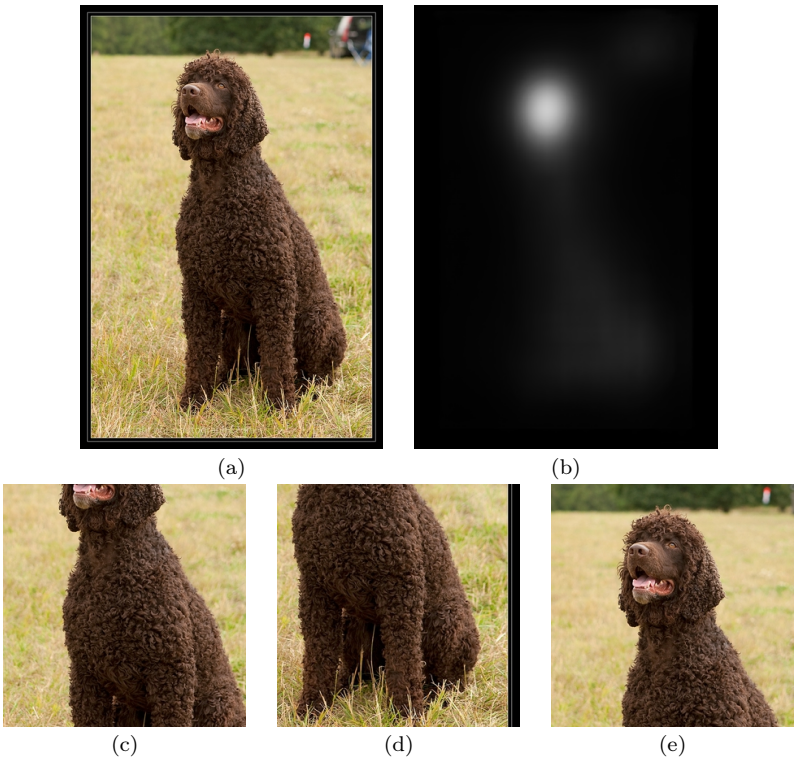

Figura 5.2: Example of image and cropping strategies. (a) is the original image, (b) is the saliency map produced by the network, (c) is the central crop, (d) is the random crop, (e) is the salient crop.

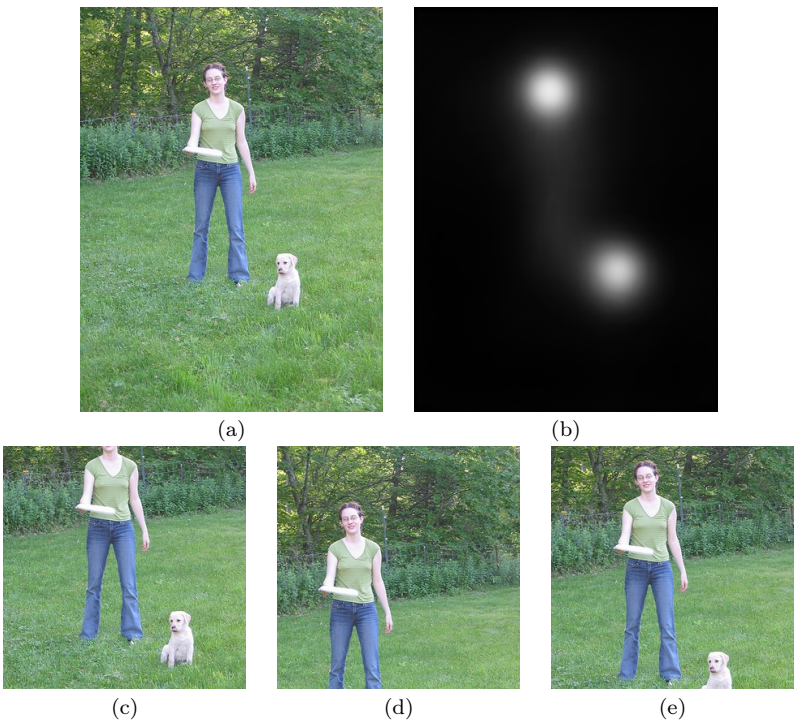

Figura 5.3: Example of image and cropping strategies. (a) is the original image, (b) is the saliency map produced by the network, (c) is the central crop, (d) is the random crop, (e) is the salient crop. Only the salient crop framed the two entities in the image.

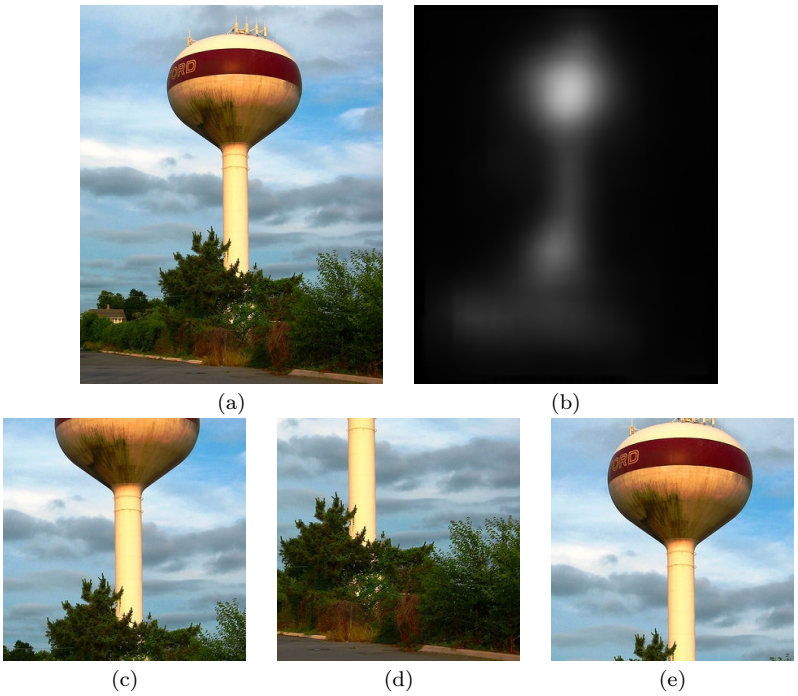

Figura 5.4: Example of image and cropping strategies. (a) is the original image, (b) is the saliency map produced by the network, (c) is the central crop, (d) is the random crop, (e) is the salient crop.

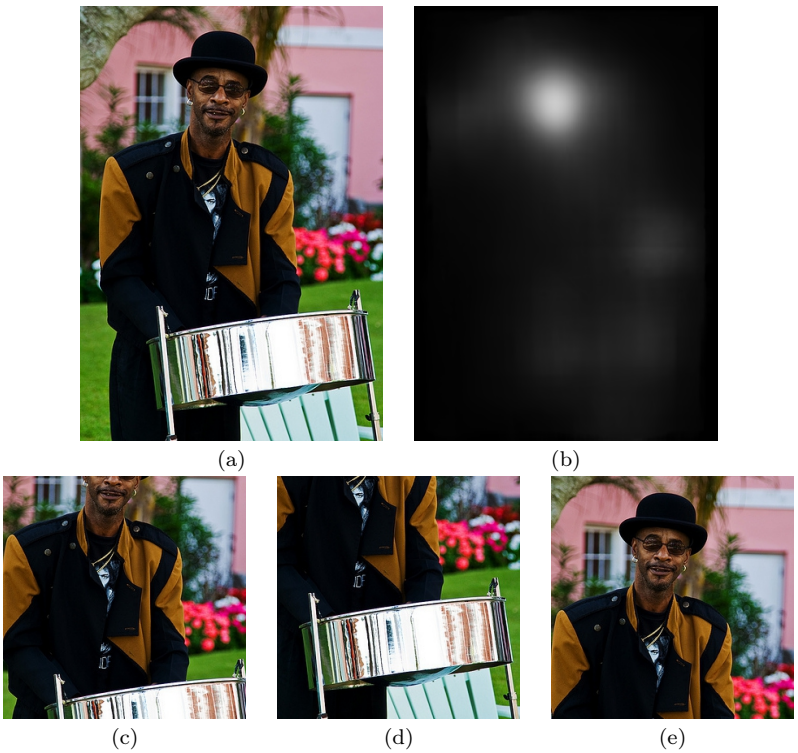

Figura 5.5: Example of image and cropping strategies. (a) is the original image, (b) is the saliency map produced by the network, (c) is the central crop, (d) is the random crop, (e) is the salient crop.

Discussion

The results show that a reasonably simple combination of a concept of attention with a Deep Learning approach was beneficial for the task, with potential of being even more useful in settings in which center bias is less frequent in images.

More work in this direction could include a combination of the visual saliency network with the classification network at an architecture level – for example, by adding one or more convolutional layers that combine the last convolutional layers of the saliency and classification networks and further training.

References

[1] Salicon dataset, 2016. `www.salicon.net`.

[2] *XII Workshop of Thesis, Dissertations and Undergraduate Research Projects*, 2017.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, Montreal, Quebec, Canada, 2009. ACM Press.

[5] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark.

[6] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *arXiv:1604.03605 [cs]*, April 2016. arXiv: 1604.03605.

[7] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell. *CoRR*, abs/1508.01211, 2015.

[8] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder–decoder networks. 17(11):1875–1886.

[9] KyungHyun Cho, Aaron C. Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *CoRR*, abs/1507.01053, 2015.

[10] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*, June 2014. arXiv: 1406.1078.

[11] E.L. Colombini, A. da Silva Simoes, and C.H. Costa Ribeiro. An attentional model for autonomous mobile robots. *IEEE Systems*, (99):1–12, 2016.

[12] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

[13] L. Deng. The MNIST Database of Handwritten Digit Images for Machine Learning Research. *IEEE Signal Processing Magazine*, 29(6):141–142, November 2012.

[14] Simone Frintrop. Vocus: a visual attention system for object detection and goal-directed search. In *IN LECTURE NOTES IN ARTIFICIAL INTELLIGENCE (LNAI)*. Springer, 2005.

[15] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. Darpa timit acoustic phonetic continuous speech corpus cdrom, 1993.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. arXiv: 1406.2661.

[18] Alex Graves. Adaptive Computation Time for Recurrent Neural Networks. *arXiv:1603.08983 [cs]*, March 2016. arXiv: 1603.08983.

[19] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[21] Helgi Helgason. General attention mechanism for artificial intelligence systems. 05 2013.

[22] Marcus Hutter. The human knowledge compression prize. `http://prize.hutter1.net`.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[24] Yann LeCun. *A Theoretical Framework for Back-Propagation*. 1988.

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[26] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *CoRR*, abs/0712.3329, 2007.

[27] Tomas Mikolov, Armand Joulin, and Marco Baroni. A roadmap towards machine intelligence. *CoRR*, abs/1511.08130, 2015.

[28] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention.

[29] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *CoRR*, abs/1406.6247, 2014.

[30] Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[31] Arvind Neelakantan, Quoc V. Le, and Ilya Sutskever. Neural Programmer: Inducing Latent Programs with Gradient Descent. *arXiv:1511.04834 [cs, stat]*, November 2015. arXiv: 1511.04834.

[32] Chris Olah and Shan Carter. Attention and augmented recurrent neural networks. *Distill*, 2016.

[33] Erik Perillo and Esther Colombini. Attention for robotic systems with deep learning, 2018.

[34] Erik Perillo and Esther Colombini. Efficient Visual Attention with Deep Learning. In *IEEE International Conference on Systems, Man, and Cybernetics*, October 2018. To be published.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[36] F. Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, pages 65–386, 1958.

[37] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747 [cs]*, September 2016. arXiv: 1609.04747.

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[39] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognit Psychol*, 1980.

[40] Alan M. Turing. Computing machinery and intelligence. *Mind*, 1950.

[41] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762 [cs]*, June 2017. arXiv: 1706.03762.

[44] Feng Wang and David M. J. Tax. Survey on the attention based RNN model and its applications in computer vision. *arXiv:1601.06823 [cs]*, January 2016. arXiv: 1601.06823.

[45] Joanna M Wolfe, K. Cave, and Steve Franzel. Guided search: an alternative to the feature integration model for visual search. *Journal of experimental psychology. Human perception and performance*, 15 3:419–33, 1989.

[46] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent Trends in Deep Learning Based Natural Language Processing. *arXiv:1708.02709 [cs]*, August 2017. arXiv: 1708.02709.