**Automated backgammon player: learning through self-play**

**1. Abstract**

Neural Network used widely in the modern society, especially in machine learning. Through Artificial Neural Network (ANN) could make machines think like a human and behave like a human. In this project, I would use ANN to train a model that would learn to play backgammon like a player through self-play. My aim is to finish a backgammon model with a simplified backgammon game.

**2. Introduction**

**Background:**

In 1992, Gerald Tesauro developed a computer backgammon program “TD-Gammon” using artificial neural networks (ANNs). The performance of TD-gammon was at a level not far below that of the best human players of the time, and the strategies explored by TD-gammon led to advances in the theory of backgammon play.

In 1997, Pollack and Blair demonstrated that a coevolutionary self-play approach to developing backgammon strategies could be successful using ANNs and hill climbing (no reinforcement learning necessary). The result demonstrated that the dynamics of backgammon are particularly suited to coevolutionary learning through self-play (unlike many other games, such as chess) (Pollack, J.B., & Blair, D., 1998). After this paper was published, Tesauro write a short comment to defend his work, arguing that hillclimbing alone is not sufficient to achieve successful learning.

Recent developments in ANNs (in particular deep learning methods) have generated international headlines through successful game playing implementations such as AlphaGo, capable of beating the best humans. In addition, based on previous professional research on backgammon through self-learning.

**Aim:**

This project will attempt to replicate a backgammon player that had been developed by Pollack and Blair, successfully using an ANN trained through self-learning, beginning with a simplified version of backgammon, and moving on to a more realistic version of the game.

**Objective:**

* Demo and Research

Firstly, I would write a demo in Java to be familiar to backgammon and read related paper on backgammon and ANN.

* Simplest AI with some basic strategies

Secondly, according to the optimal doubling written by Keeler and Spencer(1975) and the optimal strategy written by Thorp (1988), apply these strategies on demo.

* Self-learning (ANN)

Then the demo should use self-learning and start applied in ANN model.

* Tests and Reflection

Finally, testify the model and do reflection all the time. In this period, how to evaluate the board is the key point that it must satisfy the strategies noted before.

* Report

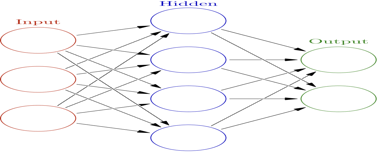
Finish a report contained above mentioned.

**3. Proposed approach**

As for the language, after scanning some papers, most of they use Java to write the UI and not mention which language they used. Then they would use Matlab to figure out the data algorithms. Consequently, I also could use Java to write most part and then use other tools to help analyze. ANN would be the main method to train the model.

* *ANN*

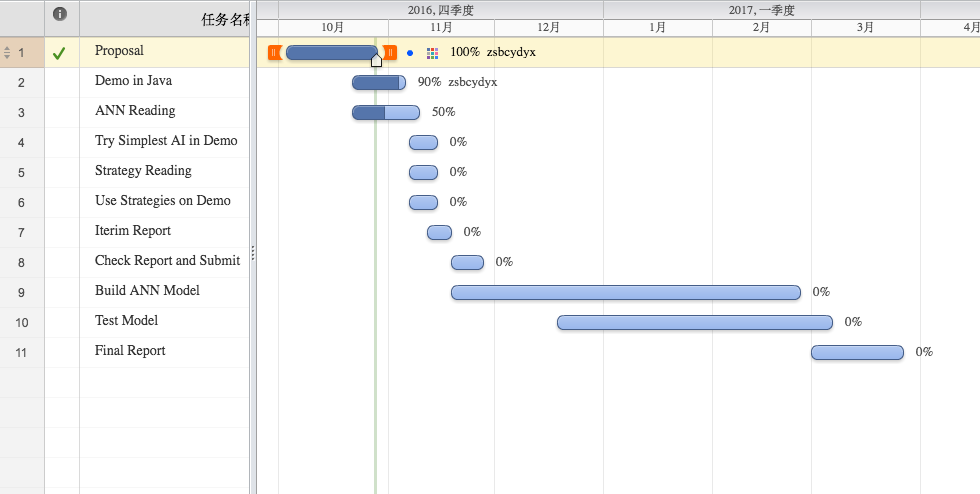
ANN contain multiply hidden layers with dynamic weights and then get a output from its learning. The weights would be modified after each train or test so only if the accuracy would drop from a high level to a lower level that this ANN model is useful.



* *Co-evolution*

Co-evolutionary algorithms are a class of algorithms used for generating [artificial life](https://en.wikipedia.org/wiki/Artificial_life) as well as for optimization, game learning and [machine learning](https://en.wikipedia.org/wiki/Machine_learning). It is widely used in self-learning that help machines to behave with human-like intelligence.

**4. Project plan**

****

**References**

1. Keeler, E.B., & Spencer, J. (1975) *Optimal Doubling in Backgammon.* Available at: <http://www.bkgm.com/articles/KeelerSpencer/OptimalDoublingInBackgammon/> (Access at 28th, October, 2016)
2. Pollack, J.B., & Blair, D. (1998). Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32.
3. Tesauro, G. (1992). Practical issues in temporal difference learning. *Machine Learning*, *8*, 257–277.
4. Tesauro, G. (1998). Comments on Co-Evolution in the Successful Learning of Backgammon Strategy. *Machine Learning*, *32*, 241–243
5. Thorp, E.O.,(1988). *Backgammon: The Optimal Strategy for the Pure Running Game* Available at: http://www.edwardothorp.com/sitebuildercontent/sitebuilderfiles/BackGammon.pdf(Access at 28th, October, 2016)

.