Thesis Title

by
Student Name

Professor SuperProf, Advisor

A thesis submitted in partial fulfillment
of the requirements for the
Degree of Bachelor of Arts with Honors
in Physics

WILLIAMS COLLEGE
Williamstown, Massachusetts
July 6, 2021

Abstract

Your abstract will summarize your thesis in one or two paragraphs. This brief summary should emphasize methods and results, not introductory material.

Executive Summary

Your executive summary will give a detailed summary of your thesis, hitting the high points and perhaps including a figure or two. This should have all of the important take-home messages; though details will of course be left for the thesis itself, here you should give enough detail for a reader to have a good idea of the content of the full document. Importantly, this summary should be able to stand alone, separate from the rest of the document, so although you will be emphasizing the key results of your work, you will probably also want to include a sentence or two of introduction and context for the work you have done.

Acknowledgments

The acknowledgment section is optional, but most theses will include one. Feel free to thank anyone who contributed to your effort if the mood strikes you. Inside jokes and small pieces of humor are fairly common here ...

Contents

List of Figures

Chapter 1

Introduction

The introduction is one of the most important pieces of your thesis. Here is a place for you to introduce the problem(s) on which you have worked and place them in the larger context of your field. You should aim to ensure that this section is completely understandable to virtually anyone - and certainly anyone with a sophomore-level grasp of physics. Presumably this will include references to the literature.

In addition to setting your work into context, a second good idea for your introduction is to give a short outline for what the rest of your thesis will discuss. This is often done in the closing paragraph(s) of the introduction with sentences like "In the following chapters ..." and "Chapter 2 discusses ..." Tremendous detail is not required in this outline, but rather just a brief road map for the rest of the document.

1.1 A section

The `\section` tag will create a new section within a chapter. Sections will be sequenced with digits following a decimal point in the table of contents, i.e. this is section 1.1.

1.2 Another section

This second section is, obviously, 1.2.

1.2.1 A subsection

Subsections are created using the `\subsection` delineate smaller pieces of your document, and will appear after a second decimal point; this is subsection 1 of section 2 of chapter 1, i.e. 1.2.1.

A subsubsection

Subsubsections are still smaller sections. By default, this is the finest subdivision of a chapter in LaTeX, and they will *not* appear in the table of contents.

1.2.2 A useful command

This is a margin note.

One command I often ask my students to use is `\marginpar`, which can be used to create a margin note. These are super helfpul if there's something to which you need to return later (say, after you've looked up a number), as notes in the margin are really easy to find quickly.

1.3 Some figures

You will surely want to add figures to your thesis to help explain your ideas. There are a number of different ways to include such things, but the most typical way would be to generate the figure in another piece of software (`MATLAB, Mathematica, Adobe Illustrator, ...` and simply include it in your LaTeX code. This will require use of the *figure* environment.[1] See this document's LaTeX code for details . . .

Here, back in the main body of the text, we can create a reference to figure 1.1. This is automatic; the actual numbers are not typed into the code, but rather the `\ref` tag has been used. Always always always use the `\ref` command to reference figures, or invariably at some point you'll move something and all of your references will be incorrect and you'll have to fix them manually.

As an alternative to the ordinary figure environment, you might deem it desirable to tuck a figure in more closely amongst the text. This has a separate environment known as *wrapfig*. Here we will include the same figure as above a second

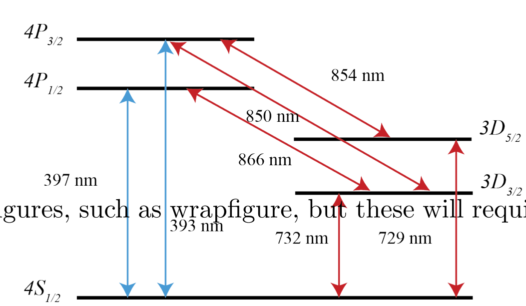


Figure 1.2: A figure included using the wrapfig environment

[1] there are many other possible environments to include figures, such as wrapfigure, but these will require including additional packages . . .

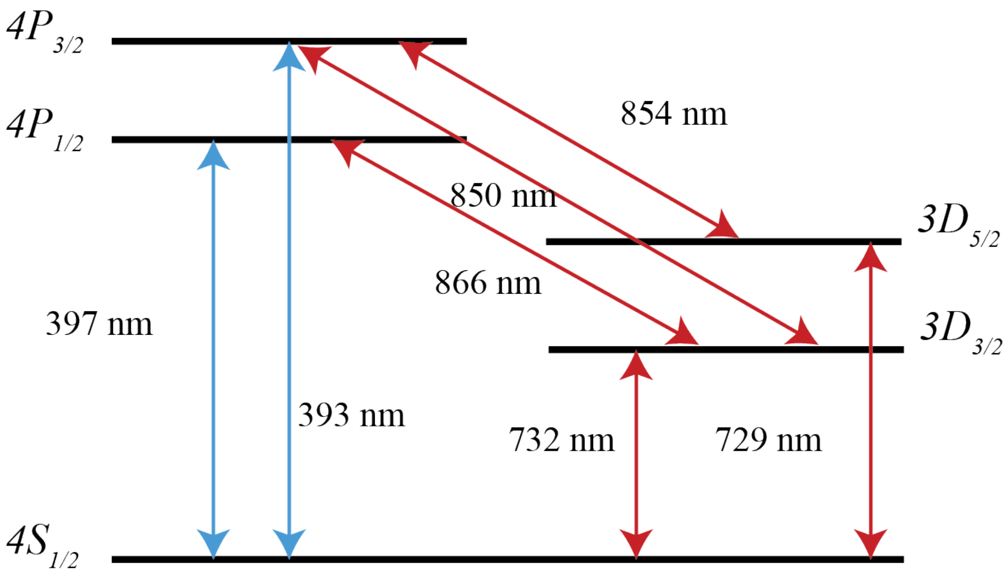


Figure 1.1: Long-form caption that appears in main body of the document

time, but this time using the *wrapfig* environment. This will insert the figure into your document with the text wrapping around the perimeter, rather than offsetting it into its own separate chunk of page, as above. As before, we can use an automated reference to the figure using the `\ref` tag; here we have figure 1.2. Working with the wrapfigure environment sometimes requires a little bit of massaging to ensure that everything lines up properly in your document, but with a small amount of work you will find that you can get the text to box the figure quite nicely.

Here I have added a table, because tables are also useful. This table has nothing to do with the rest of the material in this thesis template, but you should probably only add relevant tables.

Name	SpT	Dist. (pc)	Age (Myr)	3σ M$_{\rm dust}$ limit (M$_\oplus$)	3σ CO(3-2) limit (mJy km s^{-1})	Disk indicator
J0226	L0	46.5	45	0.01	24	Paβ, IR
J0501	M4.5	47.8	42	0.01	23	Hα, IR
J1546	M5	59.2	55	0.01	14	HeI, [OI], Hα, IR
J0446 A/B	M6/M6	82.6/82.2	42	0.027	18	Hα, IR
J0949 A/B	M4/M5	79.2/78.1	45	0.024	17	Hα, IR
LDS 5606 A/B	M5/M5	84/84	30-44	0.027	19	Hα, IR, UV

Chapter 2

A second chapter

Here is a second mock chapter. As far as the LaTeX is concerned, it is in no way different from the introduction excepting that it appears after it in the main .tex file. As before, it can be populated with sections, subsections, figures, etc. as you see fit.

In fact, you will probably write perhaps three to six chapters for your thesis depending on how your work is most effectively organized. Most theses will contain an introduction, at least one 'body' chapter, and some sort of conclusions/future directions chapter. Most theses will also include an appendix or two ...

Chapter 3

Transformers

3.1 De RNN's a Transformers

Las **Redes Neuronales Recurrentes** o **RNN** (por sus siglas en Inglés) datan del año 1986, basadas en el trabajo de Rumelhart [1]. Este tipo de redes están especializadas en el procesamiento de datos que contienen información temporal, mejorando los resultados obtenidos por otros tipos de redes como *Redes FeedForward* o *Redes Convolucionales*.

La principal idea detrás de estos modelos de red es el concepto de *Parameter Sharing*. Con *Parameter Sharing* un modelo un modelo puede generalizar mejor cuando la información que esta contenida en diferentes partes de una secuencia. Así, el modelo no necesita aprender independientemente todas las reglas que forman la secuencias, sino que ahora, la salida para cada elemento en el tiempo esta determinada por la salida del elemento anterior. Resultando en una recurrencia con las mismas reglas de actualización aplicadas a cada elemento en el tiempo. La ecuación 3.1 representa este proceso; $h^{(t)}$ es el estado de la recurrencia aplicada por alguna función f a un elemento $x^{(t)}t$ de la secuencia x en el tiempo, t y θ son los parámetros compartidos.

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta) \tag{3.1}$$

En una *RNN* vista como un *gráfo computacional dirigído y acíclico*, cada nodo representa un estado en la recurrencia y procesa la información de la secuencia x con los mismos parámetros θ en cada paso, observe la figura 3.1.

Existen diversas formas como construir *Redes neuronales Recurrentes*, pueden producir una salida en cada paso de tiempo o tener solo una al final de la recurrencia y también

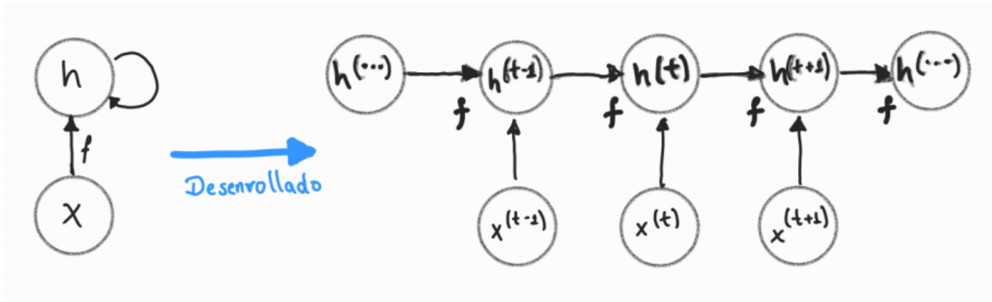
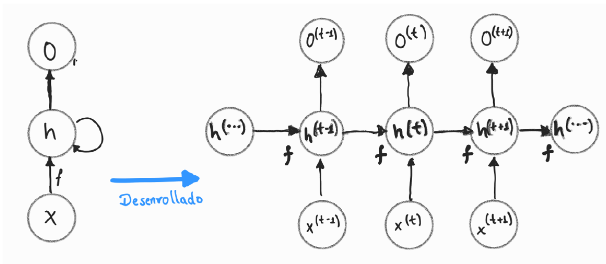


Figure 3.1: Grafo computacional generado por una *RNN* al "desenrrollar" la recurrencia. Usando los parámetros compartidos en cada nodo, con cada elemento $x^{(t)}$ de la secuencia genera un nuevo estado oculto $h^{(t)}$ para retroalimentar nuevamente la entrada del siguiente nodo.

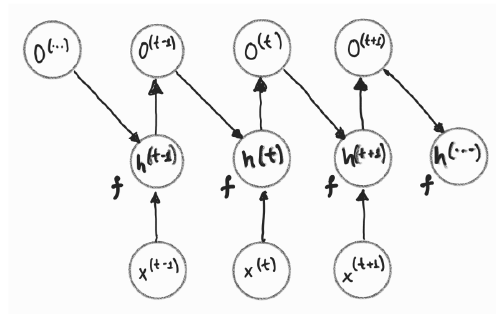
pueden tener conexiones entre unidades ocultas. La manera más común de implementar una *RNN* está ilustrada en la figura 3.2a. En eta figura, cada etapa de la recurrencia es retroalimentada por la activación del estado oculto previo. Así, $h^{(t)}$ contiene información codificada de elementos previos de la secuencia que puede ser usada en el futuro para obtener una salida $O^{(t+1)}$. En la figura 3.2b se cambia la retroalimentación de $h^{(t)}$ por $O^{(t)}$. Nótese que en este caso, la red es entrenada para obtener un valor en específico $O^{(t)}$ lo que provocaría que gran parte de la información de los estados ocultos pasados $h^{(t-1)}, h^{(t-2)}, ...$ no se transmita. En el esquema anterior 3.2a la red es entrenada para decidir que información debe transmitir en el futuro a través de los estados ocultos, en cambio, en la figura 3.2b cada estado esta conectado con el pasado a través de la predicción del paso anterior, perdiendo así gran parte de la información codificada en los estados ocultos, a menos que la salida $O^{(t-1)}$ sea lo suficientemente rica y esté en altas dimensiones.

Por otro lado, la *RNN* representada en la figura 3.2c tiene una sola salida al final de la recurrencia. Al contrario de las anteriores, este tipo de redes pueden ser usadas para resumir información contenida en la secuencia para finalmente predecir un único valor final. El *Análisis de Sentimiento* en textos es una tarea común que puede ser representada con este esquema de red. En la figura 3.2d vemos un modelo de *RNN* entrenado mediante el proceso de Teacher Forcing; durante el entrenamiento la red es retroalimentada con las salidas esperadas del modelo $y^{(t)}$ en el tiempo $t+1$. La ventaja de esta red es que al ser eliminadas las conexiones entre estados ocultos, las funciones de pérdida basadas en comparar la predicción en el tiempo t con el valor objetivo $y^{(t)}$ pueden ser desacopladas. Por tanto, el entrenamiento puede ser paralelizado al calcular el gradiente para cada tiempo t por separado, puesto que ya tenemos el valor ideal para esta salida.

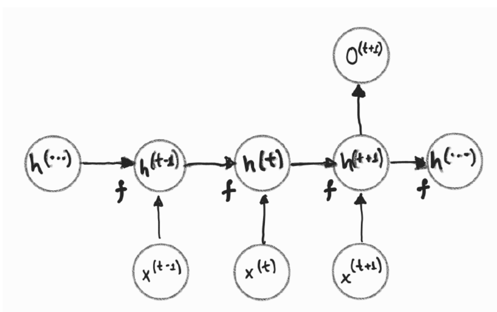
Finalmente, en la figura 3.3 la Red Neuronal Recurrente es modificada para esta vez no procesar una secuencia, sino que, procesa un solo vector en cada paso. El estado oculto previo $h^{(t-1)}$ retroalimenta al siguiente paso t así como la predicción esperada $y^{(t)}$ que también es usada para calcular la función de costo del paso anterior $L^{(t-1)}$. Esta estructura de red puede



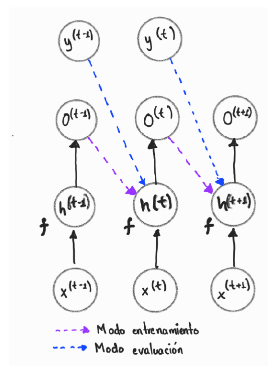
(a) Las activaciones de las capas ocultas $h^{(t)}$ alimentan al nodo siguiente, cada etapa de la recurrencia genera una salida $O^{(t)}$.



(b) Cada nodo es alimentado por las salidas de cada estado oculto $O^{(t)}$.



(c) Al final de la recurrencia solo tiene una salida $O^{(T)}$, puede ser usada para resumir/predecir un valor de una secuencia.



(d) Teacher Forcing. En modo de entrenamiento cada nodo en el tiempo t es retroalimentado por la salida correcta $y^{(t-1)}$, en modo evaluación es retroalimentado por las salidas del modelo $O^{(t-1)}$.

Figure 3.2: Distintos tipos de RNNs.

ser usada en tareas como Image Captioning, en donde la entrada es una imagen y la salida una secuencia de palabras que describen esta misma.

Si bien, los modelos ejemplificados anteriormente son construidos de forma *causal*, es decir, la secuencia es procesada en un solo sentido, estos pueden ser modificados fácilmente para que las secuencias sean procesadas en ambas direcciones, a lo que llamamos **Redes Neuronales Recurrentes Bidireccionales**.

- Celdas de memoria LSTM GRU

- Attention mechanism - Mencionar parallización y backpropragation trougth the time

- Problema de RNN Vanising gradient https://www.iro.umontreal.ca/ lisa/pointeurs/ieeetrnn94.pdf https://arxiv.org/pdf/1211.5063.pdf

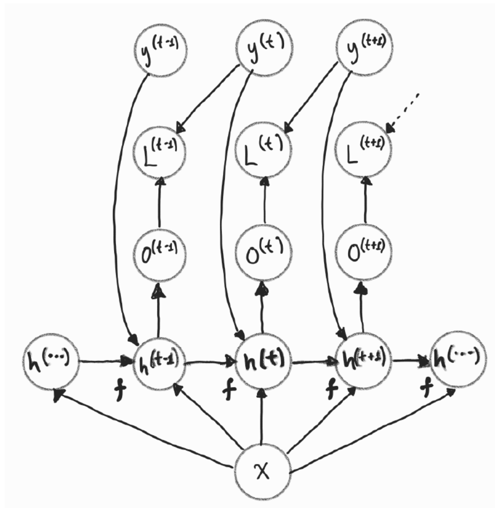


Figure 3.3: Modelo usado para tareas de *Image Captioning*, la entrada es una sola imagen y la red predice una secuencia de palabras que describen dicha imagen La salida esperada $y^{(t)}$ sirve como objetivo para la función de costo del paso anterior y como entrada en cada paso.

3.2 El modelo Transformer

A finales del año 2017 se presentó un nuevo modelo que vino a revolucionar el área de Procesamiento de Lenguaje Natural, El Transformer [2]. Una de sus principales características es la capacidad de procesar la información de alguna secuencia de forma paralela, caso contrario a las Redes Neuronales Recurrentes, donde la información se procesa recurrentemente. Gracias a ello la capacidad de *recuerdo* no se ve afectado por el problema de *El desvanecimiento del Gradiente* cuando se trabaja con secuencias bastante largas.

Appendix A

An appendix

Appendices are a good idea for almost any thesis. Your main thesis body will likely contain perhaps 40-60 pages of text and figures. You may well write a larger document than this, but chances are that some of the information contained therein, while important, does *not* merit a place in the main body of the document. This sort of content - peripheral clarifying details, computer code, information of use to future students but not critical to understanding your work . . . - should be allocated to one or several appendices.

A.1 About the bibliography

What follows this is the bibliography. This has its own separate environment and syntax; check out the comments in the .tex files for details. Worth nothing, though, is that you may find it helpful to use automated bibliography management tools. BibTeX will automatically generate a bibliography from you if you create a database of references. Other software - for example JabRef on a pc - can be used to make managing the reference database easy. Regardless, once you've created a .bib file you can cite it in the body of your thesis using the `\cite` tag. For example, one might wish to cite a reference by Bermudez [3]. If you use BibTeX, you can put the relevant information into a referencedatabase (called bibliography.bib here), and then BibTeX will compile the references into a .bbl file ordered appropriately for your thesis based on when the citations appear in the main document.

Bibliography

[1] R. D. E., H. G. E., and R. J. Williams, “Learning representations by back-propragation errors.,” *Nature*, vol. 323, pp. 533,536, 1986.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gómez, L. Kaiser, and I. Polosukhin, “Attention is all you need.,” *NIPS*, 2017.

[3] A. Bermudez, M. Bruderer, and M. B. Plenio, “Controlling and measuring quantum transport of heat in trapped-ion crystals,” *Phys. Rev. Lett.*, vol. 111, p. 04091, 2013.