

An Intelligent Gameplayer

Creating an Artificial Neural Net to Play Connect Four

Building a Basic Player

Goal: Win if there's a chance, otherwise block the opponent if needed

Strategy

- Check for own three in a row with open fourth slot
 - Win Game
- Check for opponent three in a row with open fourth slot
 - Block opponent
- Play randomly

Strength: Never misses opportunities to win or block

Weakness: No move look ahead

Building a Complex Player

Goal: Use look ahead maximize favorable board positions and minimize opponent's ability to set up good boards

Strategy

- Evaluate boards with pattern-based scoring system
 - 1000 points given for four in a row
 - 100 points given for three in a row
 - 10 points given for two in a row
- Simulate future moves to increase favorability

Strength: Score moves based on strength, ability to setup favorable board positions

Weakness: Scoring is based on abstract point system

Choosing a Move

1	2	3	4	5	6	7
=	=	=	=	=	=	=
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	R	Y	Y	0
0	0	0	Y	R	R	R
R	Y	R	R	Y	Y	Y
R	R	Y	R	R	Y	Y

Score: 85

1	2	3	4	5	6	7
=	=	=	=	=	=	=
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	R	Y	Y	0
0	R	0	Y	R	R	R
0	Y	R	R	Y	Y	Y
R	R	Y	R	R	Y	Y

Score: 145

1	2	3	4	5	6	7
=	=	=	=	=	=	=
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	R	Y	Y	0
0	0	R	Y	R	R	R
0	Y	R	R	Y	Y	Y
R	R	Y	R	R	Y	Y

Score: -4980

1	2	3	4	5	6	7
=	=	=	=	=	=	=
0	0	0	0	0	0	0
0	0	0	R	0	0	0
0	0	0	R	Y	Y	0
0	0	0	Y	R	R	R
0	Y	R	R	Y	Y	Y
R	R	Y	R	R	Y	Y

Score: 45

1	2	3	4	5	6	7
=	=	=	=	=	=	=
0	0	0	0	0	0	0
0	0	0	R	0	0	0
0	0	0	R	Y	Y	0
0	0	0	Y	R	R	R
0	Y	R	R	Y	Y	Y
R	R	Y	R	R	Y	Y

Score: 35

1	2	3	4	5	6	7
=	=	=	=	=	=	=
0	0	0	0	0	0	0
0	0	0	0	0	R	0
0	0	0	R	Y	Y	0
0	0	0	Y	R	R	R
0	Y	R	R	Y	Y	Y
R	R	Y	R	R	Y	Y

Score: 30

1	2	3	4	5	6	7
=	=	=	=	=	=	=
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	R	Y	Y	R
0	0	0	Y	R	R	R
0	Y	R	R	Y	Y	Y
R	R	Y	R	R	Y	Y

Score: 35

Results of the First Players

Win Percentages by Player Types:

	Random with Blocking (vs.) Two Move Lookahead	Random with Blocking (vs.) Four Move Lookahead	Two Move Lookahead (vs.) Four Move Lookahead	Four Move Lookahead (vs.) Six Move Lookahead
Win Percentages	Two Move: 82.5%	Four Move: 96%	Four Move: 73%	Six Move: 62.5%

Neural Net Scoring System

Heuristic Function

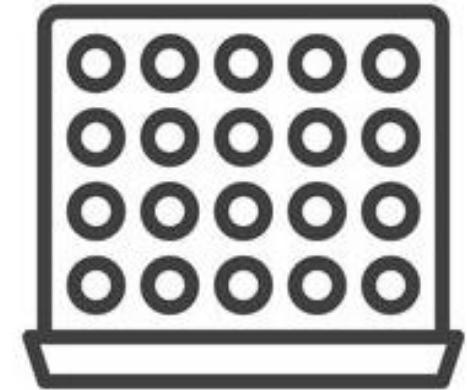
- Function estimates how good a game state is, helping the AI decide which moves are more promising

Complex Gameplayer Weakness

- Function scores board with abstract point system

Artificial Neural Net

- Artificial neural net will develop its own heuristic function



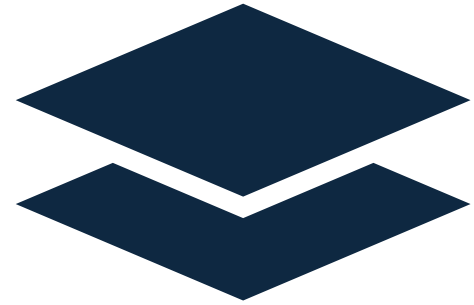
Neural Nets at a High Level



Brain-like Structure

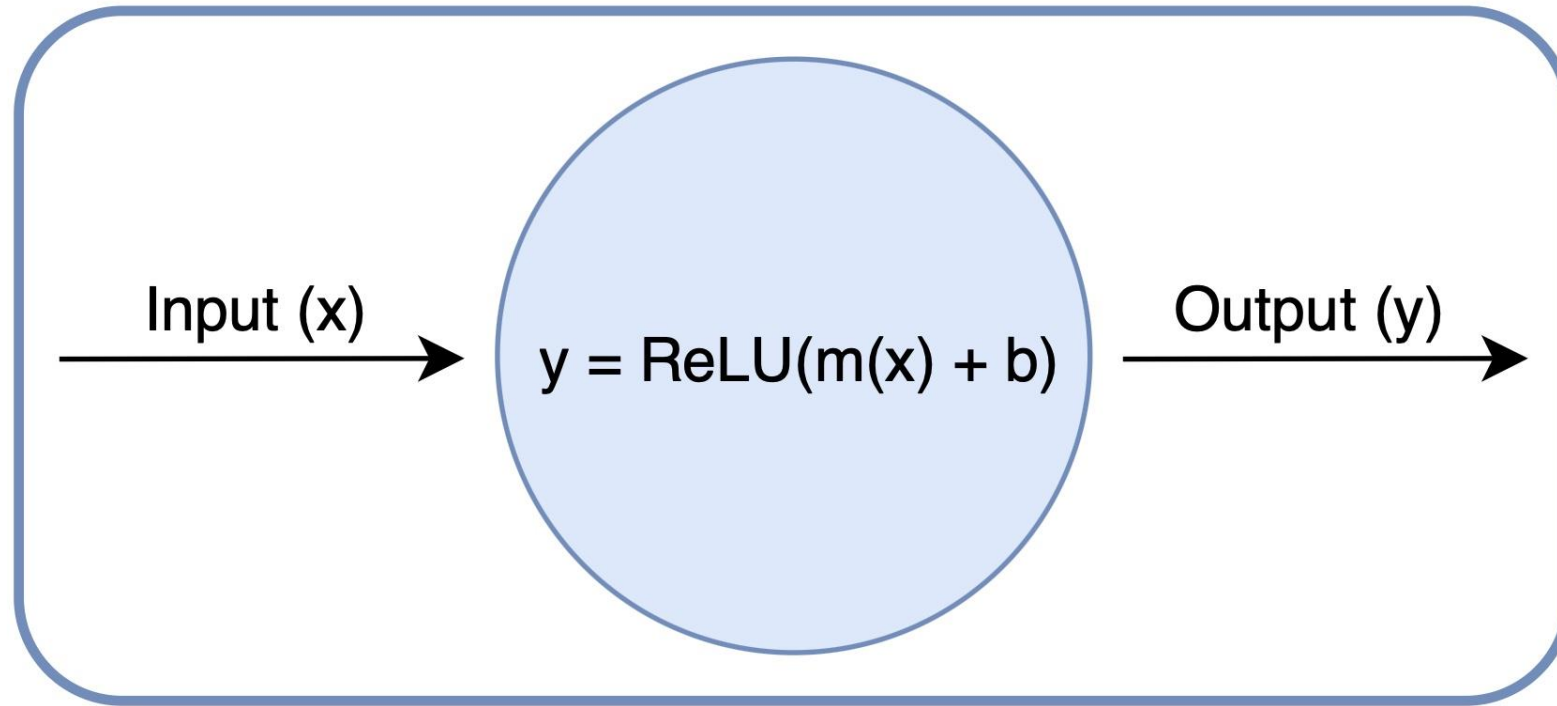


Input → Activation → Output



Multiple Layers

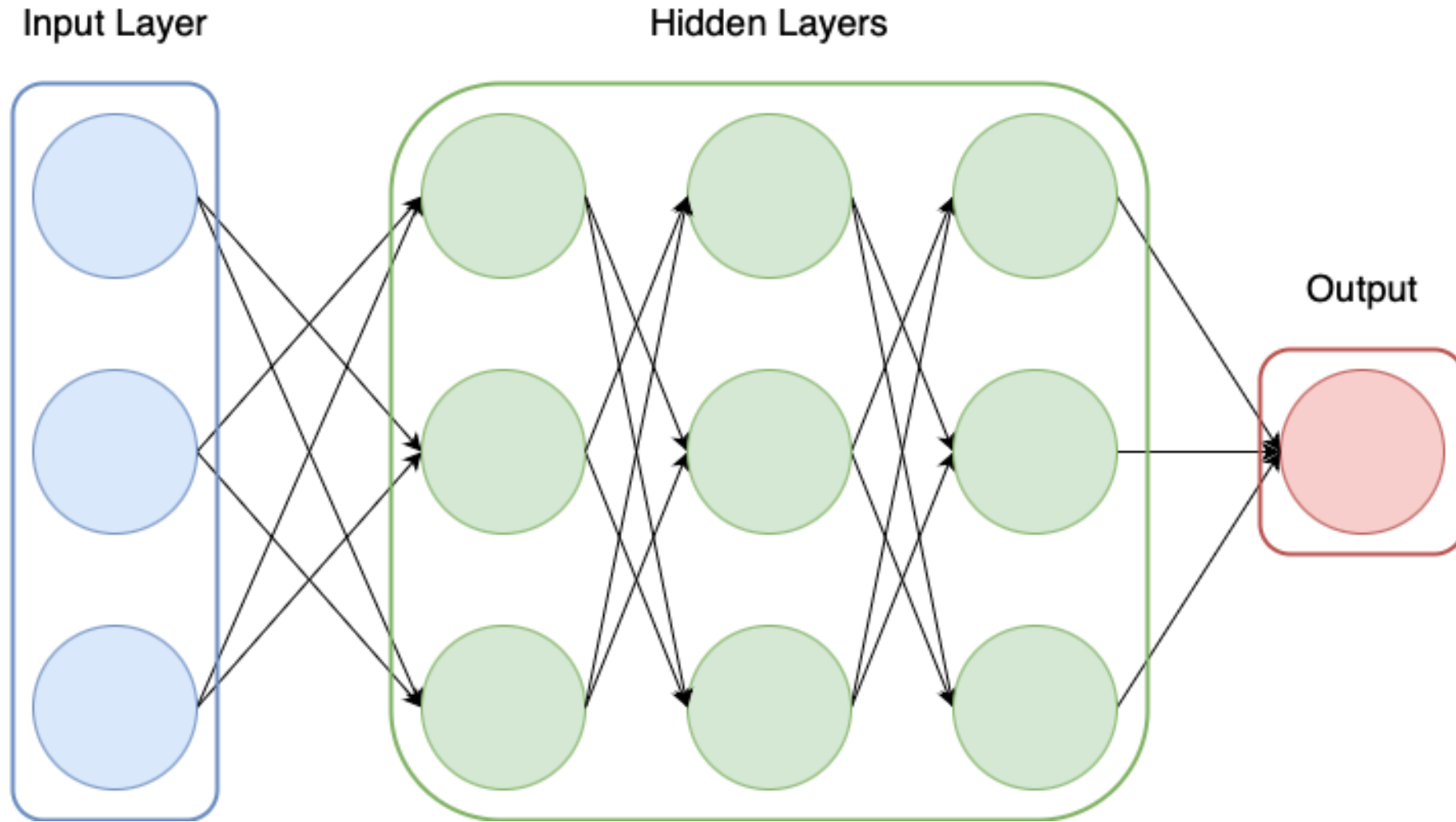
Single Node Diagram



Rectified Linear Unit (ReLU)

- Function outputs the input directly if it's positive; otherwise, it outputs zero

Neural Net Diagram



Connect Four Neural Net

Input Layer

- ± 1 or 0 representing player/spaces

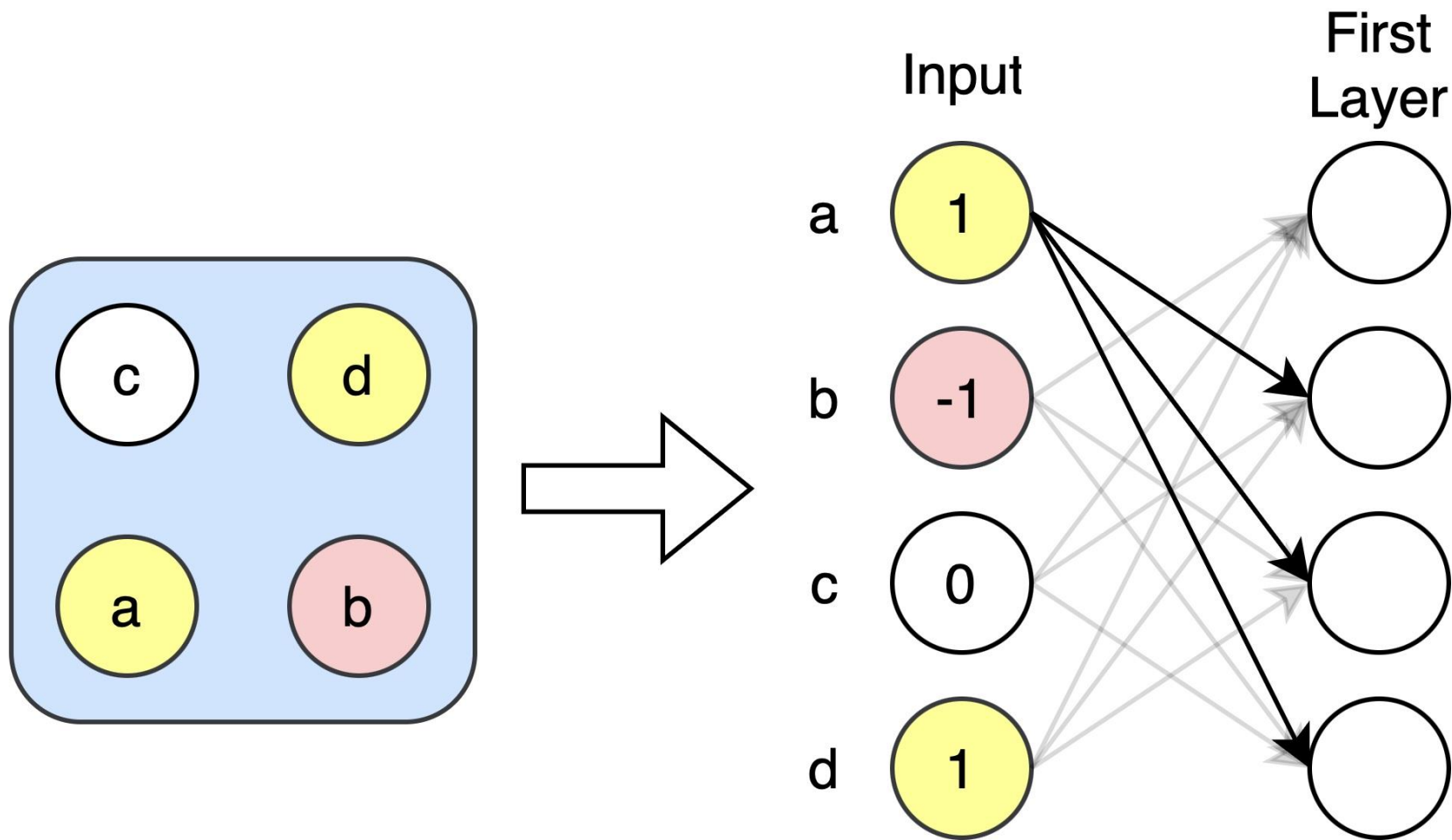
Hidden Layers

- 3 hidden layers
- Connections between adjacent positions

Output Layer

- Single value as board evaluation

Constructing the Neural Net



Improving the ANN Evaluation

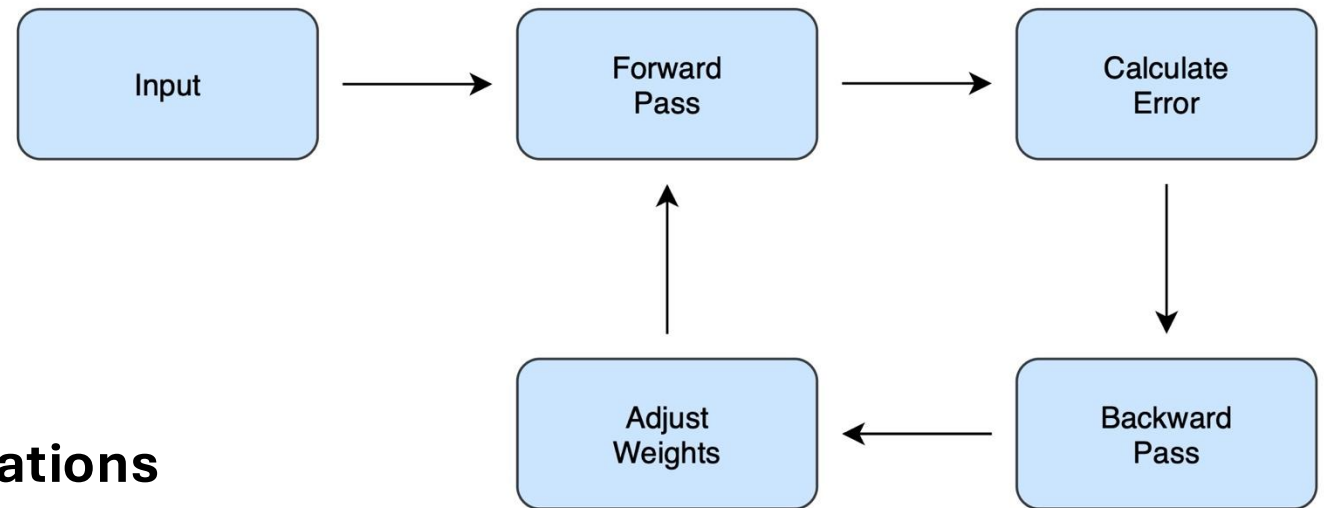
Utilizing Error

- Difference between target heuristic and ANN heuristic
- Adjust weights to reduce error

Continuous Process

- Repeat the process over many different board states
- More samples creates better heuristic function

Currently at 25,000 completed iterations



Developing a Target Value

Monte Carlo Simulation

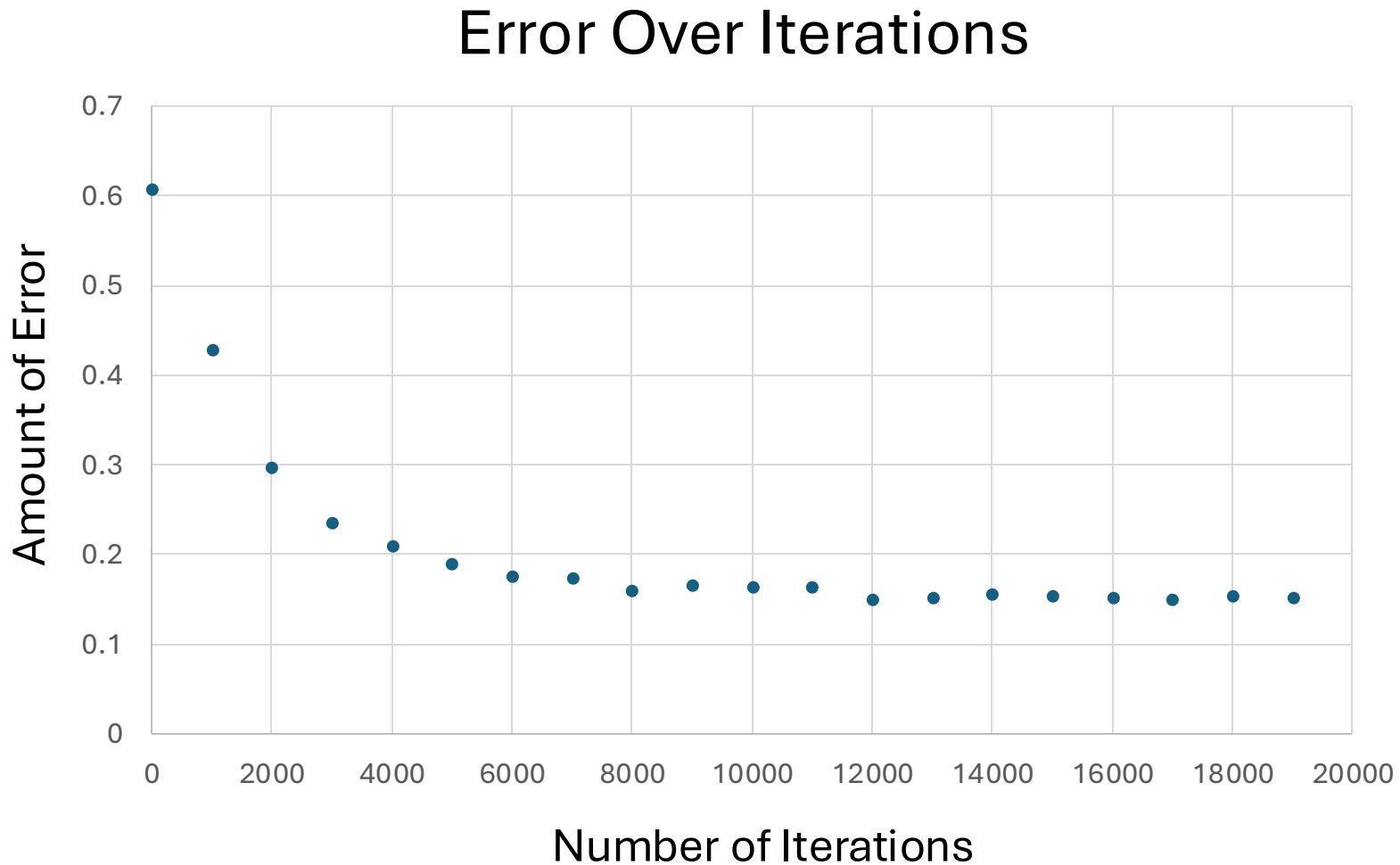
- Play large number of games from given board state
- Resulting win percentage becomes target value

Choosing a Designated Player

- Trade off between quality of player and run time
- Heuristic with four move look ahead



Decrease in Error Over Iterations



Results of the Learning Process

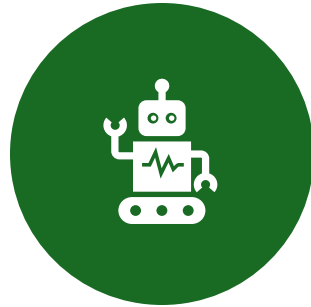
Neural Net Win Percentages:

	ANN with Starting Weights	Random Play with Blocking	Two Move Lookahead	Four Move Lookahead
Untrained	-	9%	4%	2%
Trained	58%	12.5%	7.5%	3%

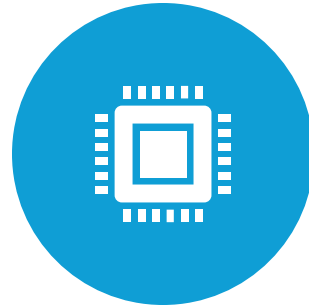
Key Takeaways and Next Steps



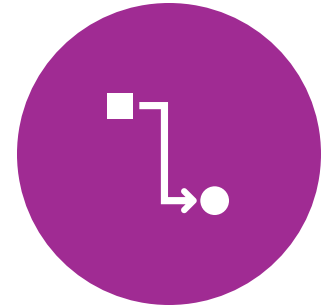
Neural networks can learn to master games **without explicit strategies**



Powerful decision-making structure used in **gameplayers and image recognition**



The training process is **time consuming and computationally** demanding



Small changes in **hyperparameters** significantly alter learning process