Undergraduate Thesis

Your Title Goes Here

Christian Leininger

Examiner: Prof. Dr. Bugs Bunny
Advisers: Terence Hill, Bud Spencer

University of Freiburg
Faculty of Engineering
Department of Computer Science
Chair/Laboratory for Thesis Templates

October 05th, 2016

Writing Period

05. 07. 2016 – 05. 10. 2016

Examiner

Prof. Dr. Bugs Bunny

Second Examiner

Prof. Dr. Wile E. Coyote

Advisers

Terence Hill, Bud Spencer

Bachelor/Master Thesis

Your Title Goes Here

Your Name

Gutachter: Prof. Dr. Bugs Bunny
Betreuer: Terence Hill, Bud Spencer

Albert-Ludwigs-Universität Freiburg
Technische Fakultät
Institut für Informatik
Lehrstuhl für Thesis-Templates

3. April 2017

Bearbeitungszeit

05. 07. 2016 – 05. 10. 2016

Gutachter

Prof. Dr. Bugs Bunny

Zweitgutachter

Prof. Dr. Wile E. Coyote

Betreuer

Terence Hill, Bud Spencer

Declaration

I hereby declare that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work.
I hereby also declare that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date	Signature

Abstract

foo bar

Zusammenfassung

German version is only needed for an undergraduate thesis.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

List of Algorithms

1 Introduction

This is a template for an undergraduate or master's thesis. The first sections are concerned with the template itself. If this is your first thesis, consider reading Section 1.3.

The structure of this thesis is only an example. Discuss with your adviser what structure fits best for your thesis.

1.1 Template Structure

- To compile the document either run the makefile or run your compiler on the file 'thesis_main.tex'. The included makefile requires latexmk which automatically runs bibtex and recompiles your thesis as often as needed. Also it automatically places all output files (aux, bbl, ...) in the folder 'out'. As the pdf also goes in there, the makefile copies the pdf file to the parent folder. There is also a makefile in the chapters folder, to ensure you can also compile from this directory.

- The file 'setup.tex' includes the packages and defines commands. For more details see Section 1.2.

- Each chapter goes into a separate document, the files can be found in the folder chapters.

- The bib folder contains the .bib files, I'd suggest to create multiple bib files for different topics. If you add some or rename the existing ones, don't forget to also change this in thesis_main.tex. You can then cite as usual [**?**, **?**, 1].

- The template is written in a way that eases the switch from scrbook to book class. So if you're not a fan of KOMA you can just replace the documentclass in the main file. The only thing that needs to be changed in setup.tex is the caption styling, see the comments there.

1.2 setup.tex

Edit setup.tex according to your needs. The file contains two sections, one for package includes, and one for defining commands. At the end of the includes and commands there is a section that can safely be removed if you don't need algorithms or tikz. Also don't forget to adapt the pdf hypersetup!!
setup.tex defines:

- some new commands for remembering to do stuff:
  - `\todo{Do this!}`: **(TODO: Do this!)**
  - `\extend{Write more when new results are out!}`:
    (EXTEND: Write more when new results are out!)
  - `\draft{Hacky text!}`: **(DRAFT: Hacky text!)**

- some commands for referencing, 'in `\chapref{chap:introduction}`' produces 'in Chapter 1'
  - `\chapref{}`

  - `\secref{sec:XY}`
  - `\eqref{}`
  - `\figref{}`
  - `\tabref{}`

- the colors of the Uni's corporate design, accessible with
  `{\color{UniX} Colored Text}`
  - UniBlue
  - UniRed
  - UniGrey

- a command for naming matrices `\mat{G}`, $\mathbf{G}$, and naming vectors `\vec{a}`, $\mathbf{a}$. This overwrites the default behavior of having an arrow over vectors, sticking to the naming conventions normal font for scalars, bold-lowercase for vectors, and bold-uppercase for matrices.

- named equations:

```
\begin{align}
    d(a,b) &= d(b,a)\\ \eqname{symmetry}
\end{align}
```

$$d(a,b) = d(b,a) \tag{1}$$

symmetry

1.3 Advice

This section gives some advice how to write a thesis ranging from writing style to formatting. To be sure, ask your advisor about his/her preferences.
For a more complete list we recommend to read Donald Knuth's paper on mathematical writing. (At least the first paragraph). `http://jmlr.csail.mit.edu/reviewing-papers/knuth_mathematical_writing.pdf`

- If you use formulae pay close attention to be consistent throughout the thesis!

- In a thesis you don't write 'In [24] the data is..'. You have more space than in a paper, so write 'AuthorXY et al. prepare the data... [24]'. Also pay attention to the placement: The citation is at the end of the sentence before the full stop with a no-break space. `... last word~\cite{XY}.`

- Pay attention to comma usage, there is a big difference between English and German. '...the fact that bla...' etc.

- Do not write 'don't ', 'can't' etc. Write 'do not', 'can not'.

- If an equation is at the end of a sentence, add a full stop. If it's not the end, add a comma: $a = b + c \quad (1)$,

- Avoid footnotes if possible.

- Use ``'' for citing, not "".

- It's important to look for spelling mistakes in your thesis. There are also tools like aspell that can help you find such mistakes. This is never an excuse not to properly read your thesis again, but it can help. You can find an introduction under `https://git.fachschaft.tf/fachschaft/aspell`.

- If have things like a graph or any other drawings consider using tikz, if you need function graphs or diagrams consider using pgfplots. This has the advantage that the style will be more consistent (same font, formatting options etc.) than when you use some external program.

- Discuss with your advisor whether to use passive voice or not. In most computer science papers passive voice is avoided. It's harder to read, more likely to produce errors, and most of the times less precise. Of course there are situations where the passive voice fits but in scientific papers they are rare. Compare the sentence: 'We created the wheel to solve this.' to 'The wheel was created to solve this', you don't know who did it, making it harder to understand what is your contribution and what is not.

- In tables avoid vertical lines, keep them clean and neat. See 1 for an example. More details can be found in the 'Small Guide to Making Nice Tables' `https://www.inf.ethz.ch/personal/markusp/teaching/guides/guide-tables.pdf`

2 Related Work

Give a brief overview of the work relevant for your thesis.

3 Background

The following work is based on the book Reinforcement Learning: An Introduction[2] from Richard S. Sutton and Andrew G. Barto

3.1 Markov Decision Processes

The Markov Decision Process (MPD) is the mathematical framework of Reinforcement Learning and is defined as a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, >$

- $\mathcal{S}$ is a number of states, s $\in \mathcal{R}^n$
- $\mathcal{A}$ is a of actions, a $\in \mathcal{R}^n$
- $\mathcal{T}$ is the transition probability function
- $\mathcal{R}$ is the reward

Every state in a Markov Decision Process needs satisfy the Markov Property. This means that "The future is independent of the past given the present". The definition is given by an State Reward pair as

$$\mathcal{P}(S_{t+1}, R_{t+1})|S_t, R_t) = \mathcal{P}(S_{t+1}, R_{t+1})|S_{t+1}, R_{t+1}, S_{t+1}, R_{t+1}) \tag{2}$$

Unfortunately for most real-world problem this assumption is violated. This is also the case for the state transition function, which is definite as

$$\mathcal{P}(S_{t+1}|R_{t+1})|s_t, a_t) = \mathcal{P}(S_{t+1}|S_t = s, A_t = a) \tag{3}$$

The Reward is a function of state action pairs

$$\mathcal{P}(S_{t+1}|R_{t+1})|s_t, a_t) = \mathcal{P}(S_{t+1}|S_t = s, A_t = a) \tag{4}$$

The goal of the agent is to find the policy that maximizes the total reward. This policy is a function that maps states to actions in the deterministic case. While a stochastic policy has a certain probability to choose an action in a state. For any MDP there is always at least one optimal policy. In case the transition probability is given, the dynamic programming-based Value Iteration Algorithm is one way to compute this policy. Which is build upon the concept of a Value Function.

3.1.1 Value Function

The value function $V^\pi : \mathcal{S} \to \mathcal{R}$, represents expeceted total reward of a policy in a given state and following the policy It is defined as follows,

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi[R_t|s_t = s] \\ &= \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s] \\ &= \sum_{a\in\mathcal{A}} \pi(s,a) \sum_{s'\in\mathcal{S}} \mathcal{T}^a_{ss'}[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')] \end{aligned} \tag{5}$$

One way to find the optimal policy is the value Iteration Algorithm. In each Iteration the value function is updated by the following equation,

$$V_{i+1}(s) := \max_a \Big\{ \sum_{s',r} P(s', r|s, a)(r + \gamma V_i(s')) \Big\} \tag{6}$$

3.1.2 Q Function

In many real-world problems, the transition function is unknown, and approximate the Value function is not feasible, because it would require evaluating all possible actions in a single state every time step. In this case the Q-value function helps by using the state action pair

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{7}$$

This update equation is used to find an optimal policy in the Q-Learning Algorithm, repesened in Algorithm 1

Algorithm 1 Q-learning

Algorithm parameters: learning rate $\alpha \in (0, 1]$, discount factor $\gamma \in (0, 1]$, exploration rate $\epsilon > 0$

Initialize $Q(s, a)$ randomly except $Q(terminal, \cdot) = 0$

while not converges **do**

  Set initial state s

  while s is not terminal **do**

    With probability ϵ:

      Pick random action a

    otherwise:

      $a = \text{argmax}_a\ Q(s, a)$

    Execute action a and observe reward r and succesor state s'

    $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a^*} Q(s', a^*) - Q(s, a)]$

    $s \leftarrow s'$

  end while

end while

The fact that the policy for interacting with the environment is different from the one to update the Q-value Function makes it to an off-policy Algorithm

3.2 Neural Network

In complex environments, the state space is continuous, which makes it infeasible to use to store all state-action pairs in a table. Advances in machine learning especially in deep learning make it possible to approximate this Q function. A basic element of Deep learning is artificial neural networks which are inspired by the human brain. Neural network have a structure of input layer connected to hidden layers followed by the output layer. A standard Network has linear many hidden layers represented

as nodes. The nodes between different layers are connected by weights. The values of the weights can be trained iteratively by using an optimization technique like stochastic gradient descent and backpropagation. This simple architecture could be seen as matrix multiplications which makes it a linear function. By adding nonlinear activation functions it is able to approximate more complex nonlinear functions.

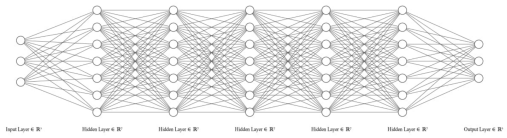


Abbildung 1: Deep Neural Network with several hidden layers

4 Approach

The approach starts with the problem definition and continues with what you have done. Try to give an intuition first and describe everything with words and then be more formal like 'Let g be ...'.

4.1 Problem Definition

Start with a very short motivation why this is important. Then, as stated above, describe the problem with words before getting formal.

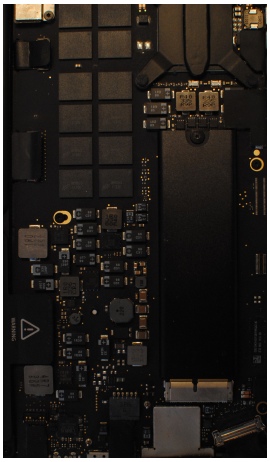
4.2 First Part of the Approach

4.3 N-th Part of the Approach

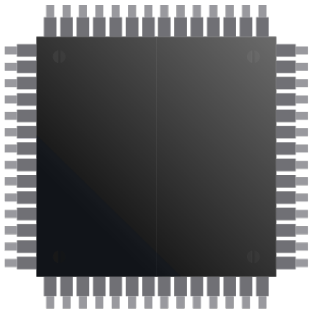
5 Experiments

Type	Accuracy
A	82.47 ± 3.21
B	78.47 ± 2.43
C	84.30 ± 2.35
D	86.81 ± 3.01

Tabelle 1: Table caption. foo bar...



(a) Some cool graphic



(b) Some cool related graphic

Abbildung 2: Caption that appears under the fig Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

6 Conclusion

7 Acknowledgments

First and foremost, I would like to thank...

- advisers
- examiner
- person1 for the dataset
- person2 for the great suggestion
- proofreaders

Literaturverzeichnis

[1] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.," *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.