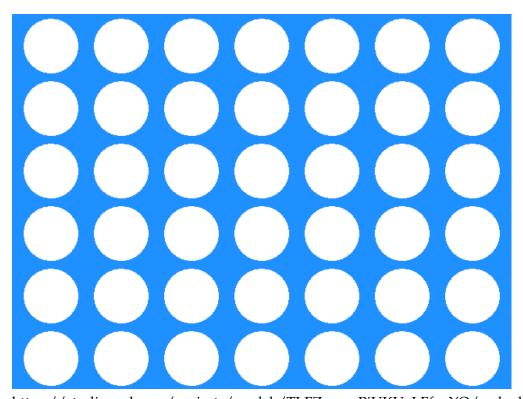# Presentation

November 29, 2017

## 1    Training a Computer to Play Connect 4

Ben Andrews & Jimmy Hickey

## 2    Connect 4



https://studio.code.org/projects/applab/TLFZogscaPiUKUzLFfvzYQ/embed

```
In [ ]: run game.py human human
```

## 3    Method

Trained a network using supervised learning methods; thus, we needed to generate data.

### 3.1 Data Generation

We implemented a minimax algorithm to play against a random player.

## 4 MiniMax

Minimax is a tree traversal algorithm used in game theory. Its goal is to minimize its possible loss and maximize its gains.

### 4.1 Adding $\alpha$-$\beta$ Pruning to Minimax

$\alpha$-$\beta$ pruning is an adversarial search algorithm that reduces the search space of a minimax tree by using optimization criteria.

It stops searching a branch when the branch is determined to be certainly worse than what has been previously discovered.

## 5 Minimax with Pruning Example

http://inst.eecs.berkeley.edu/~cs61b/fa14/ta-materials/apps/ab_tree_practice/

## 6 Minimax in Connect 4

Scores are generated based on how many doubles, triples, and wins exist on the board.

- Doubles: 1 points
- Triples: 100 points
- Wins: 100,000 points
- Opponent Wins: -100,000 points

## 7 Machine Learning vs. Minimax

Machine Learning's overhead exists in the learning processes. After that it can run very quickly. Minimax has to perform its search every move, which takes a considerable amount of time.

However, Minimax is guaranteed to make the most optimal move based on the state of the board; whereas the nerual network will only make the best move that it learned.

## 8 Supervised Learning with SKLearn

```
MLPClassifier(activation="tanh",
              solver="sgd",
              hidden_layer_sizes=(1000, 1000),
              learning_rate_init=0.01,
              learning_rate='constant',
              max_iter=5000,
              batch_size=5000,
```

```
        tol=0.00001,
        shuffle=True)
```

```
In [ ]: run game.py net_hard net_hard
```

### 8.0.1 Improvements

- Edit MiniMax.
- Use more data.
- Train differently.
- Train on perfect data. http://connect4.gamesolver.org/?pos=

## 9   Sources

https://github.com/JimmyJHickey/Physics-Club-Connect-Four
    https://github.com/erikackermann/Connect-Four
    http://blog.gamesolver.org/
    http://scikit-learn.org/stable/

```
In [8]: from notebook.services.config import ConfigManager
        cm = ConfigManager()
        cm.update('livereveal', {
                'width': 1100,
                'height': 680,
                'scroll': True,
        })
```

```
Out[8]: {'height': 680, 'scroll': True, 'width': 1100}
```