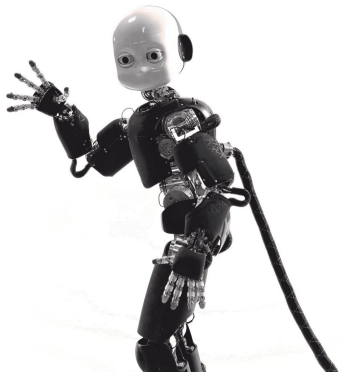
TÉCNICO LISBOA



Anticipation in Human-Robot Cooperation: A recurrent neural network approach for multiple action sequences prediction

Paul Rudolph Schydlo

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. José Alberto Rosado dos Santos Vitor
Prof. Lorenzo Jamone

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. José Alberto Rosado dos Santos Vitor
Members of the Committee: Prof. Luís Manuel Marques Custódio

November 2017

Mar Português

Ó mar salgado, quanto do teu sal
São lágrimas de Portugal!
Por te cruzarmos, quantas mães choraram,
Quantos filhos em vão rezaram!
Quantas noivas ficaram por casar
Para que fosses nosso, ó mar!

Valeu a pena? Tudo vale a pena
Se a alma não é pequena.
Quem quer passar além do Bojador
Tem que passar além da dor.
Deus ao mar o perigo e o abismo deu,
Mas nele é que espelhou o céu.

Fernando Pessoa, in Mensagem

Acknowledgments

First of all I would like to thank my supervisor, Prof. José Santos-Victor. For being there with patience and wisdom despite a full schedule and long nights on the birthday eve. Without your insightful comments and vision, none of this would have been possible.

A special thank you goes to my co-supervisor, Lorenzo Jamone, for the excellent and honest feedback during the process of this thesis. I specially thank you for taking the time to help even after moving to London.

I would also like to extend my gratitude to my colleges and friends at Vislab for making me feel welcome. It was a unique experience to be a part of this group.

I express my gratitude to my partner, Roberta Vittiglio, for being a wonderful human being and always there to support me (even more during the writing of this thesis), with valuable insights and an infinite kindness. Without you I would never have made it this far.

Last but not least, a special thank you to my friends and family who were always beside me during this journey. Thank you for making the time at Técnico the happy memory it is. Without you I wouldn't be the person I am today.

Abstract

Human-robot cooperation in a shared workspace is a key enabler for new developments in advanced manufacturing and assistive applications. This close cooperation can be studied through the joint-action framework.

Joint action defines cooperation scenarios as the interplay between three factors: shared perception of the environment and action goals, action anticipation (the focus of this thesis) and anticipative action.

Anticipating actions is possible through a series of non-verbal cues humans communicate about their intent. Close cooperation require robots that can understand human non-verbal cues. While the problem of understanding and predicting action from non-verbal cues has been approached from different angles, they normally assume limiting Markovian assumptions, recent approaches based on recurrent neural networks without Markovian priors have led to encouraging results in the human action prediction problem both in continuous and discrete spaces.

Our approach extends the research in this direction. More specifically, our contributions address two shortcomings of existing literature: 1) predicting multiple and 2) variable-length action sequences. This is achieved by introducing novel neural network topology in the action prediction problem. Here we show the ability to train the model on a action prediction dataset and explore the influence of the model's parameters. The final model anticipates human action, mitigating complexity through a pruning strategy.

We demonstrate the importance of predicting multiple action sequences as a means of estimating the stochastic reward in a human robot cooperation scenario extending the state of the art in directions that are key to enable human-robot cooperation involving non-verbal communication.

Keywords

Joint action, human-robot cooperation, action anticipation, recurrent neural networks

Resumo

Cooperação humano-robô em espaço partilhado é um elemento chave para novos desenvolvimentos em aplicações avançadas de fabricação e de assistência. A cooperação pode ser estudada pela teoria de acção conjunta que define a cooperação como interacção de três factores: percepção do contexto, antecipação da acção (o foco desta tese) e acção anticipativa.

Anticipar o movimento do outro é possível através de pistas não verbais. A cooperação próxima exige robôs que possam entender pistas humanas não-verbais. Embora o problema de prever a acção a partir de pistas não-verbais tenha sido abordado de diferentes ângulos, as soluções normalmente assumem suposições Markovianas limitantes, abordagens recentes baseadas em redes neuronais sem suposições Markovianas demonstraram resultados encorajadores no problema de predição da ação humana.

Esta dissertação estende o estado da arte nesta direção. Mais especificamente, as contribuições abordam duas falhas da literatura existente: previsão de sequências de acção 1) múltiplas e 2) de comprimento variável, através da introdução de uma nova topologia da rede neural no problema da predição de acção. Demonstramos o treino do modelo num conjunto de dados de predição de ação e a influência dos parâmetros do modelo. O modelo antecipa com sucesso a ação humana e mitiga problemas de complexidade através de uma estratégia de prunning.

Demonstramos a importância de prever sequências de acção como meio de estimar a recompensa estocástica num cenário de cooperação, estendendo o estado da arte em direções que são fundamentais para a cooperação humano-robô envolvendo comunicação não-verbal.

Palavras Chave

Acção conjunta, cooperação humano-robô, antecipação de acção, redes neuronais recorrentes

Contents

List of Figures

List of Tables

List of Acronyms

CRF Conditional Random Field

Dec-MDP Decentralized Markov Decision Process

HMM Hidden Markov Model

LSTM Long Short Term Memory

RNN Recurrent Neural Network

ToBY Theory of Body Mechanism

ToM Theory of Mind

ToMM1 Theory of Mind Module I

ToMM2 Theory of Mind Module II

Nomenclature

θ	Model parameters
θ^*	Optimal model parameters
X	Array of feature vector sequence samples
x_i	Position i in a given feature vector sequence
X_k	Feature vector sequence sample k
Y	Array of reference distribution sequence samples
y_i	Position i in a given reference distribution sequence
Y_k	Reference distribution sequence sample k
$y_{i,a}$	Probability of action a in distribution at position i of the reference distribution sequence.
R	Joint reward function
S	World state
T	Transition probabilities between states over joint actions

1

Introduction

Contents

1.1 Motivation

In a world with a growing number of autonomous systems and moving towards the coexistence and cooperation between humans and sophisticated robots, it is crucial to enable artificial systems to understand and predict human behaviour. The ability to predict how the human will behave in the near future finds applications in areas such as human robot-cooperation [3, 4], auto-mobile safety [5], elderly care [6], among many others [7].

In addition to the use of speech for communicating and coordinating their next actions, humans rely extensively on non-verbal cues for action and movement prediction [8]. Situations where fast cooperation is essential, for example cooperative assembly, benefit greatly by the understanding of subtle non-verbal cues [4] about the human intention and future action as they are an important information exchange channel and enable natural human-robot interfaces. In these fast moving scenarios it is not enough to merely recognize the current action. Instead, to guarantee a seamless cooperation, predicting human actions and anticipating intent is important. [9].

Figure 1.1: Cooperation requires the understanding of subtle non-verbal cues.

In summary, understanding and predicting human behaviour is an important milestone for enabling key usage scenarios for close human robot cooperation such as:

1. Enable safe, close-proximity, human-robot collaborative manufacturing [10].
2. Guarantee the auto mobile safety of both self driving car passengers and pedestrians surrounding the vehicle [11].
3. Assistive robots which understand intent and act naturally [12].

1.2 Topic Overview

This thesis addresses the problem of modelling and anticipating human actions during a cooperation setting using a recurrent neural network approach. To meet this goal we need to first understand topics related to social cognition and action anticipation.

While the concepts are expanded further in later chapters, this section gives a brief introduction to the context of the thesis.

1.2.1 Cooperation

Human cooperation falls under the broader field of social cognition which relates to how we perceive the surrounding agents as animate and goal driven beings. Inside of social cognition cooperation is the sub-field which studies how humans cooperate and interact to complete a shared goal [13]. This joint human-robot action will be the focus of this thesis.

Cooperation, according to Sebanz [9], can be studied as the interplay between three factors: shared perception of the environment and action goals, anticipating the actions of the other and acting according to the other agent's goal directed anticipated action [9]. For the purpose of this thesis we will focus on the second factor, anticipating other agent's actions.

Human action can either be anticipated through explicit clues like verbal communication of the next action: "I will eat the cake" or through implicit non-verbal clues such as observing the other agent looking at the action goal, e.g. looking at a cake might convey interest in eating the cake.

The first chapter, Cooperation, is focused on expanding on these notions showing how humans interact with the world through the lens of animate and inanimate entities, introducing the reasoning behind the separation of concerns between the explicit prediction of human action and the mechanistic interaction with the world [14].

So far, we looked at how humans can interact with the world through the lens of social cognition, that is, recognizing the other as a goal driven being [14]. Here we gave special attention to the joint action phenomena which underlies cooperation [13].

We introduced the main factors which influence a cooperation scenario, more specifically, the perception of shared workspace and knowledge of the action goal, understanding and anticipating the action's of the cooperation partner and acting taking the anticipated action of the other into account while maximizing the completion of a joint goal [9].

We can now look at the panorama of human-robot cooperation implementations with this general understanding of the underlying mechanisms as studied by psychologists and cognitive scientists. While later chapter go more in depth on specific implementations, here we give a general overview of the main components involved in a human robot cooperation scenario.

The first component, shared representation, is the understanding of the joint action goal. This representation , for the sake of example, can be a singular human intent or a shared plan with clear boundaries between the agents. The second component, action anticipation, goes hand in hand with the first: understanding the context it is possible to assign a probability distribution to the human's possible future actions. The third component, anticipative action, takes the information from the two previous components and decides on the best action to take in order to accomplish the joint goal. This part is generally implemented as a fixed set of actions as a function of the human intent [9].

While the focus of this thesis is the action anticipation component, the important point to retain is the importance to dynamically predict and adapt to the partners motions, instead of following a fixed action plan, for a fast and close cooperation.

(a) (b)

Figure 1.2: Joint action in a cooperation scenario: a) Humans naturally react to one another through non-verbal clues b) The gaze and movement of the human partner communicates intent. Source: Project SIMERON

1.2.2 Prediction models

The previous section hinted at the importance action anticipation plays as a core module in a human-robot cooperation implementation. This action anticipation is possible through a series of non-verbal cues humans communicate about their intent and action goals [15, 16].

The problem of understanding and predicting action from non-verbal cues has been approached from different angles, both from a generative and discriminative perspective which have shown excellent results. Nevertheless, these implementations normally assume limiting Markovian assumptions [17, 18], that is, the probability distribution over future actions is defined as a function of the non-verbal cues in the previous time steps, without taking into account potentially informative long term dependencies.

More recently approaches based on recurrent neural networks have led to encouraging results in the human action prediction problem [5, 19]. Recurrent neural networks are specially interesting since they are able to avoid the Markovian prior by keeping an internal state of the human intent, here the action prediction does not depend on the adjacent movements but through the internal state depends on a long history of past movement. The thesis expands the literature in this direction.

While there is a diversity of models that have been applied to the action anticipation problem, they are generally concerned with the so called *intent recognition* problem where the recognition of the action goal and anticipation of future actions is separated [18, 20–23]. The models capture action goals, e.g. recognizing the human intent to drink, but not patterns in human movement, that is, correlations between consecutive actions, such as understanding that grabbing a cup might be correlated with taking the cup close to the mouth and drinking right afterwards without codifying these goals explicitly into the model as action plans.

1.2.3 Overview

This section gave a general overview of the concepts which will be discussed in more detail during this thesis. We saw how humans separate social and mechanistic cognition, we looked at how social cognition relates to perceiving the other as goal directed agents and how this relates to action

anticipation. In the end we reviewed some shortcomings inherent in the existing action anticipation literature.

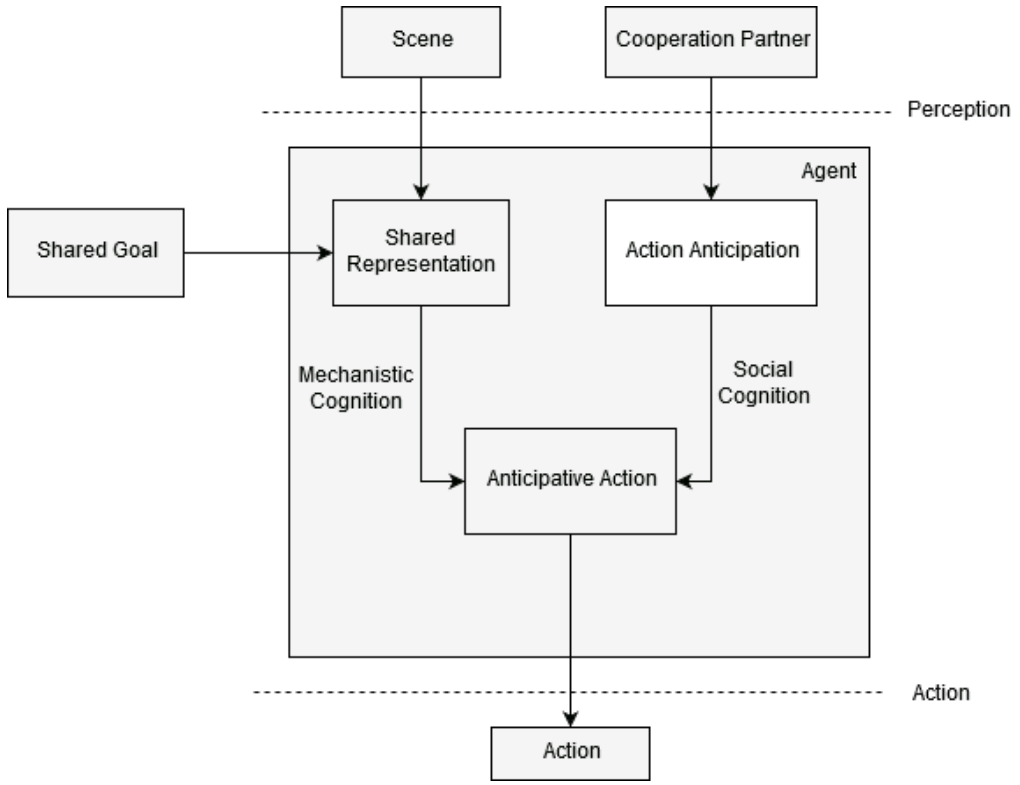


Figure 1.3: Overview of the proposed architecture. The focus of this thesis is the action anticipation module.

The diagram (fig. 1.3) seeks to clarify how all the components interact. Here we can see the clear distinction between the more mechanistic reasoning about the scene and shared goal which integrates with the reasoning about other goal oriented agent's intent and anticipated actions. The final action is obtained by choosing an action which maximizes the shared goal completion taking into account the anticipated actions, the shared plan and the current configuration of the scene.

The focus of this thesis is the action anticipation module, how we can infer future action sequences from the human movement.

1.3 Objectives

While the field of action prediction in discrete space has had a rapid evolution in the last couple of years, there are two shortcomings related to action prediction in the literature that this thesis addresses.

i) Predicting a fixed versus a variable number of time steps into the future. While models like the one introduced in [19] have a remarkable ability to condense contextual past information their scope is limited to fixed step ahead prediction length, that is, instead of predicting a sequence of future actions they predict the distribution only over the next action. This thesis extends recurrent neural network models in a classification setting with variable length action sequence prediction.

ii) Single future action sequence versus multiple future action sequences. While models like the one introduced in [24] are able to effectively use recurrent models to predict a variable number of steps into the future their scope is limited to sampling a single future action sequences. This thesis explores the prediction of *multiple* future action sequence.

1.3.1 Contributions

The main contributions of this thesis are the following:

- Extending recurrent neural network fixed time step action prediction with **variable length action prediction**, instead of predicting a single time step into the future the model is able to predict multiple time steps into the future.
- Introducing the simultaneous prediction of **multiple future action sequences**, instead of greedily predicting one action sequence the model prunes the search space and returns a subset of the most probable future action sequences.
- Formalizing a social cognition based cooperation scenario as a Markov decision process and demonstrating the importance of the two previous properties for **predicting the stochastic future reward** in a human-robot cooperation setting.

1.3.2 Outline

The thesis is organized in four main parts which together give an overview from the more broad concepts of social cognition, to the specifics of cooperation, the importance of action anticipation for the natural interaction in a cooperation scenario, the implementation details of the action anticipation model and ends with a quantitative and qualitative analysis of the model's performance metrics and parameters.

i) Chapters 2:*Social Cognition* and 3:*Prediction Models* introduce the theoretical background and state of the art.

ii) Chapter 4:*Anticipation Model* introduces the proposed model, training procedure and its challenges.

iii) Chapter 5:*Application Scenario* theoretically formulates a cooperation scenario as a Decentralized Markov Decision Process demonstrating the anticipation model's importance.

iv) Chapters 6:*Results* defines the experimental setup, quantitative performance metrics and understanding how the model parameters affect it's performance.

2

Cooperation

Contents

The previous chapter gave a general overview on how the different parts like cooperation and action anticipation inter-relate. This section looks more closely at how humans' social cognitive mechanisms work, how we reason about others and how that relates to the focus of this thesis, action anticipation in a cooperation setting.

Cooperation can be seen as belonging to the broader field of social cognition which is concerned with the understanding of the other. More specifically, social cognition is the basis for social interaction, as it studies the mental processes involved in perceiving, attending to, remembering, thinking about, and making sense of the people in our social world, how the information humans receive about others are processed, how it is integrated to form an internal model of their intentions and motives [25].

Within social cognition, cooperation can be defined as "any form of social interaction whereby **two or more individuals** coordinate their actions in space and time to bring about a change in the environment towards a **shared goal**" in [26]. Coordinating actions requires understanding the surrounding agent's as goal oriented beings with intent and belief state.

Here, another sub-field of social cognition, Theory of Mind (ToM), intensely studied by Baron-Cohen and Leslie in the context of Autism spectrum disorders, helps us understand how the human cognition distinguishes between animate and inanimate entities. ToM relates to understanding the surrounding agents as goal oriented beings rather than inanimate objects which merely react to external stimuli [27].

In this chapter, the above mentioned concepts are expanded, establishing a theoretical base and vocabulary for the thesis. More specifically, this section justifies the separation of concerns between reasoning about inanimate (mechanistic) and animate (social) entities.

2.1 Cooperation

Cooperation, or joint action, according to Sebanz can be seen as the interplay between three factors: shared representation, action prediction and anticipative action [26]. The former is concerned with establishing a common goal and frame of reference while the second relates to predicting future actions and the third factor relates to acting according to the involved agent's intentions.

As a small aside, from a developmental robotics perspective joint action is especially interesting to study as the ability to coordinate actions with others only develops at the age of around 24 months. Before this age children are only able to engage in ritualistic predetermined behaviour [28], this indicates the ability to coordinate with others is learned and not innate.

The next sub-sections look more closely at each of the components that make up a cooperation scenario when viewed through the lens of joint action.

2.1.1 Shared Representation and Intent

The first factor, shared representation, is concerned with establishing the base for the cooperation, a common world state and sub-goals. [26].

This component is concerned with perceiving the current state of the shared workspace. It is related to the agents visibility of the shared workspace and knowledge of the shared goal. This establishes a mutual understanding between the agents, the context for future actions.

In this step the cooperation partners attempt to answer questions such as "What are we trying to achieve?" and "What are the sub goals necessary to reach the goal?". This establishes an initial layout of the required sub goals, during execution this work is shared dynamically between the robot and human.

In the context of Human-robot interaction this component is related to the robot being able to perceive the environment and receiving the same goal instructions as the human.

2.1.2 Action Anticipation and Perspective Taking

The second factor, action anticipation, is concerned with perspective taking and predicting the cooperation partner's possible next actions from non-verbal clues or contextual information. [26]

In this step the cooperation partners attempt to answer questions such as "What could be the others next action given the unfinished sub goals?"

In the human brain this perspective taking ability is believed to be associated to the so called mirror neuron system which are neurons that activate during the observation of actions performed by others [29, 30]. This behaviour is believed to be linked to the ability to understand other agent's behaviour as goal directed actions.

This ability is also deeply connected to the Theory of Mind explanation for the human perspective taking phenomena first introduced by Baron Cohen linked to autism spectrum disorders [31] further explored in the next section.

Furthermore, it is this ability which makes understanding and anticipating the action partners intention possible, ensuring a smooth and fast coordination between the parties and enables tasks such as passing a ball or driving in traffic [9].

In the context of Human-robot interaction this component is related to the robot understanding the behaviour and movement patters of the human given a context or a series of non-verbal cues such as eye gaze or body posture.

This thesis focuses on modelling this mechanism, prediction human action from non-verbal and contextual clues.

2.1.3 Anticipative Action

The third factor, anticipative action, is concerned with taking the right action knowing the shared goal and the probability distribution over the cooperation partner's possible future actions. [26]

In other words, knowing the final goal of the team and the action the other is attempting to do, what action takes the team closer to achieving the final goal.

In this step the cooperation partners attempt to answer questions such as "How should I act to bring the team closer to the goal?".

This is the component which makes the interaction dynamic, that is, the action partners adapt to each others movements and actions in order to accomplish a shared goal as fast or as efficiently as possible.

In the context of human-robot interaction this component is related to maximizing the future expected joint reward function as a function of the shared goal and future human actions.

2.2 Cooperative Robotics

The previous section introduced a perspective on how cooperation can be seen as the interplay between different factors. Within the field of robotics we can say Cooperative Robotics is concerned with understanding how to apply these cooperation mechanisms in robotics, research in this direction has a long history. A keynote at the 1996 AAAI national conference by Grosz [32] titled "Collaborative Systems" captures the importance of systems which cooperate naturally with humans. Citing the authors words "*(...) a significant challenge for AI in the 1990s is to build AI systems that can interact productively with each other, with humans, and with the physical world*", curiously, this challenge remains and the sentence is as valid for the 1990s as for today. In the talk, Grosz explains how systems only following human orders stand in stark contrast to how humans cooperate naturally by understanding the task context and each other's intentions.

The previous section presented the different modules of a joint action (shared representation, action anticipation, anticipative action). This section looks at different cooperative robotics implementations and architectures through the lens of joint action.

Schrempf [33] introduces such an architecture, aptly called *A Novel Approach To Proactive Human-Robot Cooperation*, where the author captures the shared representation (configuration and shared goal) by perceiving the scene and recognizing human intent. This intent is directly related to anticipating human action as there is a clear distinction between the robot and human tasks. The anticipative action is limited to executing the pre-defined robot part.

Another interesting architecture is the joint work on creating a *Platform-Independent Cooperative Human Robot Interaction System* [34, 35]. Here the shared representation corresponds to correctly identifying the active shared plan. The action anticipation is achieved by defining so called *perceptual primitives* which are easily recognizable and aid the partner's action recognition. The shared plan is split dynamically as a function of the recognized action. This work is especially interesting since the authors seek to create a platform independent framework for the implementation of abstract actuator-independent cognition cooperative robotics scenarios.

A different take on the same problem is mentioned by Jain [5], here the publication author enumerates a scenario where the cooperation partner is the car, here the shared representation is the desired trajectory, the action anticipation is defined as understanding where the driver intends to drive based on face and gaze features, the anticipative action is concerned with preventing accidents by predicting where the driver is steering the car to.

An interesting project related to these capabilities applied to the manufacturing scenario is the

ROBO-PARTNER project [36] (fig. 2.1) where the author stresses the importance of naturally interacting human-robot cooperation systems for flexible and safe manufacturing. The shared representation in this case consists of the assembly plan and the action anticipation a mixture of non-verbal and verbal cues.

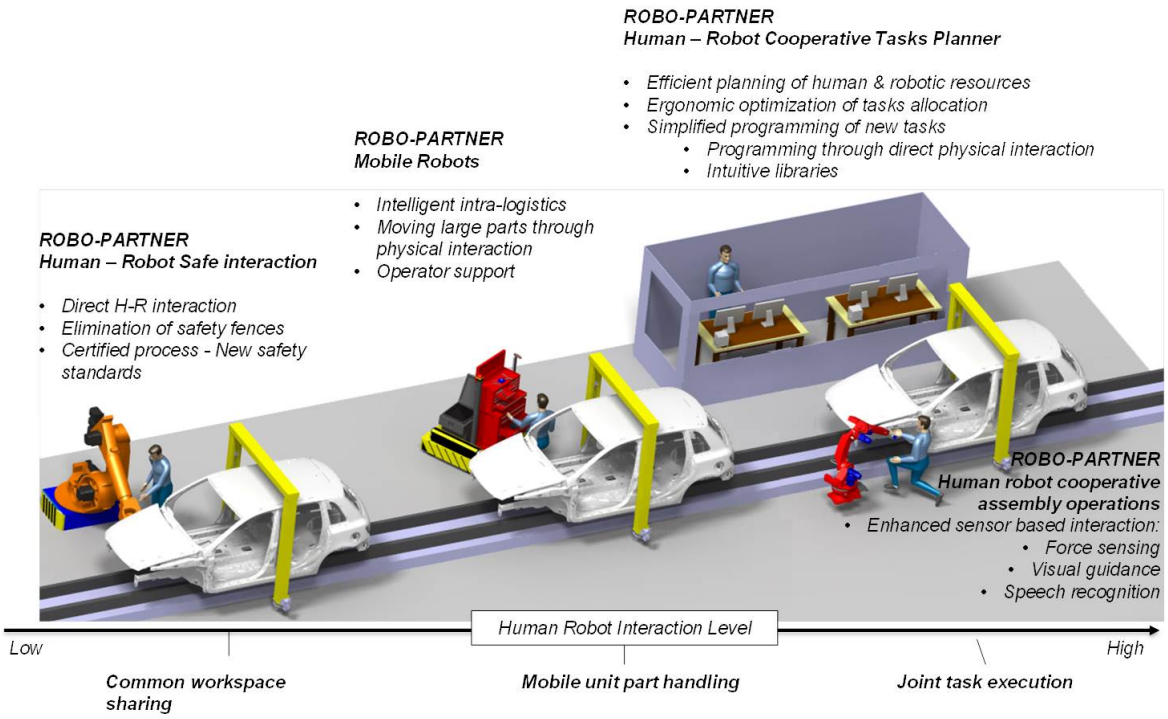


Figure 2.1: Source: ROBO-PARTNER project.

This section motivated and exemplified the importance of action anticipation in human-robot cooperation frameworks. The next section looks in more detail at the action anticipation component.

2.3 Theory of Mind

While the previous section introduced some concepts which are important for understanding the human cooperation mechanisms and the importance of modelling and understanding the intention of the cooperation partner, the question of how the human cognition separates and reasons about inanimate and animate objects remains open.

An important area of research in understanding and reasoning about goal directed entities is the Theory of Mind mechanism. This mechanism, as the name suggests, is related to forming a theory about the other's mind, theorizing about possible goals and intents.

This ability is directly related to perspective taking and social synchrony which makes social interaction possible [37]. Theory of Mind concepts were first introduced by Simon Baron-Cohen [38] and later complemented by additional views from Alan Leslie [39, 40].

2.3.1 Animate vs Inanimate entities

Before looking at how human cognition reasons about the intentions of the surrounding goal directed entities, it is important to see how the human cognition distinguishes between an inanimate object and a goal directed agent.

Alan Leslie expands on the idea of intentionality and self propelled motion, introduced by Baron-Cohen in the context of the Theory of Mind, defending that agents perceive the world in three levels of agency [41].

1. **Mechanical Agency** is related to the movement of physical bodies and is deeply connected to a module called the Theory of Body Mechanism (ToBY) which is responsible for codifying the infants knowledge of physical bodies.

   This module[42] is believed to be innate by Alan Leslie, but later work by Leslie Cohen went on to disprove this idea [41] and defending that the module rapidly develops in the child's first months of development.

   e.g. "If I push the cube it will move"

2. **Actional Agency** or Theory of Mind Module I (ToMM1), explains events or actions according to goals and intents. In contrast to the ToBY which deals with physical laws, this agency is related to psychological laws [39].

   e.g. "Mary is looking at the cake, her intent is to eat the cake"

3. **Attitudinal Agency** or Theory of Mind Module II (ToMM2) is related to understanding events through the agent's attitudes and beliefs [40].

   e.g. "Peter is not going to work today, he believes the payment at his company is unfair"

2.3.2 Mind Reading Mechanism

In the previous section we saw how the perception of the world can be decomposed in different levels of agency, in this section we will look more closely at one type of agency, actional agency, which, as it relates to the intention and action goal, has the biggest role during cooperation scenarios.

The concept of understanding the other as being a goal directed being, is first studied in the context of understanding Autism related disorders [38]. Autism spectrum disorders are especially interesting for studying this kind of phenomena as it is believed they are related to the absence of the so called *mind reading mechanism*.

Baron-Cohen was one of the first researchers looking into this problem, decomposing the *mind reading mechanism* in four modules [38] (Fig. 2.2).

1. **Intentionality Detector (ID)** interprets self propelled motion in terms of goals and desires. Here a dyadic relationship between the agent and the object is established: "Mary wants the cake".

2. **Eye Direction Detector (EDD)** identifies eye like features present in the scene after which it proceeds to classify the gaze direction and understanding the object of focus. This process results in a dyadic representation between the agent and the object: "Mary sees the cake".

3. **Shared Attention Mechanism (SAM)** combines the information from the two previous modules and reaches conclusions regarding the other agent's attentional focus and the interpretation of the own and others intentions. The result is a triadic representation between the self, the other agent and the object of focus: "I see Mary looking at the cake".

4. **Theory of Mind Mechanism (ToMM)** combines all the information to reach a coherent theory about the other agent's intentions and mental states [38].

As we can see, Baron-Cohen's theory of mind is greatly inspired by the idea that eye like features, as they convey information seeking signals, are an important clue to distinguish between inanimate objects and intentional agents [14].

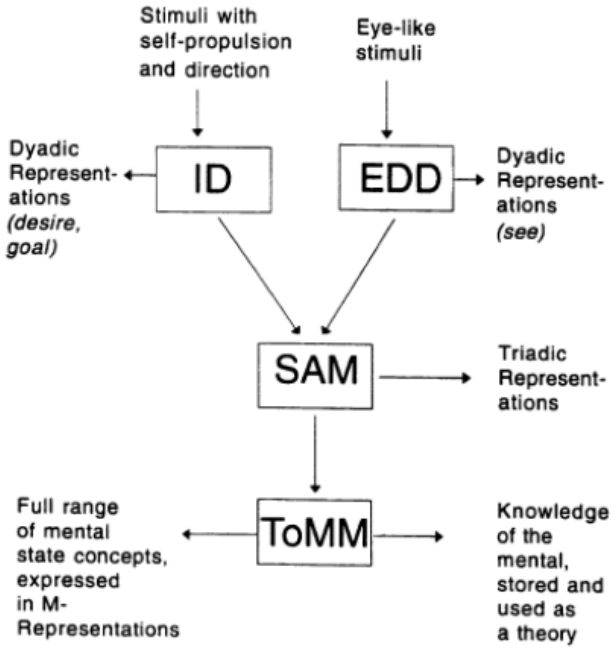


Figure 2.2: The mindreading system. **Source:** Mindblindness: An Essay on Autism and Theory of Mind

The object of focus of this thesis is this mind reading mechanism, how we can estimate the intention and action goal from non-verbal clues such as movement and eye gaze and make robots a little less autistic.

2.4 Gaze

The previous section went over how human gaze is directly related to the detection of agency and intention [38, 43], where we introduced Baron-Cohen's [38] model of seeing gaze as a means of detecting agency and intent through the *mind reading mechanism*.

While there are different human non-verbal clues, this section expands on the previous concepts by taking a closer look at how gaze conveys intent for the sake of example. Gaze, according to Tomasello [43], is so important that the human species have evolved especially white sclera to facilitate short range cooperative joint action scenarios.

Figure 2.3: Comparison between primate and human eyes.

Gaze can have different functions related to social interactions. The main behaviours associated to the human gaze fall under the categories of information seeking and signalling [16, 44].

As the eye, in opposition to the lips and ears, can both signal and perceive they are specially difficult to interpret [44]. Nevertheless it is this property that makes the eyes so important in social interaction, it can perceive the surrounding agents but at the same time communicates our point of attention and enables us to communicate the commitment to a shared task or our attention in a conversation [45].

2.4.1 Signalling

Gaze have both a information seeking and explicit signalling function. This section is concerned with the latter, how we can perceive the intent from the focus of attention.

Gaze has been studied from the perspective of making humanoid robots more predictable and intuitive to cooperate, where the main concern is understanding the different functions such as regulation, establishing joint attention and initiating/avoiding of social encounters [46].

Regulation is related to the human ability to regulate the flow of dialogue during a conversation, in these situations gaze has the role of aiding the turn taking behaviour by signalling attention or a request for response [47, 48].

The role of gaze in joint attention, that is, communicating and understanding an object of focus, is in perceiving the visual attention of the other. The protocol for joint attention includes first, observing the object of focus, second, redirecting the attention of the other to the object of focus, third, observing the other and confirming the understanding of the object of focus [49].

As we have seen so far gaze can have an information seeking and information conveying role. Nevertheless, the boundary between the two is not clear, gaze conveys information while we are

seeking information and vice versa [44].

The focus of this thesis is in using non-verbal cues like gaze, which convey information about the human intent during the interaction with the environment to anticipate the cooperation partner's next action.

2.4.2 Information seeking

The last section focused on the more explicit use of gaze for the purpose of signalling and conveying information. As we have seen in the previous section, gaze information seeking and conveying is interlinked and there is an implicit leakage of information during the interaction with the environment.

Human gaze plays an important role in collecting information about our surroundings, eyes are a vital sense the human uses to keep himself safe from danger and in track of his surroundings. As the eye only has a limited visual area it is able to cover, field of view, it continuously moves to adapt it's window into the world.

A concrete example of this type of gaze behaviour is called visuo-motor coupling. Visuo-motor coupling enables the human to gather information around the active end-effector [50]. Nevertheless, this information gathering has the side effect of communicating the action goal to the cooperation partner.

An interesting work in this direction and which highlights the relation between the action goal and gaze patterns is the work by Lukic [50] where he is able to model the connection between the end-effector and the eye gaze as coupled dynamical system with the goal of seeking information around the action target. This exemplifies one possible relation between implicit non-verbal cues and intents such as action targets.

This type of implicit cues which allow us to estimate the cooperation partners intention from his interaction with the environment, be it through gaze or body posture, is the focus of this thesis.

2.5 Conclusion

In this chapter we were able to go from the very broad field of social cognition down to the more specific details on the different factors which influence a cooperation scenario, reasoning for the importance of anticipating human action for a natural and efficient coordination between the action partners.

We introduced a line of research, followed by Baron-Cohen and Leslie in relation to autism spectrum disorders, about the topic of how the human reasons separately about entities with an intentional agency and those with a more mechanical nature. More specifically we looked at the how the human cognition reasons about other intentional agents through the so called mind reading mechanism.

We introduced the human gaze as an example of how humans convey implicit information about their intention while interacting with the environment and seeking information continuously adapting the field of view.

The next chapter will take the concepts we introduced in this chapter and give an overview of how

cooperation architectures and, more specifically, intention prediction have been implemented in the field of social robotics.

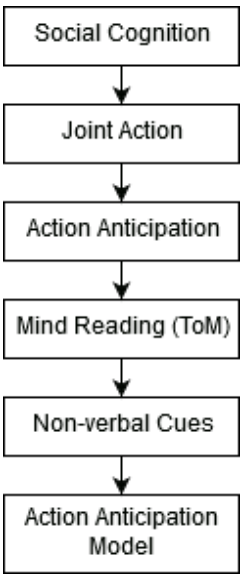


Figure 2.4

3

Action Anticipation: Theoretical Background

Contents

The previous two chapters introduced the general field of cooperation and demonstrated the importance of the action anticipation module as a means of understanding and adapting to the behaviour of the cooperation partner.

This section starts by introducing the general class of sequence prediction problems and how it relates to action anticipation. The next section moves on to introduce both more classical generative approaches like the Hidden Markov Model and discriminative models like the conditional random fields, finishing with more modern approaches like the Structural Recurrent Neural Networks and Long Short Term Memory and Attention.

3.1 Action anticipation

As mentioned in the previous chapter, the problem of action anticipation has an important role in the cooperation problem.

Human action (behaviour) anticipation (prediction) can be solved on different levels of abstraction and is concerned with estimating the set of next possible action sequences.

On a higher level of abstraction, models can predict actions in a discrete space[5, 19] where the actions are symbolic in nature and can represent underlying movement patterns, e.g. “press-button” or “grab-object”. Vernon [51] argues for the importance of this type of action anticipation under the name of intention recognition, understanding the human action goal.

On a lower level of abstraction, movement can be directly anticipated in a continuous space [24], e.g. human walking trajectories. In this thesis we will refer to this type of anticipation as movement anticipation to contrast the continuous nature with the discrete nature of the intention recognition task.

Past works explored different architectures like probabilistic models with Markovian assumptions [18] and discriminative methods such as conditional random fields [17]. Recently, recurrent neural networks which do not assume limiting Markovian assumptions have shown excellent results [19, 24, 52].

Predicting continuous actions has been addressed in the context of body posture and human trajectory prediction. Example of relevant work include the use of recurrent neural networks by Martinez [24] as a means of predicting contextually coherent future joint trajectories.

The dual problem is action prediction in discrete outcome space. Examples of related work include a Conditional Random Field based approach by Koppula [17] to capture temporal dependencies and Saponaro’s Hidden Markov Model based approach [18]. More recently, Jain [19] introduced the structural RNN as a means of encoding past contextual information and predicting a fixed number of steps in the future.

As we have seen from this short introduction to the field, action anticipation has been approached from different perspectives, both in continuous and discrete space and generative and discriminative models, this thesis is concerned with expanding the field in the direction of discrete space discriminative anticipation models.

3.2 Generative vs Discriminative

The goal of the classification problem is to assign an underlying class, y, to observations, x. The problem is to select the class, y, which maximizes the conditional probability of the class given the observation (3.1).

$$y^* = \arg\max_y P(y|x) \tag{3.1}$$

There are two possible approaches to the classification/prediction problem. The first is called the generative approach where we model the joint distribution of the underlying random process. With a model of this kind we are able to assign a probability value to every observation and class pair.

In this setting the conditional probability of the class given the observation is given as a function of the joint probability between the observation and the class (3.2).

$$y^* = \arg\max_y P(y|x) = \arg\max_y \frac{P(x,y)}{P(x)} = \arg\max_y P(x,y) \tag{3.2}$$

The second is called the discriminative approach where we avoid modelling the joint distribution and only model the conditional probability of the class given the observation.

In this setting the conditional probability of the class given the observation is given directly as a function which maps the observation to a class probability (3.3).

$$y^* = \arg\max_y P(y|x) = \arg\max_y f_y(x) \tag{3.3}$$

Both approaches to the classification problem have their advantages and disadvantages. Generative models are useful if we are interested in inverting the model and sampling a set of observations which are coherent with a given class. Discriminative models solve the classification problem directly (by not requiring the full joint probability between the observations and classes).

A more intuitive example would be a scenario where two learners, a discriminative and generative, are instructed to distinguish between different languages. A generative model would learn the full probability distribution over possible sequences (e.g. grammatical structures) while a discriminative would only extract those features from the observation which are interesting for the task at hand, discriminating between the languages.

3.3 Sequence Modelling

The theoretical framework of the classification setting is broad and can be seen as covering the sequence modelling task. The sequence modelling task is similar in nature to the classification task with the difference being in the reference y becoming a sequence.

$$y = [y_1, y_2, ..., y_N] \tag{3.4}$$

The formulation of the maximization problem over the probability remains the same. The probability of a sequence is given by the joint probability of all its entries, which might or might not be decomposable into individual factors.

3.3.1 Hidden Markov Model

While there are different generative sequential models, here we will only focus on one important sequence model, the Hidden Markov Model. Hidden Markov Model (HMM) is a type of generative directed graphical model which models the joint distribution between class labels and observations by decomposing the full joint distribution into a graphical model network topology.

The topology is made up of class nodes and observation nodes, where the nodes represent realisations of the underlying random process.

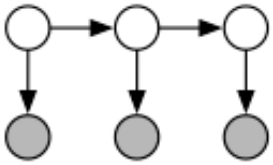


Figure 3.1: Hidden Markov model structure.

The connections between class nodes (unshaded) and between class nodes and observation nodes (shaded), are called transition probabilities and emission probabilities respectively.

The model is called hidden since it maps an underlying (hidden) state, y, to an observation, x, through the so called emission probability and Markov because it assumes the Markovian prior in that only the prior hidden state defines the next.

HMMs capture the transition dynamic between the classes and the relation between the underlying classes and the observations. The joint distribution can then be decomposed into a factor related to the transition and another related to the emission probability. 3.5

$$P(x,y) = P(y_0)P(x_0|y_0)\prod_{i=1}^{N} P(y_i|y_{i-1})P(x_i|y_i) \tag{3.5}$$

3.3.2 Conditional Random Fields

In Fig. 3.2 is a graphical representation of the models sequential structure. Here the darker nodes represent the measured states and the light nodes represent the unobserved hidden state. The hidden states are connected between them through a so called transition distribution and the measured states are connected to the hidden state through an emission distribution [53].

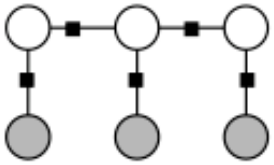


Figure 3.2: Conditional Random Field model structure.

Conditional Random Field (CRF) is a type of discriminative undirected graphical model. This kind of model is sequential by nature, the CRF is similar to the HMM in the sense that the underlying connection topology is identical.

They differ in that while the generative HMM models a joint probability between adjacent nodes in the network, the discriminative CRF considers the edges as features functions which only capture a discriminative measure between nodes ([53]).

This kind of graphical model is common in activity recognition, where the measured state corresponds to a human motion and the hidden state to the activity label [54, 55].

As this model is discriminative it assigns a probability to a specific realization of the measured and hidden variables avoiding a full generative model of the distribution. The joint distribution over the nodes (field) can be calculated from the factors between the nodes [53].

More recently this kind of structure has been applied to the action anticipation problem, in [17] the author is able to anticipate possible futures by instantiating weighted particles and searching the space of possible future sequences pruning unreasonable futures with the weight of the random field.

3.3.3 Recurrent Neural Networks

Following the recent success of neural networks in similar sequence to sequence tasks it is very interesting to explore these models further [5, 56, 57]. Recurrent neural networks are especially interesting since in contrast to the two previous models they don't include limiting Markovian assumptions, their internal state is able to capture long range dependencies.

The Recurrent Neural Network (RNN) is the sequential version of the feed forward neural network. While the feed forward neural network maps a fixed sized feature vector to a classification label, the RNN is able to classify variable length ordered sequences of feature vectors (Fig. 3.3).

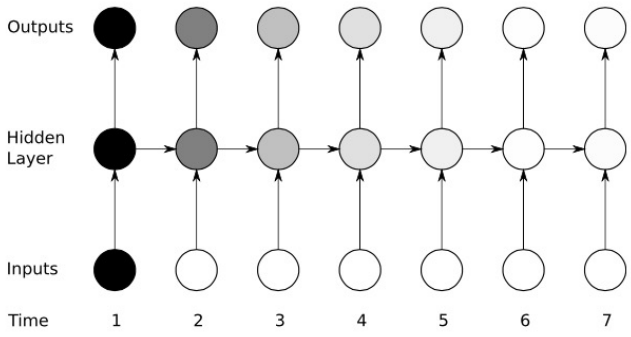


Figure 3.3: Recurrent Neural Network structure.

The RNN is a topology which can be decomposed in three layers. The bottom layer receives the feature vector representations and converts them to a higher level representation. The middle layer receives this representation from the bottom layer and feeds this vector to the RNN cell, the central part of the RNN network topology. The output layer projects the result from the previous layers into a probability space through a projection matrix and a softmax normalizing layer.

The middle layer's RNN cell is a block which repeats (unfolds) along the time dimension. This block receives an input and together with the previous state computes the next state and the output at the current time step.

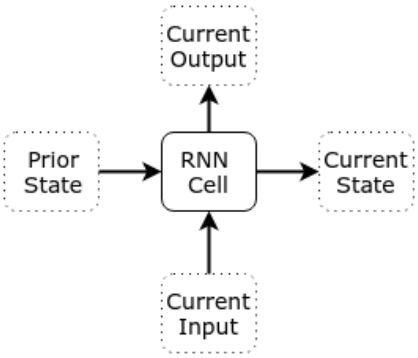


Figure 3.4: Recurrent Neural Network cell.

Through this mechanism we are able to keep a continuously changing internal state which codifies the contextual information. The cell is defined by an expression for the output at the current time step, o_t and next internal state, h_t (eq. 3.6). Here i_t, o_t and h_t represent respectively the current input, output and hidden state. The parameters W_x, U_x and b_x are part of the model parameters θ

$$\begin{cases} h_t = \sigma(W_h h_{t-1} + U_h i_t + b_h) \\ o_t = \sigma(W_o h_t + U_o i_t + b_o) \end{cases} \tag{3.6}$$

This recurrence is equivalent in structure to a discrete dynamical system with external input. One shortcoming with this structure is noticeable when propagating the error in this network.

To train these kind of models the network is unfolded to the full model as represented in (Fig. 3.3). Then the influence of the first input on the final output is propagated across the network. As the state update function operates on the input its influence vanishes along time.

The same happens with the gradient, when propagating the error along the network's time dimension the gradient can either vanish or explode, depending on whether the state update function's derivative decreases or increases the back propagation's magnitude.

This is a common issue when training this type of model and was in part mitigated by the introduction of the Long Short Term Memory (LSTM) topology by Hochreiter & Schmidhuber [58] reviewed in the next section.

3.3.4 Long Short Term Memory

The previous section introduced the general topology of a recurrent neural network and introduced the vanishing/exploding gradients problem. In this section the LSTM topology is reviewed which seeks

to mitigate this problem.

The vanishing/exploding gradients problem is mainly related to the RNN applying the same state update function in each time step. The LSTM is able mitigate this problem by selectively updating the internal state as a function of the current state and input through a gating mechanism [58] (Fig. 3.5).

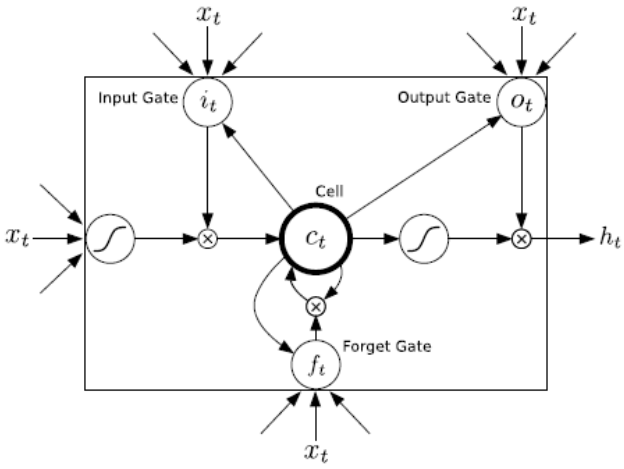


Figure 3.5: Long Short Term Memory cell.

In this topology the internal state and current input define the input, output and forget gate's flow of information.

The input gate defines the influence of the new input on the internal state. The forget gate, as the name suggests, defines which parts of the internal state are retained over time, the output gate defines which part of the internal state becomes the output of the network (3.7). Here f_t, i_t, o_t and h_t represent respectively the forget gate, current input, output and hidden state. The parameters are W_x, U_x and b_x are part of the model parameters θ

$$\begin{cases} f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ c_t = f_t \cdot c_{t-1} + i_t \cdot \sigma(W_c x_t + U_c h_{t-1} + b_c) \\ h_t = o_t \cdot \sigma(c_t) \end{cases} \tag{3.7}$$

By selectively accepting new information and erasing past information, this topology is able to effectively maintain the error flow across the whole network.

3.4 Bias-variance trade-off

Before moving on to introducing the implemented model it is first necessary to recall an important concept, the bias-variance trade-off.

Statistics defines the approximation error of a given model as the combination of two distinct error terms: bias and variance errors. Bias error is related to a systematic error in the model while variance is related to the spread (variance) of the error.

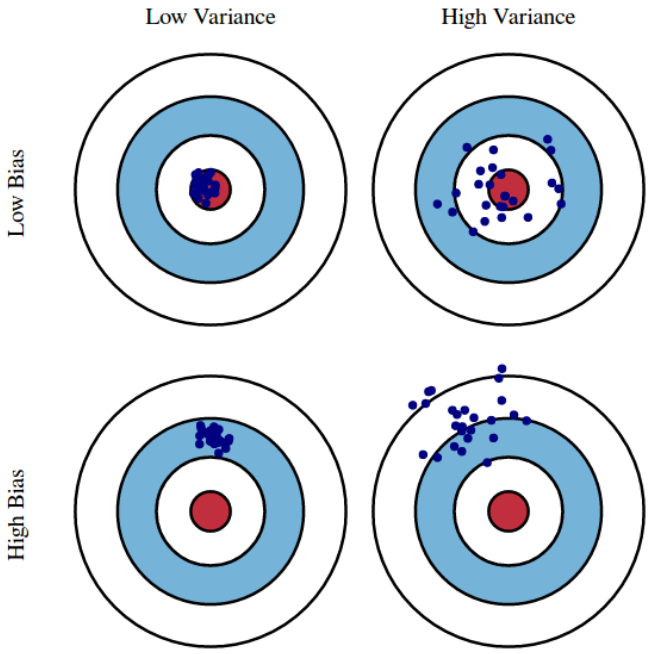


Figure 3.6: Bias-variance trade-off visualization.

Bias error positively correlated with the model's capacity, a high bias means the model does not adapt to the training data, that is, it under-fits. Variance defines the sensitivity on the samples, a high variance means the model adapts too much to the training data, that is, it over-fits.

A well tuned machine learning model strikes a balance between the two, a capacity big enough to fit the data but not to the point of over-fitting on the training data and not generalizing to new data.

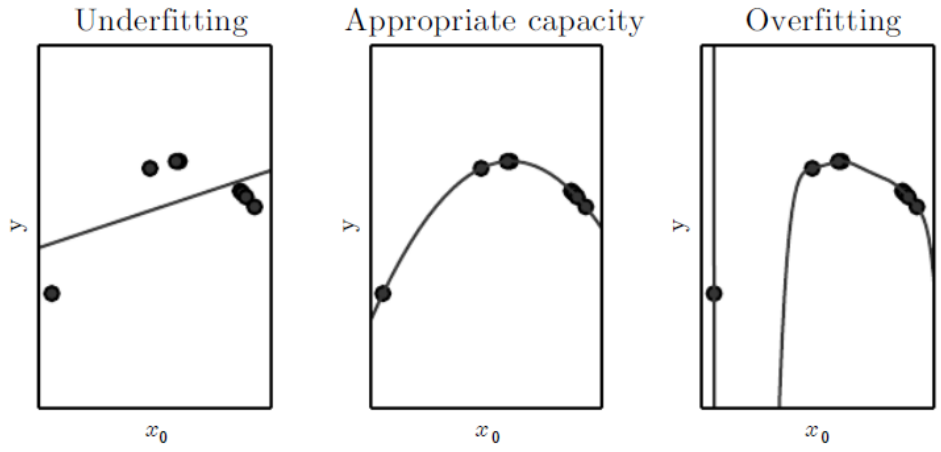


Figure 3.7: Bias-variance trade-off Source: Deep Learning [1]

The figure 3.7 captures this dynamic. The first sub-figure from the left exemplifies a situation where the model has a lower capacity than required for the correct modelling of the data, high bias. Here the model is linear but the data seems to follow a second order polynomial model.

The second sub-figure from the left exemplifies the ideal capacity. Here the model, a second order

polynomial, accurately captures the data.

The third sub-figure from the left represents an over-fitting situation, where the capacity of the model is higher than necessary, high variance. Here the model adapts too closely to the training set and will not generalize to data outside of this set.

3.5 Conclusion

We started this chapter by defining the more general classification problem, inside of the more general framework of the classification problem we introduced several sequence prediction models.

We started the review with more classical models like the Hidden Markov Model, the Conditional Random Field and moved on to more recent models like the Recurrent Neural Network and Long Short Term Memory.

The Long Short Term Memory cell due to the gating mechanism mentioned in this chapter will be the basis for our model, which is introduced in the next section.

4

Proposed Action Anticipation Model

Contents

The previous chapters started with a broad overview of the factors which influence cooperation demonstrating the importance of anticipating the cooperation partners actions to guarantee a smooth and efficient cooperation and ended with more specific details like how the human cognition separates between animate and inanimate entities, demonstrating the existence of an implicit human action anticipation module in the human mind called the mind reading mechanism.

While the previous chapter gave an overview of generic sequence prediction models as a state of the art reference, this chapter is concerned with the thesis' proposed solution.

The chapter starts with a clear problem statement, moving on to introducing the proposed models topology and finishing with defining the model's optimal parameters through formulating the search in the parameter space as a minimization optimization problem.

4.1 Problem statement

The problem of action anticipation is concerned with predicting the sequence of possible next actions given a past sequence of non-verbal cues (e.g. gaze position or skeleton configuration).

Figure 4.1: Action anticipation Given a sequence of features about the human (e.g. gaze position or skeleton configuration) we are interested in estimating the sequence of possible next actions.

Figure 4.1 exemplifies such a scenario: 1) in the first segment the human looks at the box, 2) in the second segment the human grabs the box, 3) finally, in the third frame the human looks at an empty spot in the cupboard. After seeing this sequence of actions we know the human is holding the box and is looking at an empty spot on the cupboard, from this information we could infer the human's intent is to place the box in the position he is looking at.

From this example we can see we have two variables which are inter related, the sequence of past segment representations (e.g. gaze or skeleton configurations) and the future sequence of executed actions. The focus of this thesis is to relate clues like gaze and movement with the human's intent or action target.

While this example clarifies, in a less abstract manner, the task at hand, the next section is concerned with formalizing this setting and clearly defining the variables of interest and concepts such anticipating a sequence of actions.

4.1.1 Notation

While we gave a concrete example of the problem setting, this subsection defines the terms more formally and summarizes the main notational choices made when writing this section:

- Information about a given past time instant can be represented as a fixed size vector, a so called feature vector. We will denote a sequence of such past time instant representations with the lower case letter x, the representation of the segment at past time step i is denoted by a subscript, x_i.

- As the human is stochastic in nature, the future actions are represented as a sequence of discrete distributions over the action labels. We will denote a sequence of discrete distributions with the lower case letter y, the representation of the action distribution at a future time step, i, is denoted by a subscript, y_i. The probability of a given action label, a, at a future time step, i, is given by a subscript, $y_{i,a}$

- A sequence of past time instant representations defines one sample, x, a collection of these sequences is represented with the upper case letter, X, so the collection of segment representation and action distribution sequences is represented respectively by X and Y, the sequence at position k in the collection is denoted by a superscript, X^k and Y^k respectively.

- The action anticipation model associated to a given set of parameters, θ, is defined as the probability of future action sequences given a sequence of past segment representations, x. We will denote the model with the letter p_θ for probability as a function of the model parameters, so the final expression for the model becomes $p_\theta(\hat{y}|x)$. It is important to distinguish between the reference, y, and the estimated, $\hat{y}$, sequence of distributions

4.2 Prediction model

The last section defined the problem and we have seen that our objective is to approximate the distribution $p_\theta(\hat{y}|x)$, that is, the probability over future action sequences.

Estimating this probability is equivalent to estimating the probability of every possible action sequence, this is computationally infeasible in practice, in the next section we will look more closely into this problem.

In our proposed approach, we will divide the problem in two parts, the first, the encoder, is concerned with condensing sequence of segment representations, x, by into a fixed sized representation which we will call the context vector. Through this encoder we are able to convert a variable length sequence of past time instants into a fixed vector representation which condenses the past information into some internal representation learned by the network.

The second part, the decoder, acts upon the vector representation of the past information, returned by the encoder, and expands the fixed sized context vector into multiple future action sequences and their respective probabilities.

4.2.1 Encoder

The first part of the model is a contextual information encoder. The encoder condenses past information into a fixed length context vector through a recurrent Long Short Term Memory (LSTM)

cell. The LSTM cell condenses the information by keeping and updating an internal state as it receives new input.

The embedding is a fully connected layer (FeatureVectorDim x EmbeddingDim), where FeatureVectorDim is the size of the time instant representation vector, the so called feature vector. The EmbeddingDim parameter defines the dimensionality of the hidden state, for the purpose of this implementation the EmbeddingDim was chosen such that it reduces the dimensionality of the input vector, x_i.

The embedding layer includes dropouts which act as a regularization to the model [59], later sections look with more detail into the effect the dropout layer has on the network.

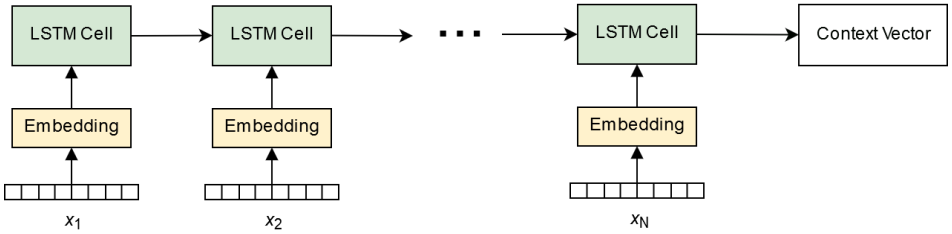


Figure 4.2: Encoder condenses all the relevant information into a context vector.

In practice the encoder LSTM's hidden state dimension was chosen to be 20, how to choose the parameters is analysed in more detail in the results section. This context vector is the initial state of the second part of the model, the decoder, which we will look at in the next section.

4.2.2 Decoder

The decoder is responsible for generating a coherent future sequence of actions from the contextual information given by the encoder network.

At each step the recurrent decoder cell, a LSTM cell, iterates upon the internal state and returns a discrete distribution over the action labels.

The distribution over action labels is obtained by transforming the decoder's internal state vector into a space with a dimension identical to the number of action labels through a dense layer. This output vector is turned into a distribution by normalizing it using a softmax layer.

The full output sequence is generated by iterating this cell through a decoding process. The decoding process samples multiple actions (e.g. the most probable actions) from the discrete action probability distribution, branching the decoding process and feeding the actions as an input to the next decoding iteration.

The projection into the action label space is a fully connected layer (HiddenStateDim x VocabDim), where HiddenStateDim is the dimension of the decoder's hidden state and VocabDim the number of action labels. The decoder LSTM's hidden state dimension is 20.

The softmax layer corresponds to a normalization of the projected output, z into a normalized probability space.

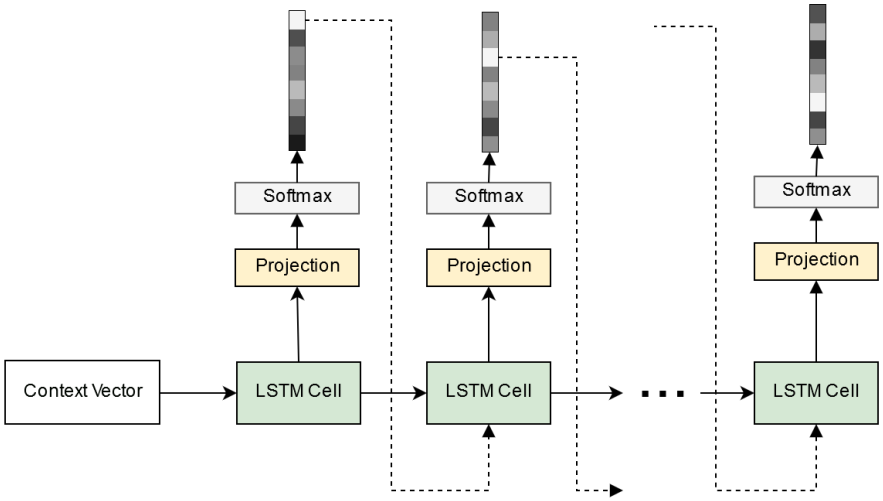


Figure 4.3: Decoder expands the context vector into the future action sequences.

$$softmax(z)_i = \frac{exp(z_i)}{\sum_j exp(z_j)} \tag{4.1}$$

4.2.3 Final model

The previous two sections looked in more detail into both parts of the model: a) the encoder condenses past information into a fixed sized context vector b) the decoder expands this context vector into all possible future action sequences.

Figure 4.4 shows the final topology, here we can clearly see how the encoder and decoder function together to generate the future action sequences.

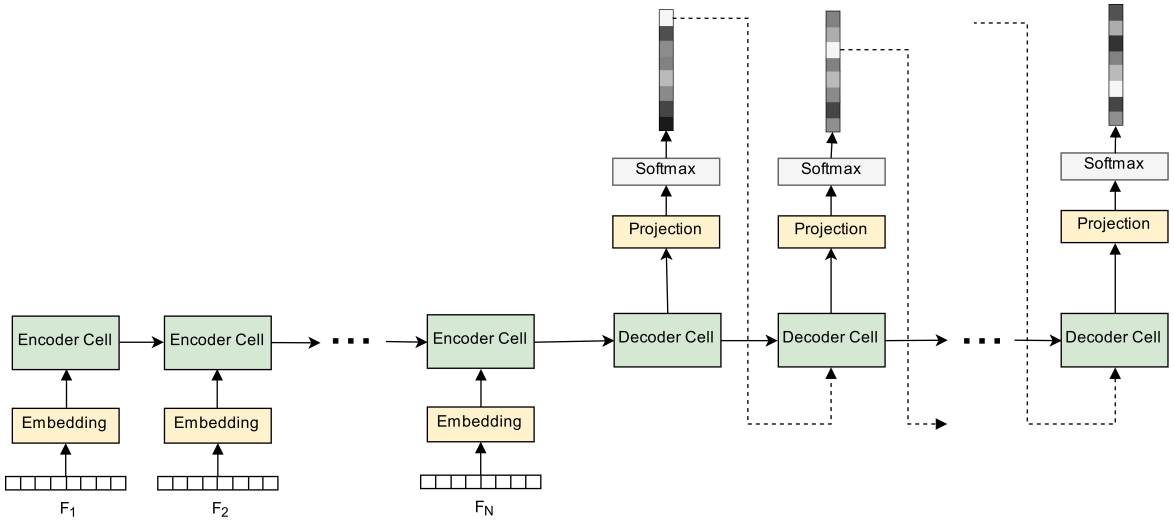


Figure 4.4: Encoder-decoder model. Left part summarises past information into a fixed length context vector. Right part expands this context vector into future action sequences.

After training, the decoding process allows for variable length action sequence prediction. Expanding every possible future action sequences becomes NP hard and computationally intractable.

The next section looks more closely at this issue and introduces one possible solution to the problem.

4.3 Complexity issues

The previous section hints at the complexity underlying the decoding process. At every decoding step the decoder samples one or more actions from the output distribution as possible actions at a given time step, it then expands these actions by branching and feeding them individually as input to the next prediction step decoder iteration. There are two strategies that could be applied to this decoding process.

Naively expanding the space of all possible action sequences and selecting the most probable action sequence in the end seems like a reasonable idea. Nevertheless, expanding the actions at each step results in a vocabulary sized multiplier in the number of possible action sequences at every prediction step. In terms of complexity this means the number of action sequences increases exponentially with the number of prediction steps.

Considering a 10 actions vocabulary size, the first decoding step results in 10 action sequences, expanding the 10 action sequences results in 100 possible action sequences for a two step ahead prediction, a N step ahead prediction would result in 10^N action sequence alternatives. Expanding all possible future action sequences becomes computationally infeasible.

Greedily expanding only the best option, could be a solution to the exponentially expanding trajectory space, nevertheless it has the shortcoming that this method only returns one action sequence prediction.

A common solution to these two problems is the implementation of a *beam search* based decoder [60]. This method keeps a set of the top K best future action sequences at every decoding step, expanding by the action vocabulary size and pruning the action sequence set back to the top K best future action sequences. The end result is a sample of the top K most probable future action sequences ordered by likelihood. These action sequences are called beams and K is called the beam width parameter.

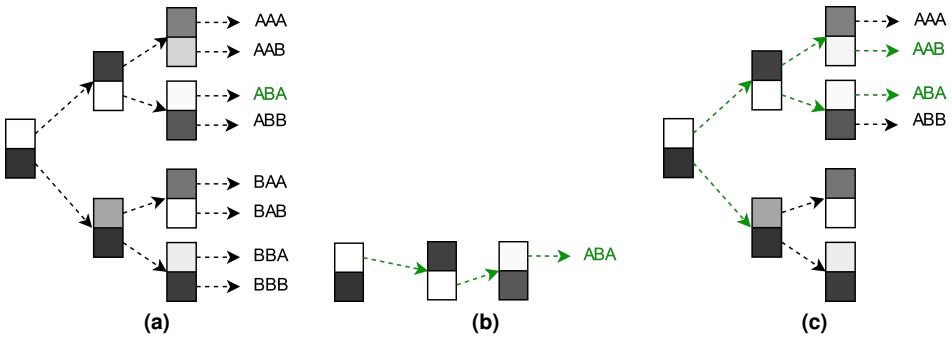


Figure 4.5: Search methods comparison. a) Exhaustive search expands all possible action sequences. b) Greedy search picks the most probable action at every step. c) Beam search keeps a set of the best K action sequences, expanding and pruning the set at every step.

4.4 Model Parameter optimization

The optimal model parameters, θ^* are those which best approximate the dataset distribution. Formulating this setting as an optimization problem is possible through a so called cost function, which measures the difference between the reference distribution, y, and the predicted distribution $p_\theta(\hat{y}|x)$ as a function of the model parameters, θ.

Finding the optimal model parameters corresponds to solving a minimization problem on the cost function (eq 4.2). Here N_s represents the number of samples, X^k a sequence of feature vectors and Y^k a sequence of reference distributions.

$$\theta^* = \arg\min_\theta \sum_{k=0}^{N_s} Cost(\hat{Y}^k(\theta, X^k), Y^k) \tag{4.2}$$

In this section we look at how the cost function for this problem is formulated, the algorithm adopted for the minimization and how the non-convexity of the problem is handled using regularizer terms.

4.4.1 Cost function

The cost function is a measure on the parameter's "quality". In the case of the thesis it is defined as the difference between the reference distribution, y, and a estimated distribution, $\hat{y}$ obtained from the model ($p_\theta(\hat{y}|x)$) through some decoding process, P. Minimizing this cost function as a function of the parameters leads to the optimal model parameters.

Taking into account that the model has to approximate a discrete distribution, a sequential cross entropy loss function is selected which is a measure of divergence (difference) between two distributions (4.3). Here $\hat{y}_{i,a}(\theta, x)$ and $y_{i,a}$ represent respectively the action's (a), estimated and reference probability and A the set of all actions.

$$H(\hat{y}_i(\theta, x), y_i) = -\sum_{a \in A} \hat{y}_{i,a}(\theta, x) \log(y_{i,a}) \tag{4.3}$$

The sequential cross entropy is obtained by summing the cross entropy cost over the prediction steps. Here $\hat{y}_i$ and y_i represent respectively the estimated and reference discrete distribution and L the prediction length.

$$Cost(\hat{y}(\theta, x), y) = \sum_{i=0}^{L} H(\hat{y}_i(\theta, x), y_i) \tag{4.4}$$

4.4.2 Adam: Adaptive moment estimation

As the function is non-linear in nature it is highly non-trivial to find a closed form solution for this optimization problem, tt is therefore necessary to adjust the parameters iteratively. An algorithmic approach for solving this class of continuous problems is the class of gradient descent optimizers.

Gradient descent algorithms are iterative and work on the principle of changing the parameters in the direction of minimizing the total cost function. Here X and Y correspond respectively to the

collection of feature and reference distributions, θ corresponds to the model parameters, n the current iteration of the algorithm and η the so called step parameter.

$$\theta(n+1) = \theta(n) - \eta \nabla Cost(\hat{Y}(\theta(n), X), Y) \tag{4.5}$$

Although the basic gradient descent algorithm formulation is simple in nature (eq. 4.5), in practice more complicated formulations are more efficient and help to mitigate common problems.

According to a recent survey, executed by Ruder [61], Adam is a gradient descent formulation which has an excellent convergence rate in comparison to other algorithms.

According to [61], the Adam algorithm seeks to merge two algorithms: AdaGrad, which works well with sparse gradients and RMSProp, which works well in on-line and non-stationary settings [62].

Algorithm 1 Adam: f(θ): Stochastic objective function with parameters θ, θ_0: Initial parameter vector, α: Stepsize, $\beta 1$, $\beta 2 \in$ [0, 1): Exponential decay rates for the moment estimates, $\epsilon = 10^{-8}$. Source: Adam: A Method for Stochastic Optimization [62].

function ADAM($f(\theta)$, θ_0, α, β_1, β_1)
  $0 \rightarrow m_0$ (Initialize 1st moment vector)
  $0 \rightarrow v_0$ (Initialize 2nd moment vector)
  $t \rightarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t + 1 \rightarrow t$
    $\nabla_\theta f_t(\theta_{t-1}) \rightarrow g_t$ (Get gradients w.r.t. stochastic objective at timestep t)
    $\beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \rightarrow m_t$ (Update biased first moment estimate)
    $\beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \rightarrow v_t$ (Update biased second raw moment estimate)
    $m_t / (1 - \beta_1^t) \rightarrow \hat{m}_t$ (Compute bias-corrected first moment estimate)
    $v_t / (1 - \beta_2^t) \rightarrow \hat{v}_t$ (Compute bias-corrected second raw moment estimate)
    $\theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon) \rightarrow \theta_t$(Update parameters)
  end while
return θ_t (Resulting parameters)
end function

This algorithm is specially interesting since its learning rates are parameter specific which improves the convergence rate.

This section introduced the gradient descent class of algorithms for iteratively computing the optimal parameters. This class of algorithms, since it works on the principle of changing the parameters parallel to the direction of the gradient, requires the gradient of the cost function. The next section looks at computational graphs and how they enable us to calculate this gradient efficiently.

4.4.3 Computational graphs

Calculating the derivative of the cost function is non-trivial as the cost function results from the composition of non-linear functions. A recent method for representing function compositions is the computational graph. The computational graph represents the calculations on input variables called tensors as nodes in a network topology.

These representations are especially useful for computing the back-propagation, a systematic way of calculating the derivative by decomposing the function using the chain rule. This computation is split in two steps.

The first step is called the forward pass and consists of updating all the nodes in the graph with the current configuration of input and state tensors. After computing all the nodes' values the cost function is recomputed.

After the forward pass, the back-propagation propagates the error backward along the network. Propagating the error backwards corresponds to computing the derivative and relating the output error with the current parameter values.

The end result is the gradient vector over the parameters which codifies how a change in the parameters affects the cost function.

During this thesis the Tensorflow framework is used to represent and compute the computational graphs.

4.4.4 Dataset preparation

Reiterating the initial problem setting, the objective of the model is to predict a sequence of future human actions given a history of past human movements.

Looking closely at the problem formulation and remembering the concepts introduced in the cost function, we require a reference distribution for the future actions.

This setting can be easily extracted from an action recognition dataset. Here a sequence of human movements is mapped one to one to a sequence of action labels. Mapping a sequence of past human movements to a sequence of future action labels results in the required format.

Iterating through a sequence of body posture, X_a, and action label pairs, L_a, the action prediction label vector, L_p, and feature vector, X_p, at a given time step, t, are given respectively by taking the labels for the F next action labels and the human body posture features from the past P entries.

$$\begin{cases} X_p(t) = X_a[t-P:t] \\ L_p(t) = L_a[t:t+F] \end{cases} \tag{4.6}$$

The reference distribution corresponding to a given label is given by so called one-hot function (eq. 4.7) which expands a scalar label into a the reference distribution vector. The reference distribution vector has a probability of 1 in the label position and 0 anywhere else. Here v represents a discrete output distribution in the form of a vector and the subscript, v_j, an index to the entry at the position j of the vector.

$$v = onehot(l) = \begin{cases} v_j = 1, & j = l \\ v_j = 0, & j \neq l \end{cases} \tag{4.7}$$

After extracting the desired feature vector X the input is centred and normalized along the feature dimensions.

$$X_{normalized} = \frac{X_p - \mu(X_p)}{\sigma(X_p)} \tag{4.8}$$

In the following sections X refers to the normalized feature vectors and Y to the one-hot encoding of the prediction labels. The dimensions of X and Y are respectively (samples x sequence length x feature vector dimension) and (samples x sequence length x number of labels).

4.5 Regularization

The family of functions which define neural network topologies are generally non-convex [1] and therefore the gradient descent algorithm does not give us a formal guarantee of convergence to the optimal solution. For this reason it is important to aid the optimization process by imposing regularization terms.

Regularization refers to introducing additional constraints, potentially derived from problem specific information, in order to solve an ill-posed problem or avoid over-fitting [1].

More specifically it helps reduce the generalization error, the gap between the error observed while adjusting the parameters to the training set and the validation set. At the same time by introducing additional information it increases the convergence rate [1].

4.5.1 Drop-out layer

Drop-outs are an effective solution to mitigate one problem in the optimization process, over-fitting. Over-fitting in this context refers to the final model parameters modelling the samples too closely which results in the model not generalizing to unseen samples.

Adding drop-out layers to the network corresponds to randomly setting an entry of an internal layer to zero during training. This procedure has the effect of inducing redundant thin networks in the topology as the network seeks to find alternative more stable representations for internal vectors which might loose information [2].

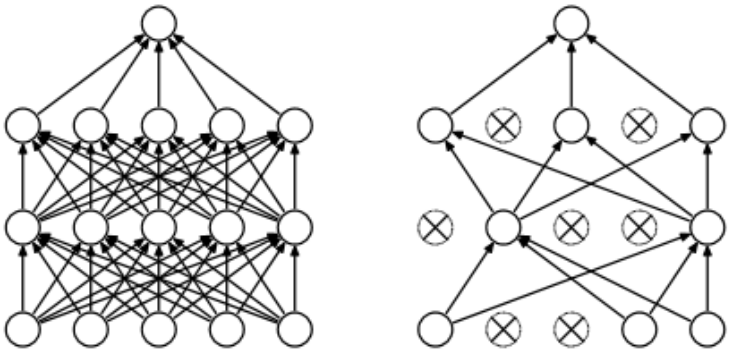


Figure 4.6: Effects of drop-out on the network topology [2]

In practice this means that internal network structures form, these get averaged at the output which has the side effect of averaging out the predictions, reducing the variance and avoiding over-fitting. For the more experienced reader, this creates an internal ensemble of sub-networks.

Empirically, the change in network structure during the gradient descent iterations, while improving the generalization of the model also helped the parameters escape local optima.

4.5.2 L2 regularization

Another method that is employed to avoid over-fitting is the addition of a L2 weight regularization term which penalizes parameters with a high L2 norm.

$$RegularizedCost = Cost + \beta \cdot \|\theta\|_2^2 \tag{4.9}$$

This L2 regularization term is sometimes also called weight decay term because the gradient of the regularizer has the effect of decreasing the parameters magnitude.

Adding the regularization term improves the model's generalization as it encourages a low variance which helps avoid over-fitting.

4.5.3 Norm stabilization

Depending on how you train a Recurrent Neural Network (RNN) it can tend to have fast changes in the output variables. A possible strategy for stabilizing the output is penalizing the difference between consecutive outputs. This has the effect of leading to parameters whose model output does not change abruptly.

In practice this implies the addition of a term which measures the L2 distance between consecutive internal state vectors.

A similar problem is the decrease in activity in internal layers. A common solution to this problem is similar to the output stabilization and is called norm stabilization [63]. In norm stabilization, instead of comparing the values directly, the distance between consecutive output norms is penalized.

4.6 Convergence

The above mentioned techniques are important for ensuring a low generalization error (difference between training and validation set loss), this section looks at important steps to aid the gradient descent optimization algorithm convergence to a solution (not necessarily optimal since the cost function is non-convex and therefore we have no optimality guarantees).

4.6.1 Gradient norm clipping

When optimizing strongly non-linear functions there is the problem of abrupt and large changes in derivatives. A large gradient can have the effect of drastically changing the parameters and hindering the convergence. A common technique for dealing with this problem and which can greatly improve convergence rate is the gradient clipping method.

The gradient clipping method deals with large gradients by setting an upper bound on the gradient norm and scaling the gradient norm down to the threshold if it surpasses this limit [64].

Empirically this has the effect of avoiding large parameter changes which might hinder convergence to the optima in highly non-linear parts due to exploding gradients.

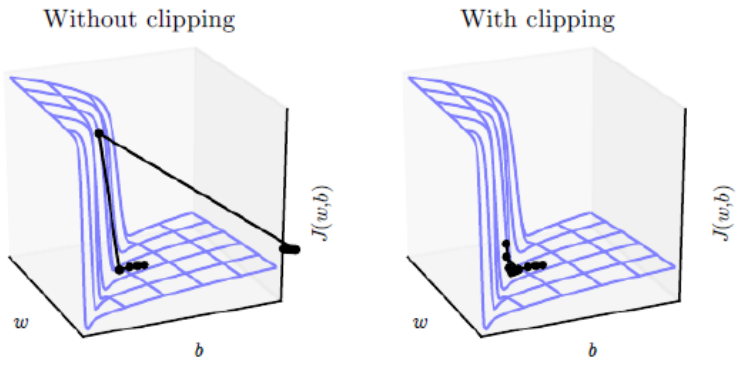


Figure 4.7: Gradient norm clipping

Algorithm 2 Gradient Norm Clipping

procedure GRADIENTNORMCLIPPING($\hat{g}$)
  if $\|\hat{g}\| \geq threshold$ **then**
    $\hat{g}\frac{threshold}{\|\hat{g}\|} \rightarrow \hat{g}$
  end if
return $\hat{g}$
end procedure

4.6.2 Xavier weight initialization

Another important factor of the optimization problem is the starting region in the parameter space, also called the weight initialization. A recent paper by Glorot et al. [65] introduced a novel initialization method which guarantees the input signal is not attenuated or grows too large when it propagates along the network [65]. Empirically this initialization performs better than randomly initializing the weights.

This initialization, also called Xavier initialization, suggests the sampling of the layer weights in a way that preserve the input variance and therefore the information flow along the network topology.

$$Y = \sum_{Nin} W_i X_i \Leftrightarrow Var(Y) = Var\left(\sum_{Nin} W_i X_i\right) \tag{4.10}$$

Taking as an assumption that the input variables are uncorrelated.

$$Var\left(\sum_{Nin} W_i X_i\right) = \sum_{Nin} Var\left(W_i X_i\right) \tag{4.11}$$

Assuming the inputs and weights are uncorrelated and that the variance of inputs and weights is identically distributed.

$$\sum_{Nin} Var\left(W_i X_i\right) = \sum_{Nin} Var\left(W_i\right) Var\left(X_i\right) = N_{in}\left(Var(X_i) Var(W_i)\right) \tag{4.12}$$

The input variance, $Var(X_i)$, is preserved if the weights have a variance, $Var(W_i)$ given by $1/N_{in}$The proposed sampling distribution is a Gaussian with zero mean and a variance given by $2/(N_{in} + N_{out})$ which corresponds to the geometric mean between preserving the forward, $1/N_{in}$, and back propagation, $1/N_{out}$, pass variance.

4.7 Conclusion

In this chapter we introduced the thesis' problem statement, explained the proposed model's topology and guaranteed its feasibility handling the implicit complexity issues using a pruning method.

After introducing the model we formulated the parameter search as the minimization of a so called cost function. The cost function, being non-convex, has no guarantee of convergence to the optimal parameters.

To aid the convergence to reasonable parameters (generalizable and stable internal state) we introduced problem specific regularizer such as drop-out layers and state stabilization.

At the same time we introduced two techniques to aid in the convergence of the highly non-convex cost function.

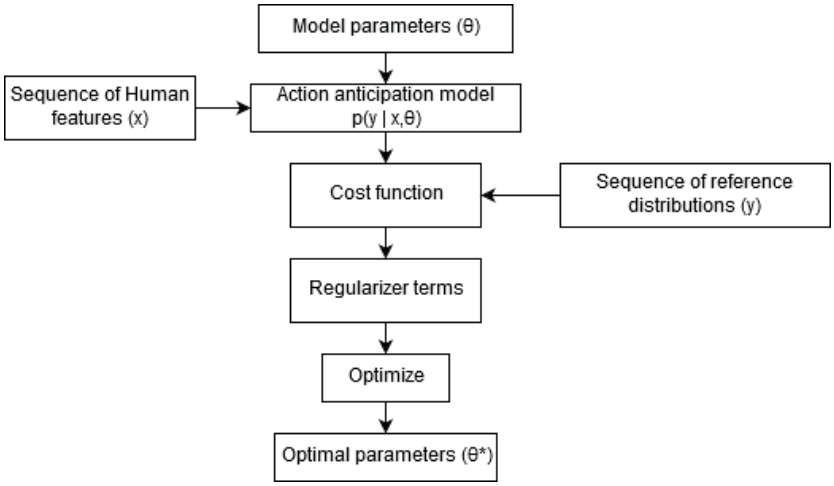


Figure 4.8: Overview of the model and parameter selection process.

After introducing the model, the next section demonstrates how the prediction of multiple and variable length action sequences is crucial to estimate the expected future reward in a human-robot cooperation scenario, the focus of this thesis.

5

Application scenario: Decentralized Markov Decision Process

Contents

Anticipating a set of possible future actions is important in cooperative assembly scenarios where two agents work together in a fast paced joint action setting. This scenario aims to clarify the importance and some caveats inherent to the action prediction problem in human robot cooperation scenarios.

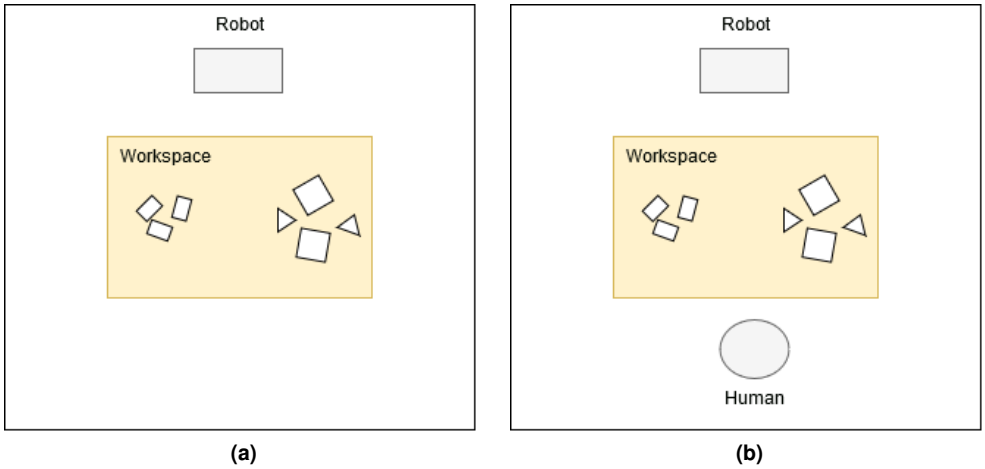
This chapter we start by defining the cooperation scenario through an example, next we formulate the setting as a Decentralized Markov Decision and demonstrate how the Action Anticipation enables us to estimate the expected future reward.

This chapter is theoretical in nature and seeks to demonstrate the importance of anticipating the cooperation partner's actions to estimate the expected future reward in a joint action scenario.

5.1 Overview

A possible scenario for joint action is cooperative assembly, where two agents work together in a fast paced environment to accomplish a shared goal.

While the nature of the task at hand is not important, one could imagine a concrete example, e.g., a automotive assembly task. One such scenario is illustrated in Figure 5.1b. In this setting, the human and robot work together in a shared workspace towards a final shared goal.



(a) (b)

Figure 5.1

A basic formulation of a Markov decision process involves the interaction of the robot with the environment (fig. 5.1a). In joint action we have an additional factor which is the human action (fig. 5.1b) that has to be taken into account.

Both the human and robot interact and influence the environment Markov process, to allow for a natural and efficient interaction between the human and robot it is important for the robot to understand and anticipate the human's actions.

This setting (Fig. 5.1b) is called a Decentralized Markov Decision Process (Dec-MDP) and was first introduced in [66] as an extension to the Markov Decision Process scenarios where more than one agent may be acting upon the environment.

Definition 5.1.1. Decentralized Markov Decision Process (Dec-MDP)

A Dec-MDP is a tuple $< S, A_s, T, R >$ with:

- Set of possible world states, S
- Set of joint human and robot actions, A_s
- Transition probabilities between states over joint actions, $T : S \times A_s \times S \rightarrow [0, 1]$
- Imediate joint reward function $R(S, a_H, a_R)$: $S \times A \rightarrow \mathbb{R}$

For the sake of this example, the world state is a set of pre-conditions, C, the transition probabilities a set of action-effect axioms, A, and a state dependent joint reward function, R.

Both the human and the robot choose their actions, a_H and a_R respectively, independently maximizing a joint reward function $R(S, a_H, a_R)$ and influencing the environment Markov process. We assume a leader-follower paradigm where the robot chooses his action dynamically as a function of the possible next human actions and the humans actions are not influenced by the robots action choice.

In this section we defined the cooperation scenario more formally, the next section looks at this formal setting from the perspective of the joint action framework.

5.2 Shared representation

The shared representation in this setting is the visibility of the joint workspace, S, and the knowledge of the joint goal.

The joint goal can be independently decomposed into a sub-goal plan through a partial order plan, ideally the robot and human share the same sub-goal plan, that is, both agent's have knowledge of the tasks which need to be accomplished.

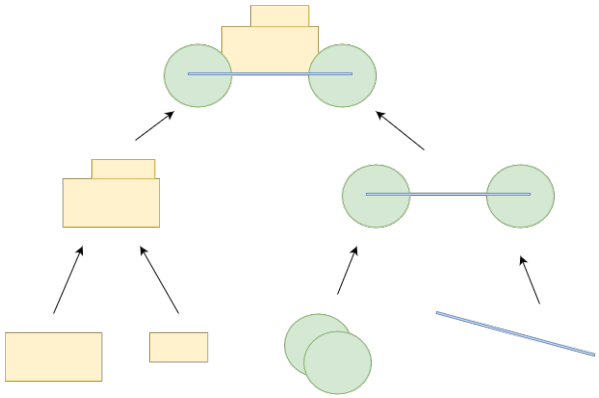


Figure 5.2: Partial order planning decomposition of the action goal.

The joint reward function, $R(S, a_H, a_R)$, is defined as a function of the sub-goal completion. It is therefore important for the robot to understand the human intent and act in a way that maximizes the joint reward (not completing the same action goal as the cooperation partner)

5.3 Anticipation

In the previous section we saw that a simple MDP models an agent's interaction with the environment and that the Dec-MDP was introduced as a natural extension to this setting when the agent does not act alone on the environment.

We reached the conclusion that in a Dec-MDP in addition to the interaction with the environment in the form of transition probabilities, as is the case in the simple MDP, it is necessary to model the cooperation partner's actions. This model of the cooperation partner's actions is similar in nature to the ability that has been called *mind reading mechanism* in the human cognition. It is through this mechanism that the robot becomes empathic, starts to understand the human agent's temporal behaviour and learns to model his actions.

Seeing the human reaching for one set of objects the robot should understand which sub goal the human is attempting to complete. That is, more formally, given the past human history the model returns a distribution over possible future action sequences, $P(a_H|a_{Hp})$. The next section formulates the expected reward as a function of this probability distribution. The probability distribution is estimated using the model introduced in the previous section.

Before moving on to the next section, it is important to remind the reader that the estimated sequences consist of the K most probable future action sequences, understanding this format will be important for the next section. Where x_i is one time instant in a sequence of past time instant representations (e.g. the skeleton joint positions at a past time instant), x, N the number of steps the model predicts into the future and K the number of action sequences the model estimates.

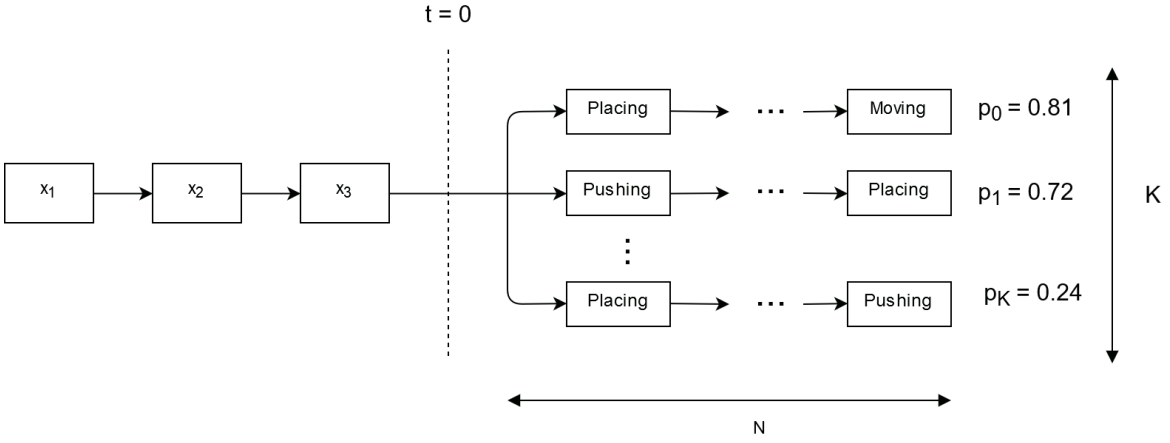


Figure 5.3: Action anticipation format.

5.4 Anticipative action

The anticipative action module is responsible for selecting an action that maximizes the joint reward. This section formalizes the decision process and introduces the action anticipation in the reward estimation.

Given an initial state, S_0, and an action sequence, A_s, which is a series of action pairs (A_H, A_R),

the total reward after N steps is given by (5.1). Where S_i, $A_{H,i}$ and $A_{R,i}$ correspond respectively to the state and the action performed by the human and robot at time step i and N the length of the sequence.

$$R(S_0, A_s) = \sum_{i=0}^{N} R(S_i, A_{H,i}, A_{R,i}) \tag{5.1}$$

Since the behaviour of the human is non-deterministic from the perspective of the robot the future reward associated to a chosen action sequence can only be determined stochastically as an expected reward.

This expected reward of a chosen robot action sequence, A_R, is given by the marginalization of the stochastic reward along the space of possible human actions (5.2), where V is the dimension of the action vocabulary, $f_{H,i}$ represents the human action distribution at time step i, a_j an action in the human action vocabulary and N the action sequence length.

$$\mathbb{E}\left[R(S_0, A_s)\right] = \sum_{i=0}^{N} \sum_{j=0}^{V} R(S_i, H_j, A_{R,i}) f_{H,i}(H_j) \tag{5.2}$$

Estimating the complete future action distribution would require expanding all possible action sequences which is NP hard and computationally intractable. The beam search introduced in the previous section enables us to approximate the future action distribution.

Considering the set of most probable action sequences returned by the multiple action sequence prediction as representative of the future human behaviour, the marginalization and therefore the expected reward can be approximated by summing and weighting each human and robot action sequence's reward by the action sequence probability (5.3).

Increasing the number of beams in the decoder, K, approximates the reward better but is computationally more demanding. Here $p(b_k)$ represents the beams probability, $a^k_{H,i}$ the action performed by the human at time step i in the beam k and $a_{s,i}$ the action performed at step i in the robot action sequence a_s.

$$\mathbb{E}\left[R(S_0, a_s)\right] = \frac{\sum_{k=0}^{K} \sum_{i=0}^{N} \left[R(S_i, A^k_{H,i}, A_{R,i})\right] p(b_k)}{\sum_{k=0}^{K} p(b_k)} \tag{5.3}$$

As the beam count tends to the total number of possible action sequence combinations this expression approximates the expected reward (5.2). Choosing an action sequence corresponds to maximizing this expected reward (5.4).

$$a^*_R = \arg\max_{a_R} \mathbb{E}\left[R(S_0, A_s)\right] \tag{5.4}$$

Maximizing the general expression is identical to the problem faced in the classic Markov Decision Process, we were able to reduce the Decentralized Markov Decision Process to a Markov Decision Process anticipating the human action and admitting a leader follower paradigm.

Optimizing the policy as a function of the predicted human behaviour induces a coupling between the agents similar to [67] where an improvement in task completion time was observed.

5.5 Conclusion

In this chapter we went from a broad overview on the general cooperative manufacturing setting to more detailed topics like the distinction between a Markov Decision Process and a Decentralized Markov Decision Process.

We introduced the additional challenge, of prediction the human behaviour, we face when solving the decision problem in a Dec-MDP setting.

We concluded the chapter by effectively dealing with the additional challenges we face in a Dec-MDP by approximating the stochastic future reward with the predicted action sequences and their respective probabilities.

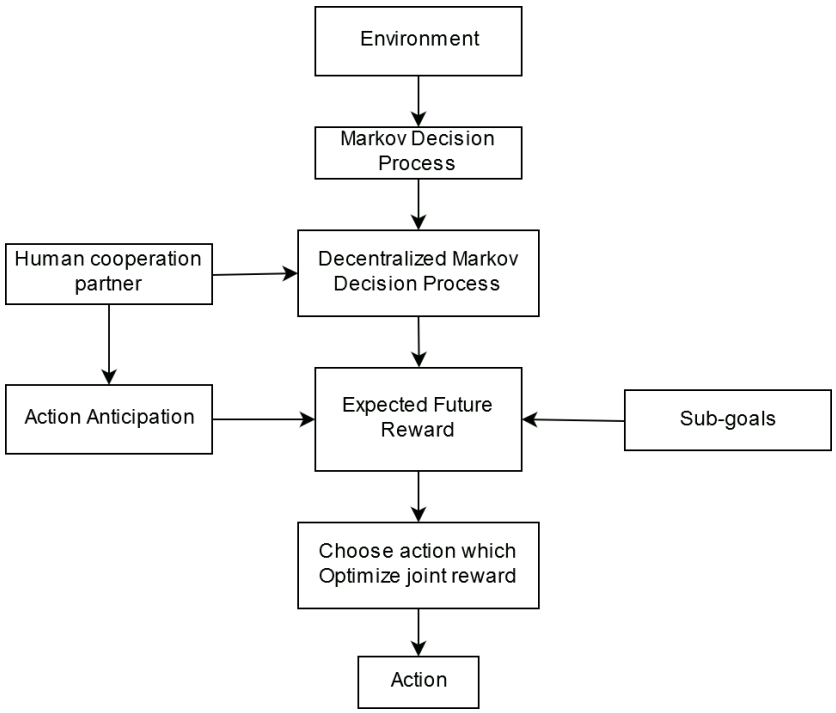


Figure 5.4: Overview of the proposed solution to the Dec-MDP problem.

6

Results

Contents

The previous chapters went from a broad overview of cooperation to more specific details like the importance of action anticipation and the so called "mind reading mechanism" through which humans reason about other agents' intent. We mentioned the importance of different non-verbal cues but never qualitatively compared them, in this section one of the things we will study is the relative importance of gaze and movement cues.

We also looked at different prediction models to model the so called "mind reading mechanism" and introduced the proposed action anticipation model. This action anticipation model is in the form of a recurrent neural network topology which is able to predict multiple variable length sequences and their respective probability. The model has different parameters which we will study in more detail in this section.

We finished with demonstrating theoretically how the predicted human action sequences enable us to estimate the expected future reward in a cooperation scenario. When deducing the formula for estimating the future reward we mentioned briefly that when the number of predicted sequences tends to the number of possible action sequence combinations the estimated reward tends towards the expected future reward. This section looks closer into this claim.

The section starts by describing the datasets used to evaluate the action anticipation model, then, we will look more closely at the relative importance of non-verbal cues and at the different model parameters, next we will list quantitative metrics of the models performance and end with studying the action sequences cumulative probability.

6.1 Action Anticipation Dataset

The multiple action sequence prediction model is evaluated on the CAD120 dataset (Fig. 6.1) which was developed by the Group of A. Saxena at the University of Stanford [68]. This dataset consists of a human actor's skeleton movement while performing a sequence of actions like "pouring" and "eating".

This dataset consists of body posture features in a binary feature format together with the respective action labels at a sample frequency of 5Hz. In this context body posture features refers to the skeleton joint angles.

The labels are a total of 10 with the names: "reaching", "moving", "pouring", "eating", "drinking", "opening", "placing", "closing", "scrubbing" and "None". The total number of action sequences is 120 and the sequences have an average length of 25 time steps.

This dataset is of especial interest since it covers the scope of action sequences and it is not limited to one action per video segment. It is one of the few datasets which has varying order of action sequences. One limitation to this dataset is that it is missing a very important human non-verbal cue, gaze. The potential impact of this limitation will be studied in more detail in the next section.

(1) (2) (3)

Figure 6.1: Datasets. CAD120 RGB-D motion dataset sample image.

6.2 Gaze feature importance

In previous chapters we have shown the importance of gaze features to predict human intent, nevertheless, the CAD120 dataset only contains body posture and does not have any gaze information about the human. This is a common limitation in existing action labelled datasets [68, 69] and its potential impact will be the focus of this section.

One of the most salient factors in the models performance are the model's features. That is, how the model perceives the scene and the surrounding agents. This section seeks to introduce a qualitative metric for the relative gaze and body posture cues importance, two commonly used features in non-verbal communication [15]. Hereby we seek to understand the impact the absence of gaze information has on the action anticipation model.

There are different feature selection methods which can be categorized into *filter*, *wrapper* and *embedded* classes [70]. Since the relation between the features (gaze and body posture) is unknown, it is assumed to be non-linear in nature. Following the non-linearity assumption the focus of this section will be on the *wrapper* class of feature selection methods.

This class of methods captures non-linear relation between the variables through a black-box model. It starts by training the model on subsets of the feature space and then ranks the features according to the models' accuracy[70].

For the sake of this experiment, the black-box model is the intention recognition model, a Recurrent Neural Network (RNN) sequence to sequence model. The structure of the model is defined by an embedding layer which at every step transforms the feature vector into an intermediary representation acting as an input to the model's RNN. For every input this RNN returns a discrete distribution over intentions. This distribution is obtained by projecting the recurrent neural network's internal state and normalizing it through a softmax layer.

The prediction accuracy of the model with and without a given feature can be considered a proxy for the feature's added information.

The feature importance is evaluated on a combined gaze and skeleton dataset which was acquired and published in the ISR Vislab ACTICIPATE[1] project.

The ACTICIPATE project is a currently active project at the Vislab research group. The project seeks to understand dyadic interactions between humans and robots during task execution in order

[1]The ACTICIPATE dataset can be downloaded from the following web page: http://vislab.isr.tecnico.ulisboa.pt/datasets/

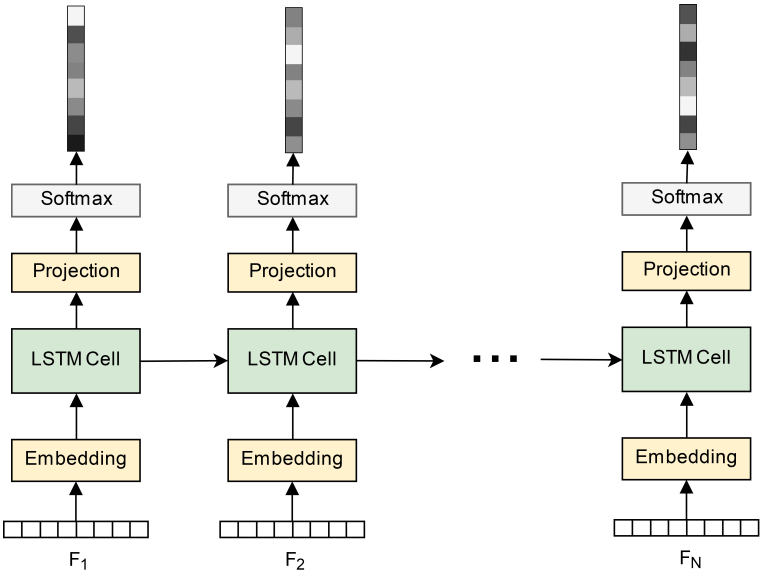


Figure 6.2: Intention recognition model. This model maps a sequence of input features to a sequence of discrete distributions over the action vocabulary.

to design systems that can share workspaces and co-work with humans.

This dataset consists of a human actor's gaze and skeleton movement while performing either one of six actions (Place Left, Place Center, Place Right, Give Left, Give Center, Give Right).

(1) (2) (3)

Figure 6.3: Datasets. ACTICIPATE joint gaze and body posture dataset.

This dataset was recorded using the Optitrack motion capture system, and Pupil Labs binocular eye gaze tracking system, synchronised at a 120Hz frequency. The total number of action sequences is 120. The sequences have an average length of 220 frames. Every sequence corresponds to one action and is labelled accordingly.

The first experiment seeks to validate the models correct behaviour. In this setting, we train the model on the combined body posture and gaze features using the place position number as a reference categorical label.

Each segment of the dataset consists of the sequence of movements and gaze patterns it takes the human actor to take the ball from the resting position to the desired target location numbered from 1 to 6. The expected behaviour is for the model to gather information during the movement and predict the right action target as soon as possible. Predicting the right action target means the probability distribution over all the locations converges to a single action target 6.4.

What we observe is that, as the movement progresses, the model receives more information and identifies the intention, correctly converging to the true label, 5. The movement takes 220 frames (about 2 seconds), the model is able to predict the intention target after seeing less than half of the total trajectory, about 100 frames, confirming that it yields the expected behaviour.

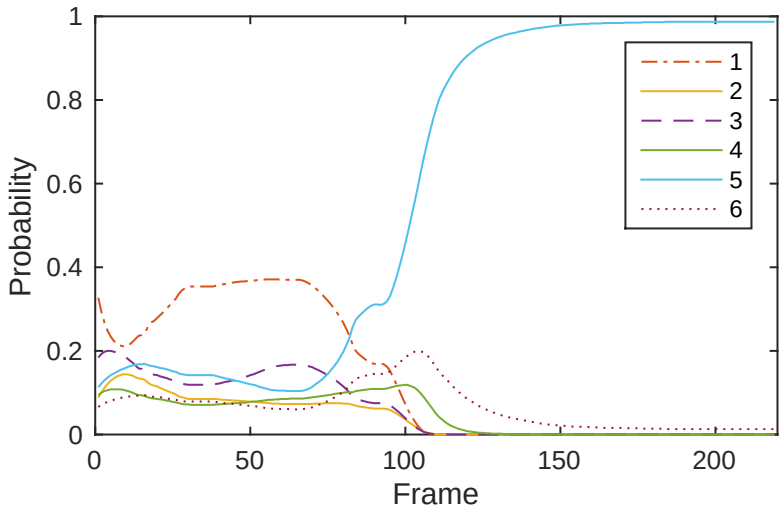


Figure 6.4: Action distribution temporal dynamic. Action probability temporal evolution. The model starts with uniform probability and after about 100 frames converges to the correct label.

The second experiment is concerned with quantifying the relative importance of the different non-verbal cues in predicting human intent. In this experiment the model is trained on two sets of features: (i) combined gaze and body posture cues, and (ii) body posture only.

Figure 6.5) shows the model performance under these two conditions. The relative difference between the performance of the model trained only on the body posture and the model trained on the combined body posture and gaze features is a quantitative metric for the gaze features importance in anticipating human actions.

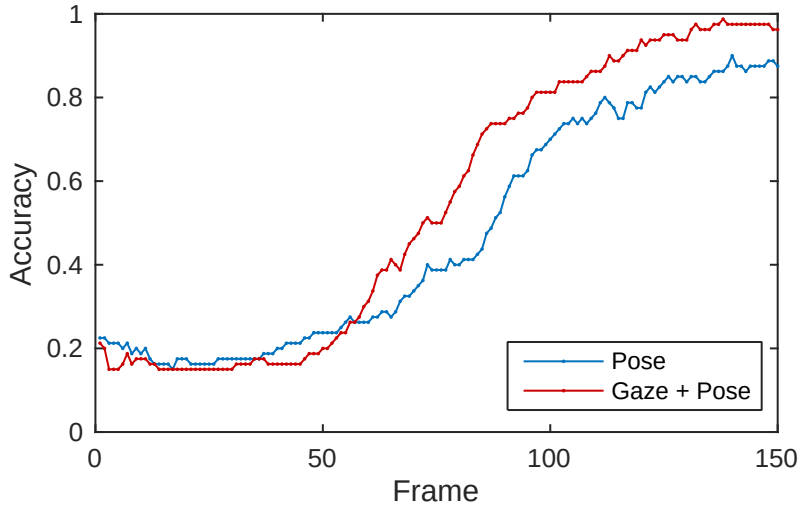


Figure 6.5: Gaze and body posture accuracy. Accuracy of a model trained on (i) gaze only features, and (ii) trained on combined gaze and body posture (pose) features.

The difference in accuracy between the two sets of cues hints at the importance of gaze. Despite the model performing similarly with and without gaze, the results show that gaze has an important role in early prediction of human activity.

The model trained on both gaze and body posture cues predicts the correct action 92 ms before the model with only body cues. An interesting result is that this delay in predictive ability coincides with the range of delays between eye and hand movement observed in research on eye-arm movement coupling [71].

As the relative difference in accuracy between using only body posture features and using gaze in addition to the body posture features is about 11%, we can say use of body posture features without gaze information for anticipating human action is supported by this dataset.

6.3 Performance metrics

The previous section looked at the relative importance of gaze and body posture features in the action anticipation task establishing an empirical base for using body posture in the action anticipation problem.

Following this line of thought, having established the role of body posture features, we will now train and evaluate the model on the action anticipation dataset introduced earlier (CAD120).

This section looks at defining performance metrics for evaluating the model. This will be important for the next section where we analyse the influence of the model parameters on the final accuracy.

The model takes the body posture features, observed over three time steps, as input in order to predict future actions as accurately as possible. This section introduces quantitative metrics for the models performance evaluated on the CAD120 dataset.

For the sake of performance measurement, out of the set of action sequences returned by the anticipation model we will consider the most probable sequence as the predicted sequence.

6.3.1 F1 Score

Since the action labels have a slight class imbalance a simple accuracy measurement (dividing the correctly predicted samples by the total number of samples) does not calculate accurate performance metrics [72].

To circumvent this issue an alternate metric, the F1 score [72], which is more robust to class imbalance will be used. The F1 can be seen as the geometric mean between the models precision and recall, two key indicators of the models performance:

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} \tag{6.1}$$

To guarantee validity of the final results the F1-score is evaluated on a four fold cross validation scheme. The cross validation scheme takes 20% of the data as an unseen test set and trains the model on the other 80% of the data, this would be one fold. Folding the data again would mean

returning to the original dataset and randomly sampling another 20% of the dataset and repeating the earlier procedure. A four fold cross validation scheme means repeating this procedure 4 times.

In practice this means the results become more robust to accidentally skewing the distribution by sampling the test set in a very unfortunate way.

The final score then becomes the average over the folds results. As it might happen that we sample folds without instances of a specific action label, the F1 score is calculated directly on the true positive, false negative and false positive rate (6.2).

$$F1 = \frac{2 \cdot TruePos}{2 \cdot TruePos + FalseNeg + FalsePos} \tag{6.2}$$

For the sake of performance evaluation the predicted sequence is the most probable sequence out of the set of action sequences returned by the action anticipation module.

Training the model on a 8 step prediction length (this parameter is studied in more detail in the next section) it is possible to observe the following sequence of F1 scores.

Prediction step	Anticipation F1 (%)	Baseline - random (%)
1	74.49	10
2	63.40	10
3	54.23	10
4	47.36	10
5	36.39	10
6	18.63	10

Table 6.1: F1 Accuracy calculated by comparing the most probable predicted sequence with the reference label sequence element wise.

The accuracies are calculated by comparing the predicted and reference label sequence element-wise.

Here we can observe a high accuracy in the first prediction steps and a naturally decaying prediction accuracy as the model predicts further into the future and the uncertainty increases. As a baseline the random classifier was selected, this classifier outputs a random prediction at every step.

This result is comparable to related work in the same dataset, more specifically the recent work by [19] shows a slightly better performance and our model is about twice as accurate as the performance obtained by [68] (the original paper where the dataset was first made publicly available).

6.3.2 Confusion matrix

An important metric to understand a classifier's performance is the confusion matrix. This table summarizes the four main types of classifications: false positives (FP), true positives (TP), true negatives (TN), false negatives (FN).

This table is constructed by comparing the reference labels with the labels of the first prediction step of the most probable action sequence. Here the vertical dimension represents the real label and the horizontal the predicted.

60	0	0	0	0	0	0	0	0	0
0	27	0	1	0	0	0	0	0	0
1	4	39	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	0	0	0	0	0
0	0	0	0	7	1	0	0	0	0
0	0	1	0	0	0	11	1	0	0
0	0	0	0	1	0	0	0	0	0
17	2	0	0	10	0	0	2	0	0
0	0	0	0	2	0	0	2	0	2

Table 6.2: Confusion matrix calculated for the first prediction step of the most probable action sequence. Vertical direction is the reference label and horizontal the predicted.

The labels correspond respectively to the actions: "reaching", "moving", "pouring", "eating", "drinking", "opening", "placing", "closing", "scrubbing" and "None". The vertical axis represents the true class (top to bottom), while the horizontal axis represents the predicted class (left to right). The elements belonging to the diagonal correspond to correct sample classifications (TP) while the others are misclassification of different types (FP, FN).

More specifically, classification errors related to the label "scrubbing" (8) and "opening" (4) are specially interesting. Both, scrubbing and opening, have an abnormal number of false positive predictions.

The firs, "scrubbing" is related to an error along the rows, which means the classifier outputs a different label for the true label scrubbing, the same behaviour has been observed by Koppula using the same dataset [17] and seems to be related to a lack of information for this label in the dataset. This label's main confusion is with the action "reaching" and "drinking".

The second, "drinking", is an error along the column which means the classifier outputs the label "drinking" too often, this might be related to a small class imbalance or similarity. This action is misclassified mainly with action "scrubbing" and "opening". From a qualitative analysis this seems to be related to a similar problem as the "scrubbing" action.

6.4 Hyper Parameters

While the previous section validated the use of body features as a means to anticipate human intent, this section looks at the second part of the equation, the action anticipation model's hyper parameters.

Hyper parameters are the parameters which define the structure of the model, in this section we will look at two of them: 1) Train prediction length and 2) Context vector dimensionality.

6.4.1 Train prediction length

Recalling, the structure of the decoder is defined by a LSTM cells, the cell iterates upon the current state and returns a new output and the next state. The first internal state is given by the context vector (encoding of past information), the next states are obtained by iterating upon this first state.

As there are no inputs or any other information, the only information available to generate the sequence of future outputs is the internal state. In this section we will look at how training the model on longer sequences improves the short range accuracy (Fig. 6.6).

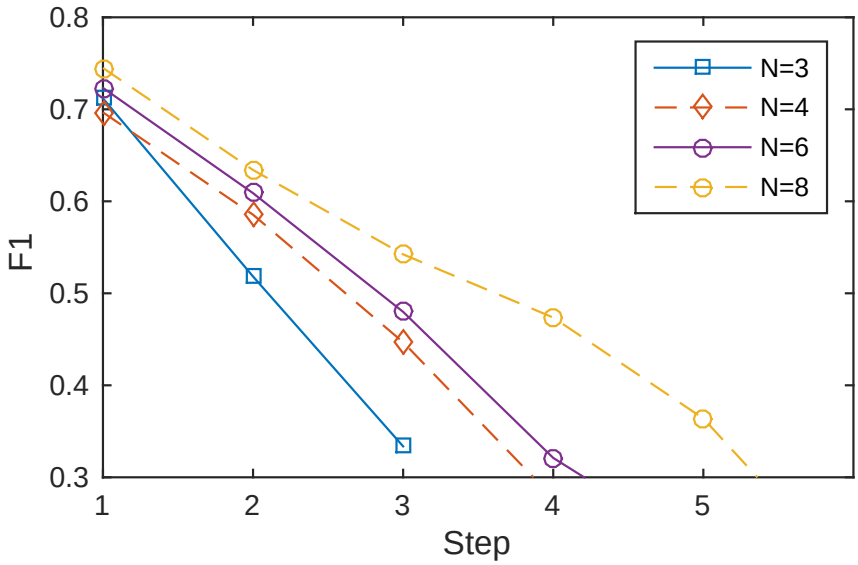


Figure 6.6: Accuracy as a function of prediction length. Prediction accuracy across time steps is positively correlated with the prediction length the model is trained on. ("N" corresponds to the prediction length used for training the model, and "Step" the position in the predicted sequence.)

In this setting the future sequence of actions corresponds to the sequence of highest probability returned by the anticipation model. The label sequence is compared, step by step, with the respective reference label sequence to obtain the F1 accuracy.

While the model is dynamic in its ability to predict variable length action sequences, we can see that the accuracy of short range action sequence prediction is influenced by the prediction length the model is trained on.

This correlation is related to the ability of the decoder to manage its internal state. When the network is trained on a long future action sequence, it learns to iterate non-destructively on the decoder's internal state, predicting longer sequences with more accuracy.

From this experiment we conclude that, even if we are only interested in short range predictions, the accuracy is influenced by the prediction steps we train the model upon and therefore it is important to train the model to predict longer time sequences than are necessary in practice.

6.4.2 Context vector dimensionality

In the anticipation model, the dimensionality of the context vectors, the bottle-neck between the encoder and decoder, can be used as a parameter to define the model's capacity.

This experiment is concerned with understanding how this parameter affects the models convergence. The model is trained on a context vector with dimension 20 and 80.

Looking at the anticipation network topology it is possible to see that the context vector is an informational bottleneck. Looking at the problem from a trajectory clustering perspective the first part of the network identifies the current action trajectory and the second expands the future steps of the trajectory. Here the context vector dimensionality defines the space of possible trajectories. Increasing this space more trajectories can be captured, reducing this space we limit the representational power of the network.

Increasing the context vector dimension increases the model capacity this capacity is related to the generalization error [1].

Increasing the capacity increases the generalization error. Increasing the generalization error makes the model prone to over fitting to the training set and not generalizing to new samples (Fig. 6.7).

Hence, the context vector dimensionality acts as a regularizer of the model.

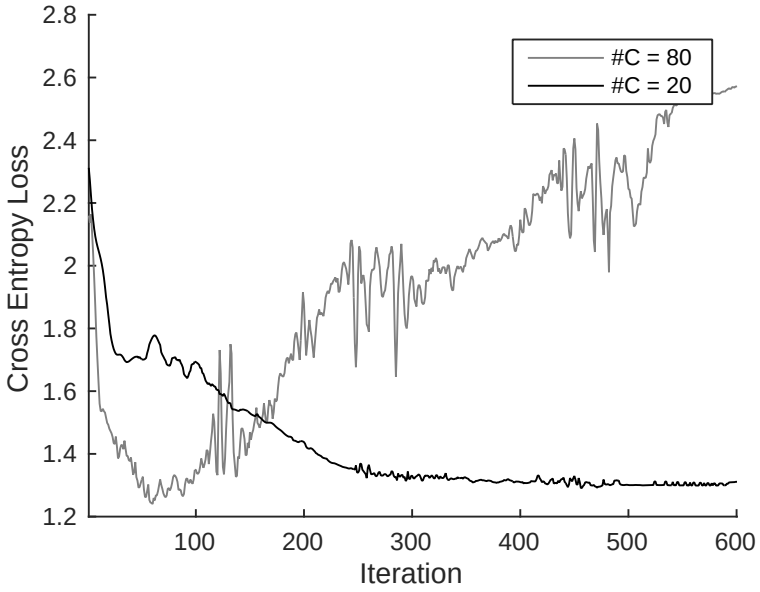


Figure 6.7: Validation loss as a function of the context dimensionality. The iteration represents the number of training steps while #C represents the dimensionality of the context vector parameter. As the dimensionality parameter is increased, the network starts to overfit to the training data.

6.5 Action sequences

So far we looked into the relevance of using only body posture features for the action anticipation problem and looked how the model's parameters influence its performance.

In the previous chapter we defined the expected reward as a function of the sequences predicted by the action anticipation model and their respective probabilities. Nevertheless, we did not have any way of showing that the most probable sequences are indeed representative of the possible future human actions.

For this reason in this section we will look into the number of beams (action sequences) parameter. Recalling, the number of beams parameter defines the number of sequences the sequence expansion problem is pruned to.

These beams determine the space of action sequences the model is able to capture (Fig. 6.8). The cumulative sum of the beams' probabilities is a measure of the solution space we are able to cover with a given number of beams.

Here the action sequence probability, $P(s)$, is calculated by factoring the joint probability, into the joint probabilities between consecutive actions (eq. 6.3).

$$P(s_j) = P(a_0|c_0) \prod_{i=1}^{N_S} P(a_i|a_{i-1}, c_i) \tag{6.3}$$

Knowing the sequences probability is independent, the disjoint beams probability is given by the sum of the beam search sequences probability.

$$P(s_1 \cup s_2 \cup \quad ... \quad \cup s_N) = \sum_{j=0}^{N_B} p(s_j) \tag{6.4}$$

The space of possible solutions grows exponentially with the number of prediction steps. While a beam width of 11 beams is able to capture 100% of the outcome probability space in a one-step ahead prediction scenario, the same number of beams only captures around 75% of the outcome probability space in the two-step ahead prediction scenario. As the solution space grows, a fixed number of beams captures a cumulative probability outcome space that decays with the number of prediction steps.

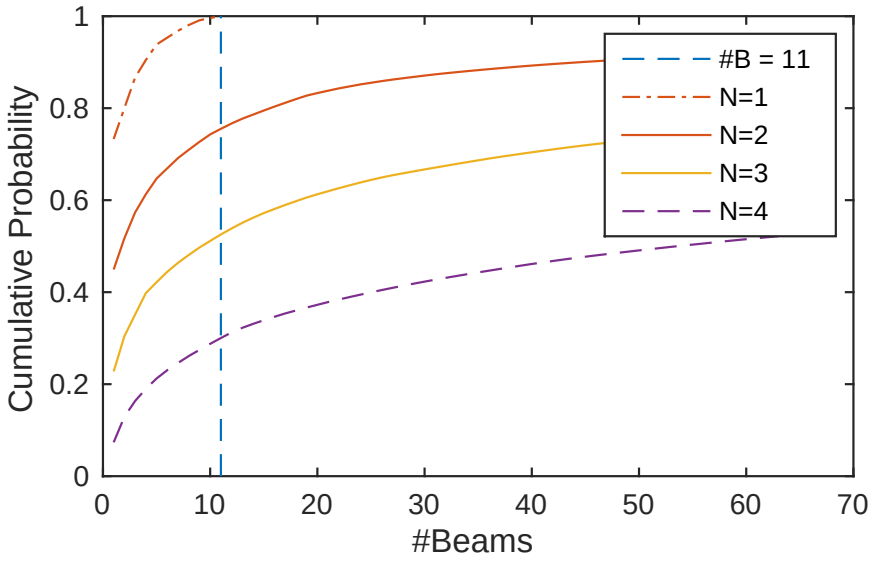


Figure 6.8: Beam cumulative probability. Cumulative probability of the outcome space the model is able to capture. "N" represents the length of the predicted trajectory, and "#Beams" the length of the predicted action sequences.

From the cumulative probability we can see that the action sequences follow a long tail distribution, that is, a set of the top more probable action sequences has a large disjoint probability. This indicates

that the top more probable action sequences are indeed representative of the future human actions and the expected future reward can be approximated using the output of the action anticipation model.

6.6 Conclusions

In this chapter we followed a structure similar to the thesis, revisiting open questions and introducing additional performance metrics when necessary.

The first open question is concerned with the action anticipation dataset and an earlier comment about the importance of gaze clues. Existing action labelled datasets do not contain any gaze information, the natural open question is: How important is gaze the action anticipation problem? The experiments on a combined gaze and body posture dataset showed the importance of gaze for early prediction nevertheless the model's final accuracy with and without gaze was similar, supporting the value of body pose features for the action anticipation problem.

The second open question concerned the models parameters, that is, how does the model performance change as a function of the parameters? The experiments showed a clear correlation between the accuracy and the prediction length the action anticipation is learned on, furthermore we were able to effectively use the context vector dimension as a regularizer for the model's capacity.

The third and last open question, concerned the estimation of the expected reward. In the theoretical formulation of the cooperation scenario we argue for the approximation of the reward through considering the set of predicted action sequences representative of the humans future actions. Does this hold true in this case? The experiments showed that the predicted action sequences follow a long tail distribution which indicates that the set of most probable sequences are indeed dominant in terms of cumulative probability and the approximation is valid.

7

Conclusions

Contents

Close human-robot cooperation is an important research topic in a world with a growing number of automated tasks. Developing empathy mechanisms in artificial systems for understanding human movement is a key milestone in both safety and task cooperation efficiency.

We started this thesis with a broad overview of cooperation by introducing the joint action framework. Through this initial view into cooperation we realized the importance of action anticipation and established the general framing of this thesis.

After this initial look into cooperation and the importance of action anticipation, we briefly looked at how the human cognition separates inanimate (Theory of Body) from animate (Theory of Mind) entities and how it reasons about other goal driven agents through the so called "Mind Reading Mechanism". In the end of this first chapter we showed the importance of non-verbal cues to estimate the human intent.

Having established a good theoretical background and the relevance of a human action anticipation model based non-verbal cues, we introduced a recurrent neural network topology designed to predict multiple and variable length future action sequences.

Predicting action sequences introduces combinatorial complexity issues which were successfully mitigated using a pruning method. The end result is the prediction of a collection with multiple and the most probable action sequences.

Having modelled the relation between non-verbal cues and future human action sequences, we demonstrated the theoretical value of predicting multiple and variable action sequences in a human robot cooperation scenario modelled as a Decentralized Markov Decision Process.

We introduced this scenario by demonstrating the difference between a one agent Markov Decision Process and a multi agent Decentralized Markov Decision Process showing the need to model the human cooperation partner to estimate the future reward.

We studied how different training procedure and parameter combinations affect the model performance. All tests were carried out on real publicly available datasets. More specifically, we found that the training prediction length is positively correlated with the model accuracy, the number of predicted action sequences define the outcome space that can be captured, and the context vectors dimensionality is a parameter defining the model capacity.

Our approach extends the state of the art in directions that are key to enable more efficient human-robot cooperation, particularly involving non-verbal communication.

7.1 Future Work

This work establishes a strong base for the implementation of a joint action scenario on a humanoid robotics platform such as the iCub.

More specifically, there are different directions this work could be expanded.

- Extend the model by exploring the connection between non-verbal cues and semantic features related to the context, through composing graphical probabilistic models.

- Explore the prediction of sequences with continuous output values as an extension to the discrete labels by predicting a continuous distribution and sampling the next steps inputs from this distribution.

7.2 Material Contributions

During the development of this thesis there were several material contributions to the field of human robot-cooperation. Here I would like to list the more important ones.

1. **ICRA submission**. A paper with the thesis conclusions was submitted to ICRA 2018 under the name "Anticipation in Human-Robot Cooperation: A recurrent neural network approach for multiple action sequences prediction" and is awaiting review.

2. **Code base**. The full code base associated to this thesis is available at [73] for future use. This directory contains the Tensorflow computation graph implementation of both the Action Anticipation (main contribution of this thesis) and Action Recognition model (used for the feature importance section). This source code is the sole authorship of the author of this thesis.

References

[1] Y. Bengio, I. J. Goodfellow, and A. C. Courville, *Deep Learning*. The MIT Press, 2016.

[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[3] C. M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration," *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, vol. 2016-April, no. Section V, pp. 83–90, 2016.

[4] K. Sakita, K. Ogawara, S. Murakami, K. Kawamura, and K. Ikeuchi, "Flexible cooperation between human and robot by interpreting human intention from gaze information," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 846–851, 2004.

[5] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent Neural Networks for driver activity anticipation via sensory-fusion architecture," *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2016-June, 2016.

[6] T. Yonezawa, H. Yamazoe, A. Utsumi, and S. Abe, "Attractive, Informative, and Communicative Robot System on Guide Plate as an Attendant with Awareness of User's Gaze," *Paladyn, Journal of Behavioral Robotics*, vol. 4, no. 2, pp. 113–122, 2013.

[7] H. Admoni and S. Srinivasa, "Predicting user intent through eye gaze for shared autonomy," *AAAI Fall Symposium - Technical Report*, vol. FS-16-01 -, pp. 298–303, 2016.

[8] C.-M. Huang, S. Andrist, A. Sauppé, and B. Mutlu, "Using gaze patterns to predict task intent in collaboration." *Frontiers in psychology*, vol. 6, no. July, p. 1049, 2015.

[9] N. Sebanz and G. Knoblich, "Prediction in Joint Action: What, When, and Where," *Topics in Cognitive Science*, vol. 1, no. 2, pp. 353–367, 2009.

[10] P. A. Lasota, G. F. Rossano, and J. A. Shah, "Toward safe close-proximity human-robot interaction with standard industrial robots," *IEEE International Conference on Automation Science and Engineering*, vol. 2014-Janua, pp. 339–344, 2014.

[11] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, "Active pedestrian safety by automatic braking and evasive steering," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1292–1304, 2011.

[12] A. Tapus, M. J. Matarić, and B. Scassellati, “The Grand Challenges in Socially Assistive Robotics,” *Robotics and Automation Magazin*, vol. 14, no. 1, pp. 35–42, 2007.

[13] S. S. Obhi and N. Sebanz, “Moving together: Toward understanding the mechanisms of joint action,” *Experimental Brain Research*, vol. 211, no. 3-4, pp. 329–336, 2011.

[14] S. Baron-Cohen, *Mindblindness: An Essay on Autism and Theory of Mind*. MIT Press, 1997.

[15] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, “Effects of Nonverbal Communication on Ef ciency and Robustness of Human-Robot Teamwork,” *International Conference on Intelligent Robots and Systems (IROS)*, 2005.

[16] M. Argyle, R. Ingham, F. Alkema, and M. McCallin, “The Different Functions of Gaze,” pp. 19–32, 1973.

[17] H. S. Koppula and A. Saxena, “Anticipating Human Activities Using Object Affordances for Reactive Robotic Response,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 1, pp. 14–29, 2016.

[18] G. Saponaro, G. Salvi, and A. Bernardino, “Robot anticipation of human intentions through continuous gesture recognition,” *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013*, no. Cts, pp. 218–225, 2013.

[19] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, “Structural-RNN : Deep Learning on Spatio-Temporal Graphs,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] G. Gras and G.-Z. Yang, “Intention recognition for gaze controlled robotic minimally invasive laser ablation,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 2431–2437, 2016.

[21] C. Perez-D’Arpino and J. A. Shah, “Fast Target Prediction of Human Reaching Motion for Cooperative Human-Robot Manipulation Tasks using Time Series Classification,” *International Conference on Robotics and Automation (ICRA)*, pp. 6175–6182, 2015.

[22] C. M. Huang and B. Mutlu, “Anticipatory robot control for efficient human-robot collaboration,” *ACM/IEEE International Conference on Human-Robot Interaction*, vol. 2016-April, no. Section V, pp. 83–90, 2016.

[23] P. F. Dominey, G. Metta, F. Nori, and L. Natale, “Anticipation and initiative in human-humanoid interaction,” *2008 8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008*, pp. 693–699, 2008.

[24] J. Martinez, M. J. Black, and J. Romero, “On human motion prediction using recurrent neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] G. Moskowitz, *Social Cognition: Understanding Self and Others*, 2004.

[26] N. Sebanz, H. Bekkering, and G. Knoblich, “Joint action: Bodies and minds moving together,” *Trends in Cognitive Sciences*, vol. 10, no. 2, pp. 70–76, 2006.

[27] M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll, “Understanding and sharing intentions: the origins of cultural cognition.” *The Behavioral and brain sciences*, vol. 28, no. 5, pp. 675–91; discussion 691–735, 2005.

[28] F. Warneken, F. Chen, and M. Tomasello, “Cooperative activies in young children and chimpanzees,” *Child Development*, vol. 77, no. 3, pp. 640–663, 2006.

[29] V. Gallese and A. Goldman, “Mirror neurons and the mind-reading,” *Trends in Cognitive Sciences*, vol. 2, no. 12, pp. 493–501, 1998.

[30] M. Iacoboni, I. Molnar-Szakacs, V. Gallese, G. Buccino, and J. C. Mazziotta, “Grasping the intentions of others with one’s own mirror neuron system,” *PLoS Biology*, vol. 3, no. 3, pp. 0529–0535, 2005.

[31] S. Baron-Cohen, “Precursors to a theory of mind: Understanding attention in others.” *Natural theories of mind: Evolution, development and simulation of everyday mindreading*, pp. 233–251, 1991.

[32] B. J. Grosz, “Collaborative Systems (AAAI-94 Presidential Address),” *AI Magazine*, vol. 17, no. 2, p. 67, 1996.

[33] R. Alami, A. Clodic, V. Montreuil, and E. Sisbot, “Toward human-aware robot task planning,” *AAAI Spring Symp.*, pp. 39–46, 2006.

[34] S. Lallee, U. Pattacini, S. Lemaignan, A. Lenz, C. Melhuish, L. Natale, S. Skachek, K. Hamann, J. Steinwender, E. A. Sisbot, G. Metta, J. Guitton, R. Alami, M. Warnier, T. Pipe, F. Warneken, and P. F. Dominey, “Towards a platform-independent cooperative human robot interaction system: III An architecture for learning and executing actions and shared plans,” *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 3, pp. 239–253, 2012.

[35] S. Lallée, K. Hamann, J. Steinwender, F. Warneken, U. Martienz, H. Barron-Gonzales, U. Pattacini, I. Gori, M. Petit, G. Metta, P. Verschure, and P. F. Dominey, “Cooperative human robot interaction systems: IV. Communication of shared plans with Naive humans using gaze and speech,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 129–136, 2013.

[36] G. Michalos, S. Makris, J. Spiliotopoulos, I. Misios, P. Tsarouchi, and G. Chryssolouris, “ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future,” *Procedia CIRP*, vol. 23, no. C, pp. 71–76, 2014.

[37] A. Baimel, R. L. Severson, A. S. Baron, and S. A. J. Birch, “Enhancing "theory of mind" through behavioral synchrony.” *Frontiers in psychology*, vol. 6, no. June, p. 870, 2015.

[38] S. Baron-Cohen, A. M. Leslie, and U. Frith, “Does the autistic child have a “theoryof mind”?” *Cognition*, vol. 21, no. 3, pp. 7–44, 1985.

[39] B. Scassellati, “Theory of mind for a humanoid robot,” *Autonomous Robots*, vol. 12, no. 1, pp. 13–24, 2002.

[40] A. M. Leslie, “Spatiotemporal continuity and the perception of causality in infants.” *Perception*, vol. 13, no. 3, pp. 287–305, 1984.

[41] A. Leslie, “The peception of causality in infants,” *Perception*, vol. 11, no. April 1981, pp. 173–186 ST – The peception of causality in infant, 1982.

[42] L. B. Cohen and G. Amsel, “Precursors to infants’ perception of the causality of a simple event,” *Infant Behavior and Development*, vol. 21, no. 4, pp. 713–731, 1998.

[43] M. Tomasello, B. Hare, H. Lehmann, and J. Call, “Reliance on head versus eyes in the gaze following of great apes and human infants: the cooperative eye hypothesis,” *Journal of Human Evolution*, vol. 52, no. 3, pp. 314–320, 2007.

[44] M. S. Gobel, H. S. Kim, and D. C. Richardson, “The dual function of social gaze,” *Cognition*, vol. 136, pp. 359–364, 2015.

[45] D. Heylen, “A Closer Look at Gaze,” *AAMAS Workshop on Creating Bonds with Embodied Conversational Agents*, pp. 3–9, 2005.

[46] B. Mutlu, “The design of gaze behavior for embodied social interfaces,” *Proc. CHI ’08, ACM Press*, p. 2661, 2008.

[47] D. G. Novick, B. Hansen, and K. Ward, “Coordinating Turn-taking with Gaze,” *Proc. ICSLP ’96*, vol. 3, pp. 1888–1891, 1996.

[48] S. Ho, T. Foulsham, and A. Kingstone, “Speaking and listening with the eyes: Gaze signaling during dyadic interactions,” *PLoS ONE*, vol. 10, no. 8, pp. 1–18, 2015.

[49] A. Frischen, “Gaze Cueing of Attention,” *Differences*, vol. 133, no. 4, pp. 694–724, 2007.

[50] L. Lukic, J. Santos-Victor, and A. Billard, “Learning robotic eye-arm-hand coordination from human demonstration: A coupled dynamical systems approach,” *Biological Cybernetics*, vol. 108, no. 2, pp. 223–248, 2014.

[51] D. Vernon, S. Thill, and T. Ziemke, “The role of intention in cognitive robotics,” *Intelligent Systems Reference Library*, vol. 105, pp. 15–27, 2016.

[52] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human Trajectory Prediction in Crowded Spaces,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–971, 2016.

[53] C. Sutton, K. Rohanimanesh, and A. McCallum, “Dynamic conditional random fields,” *Twentyfirst international conference on Machine learning ICML 04*, vol. 8, no. x, p. 99, 2004.

[54] D. L. Vail, M. M. Veloso, and J. D. Lafferty, “Conditional random fields for activity recognition,” *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, vol. 5, pp. 1–8, 2007.

[55] Q. Gao and S. Sun, “Trajectory-based human activity recognition using Hidden Conditional Random Fields,” *2012 International Conference on Machine Learning and Cybernetics*, pp. 1091–1097, 2012.

[56] B. Shi, X. Bai, and C. Yao, “An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8828, no. c, pp. 1–8, 2015.

[57] G. Neubig, “Neural Machine Translation and Sequence-to-sequence Models: A Tutorial,” pp. 1–65, 2017.

[58] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[60] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches,” 2014.

[61] S. Ruder, “An overview of gradient descent optimization algorithms,” pp. 1–14, 2016.

[62] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” pp. 1–15, 2014.

[63] D. Krueger and R. Memisevic, “Regularizing RNNs by Stabilizing Activations,” pp. 1–9, 2015.

[64] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training Recurrent Neural Networks,” no. 2, 2012.

[65] G. Lavanchy, M. Strehler, M. N. Llanos Roman, M. Lessard-Therrien, J. Y. Humbert, Z. Dumas, K. Jalvingh, K. Ghali, A. Fontcuberta García-Cuenca, B. Zijlstra, R. Arlettaz, and T. Schwander, “Habitat heterogeneity favors asexual reproduction in natural populations of grassthrips,” *Evolution*, vol. 70, no. 8, pp. 1780–1790, 2016.

[66] D. S. Bernstein, S. Zilberstein, and N. Immerman, “The complexity of decentralized control of Markov decision processes,” *Uncertainty in Artifical Intelligence Proceedings*, pp. 32–37, 2000.

[67] G. Hoffman and C. Breazeal, “Cost-based anticipatory action selection for human-robot fluency,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 952–961, 2007.

[68] H. S. Koppula, R. Gupta, and A. Saxena, “Learning Human Activities and Object Affordances from RGB-D Videos,” *International Journal of Robotics Research*, 2012.

[69] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis,” pp. 1010–1019, 2016.

[70] Y. Saeys, I. Inza, and P. Larrañaga, “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[71] R. W. Angel, W. Alston, and H. Garland, “Functional relations between the manual and oculomotor control systems,” *Experimental Neurology*, vol. 27, no. 2, pp. 248–257, 1970.

[72] Cornelist J. Van Rijsbergen, *Information Retrieval*. Butterworth-Heinemann, 1979.

[73] P. Schydlo, “Thesis source code repository,” 2017. [Online]. Available: http://github.com/pschydlo/ActionAnticipation