# ADAPTIVE POKER BOT BASED ON PLAYER-MODELLING

Bjrn Hasager Vinther & Nicolai Guldbk Holst
bhas@itu.dk ngul@itu.dk

A thesis submitted for the degree of

Bachelor Softwaredevelopment

May 2014

**Abstract**

Poker is a game which have a lot of different aspects which makes it a very interesting game to implement artificial intelligence (AI) into. The game is unlike games like chess and backgammon as poker has the problem that the information is imperfect. This is also one of the reasons why poker is interesting topic in regards to artificial intelligence as it opens up for some problematics as how to handle the information we have in regards to what we do not have. Because of this hidden information there are many decision to decide from, as we have to take into account all the information that we have access to and the possible information which is hidden from us. Another thing that makes poker interesting to have a look at is that we now have opponents who has the opportunity to trick us into thinking that he is in another state than what he actually is. In this bachelor thesis we will present a way to implement a neural network in a poker game to teach the bots how to play Texas Holdem Poker. The neural network will keep evolving along the opponents to ensure that the bot has the best possible chances of winning or minimize the loses. We will test the bots against real life human players in many suitable situations and from these results we can determine how effective our neural network is at teaching our bots how to play the game.

# Introduction

The game of poker is already a pretty complex game, put that together with artificial intelligence and you have a situation that is very challenging. To create an AI player that is producing good results against other players bring many problematics along the way. As far as implementing some artificial intelligence we are facing problems regarding the hidden information which is our opponents cards. Furthermore deception, and how many players we are playing against will also be obstacles that we will have to overcome. If we want to produce the best possible results we will have to player model each of the players that we are playing against, to make sure that the bots gameplay will be the most optimal, considering all the opponents and their gameplay. One of the reasons why it is necessary to player model every player is because we need to have the opportunity to exploit a weak player. If one keeps making weak moves that we can predict it would be a wise consideration to exploit this and beat the player. Even if we are playing a game of poker there will always good ways of handling each situation and thereby each opponent in the best way possible. The best players in the world are very efficient at adapting to each and every player of the board, even if the opponents changes their gameplay throughout the game, one must be able to adapt to this to be able to beat them.

# Preface

This is a bachelor thesis consisting of 15 ETCS points. The thesis has been written exclusively by Nicolai Guldbk Holst and Bjrn Hasager Vinther, both studying *Software development* at the IT University of Copenhagen. It has been written in the spring semester 2015. Our supervisor was Kasper Sty.

# Contents

## List of Figures

# Part I
# Project overview

## 1 Problem statement

## 2 Scope

## 3 Glossary & abbreviations

## 4 Method

## Texas Hold'em

As there are different kinds of poker, there is one which is the most common, which is Texas Holdem. Therefore this will be the kind of poker this bachelor thesis will be evolving around. This type of poker is also considered to be the type that is the most strategically complex type of the game. The rules of the game however are fairly simple. Each player is dealt 2 hole cards, which only the one receive them must be able to see. After the players have been dealt their cards, the two players who are sitting next to the dealer must be then ones to lay the blinds. The players after that can then either fold, call the bet or raise with either big blind or if we are playing no-limit then he can go all the way up to all-in. When the first round of betting are finished the dealer will deal 3 community cards which is also known as the flop. These cards are dealt so every player can see what their value is, and then we have another round of betting. When that is done, the dealer will burn a card, which means that that card will not be in play, and then he will draw another and put it face up like the rest of the community cards. This card is known as the turn. The next betting round begins and then the dealer will burn another card and then deal the river card with face up. Another round of betting, and then finally the players try to make the best hand they can with 5 cards with a combination of their hole cards and the

community cards. The player with the best hand will be the winner.

# Part II
# Theoretic part

## 5 Theory overview

### 5.1 Bayes theorem

### 5.2 Monte Carlo simulation

We make decisions every day and before during that, whether conscious or subconscious we are calculating risks of doing just that. It is impossible to predict what the future holds for us, but the Monte Carlo simulation will let you see which possible outcomes you can expect from the different decisions. A Monte Carlo Simulation will let you know all possible outcomes which can occur and how possible it is that that outcome will be the one. Basicly it is like saying what if thousands of times and seeing what happens and how often something happens. If we were to roll a dice and find out what the possibility is that two dices rolls 8. We would throw the dices 100 times, and for each time we would record the throw. That way we have how many times we roll each of the outcomes and therefore we can predict the possibility of that outcome. Lets say we rolled 8 12 times out of a 100 rolls. The possibility of rolling 8 is 18In poker this could maybe be used to predict what cards are popping up in the flop and so what are our chances of having a hand that is good. Another alternative could be to determine whether a player is an experienced player or a beginner. If we were to look how often a player wins versus how many times you win, it could be a guideline to see how big a chance you have of winning the game.

### 5.3 Nash equilibrium

*"Nash equilibrium refers to a condition in which every participant has optimized its outcome, based on the other players expected decision."*

This theory works with the assumption that a party knows the other parties strategies. If all parties have an optimized strategy compared to the other parties strategies then this situation is said to be in Nash equilibrium.

In poker this is useful in a situation where your opponent plays really aggressive and goes all-in a lot. Depending on the size of your stack compared to the big blind different hands is profitable to call in order to win the big blind. For this purpose a chart has been created to show which hands are profitable to call depending on the size of your stack. Likewise a

## 5.4 Neural network

Neural networks is models which are somewhat inspired by biological neural networks. Neural networks is considered to be one of the best methods to classify input-patterns. Neural network is for example used to recognize and reading handwriting and speech recognition. Humans are very good at recognizing visual patterns, but if we were to write a program that could do just that, it becomes alot more difficult. We have simple intuitions when we are trying to recognize different shapes. But to explain to a program how to recognize for example a Y would be something like a straight line with two lines pointing out from it at the top to each side at a 35 degree. When we try to make rules like this to let the program recognize letters we can quickly become lost in what we expect it to look like and the special cases. Neural networks has a different approach to the program which is much more suitable than describing each letter in some mathematical algorithm formula. We give the neural network a very large amount of handwritten letters, which

shall be used in the training of the neural network.

The neural network will then use the examples to automatically determine some rules for reading the handwritten letters. Of course this example doenst have a lot of different types of the letter A so if we want to let the neural network become better at reading the handwritten letter A we would have to give it an example with many more examples of the letter. It would improve

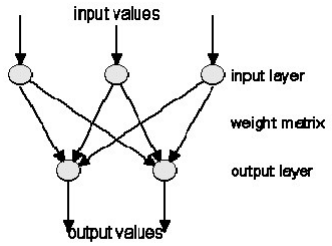
the accuracy, therefore it is better to provide the neural network with a thorough example.

*"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.* A neural network consists of neurons which can send signals to each other. Each neuron processes its input signals and determines if the signal needs to be sent further. A neural network is not being fed with rules but instead examples that it can make rules from. With that a neural network is able to learn new skills, which is something that the traditionel computer cant do.

There are several types of neural networks and are being distinguished by their type (feedforward or feedback), their structure, and the learning algorithm that they use. Feedforward neural networks will only allow the neurons to connect between two different layers. The feedback type of neural network will have a connection between neurons which are of the same layer but also between the different layers.

## 5.5 Perceptron

This neural network is considered to be one of the simplest as it has two neuron layers which only accepts binary values for the input and output. With only two layers (input/output) there are no hidden layers. The learning process of this neural network is supervised . The network can be used to solve basic logical operations.



Perceptron