

# A Simple Neural Network for Playing Othello

Ascent Robotics

Tokyo, Japan

Hello! Thank you for interviewing with us at Ascent Robotics. We would like to know a bit more about you and your technical skills by guiding you through a simple project.

We are going to develop a simple program that plays the game of Othello by using a small neural network. The overall process will be broken down into a 3 stages.

- **Do your best and finish as much stages as you can.** It is not necessary to complete all the stages.
- It is up to you to decide upon any unspecified parameters.
- Feel free to use any development tool or platform of your choice.
- You have **one week** to go through this project.
- We would like you to submit your code and a simple description at the end of the project. We would also like schedule a session with you to walk us through your result.
- Please feel free to reach out to us at [hello@ascent.ai](mailto:hello@ascent.ai) if you have any questions.

We hope you will have some fun during the process, and looking forward to see your result!

## 1 Feedforward Neural Network from Scratch

We will first start by developing a simple feedforward neural network and train it on the iris data set. Please develop the neural network **from scratch**

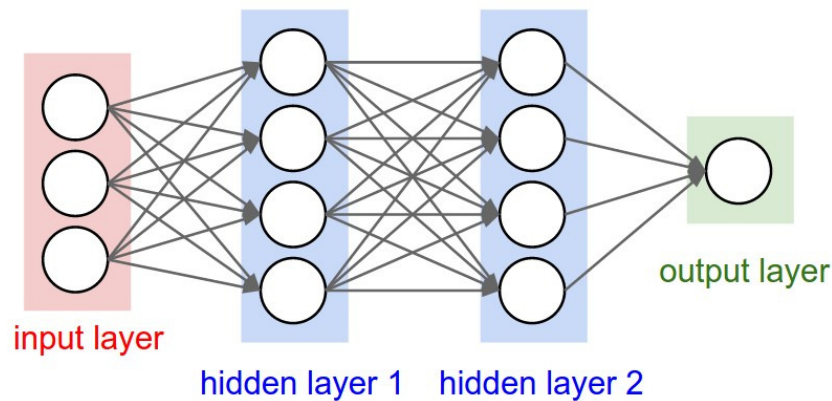


Figure 1: Simple feedforward neural network.

without using any existing framework or packages.

### Requirements:

#### 1. Simple feedforward neural network:

- One input layer, 2 hidden layers, and one output layer.
- The activation function will be the **sigmoid** function.
- The number of neurons in each layer and the type of output layer are your decision.

#### 2. Backpropagation training algorithm.

You will need to decide upon the appropriate loss function.

#### 3. Train the neural network on the iris dataset.

You will need to decide on how will you divide up the dataset to training and test set.

Figure 1 is an example of the simple feedforward neural network. Any unspecified parameters that are not mentioned above are all subject to your choice. The following are some references that will help you along your way:

1. Iris Dataset  
<https://archive.ics.uci.edu/ml/datasets/Iris>
2. Neural Networks and Deep Learning  
<http://neuralnetworksanddeeplearning.com/>
3. CS231n: Backpropagation  
<http://cs231n.github.io/optimization-2/>

## 2 Othello Game Playing Agent

We will now proceed to develop a simple program that will be able to play the game of Othello against a human opponent.

### Requirements:

1. **A simple interface** for playing a game of Othello with the agent. Simple command line interface is fine, there is no need to implement a GUI. The program will need to be able to set which side will the agent or human player be playing and also be able to take moves from the human player. The program will also need to display relevant information, such as current board state, the number of moves that has been made, whose turn is it and so on.
2. **The agent only considers one move ahead**, that is a brute force search with the search depth of 1. The agent looks through all of its possible moves that it is able to play on its turn, and picks the one that has the highest value calculated by the evaluation function to play.
3. **Use a simple evaluation function.** Figure 2 is a simple position evaluation function which you will be able to use. Each cell corresponds to a position on the board in Othello, and each value in the cell corresponds to the value of placing one's piece on the corresponding position. The higher the value is, the more advantageous will it be to place one's own piece in that position. You are free to design any evaluation function of your own, it is not necessary to use the function depicted in Figure 2.

Any unspecified design decision or parameters are subject to your choice. Any application of more advanced algorithms or techniques, such as  $\alpha\beta$  pruning, are a plus but not required. Some references that will be useful along your way:

1. Othello in Wikipedia  
<https://en.wikipedia.org/wiki/Reversi>
2. Playing othello online  
<http://www.othelloonline.org/>

120	-20	20	5	5	20	-20	120
-20	-40	-5	-5	-5	-5	-40	-20
20	-5	15	3	3	15	-5	20
5	-5	3	3	3	3	-5	5
5	-5	3	3	3	3	-5	5
20	-5	15	3	3	15	-5	20
-20	-40	-5	-5	-5	-5	-40	-20
120	-20	20	5	5	20	-20	120

Figure 2: An evaluation function for Othello.

### 3 Neural Network Agent for Playing Othello

Finally, we are ready to put everything together by integrating the neural network developed in stage 1 with the Othello game playing program developed in stage 2 into a single neural network game-playing agent.

#### Requirements:

1. **Integrating the neural network and the game playing program.** The integration is essentially replacing the simple evaluation function with the neural network developed in stage 1.
2. **Train the neural network as an evaluation function.** The neural network needs to be trained by either the **TD-Learning** or the **Q-Learning** algorithm.

This stage will involve significantly more design decisions than the previous two stages, and you will need to determine any unspecified parameters or specifications. The following are some referenees to get you started.

1. TD Learning in Wikipedia  
[https://en.wikipedia.org/wiki/Temporal\\_difference\\_learning](https://en.wikipedia.org/wiki/Temporal_difference_learning)
2. Q Learning in Wikipedia  
<https://en.wikipedia.org/wiki/Q-learning>