

# TUT4

Sakura Noskor

## Table of contents

<b>Introduction</b>	<b>1</b>
<b>Content</b>	<b>1</b>
Day 1- Transcribe the entire Introduction . . . . .	1
Day 2- Rewriting the Introduction . . . . .	3
Day 3- Transcribe the Abstract . . . . .	3
Day 4- Rewrite the Abstract in 4 sentence . . . . .	3
Day 5- Re-rewriting the abstract using popular words . . . . .	3
Day 6- 3 likable things about structure . . . . .	4
Day 7- 3 unlikable things about structure . . . . .	4

## Introduction

This is a journal of writing for seven days as the tutorial had instructed. The paper I have chosen is *Some Studies in Machine Learning Using the Game of Checkers* by A.L. Samuel. This paper is available on IEEE Xplore through the link [here](#). This paper really stood out to me as I am currently taking a machine learning course where we are learning the basics of decision trees; a concept that is used in the figures of this text.

## Content

### Day 1- Transcribe the entire Introduction

The studies reported here have been with the programming of a digital computer to behave in a way which, if done by human beings or animals, would be described as involving the process of learning. While this is not the place to dwell on the importance of machine-learning procedures, or to disclose on the philosophical aspects, there is obviously a very

large amount of work, now done by people, which quite trivial in its demands on the intellect but does, nevertheless, involve some learning. we have at our command computers with adequate data-handling ability and with sufficient computational speed to make use of machine-learning techniques is still rudimentary. Lacking such knowledge, it is necessary to specify methods of problem solution in minute and exact detail, a time-consuming and costly procedure. Programming computers to learn from experience, should eventually eliminate the need for much of this detailed programming effort.

- General methods of approach At the outset it might be well to distinguish sharply between two general approaches to the problem of machine learning. One method, which might be called the NeuralNet Approach, deals with the possibility of inducing learned behavior into a randomly connected switching net (or its simulation on a digital computer) as a result of a reward-and-punishment routine. A second, and much more efficient approach, is to produce the equivalent of a highly organized network which has been designed to learn only certain specific things. The first method should lead to the development of general-purpose learning machines. A comparison between the size of the switching nets that can be reasonably constructed or stimulated at the present time and the size of the neural nets used by animals, suggest that we have a long way to go before we obtain practical devices. the second procedure requires reprogramming for each new application, but it is capable of realisation at the present time. The experiments that were described here were based on the second approach.

- Choice of problem For some years the writer has devoted his spare time to the subject of machine learning and has concentrated on the development of learning procedures as applied to games. A game provides a convenient vehicle for such study as contrasted with a problem taken from life, since many of the complications of detail are removed. Checkers, rather than chess, was chosen because the simplicity of its rules permits greater emphasis to be placed on learning techniques. Regardless of the relative merits of the two games as intellectual pastimes, it is fair to state that checkers contains all of the basic characteristics of an intellectual activity in which heuristic procedures and learning processes can play a major role and in which these processes can be evaluated. Some of the characteristics might well be enumerated. They are:

1. The activity must not be deterministic in the practical sense. there exists no known algorithm which will guarantee a win or a draw in checkers, and the complete exploration of every possible path through checker game would perhaps  $10^{40}$  choices of moves which, at 3 choices per microsecond, would still take  $10^{21}$  centuries to consider.
2. A definite goal must exist-the winning of the game-and at least one criterion or intermediate goal must exist which has a bearing on the achievement of the final goal and for which the sign should be known. In checkers the goal is to deprive the opponent of the possibility of moving, and the dominant criterion is the number of pieces of each color on the board. The importance of having a known criterion will be discussed later.
3. The rules of the activity must be definite and they should be known. Games satisfy this requirement. Unfortunately, many problems of economic importance do not. While in principle the determination of the rules can be a part of the learning process, this is a complication which might well be left until later.
4. There should be a background of knowledge concerning the activity against which the learning process can be tested.
5. The activity should be one that is familiar to a substantial body of people so that the behavior of the program can be made understandable to them. The ability to have the program play against human opponents (or antagonists) adds spice to

the study and, incidentally, provides a convincing demonstration for those who do not believe that machines can learn. Having settled on the game of checkers for our learning studies, we must, of course, first program the computer to play legal checkers; that is, we must express the rules of the game in machine language and we must arrange for the mechanics of accepting an opponent's moves and of reporting the computer's moves, together with all pertinent data desired by the experimenter. The general methods for doing this were described by Shannon in 1950 as applied to chess rather than checkers. The basic program used in these experiments is quite similar to the program described by Strachey in 1952. The availability of a larger and faster machine (the IBM 704), coupled with many detailed changes in the programming procedure, leads to a fairly interesting game being played, even without learning. The basic forms of the program will be now described.

## **Day 2- Rewriting the Introduction**

## **Day 3- Transcribe the Abstract**

Two machine learning procedures have been investigated in some detail using the game of checkers. Enough work has been done to verify the fact that a computer can be programmed so that it will learn to play a better game of checkers than can be played by the person who wrote the program. Furthermore, it can learn to do this in a remarkably, short period of time (8 or 10 hours of machine playing time) when given only the rules of the game, a sense of direction, and a redundant and incomplete list of parameters which are thought to have something to do with the game, but whose correct signs and relative weights are unknown and unspecified. The principles of machine learning verified by these experiments, of course, applicable to many other situations.

## **Day 4- Rewrite the Abstract in 4 sentence**

Computers are great at following instruction hence with proper datasets, they can be trained to win games. This paper applied 2 machine learning algorithms to train a machine into winning a game of checkers, even against the person who programmed it. The machine itself can learn to play in 8 to 10 hours when given the rules of the game, a general direction and incomplete list of parameters related to the game but their weight and correct signs are not specified. The 2 algorithms will be looked at carefully enough to be applied to various other situations.

## **Day 5- Re-rewriting the abstract using popular words**

Computers are great at following orders so, with the right sets of information, they can be trained to win games. This paper put 2 machine learning system to train a machine into winning a game of checkers, even against the person who gave orders to the machine. The

machine itself can learn to play in 8 to 10 hours when they know how to play the game, a little push in the right direction and little information about cases they might run into in the game but their weight and signs are not written. The 2 systems will be looked at carefully enough to be used in other situations.

### **Day 6- 3 likable things about structure**

- I liked the bolded and italicized headings as they helped understand the shift in topic.
- I also liked that some of the decision trees provided had instruction in a black box of how the machine worked. Made it easier to understand.
- They also explain every one of their terminology well. So reader doesn't have to know machine learning to understand. Words like rote-learning and ply were explained in text or in the figures.

### **Day 7- 3 unlikable things about structure**

- One of the things is how some of the pages and results are cut off by a page of picture. The reason why I don't like this style is that while examining the picture I tend to forget about the content of the paragraph before, so when the paragraph continues on the next page I feel a bit lost. It breaks cohesion.
- There is no clear structure, that we discussed in class, in this paper. It may abruptly change from the data section to result or from result to discussion. The only clear indication is Introduction and Conclusion and the rest is headings based on the topic they are discussing. This makes things confusing as I am already used to reading papers with a strict structure where procedure section follows data section then result.
- All the figures used in the main body of the paper doesn't seem to be referenced in the actual body of the paper. I feel that this bad structure as many of the pictures are too big and they can be in the appendix.