Demystifying Artificial Intelligence and Machine Learning

Aran Glancy
Hill-Murray School

Contact: aglancy@hill-murray.org

Slides

Teach the Mechanics

How do we make AI accessible to K-12 students?

Demystify

Empower

Inspiration

MENACE


MACHINE EDUCABLE NOUGHTS AND CROSSES ENGINE
STANDUP MATHS
MENACE
13:59

Hill-Murray Machine Learning Club

- Students in grades 6-12
- Meet during “WIN” time

Professor Nimble

Midas

The Game of Nim

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Sample Game

Sample Game

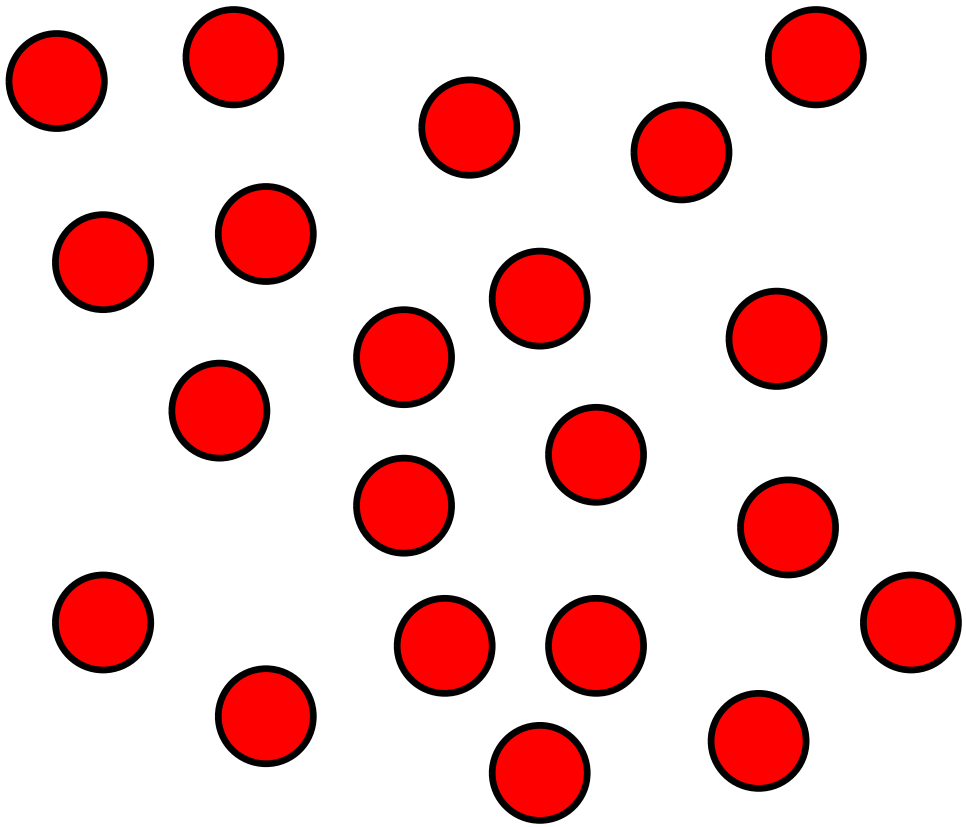
Play slideshow to see animations
Note that the “players” don’t play with any strategy until about 10 tokens left.

The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

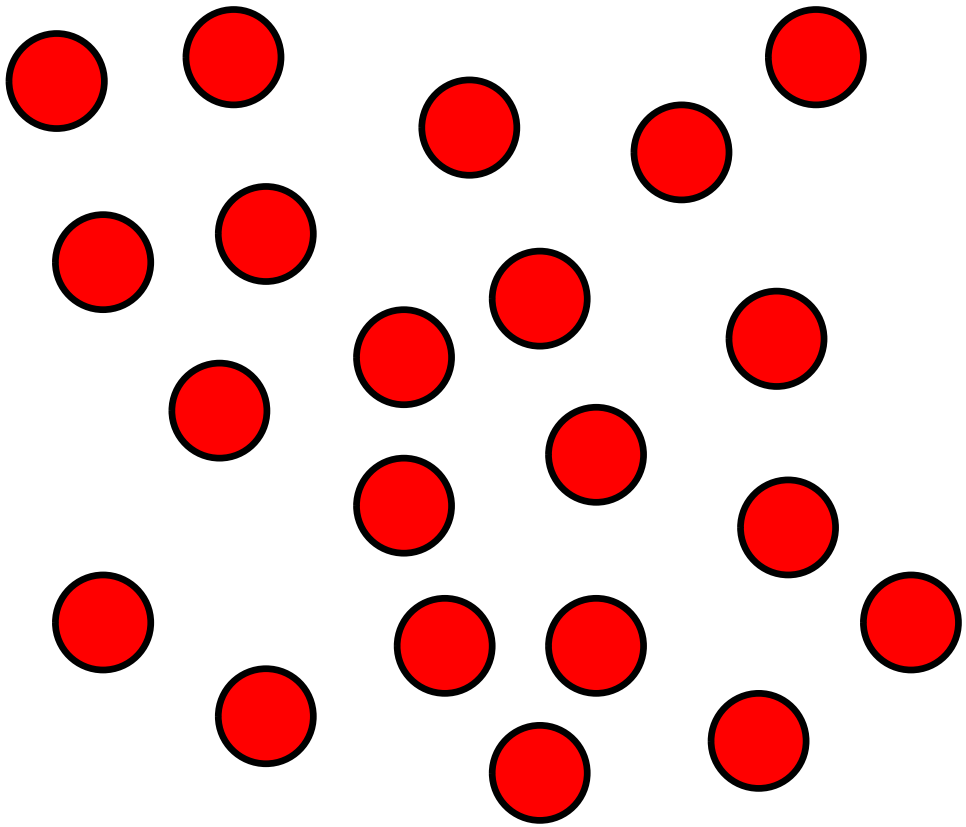


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One: 3

Player Two:

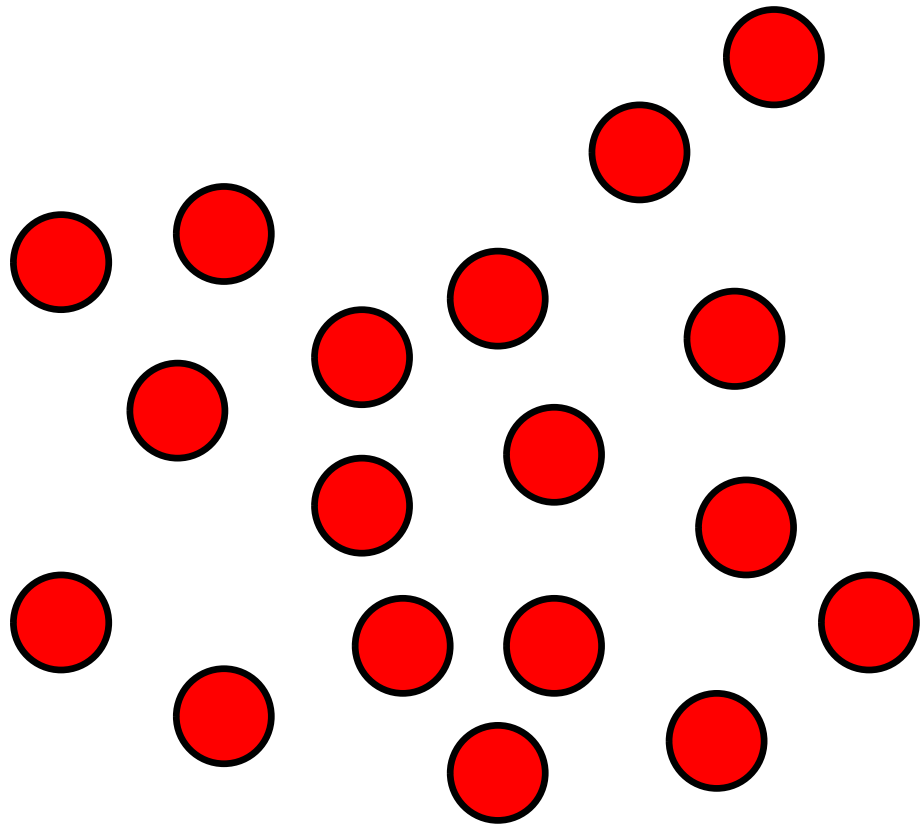


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

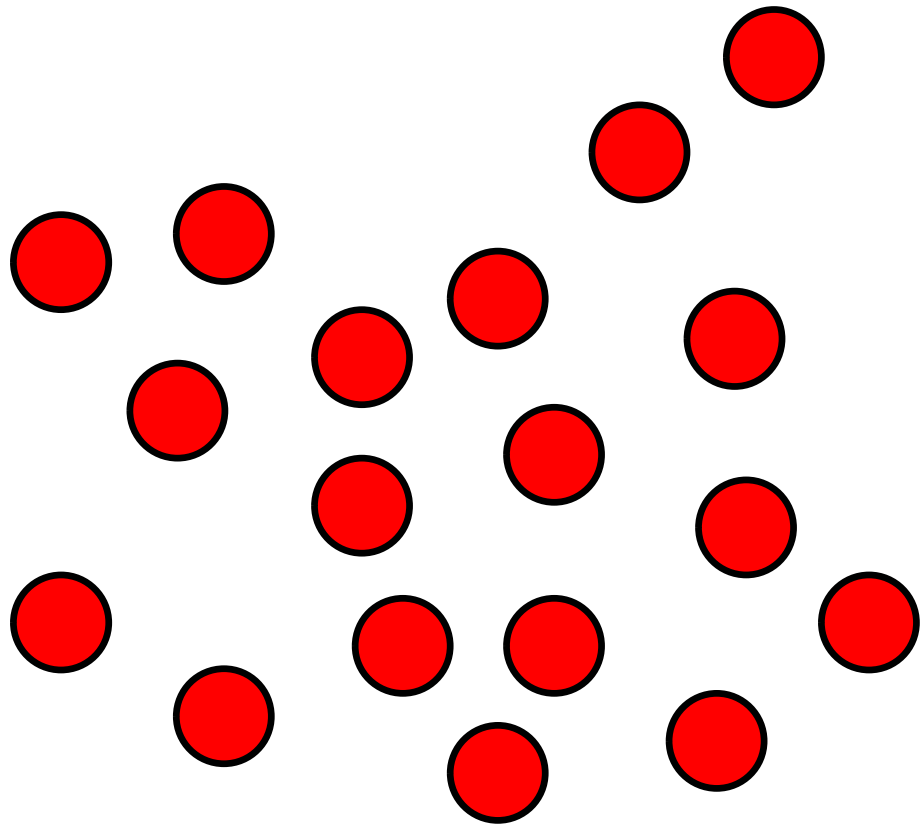


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two: 2

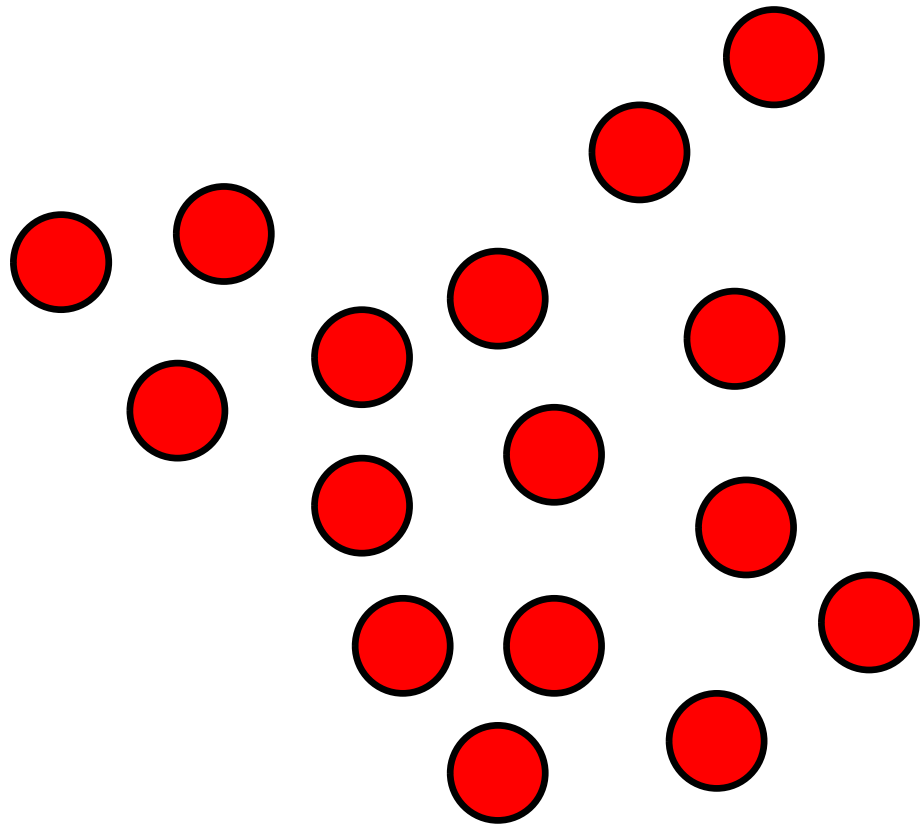


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

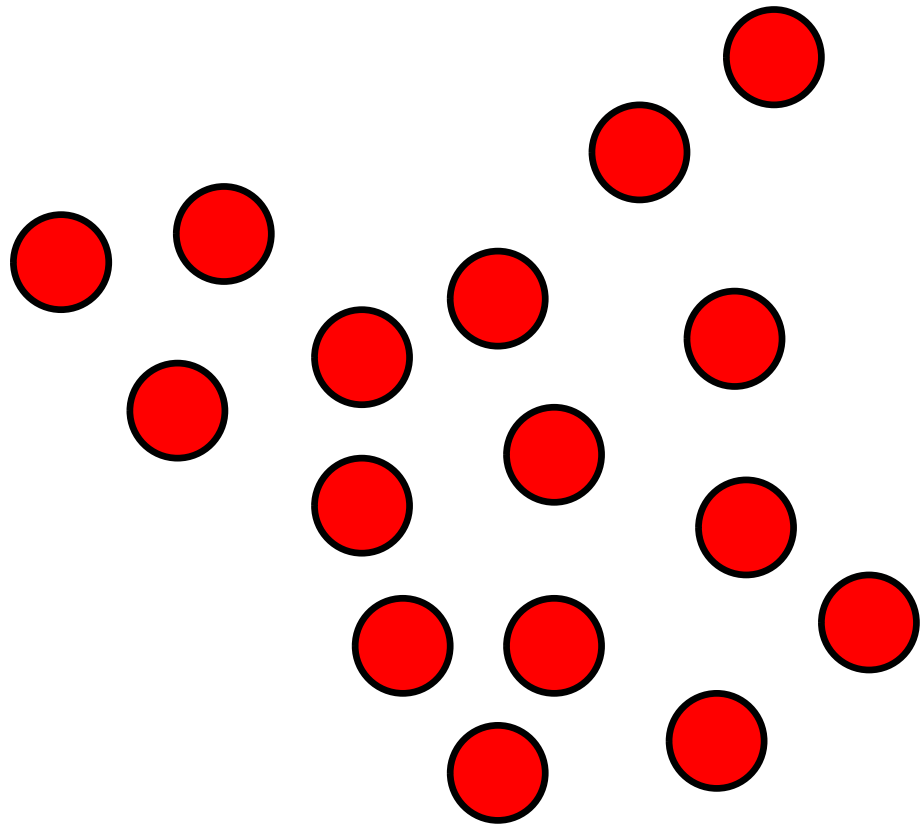


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One: 2

Player Two:

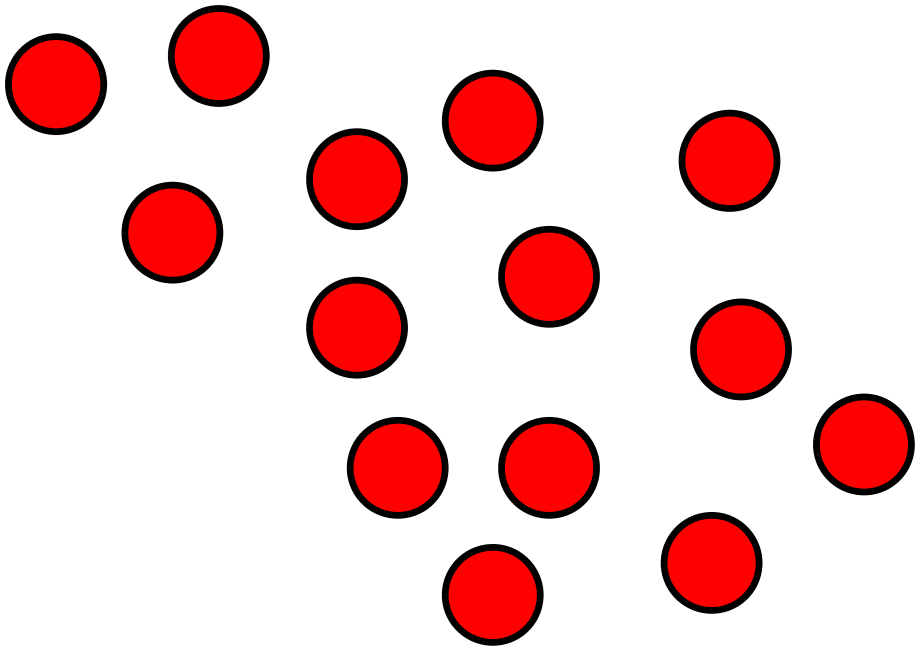


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

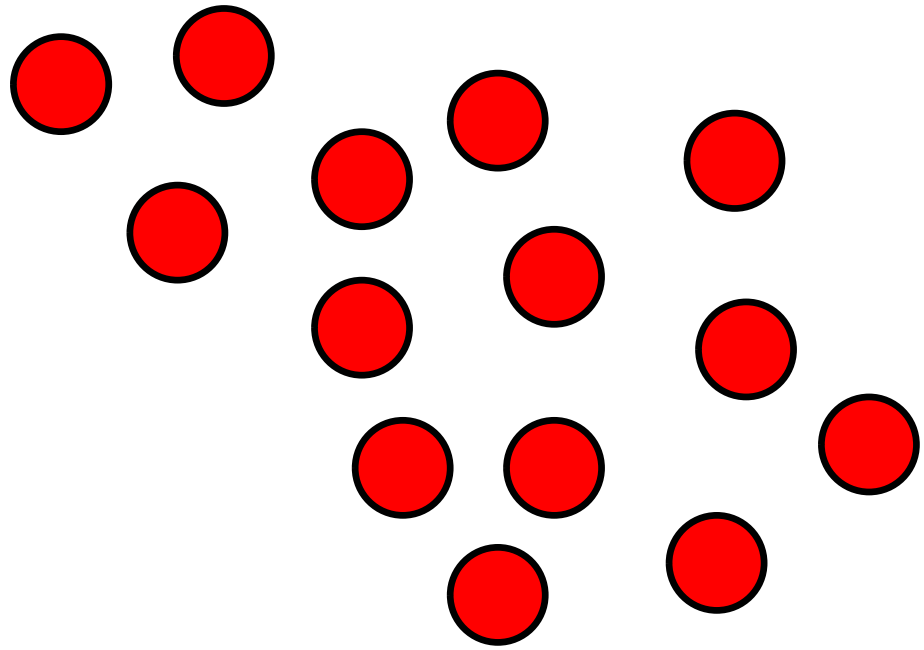


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two: 3

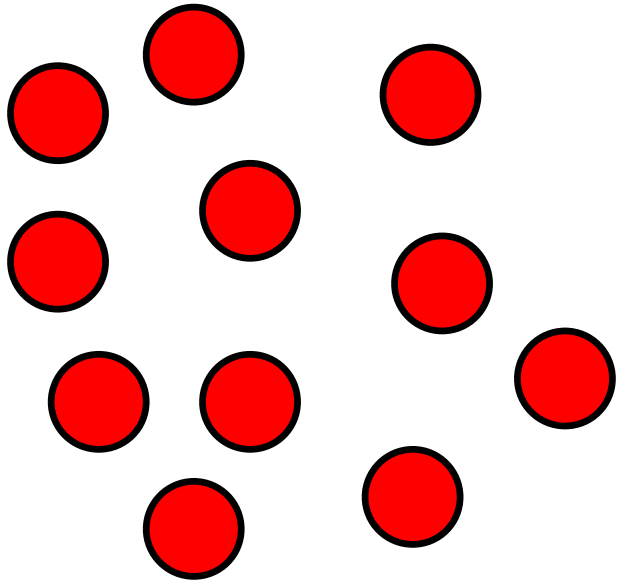


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

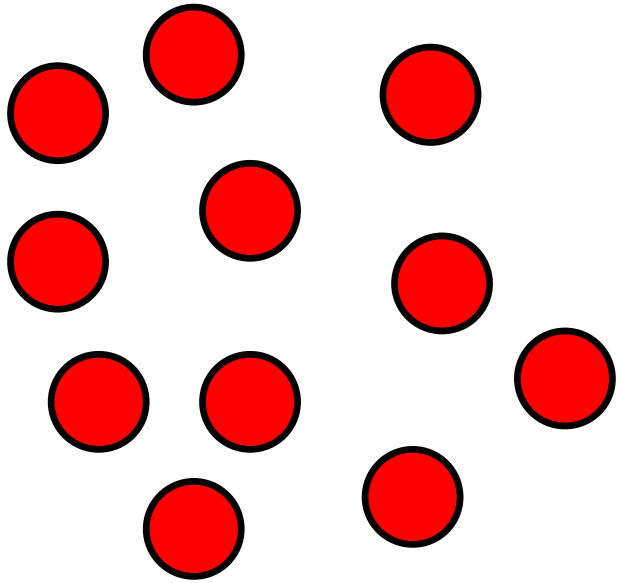


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One: 1

Player Two:

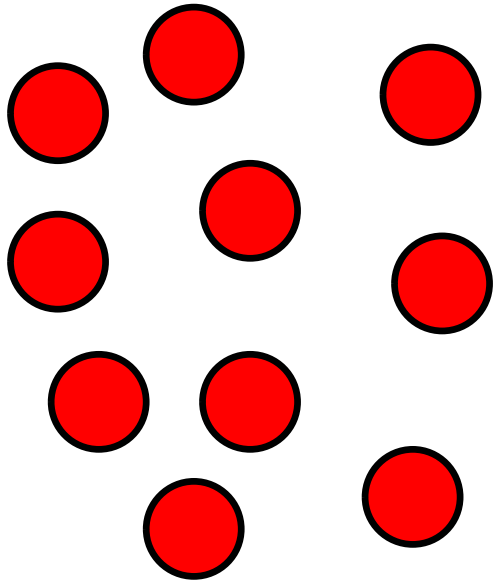


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

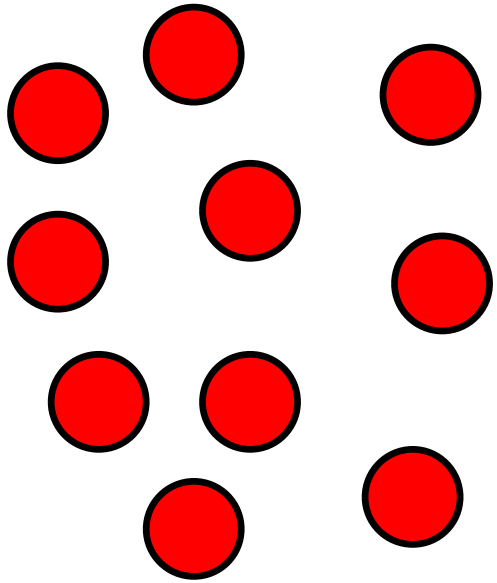


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two: 1

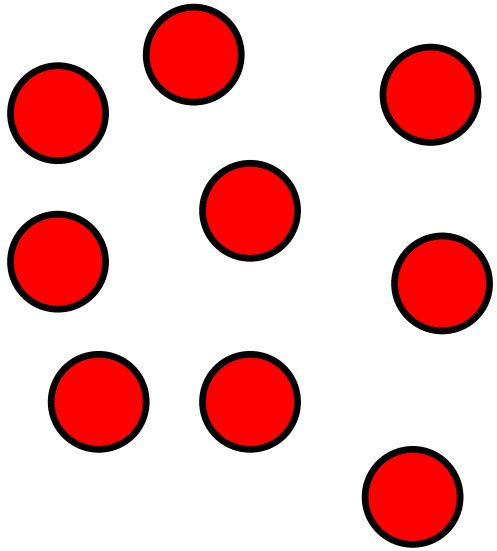


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

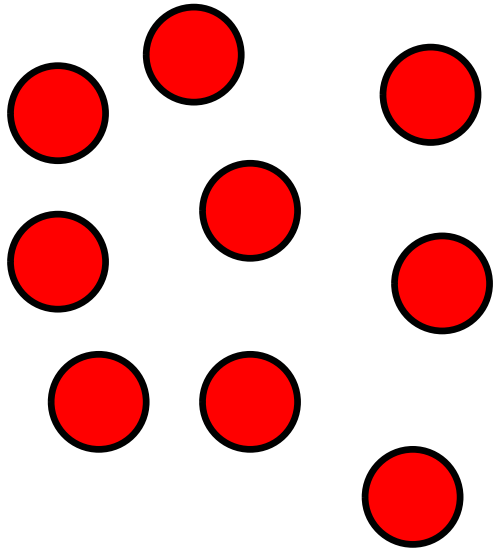


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One: 2

Player Two:

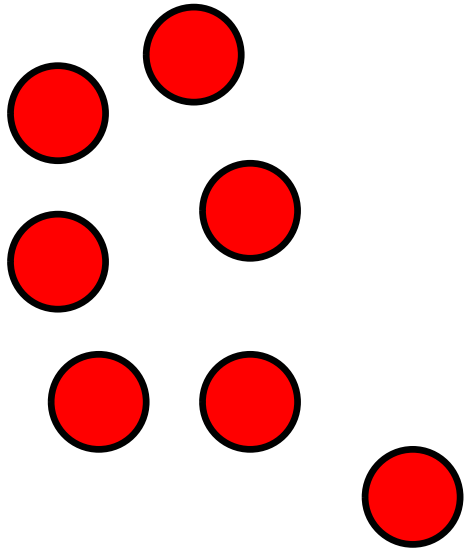


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

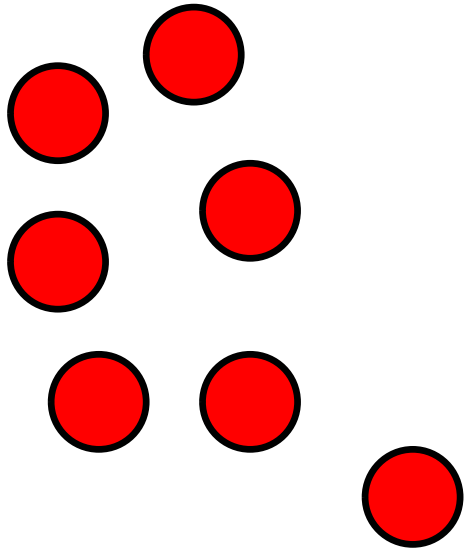


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two: 2

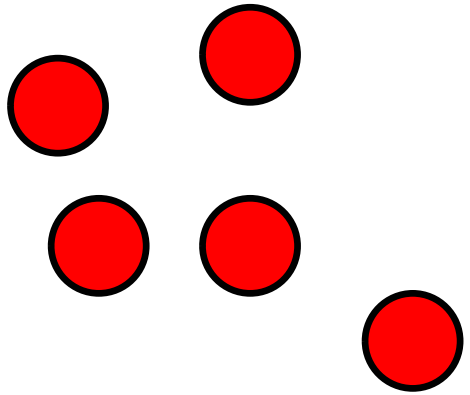


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

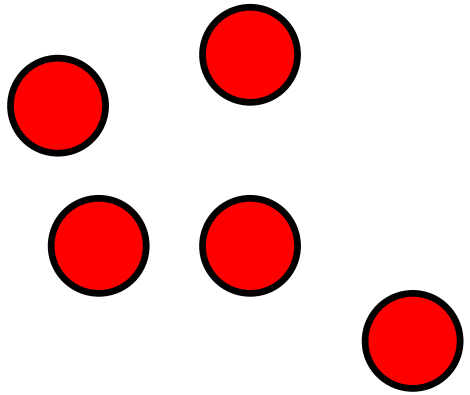


The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One: 3

Player Two:



The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:

The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

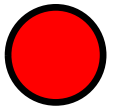
Player Two: 1

The Game of Nim - Sample Game

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One:

Player Two:



The Game of Nim - Sample Game

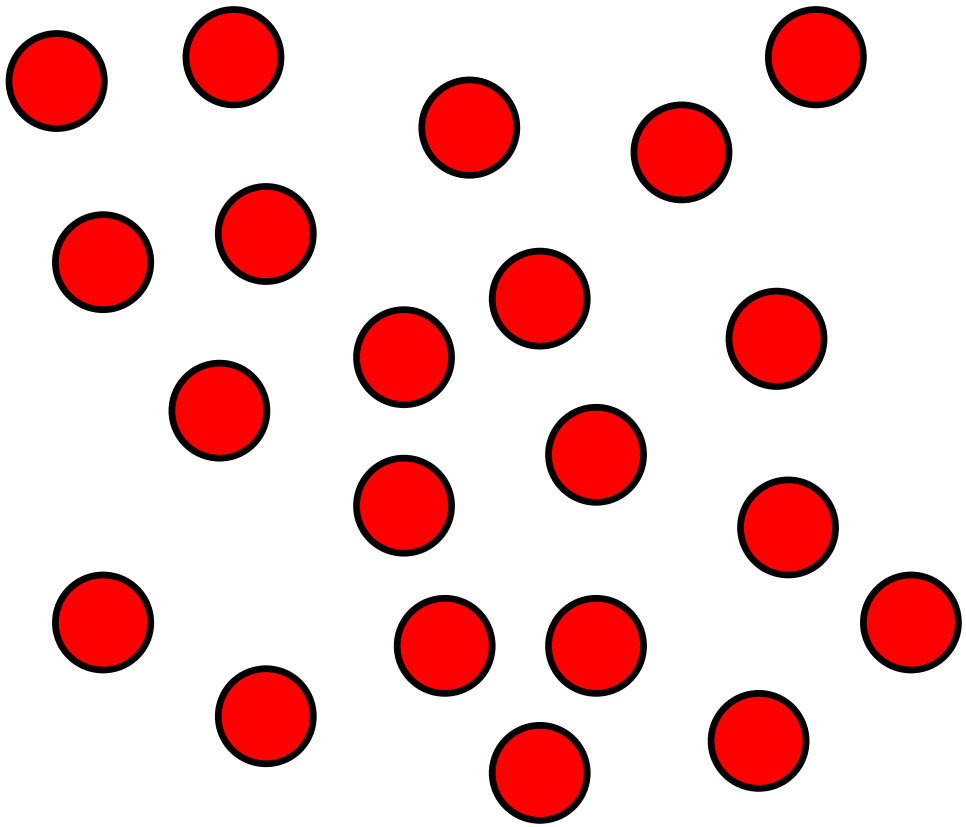
1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

Player One: Must take the last one!

Player Two:

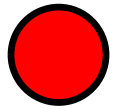
The Game of Nim

1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!

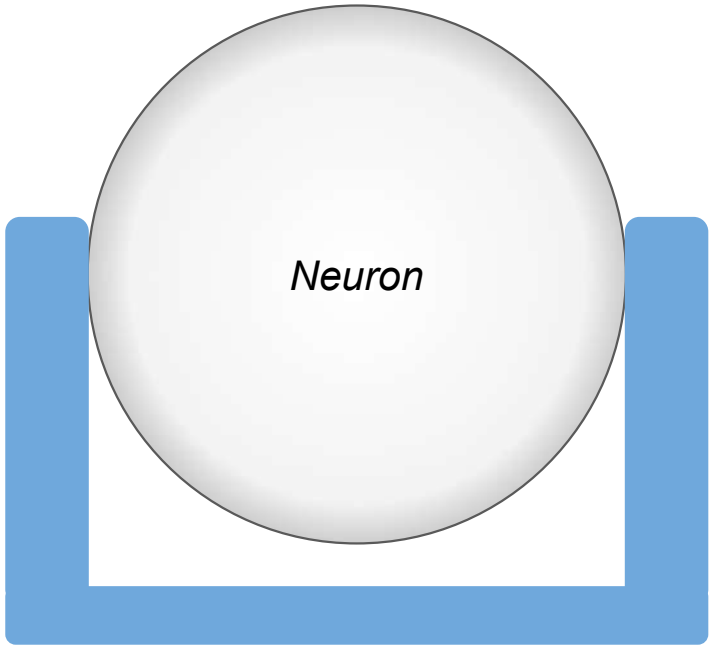


The Game of Nim

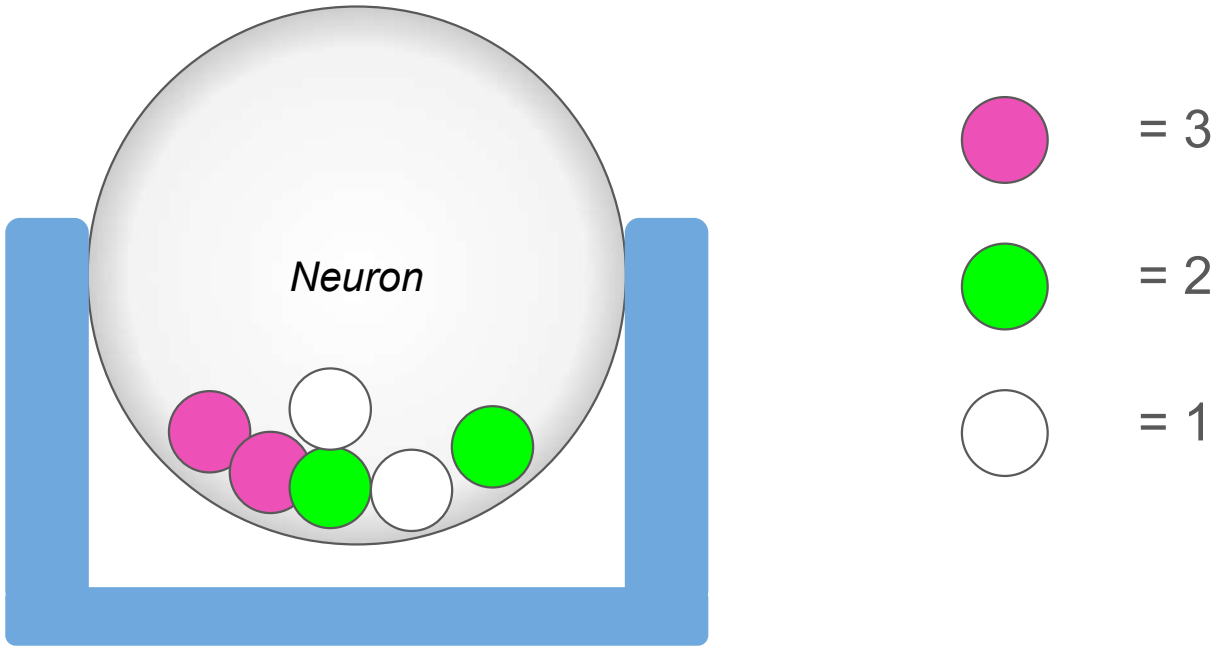
1. Players alternate
2. Remove 1, 2 or 3 items
3. Don’t remove the last one!



Professor Nimble - The Analog AI



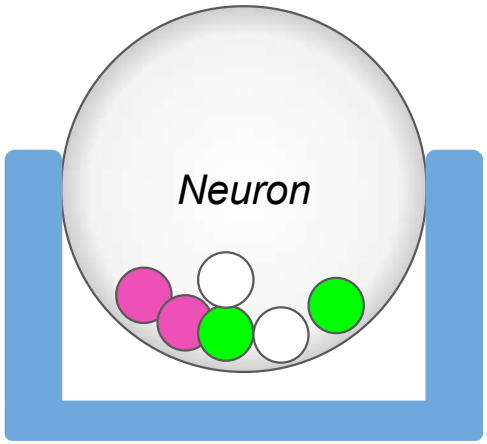
Professor Nimble - The Analog AI



How Professor Nimble Plays

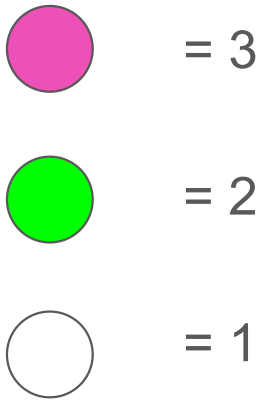
1. Find the Neuron

We use the neuron that correspond with the number of tokens remaining



2. Choose A Move

The 3 colored balls represent Prof. Nimbles Moves. It chooses randomly.

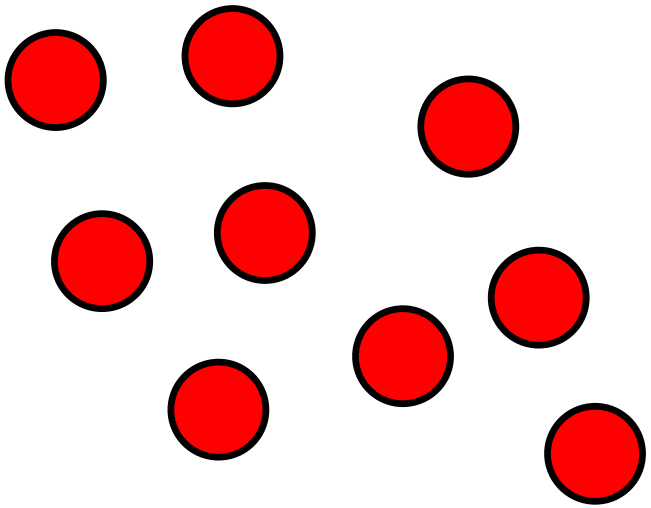


3. Learn

If Prof. Nimble wins, we replace the balls it chose and add another of that color.

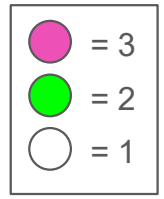
If Prof. Nimble loses, we remove the balls it chose without replacing them.

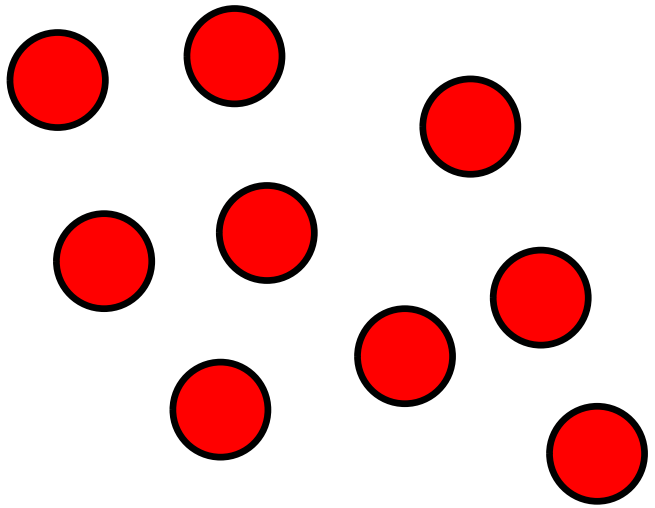
Professor Nimble



Example (Game 1)

Professor Nimble

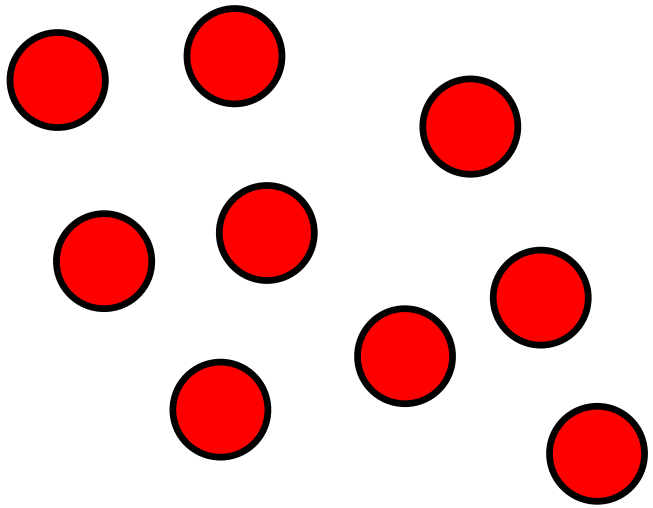




Professor Nimble

= 3
= 2
= 1

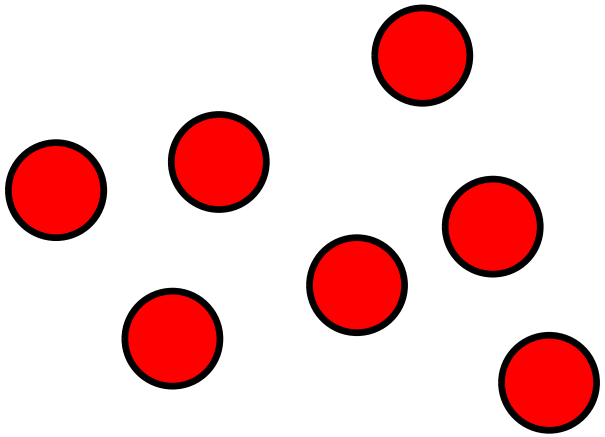
Player Move: 2



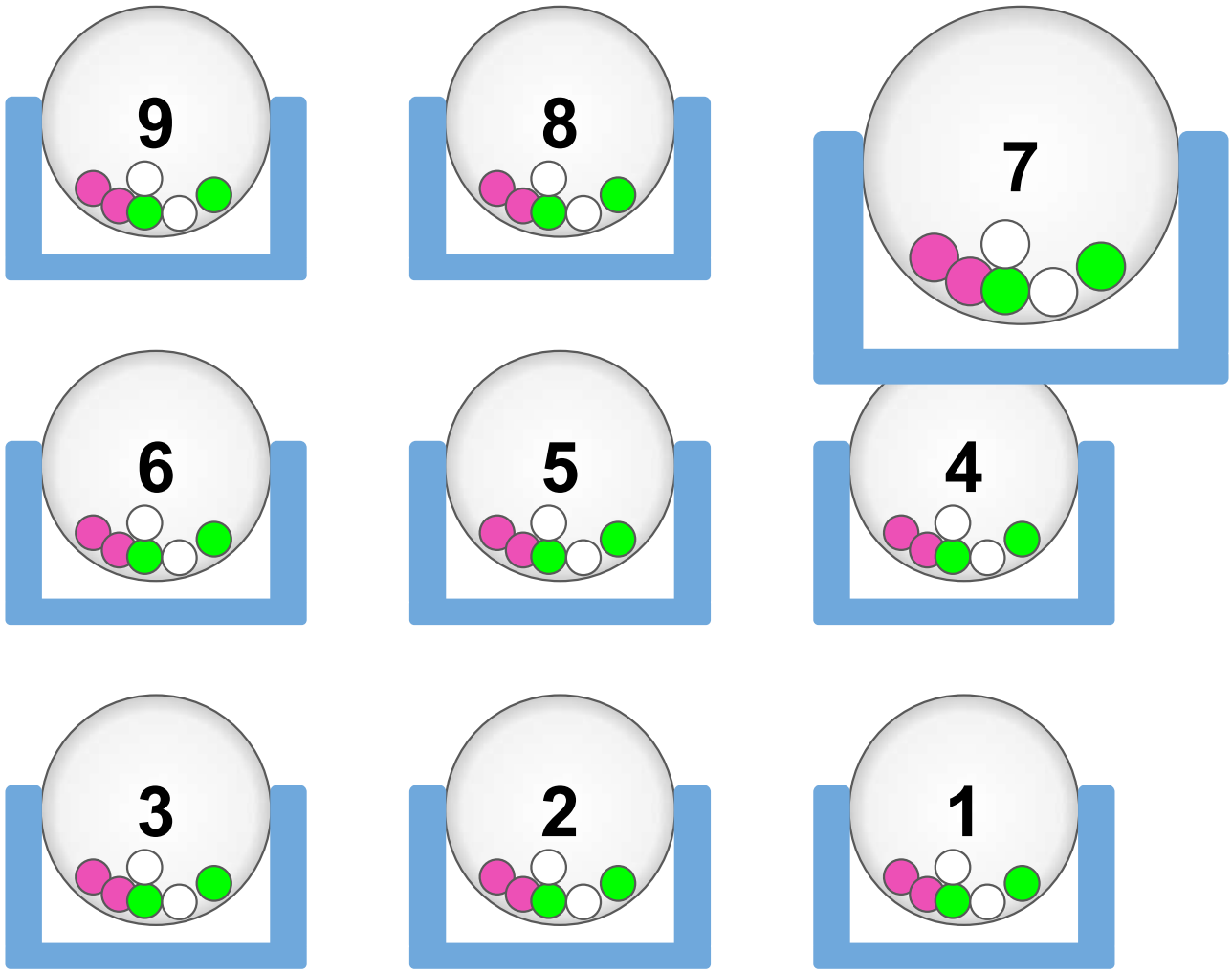
Professor Nimble

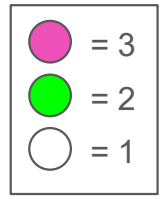
= 3
= 2
= 1

Prof. Nimble:

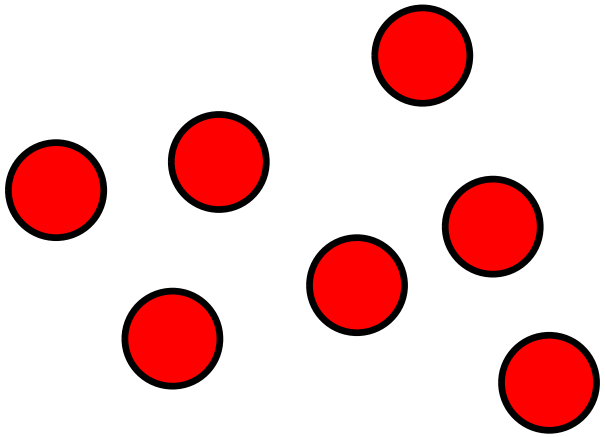


Professor Nimble

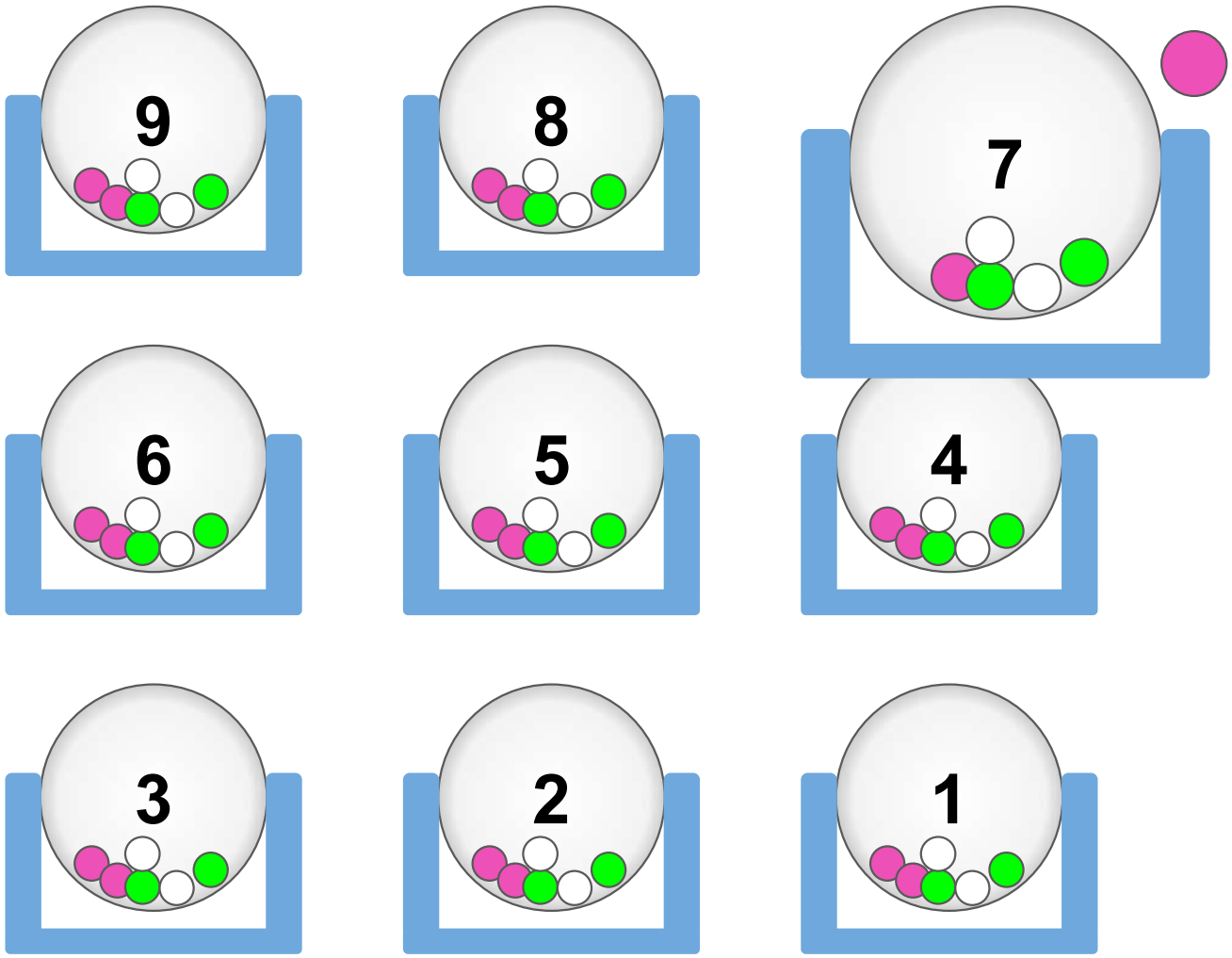




Prof. Nimble:

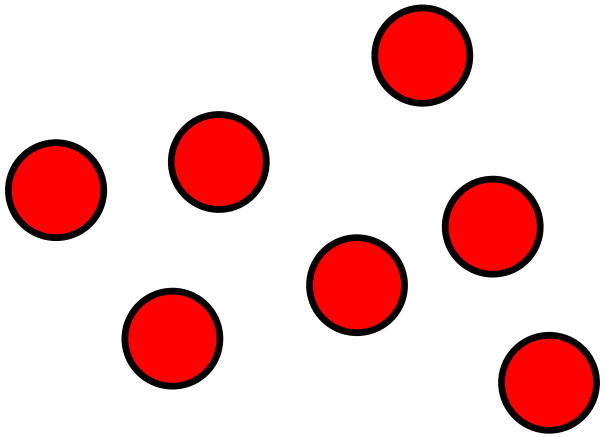


Professor Nimble



= 3
= 2
= 1

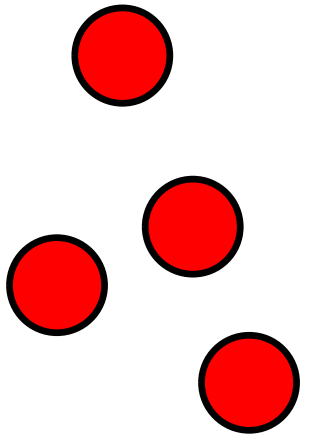
Prof. Nimble: 3



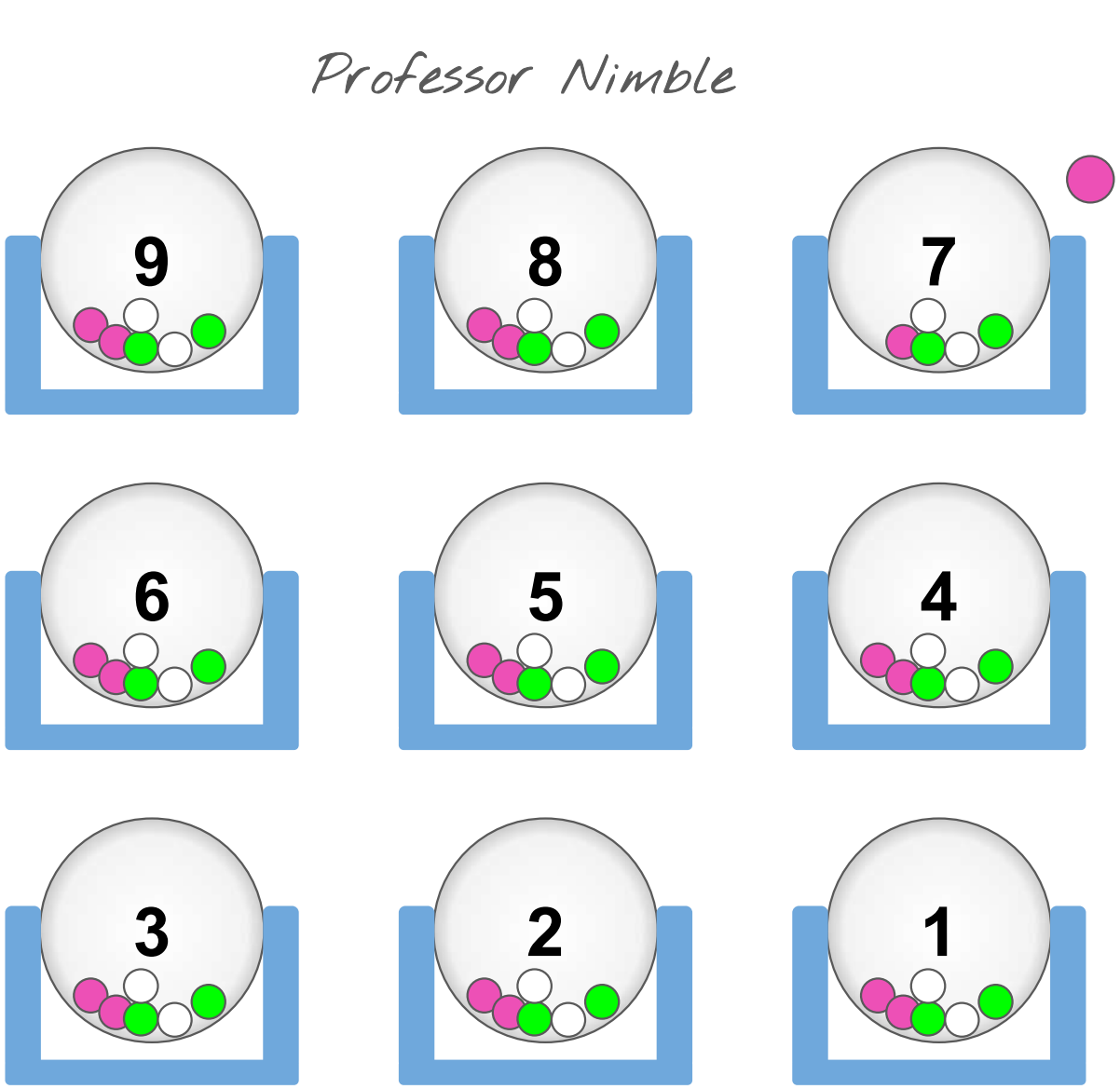
Professor Nimble

= 3
= 2
= 1

Player Move: 2

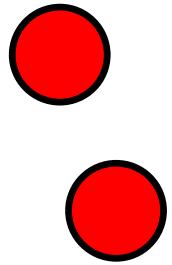


Professor Nimble

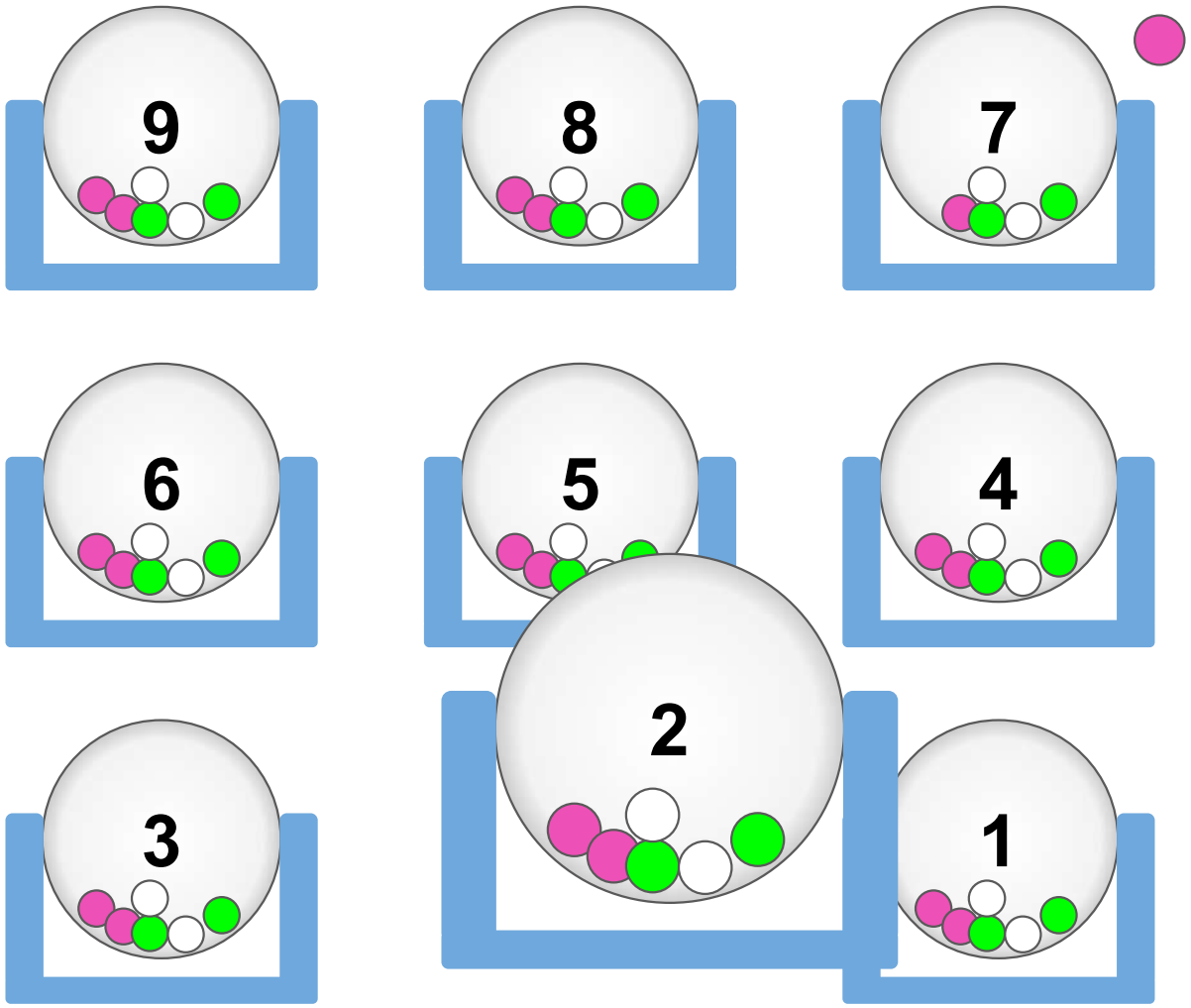


= 3
= 2
= 1

Prof. Nimble:

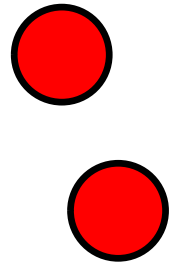


Professor Nimble

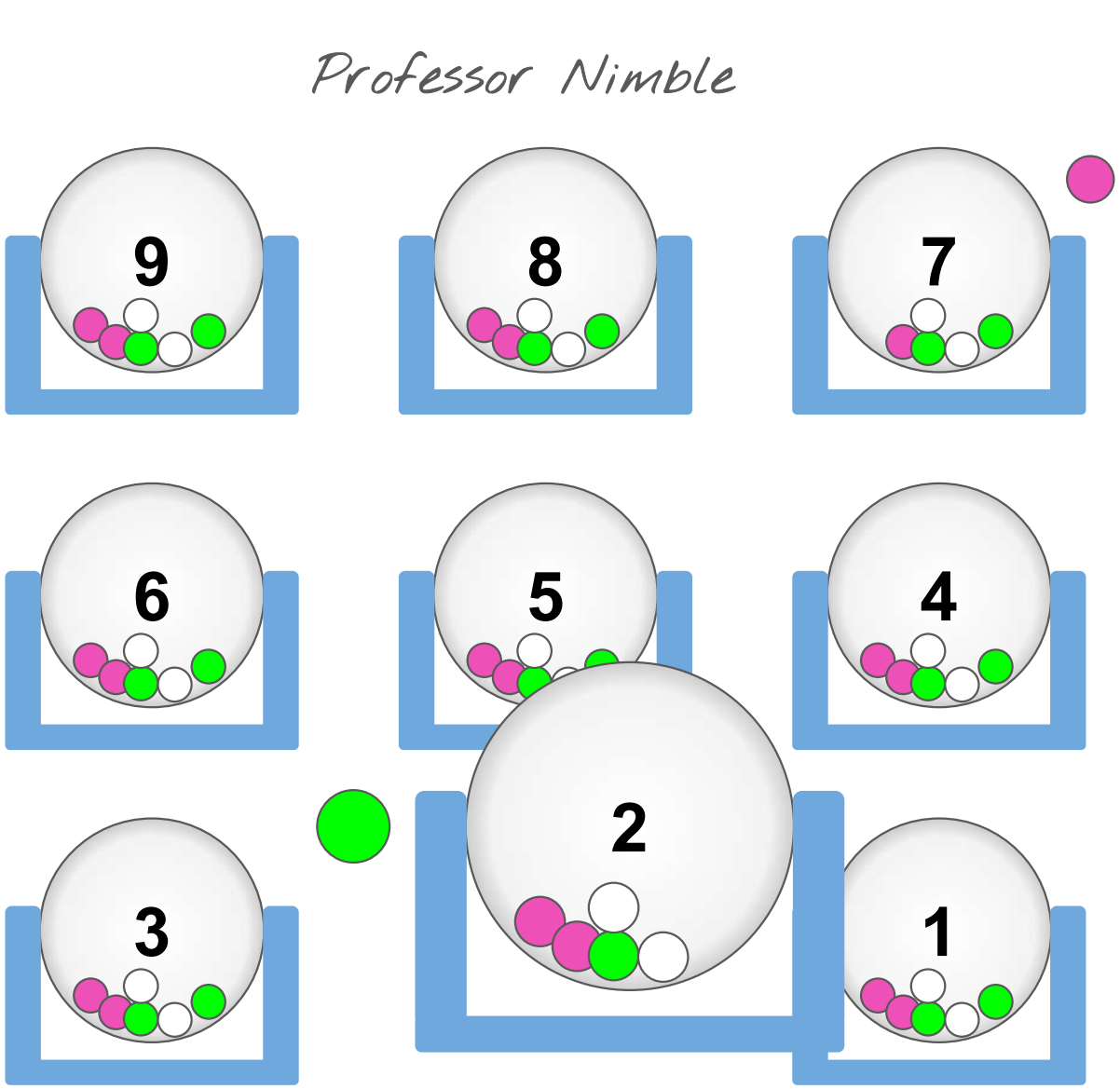


= 3
= 2
= 1

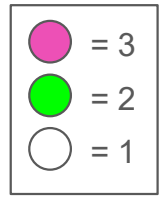
Prof. Nimble:

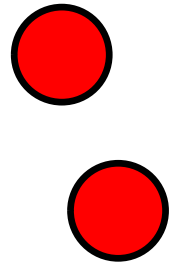


Professor Nimble

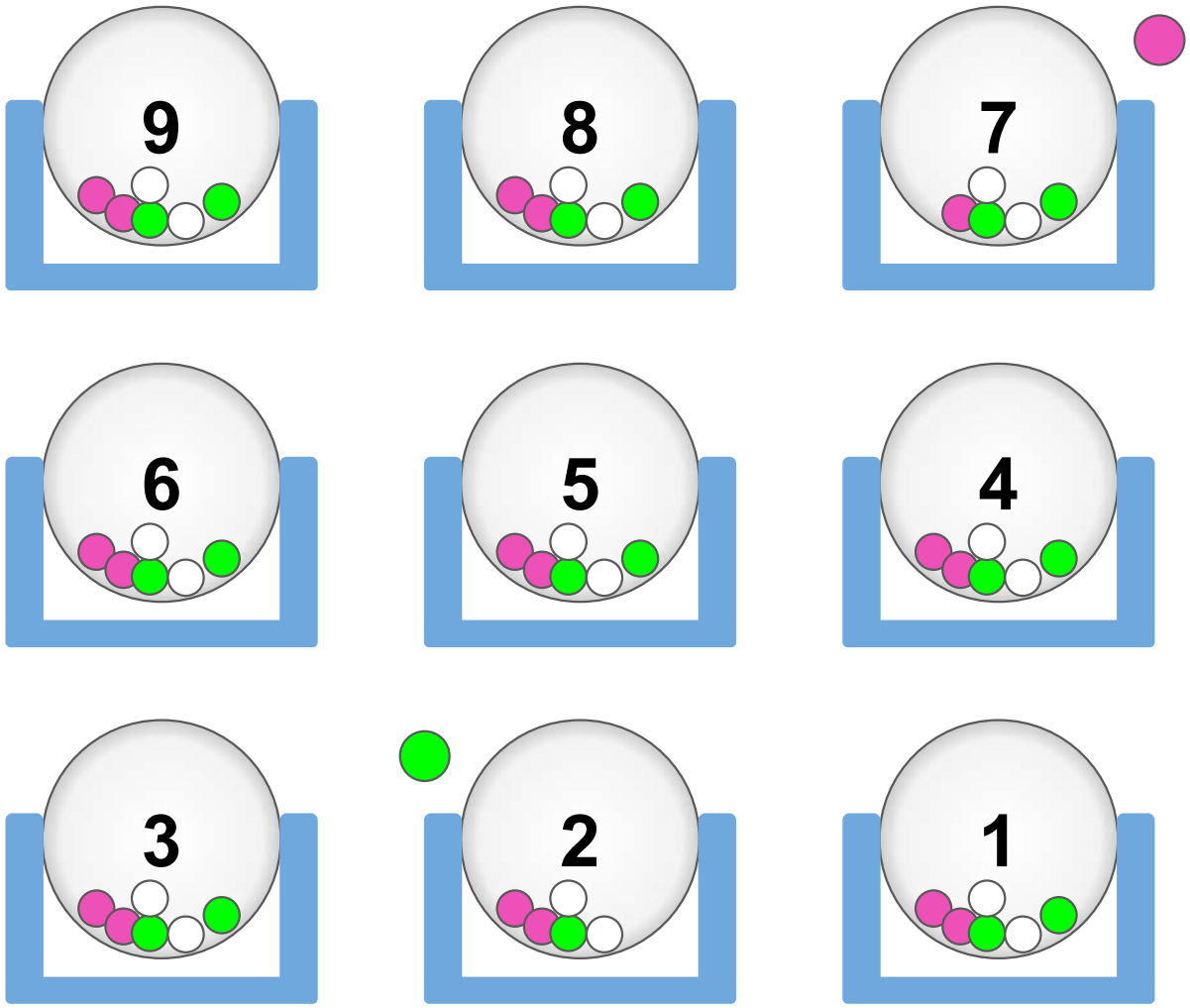


Prof. Nimble: 2

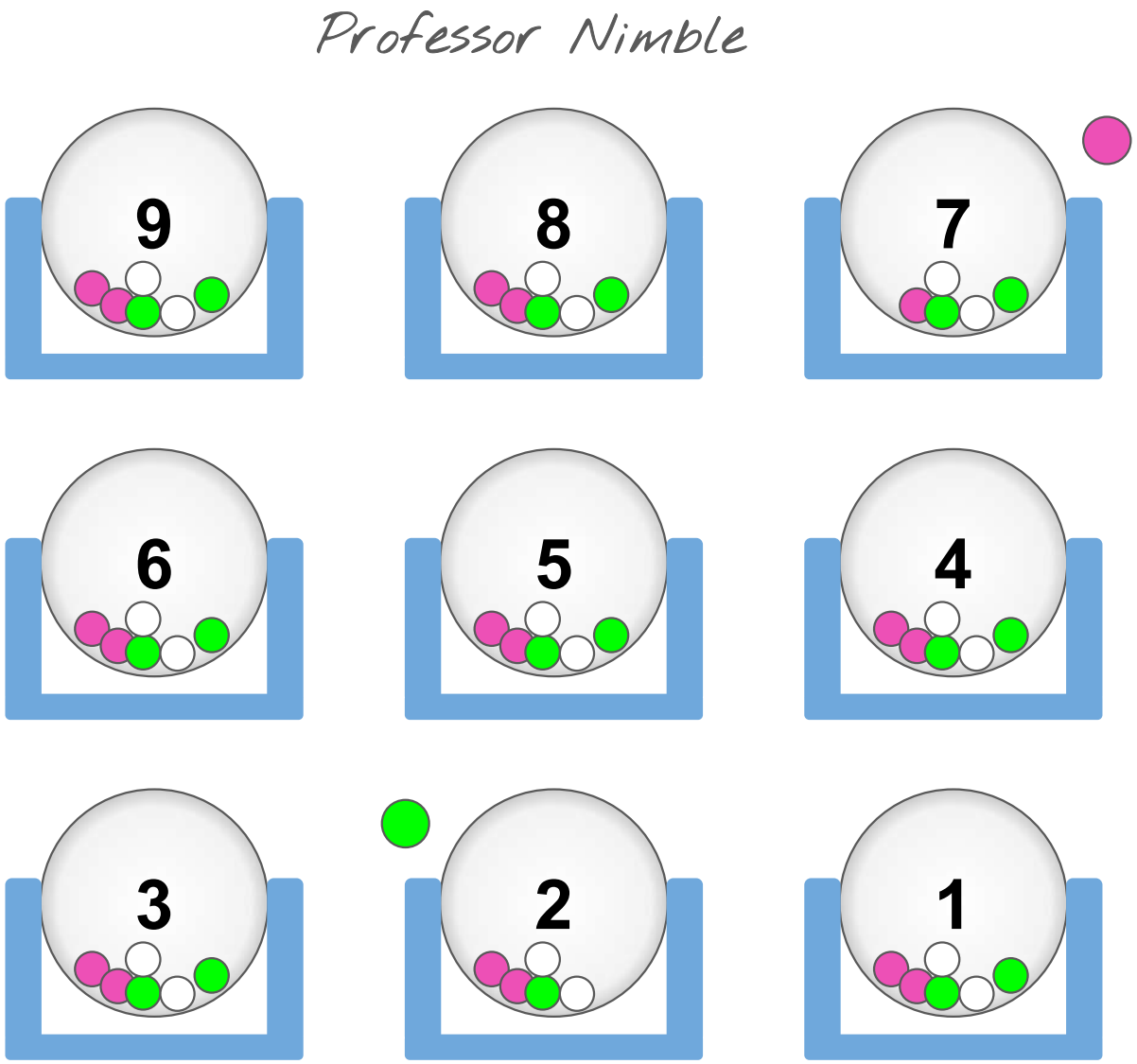




Professor Nimble



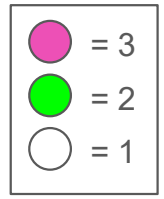
Player Wins!

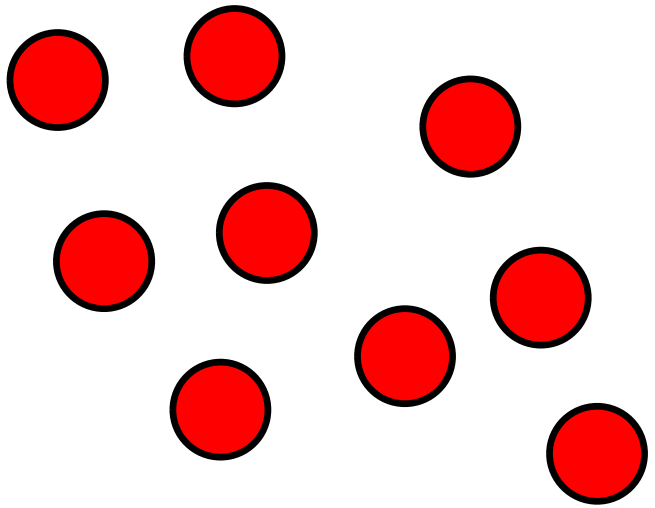


Professor Nimble "learns" that those were bad moves.

Example (Game 2)

Professor Nimble

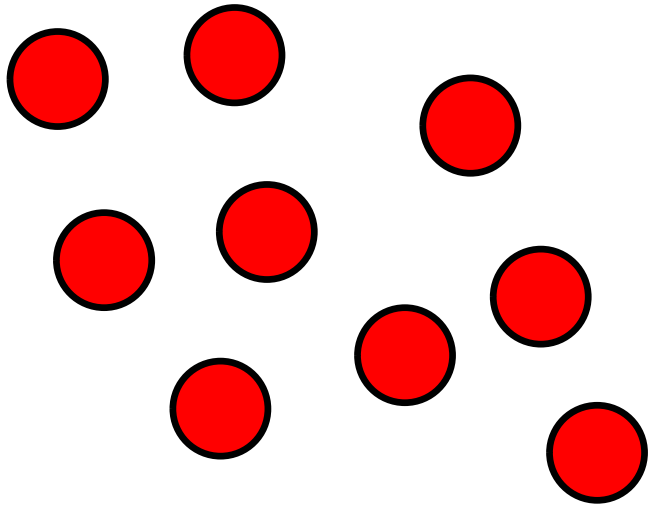




Professor Nimble

= 3
= 2
= 1

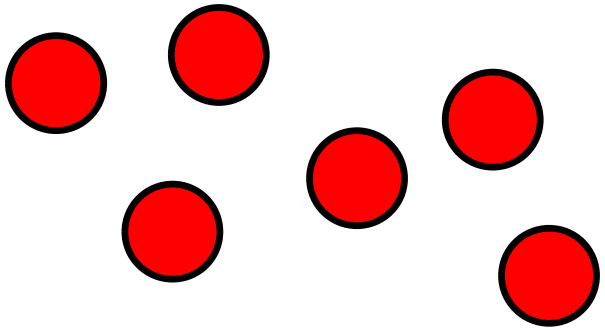
Player Move: 3



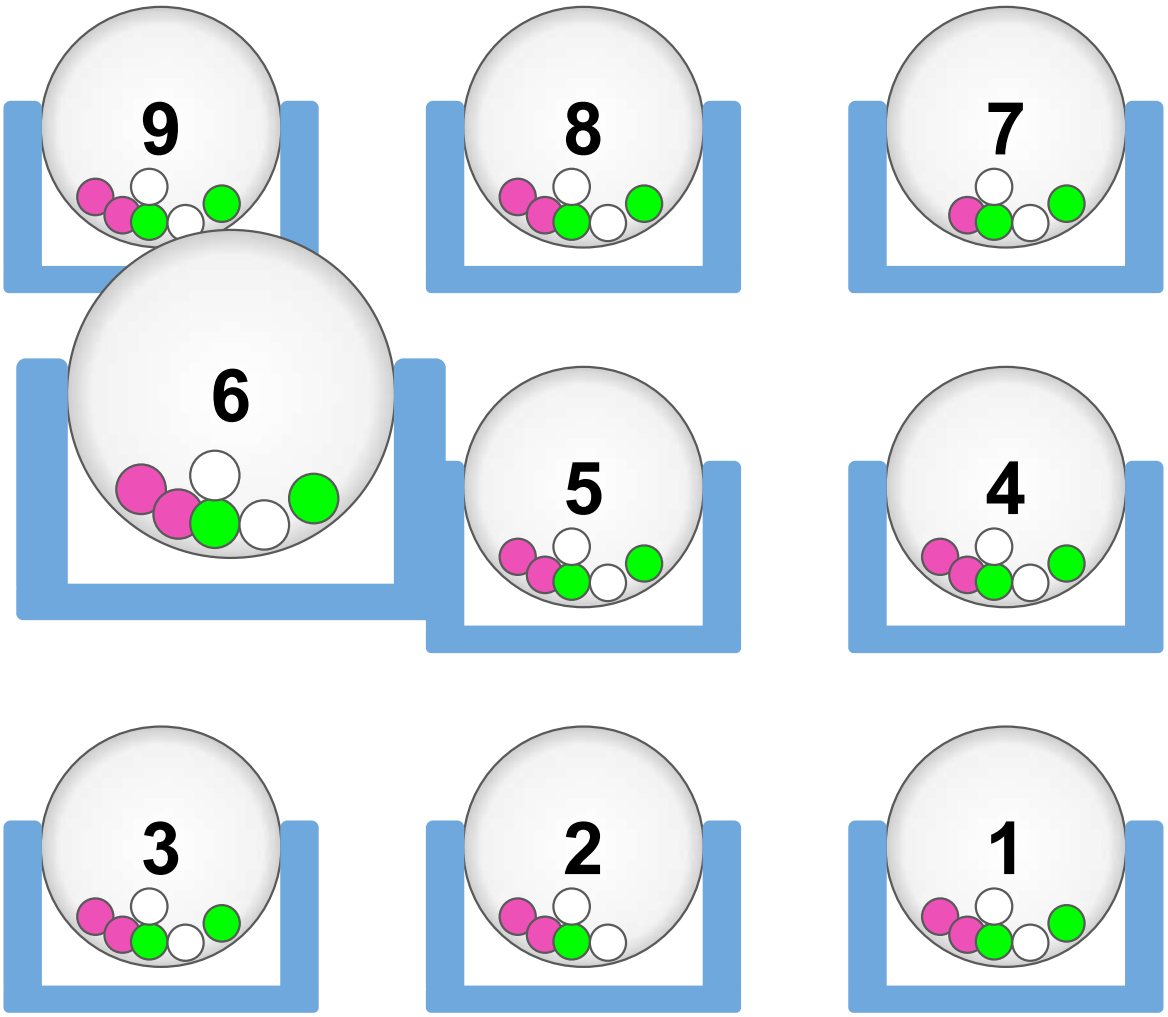
Professor Nimble

= 3
= 2
= 1

Prof. Nimble:

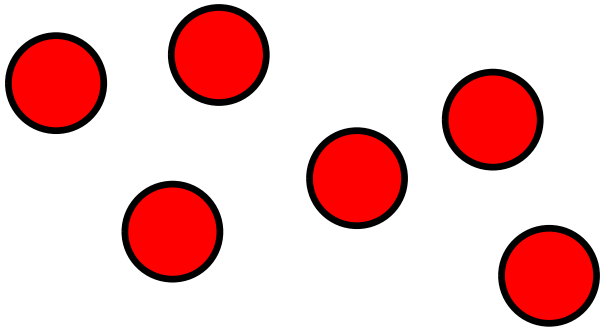


Professor Nimble

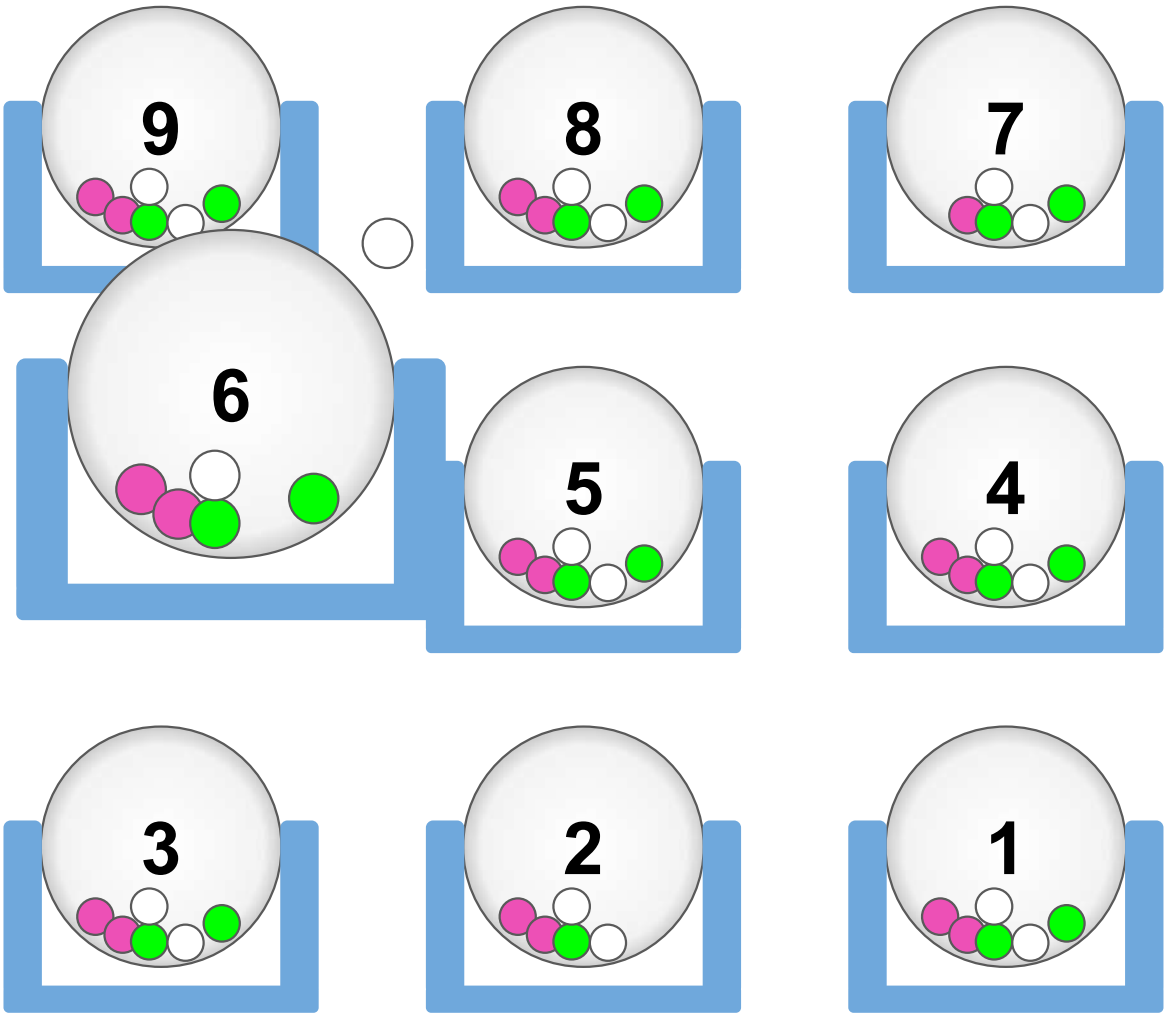


= 3
= 2
= 1

Prof. Nimble:

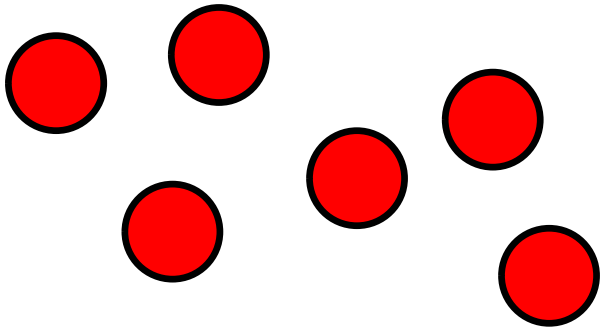


Professor Nimble



= 3
= 2
= 1

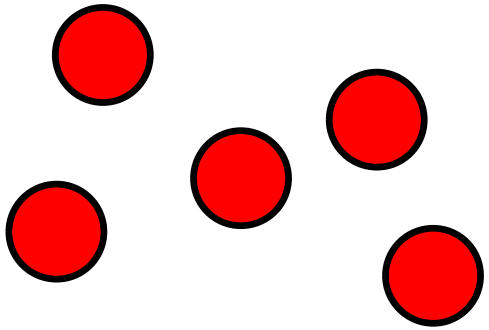
Prof. Nimble: 1



Professor Nimble

= 3
= 2
= 1

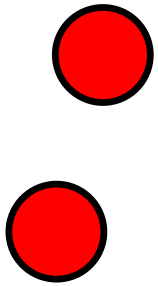
Player Move: 3



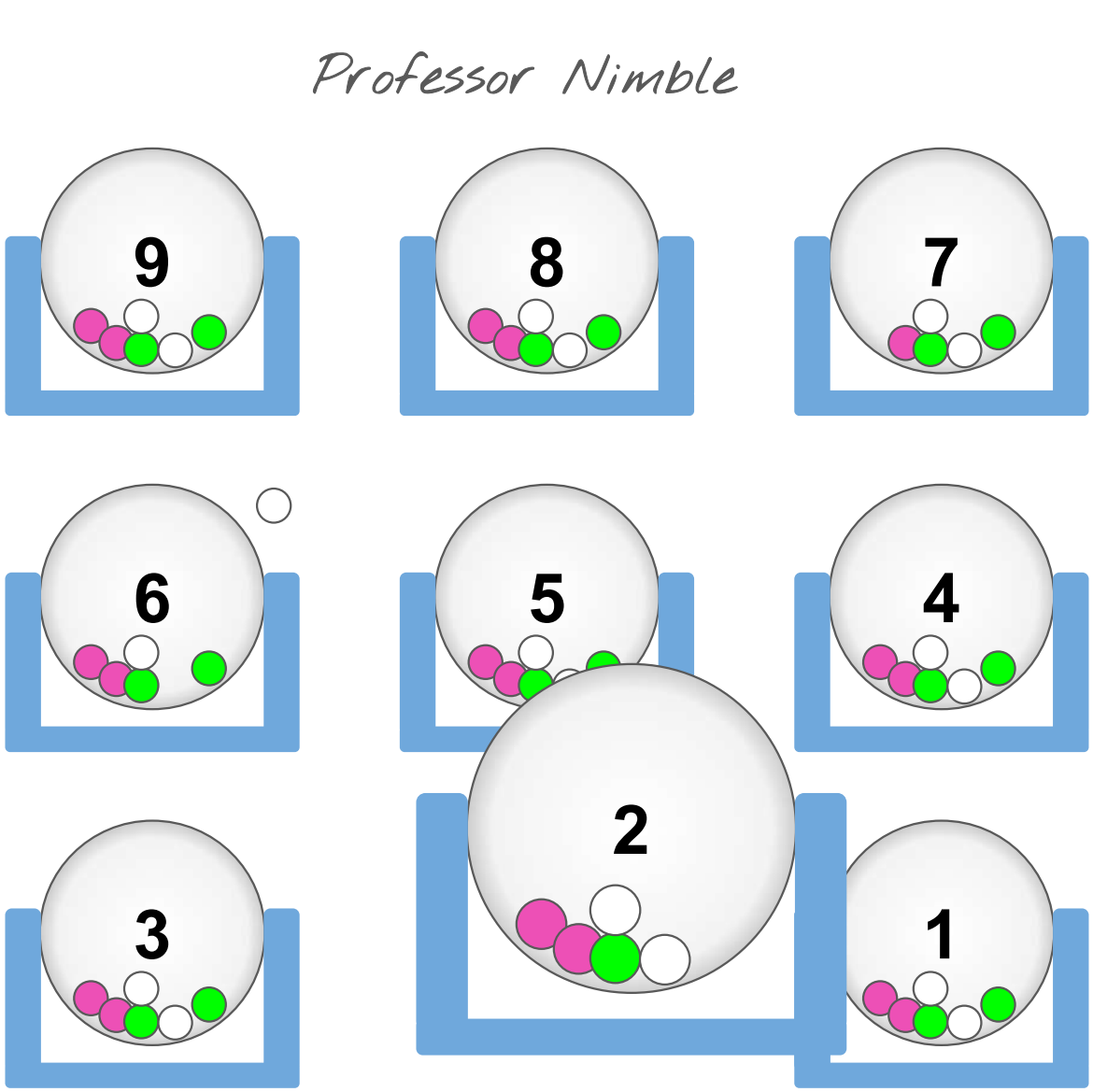
Professor Nimble

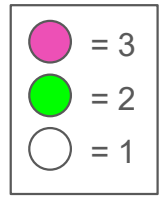
= 3
= 2
= 1

Prof. Nimble:

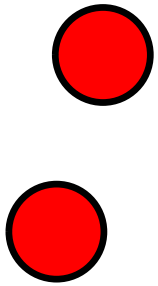


Professor Nimble

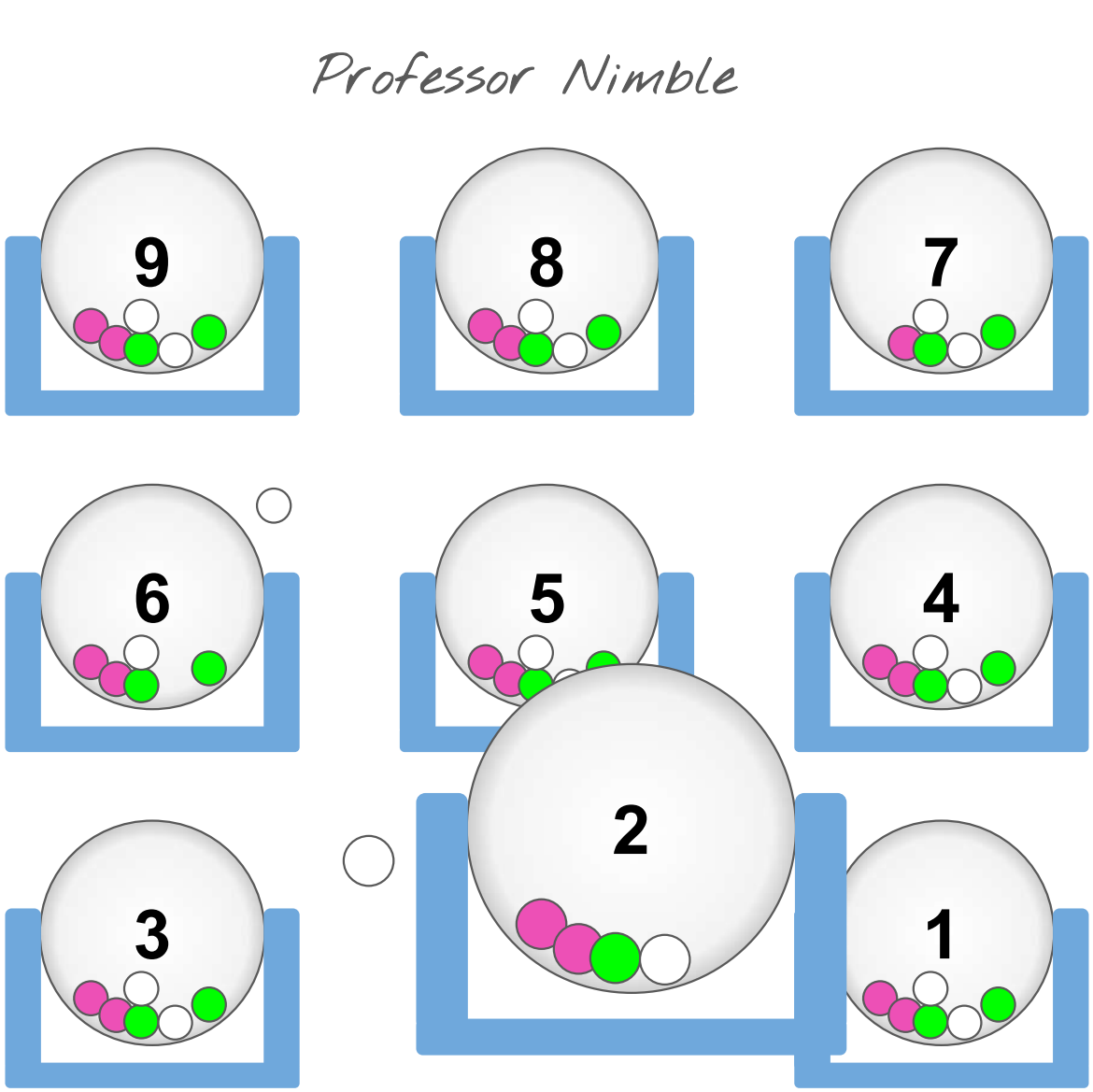


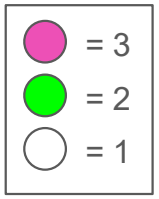


Prof. Nimble:

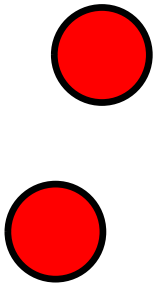


Professor Nimble





Prof. Nimble: 1

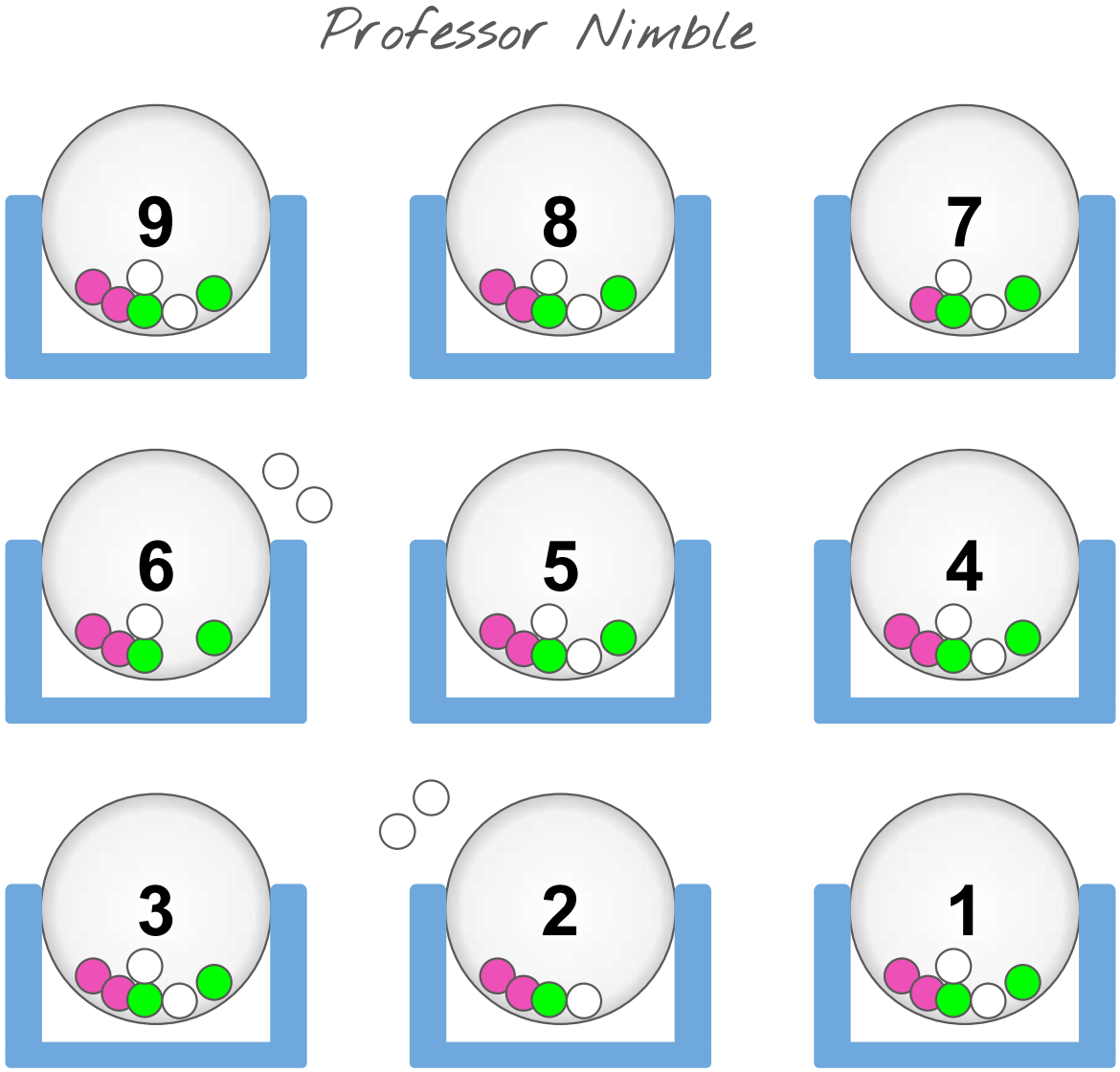


Professor Nimble

= 3
= 2
= 1

Prof. Nimble: 1

Nimble Wins!



Professor Nimble "learns" that those were good moves.

Professor Nimble

Professor Nimble "learns" that those were good moves.

Watch the Learning

Professor Nimble

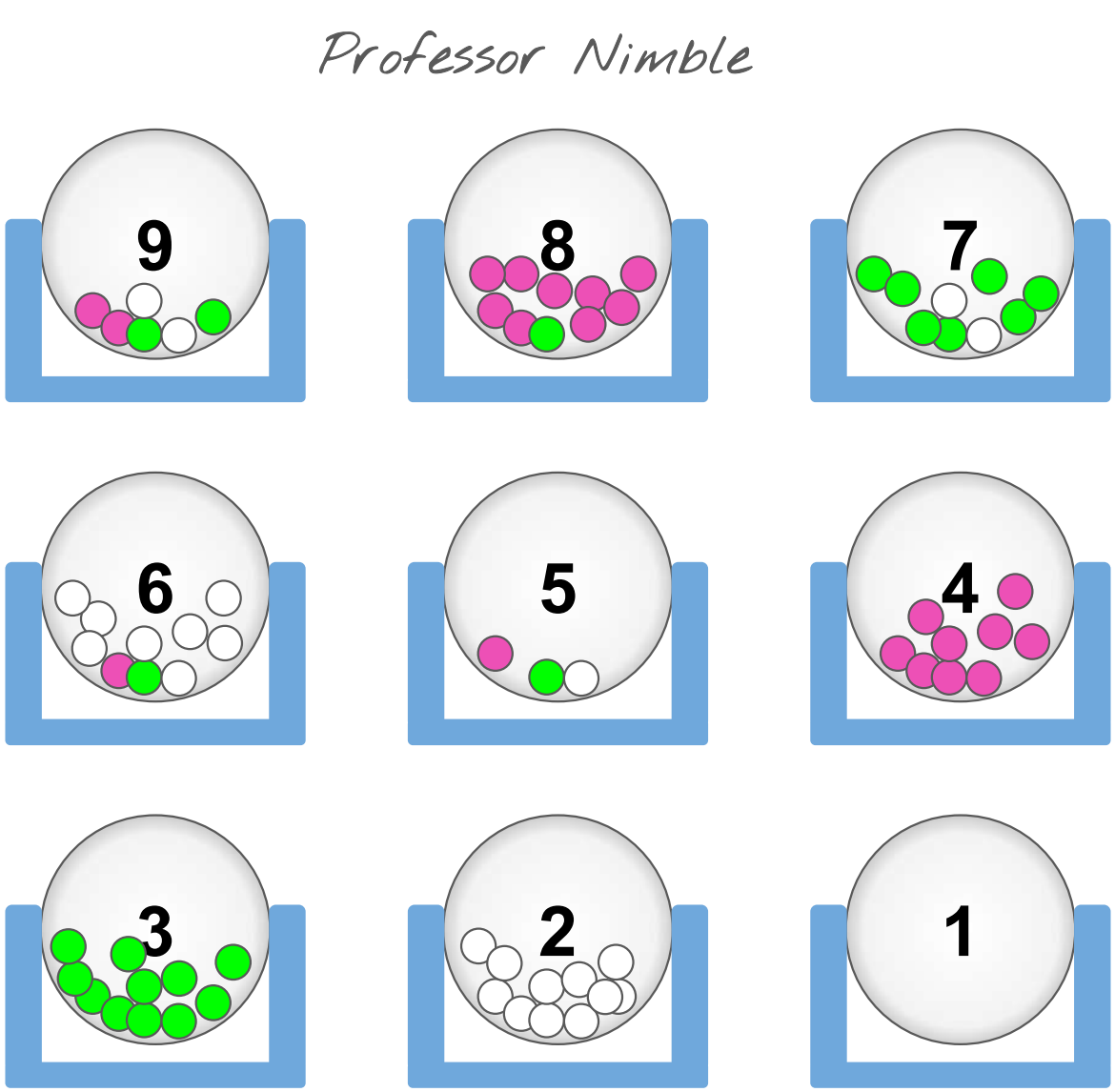
= 3

= 2

= 1

Professor Nimble after a few games

(some wins, some losses)



= 3

= 2

= 1

Professor Nimble after many games

The Real Professor Nimble


2025
Professor
Nimble
TEXAS
VIKINGS
HATS ALLOWED
Trivia during Commercials

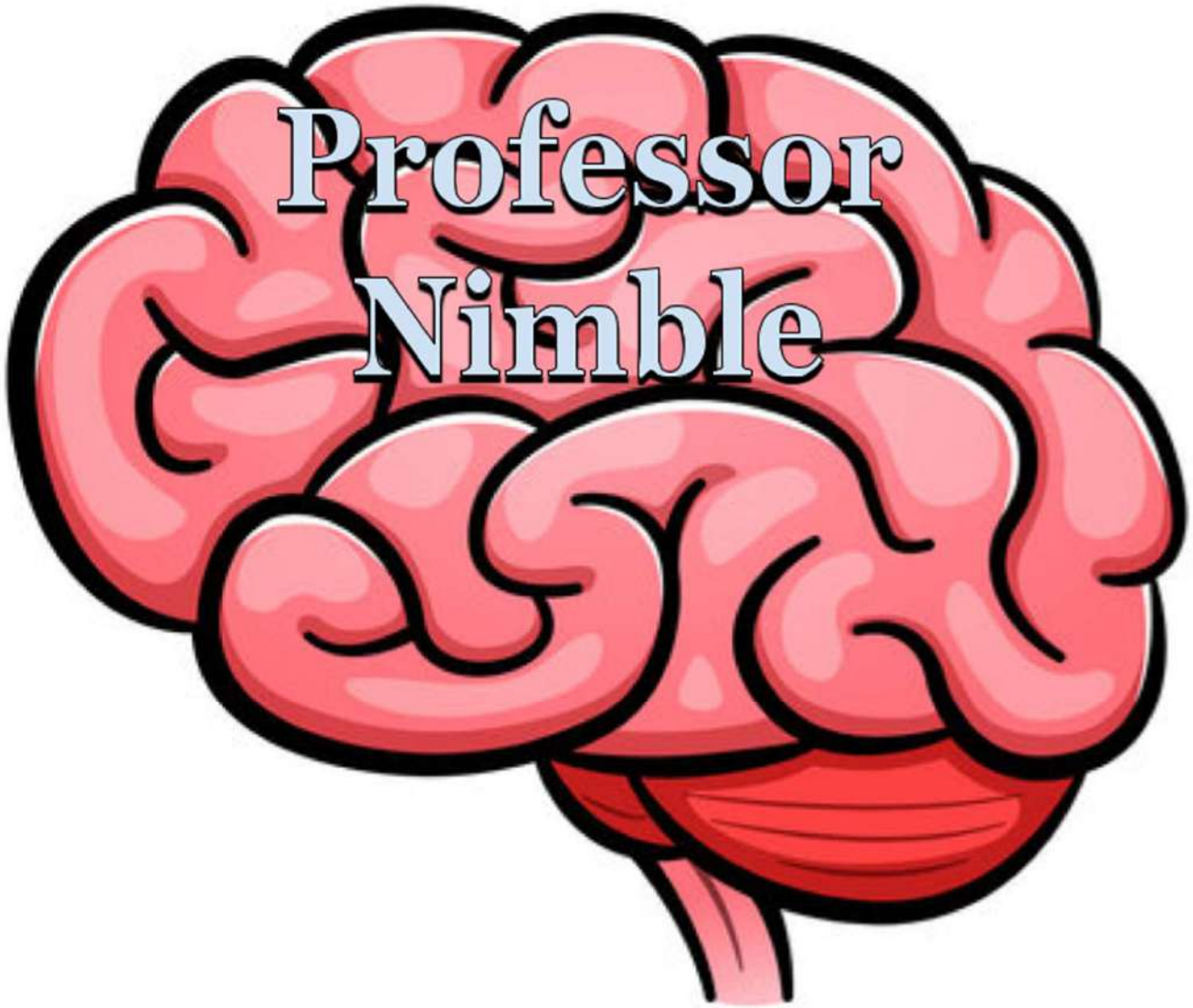

Day 1
12
11
10
9
8
7
6
5
4
3
2
1


Day 2
12
11
10
8
7
6
5
4
3
2
1

Day 3

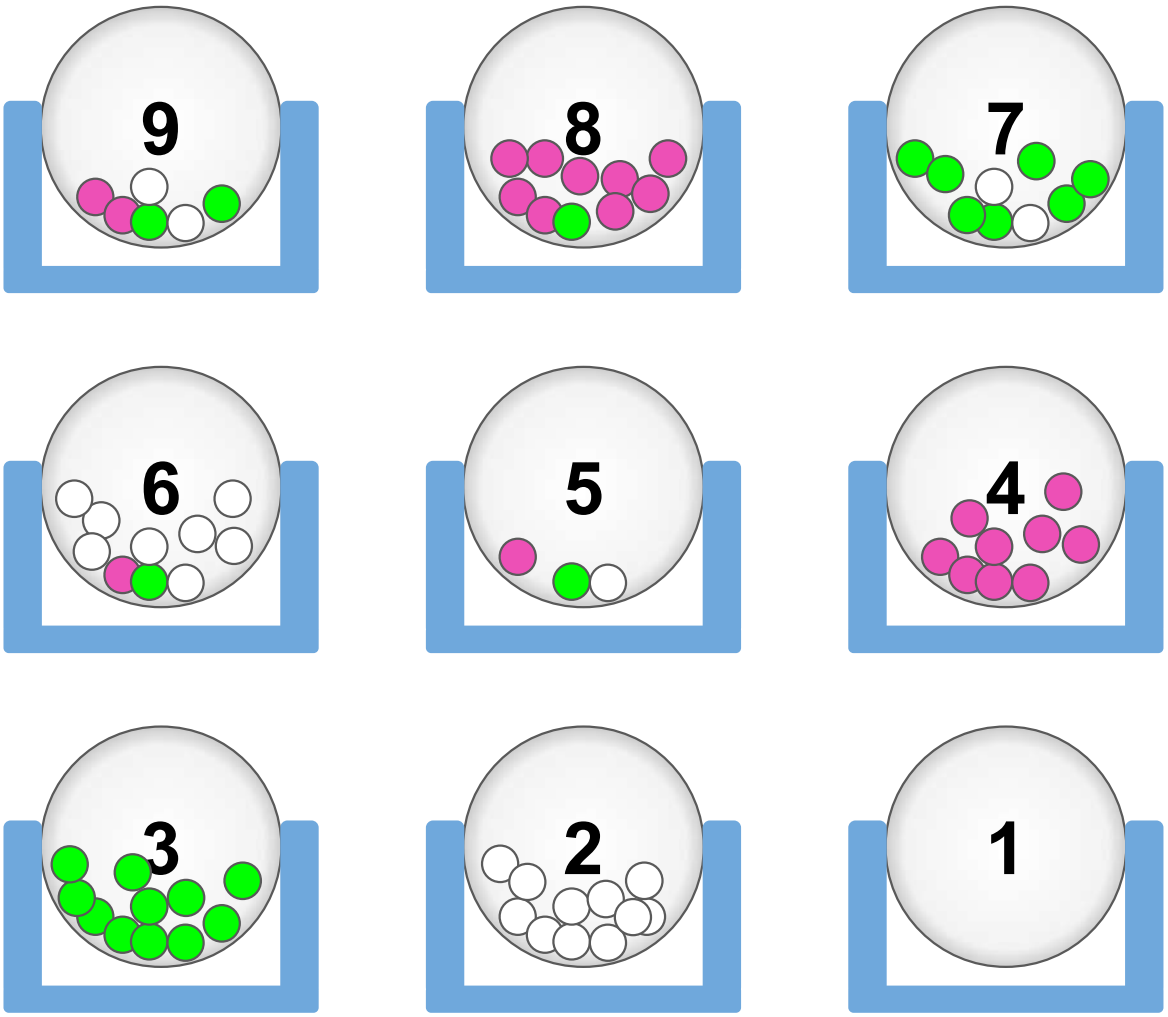
Day 4

Day 5


Professor
Nimble

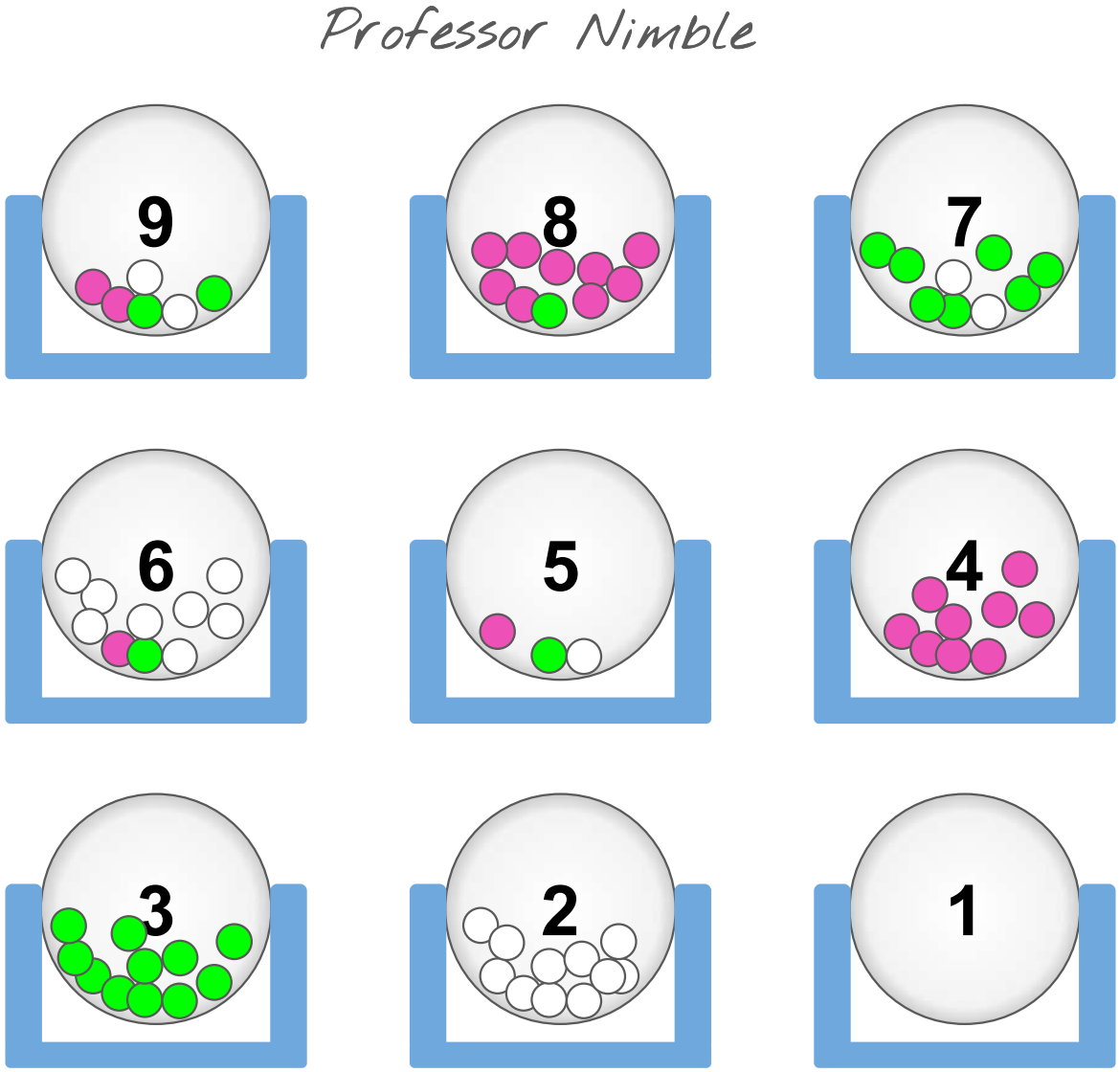
Learning from Professor Nimble

Professor Nimble



Math Concepts:

- Randomness
- Probability
- Proportions
- Ratios



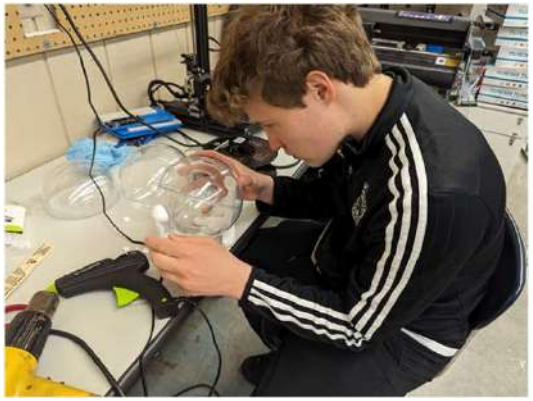
Machine Learning Principles:

- Supervised Learning
- Model Parameters
- Gradient Descent
- Backpropagation

Creating the Project

- Designing the “neurons”
- Simulating “Professor Nimble” in R
- Choosing Starting Conditions
- Creating the Public Display of Math
- Choosing Prizes


oNerds



nimMLv2.R*

Source on Save | Run | Source

```
150  # params
151  Start <- 13
152  init_balls <- 2
153  cap <- 50
154
155  trials <- 200
156  param_anal <- tibble(Trial = 1:trials, TotWinPct = 0, LastDayWinPct = 0, LastD
157
158
159  for(k in 1:trials){
160
161    # reset
162    mlm <- tibble(state = 1:Start,
163                  p1 = init_balls,
164                  p2 = init_balls,
165                  p3 = init_balls
166    )
```

430:1 | End Play Function | R Script

Console | Terminal | Background Jobs

R 4.3.2 · ~/

```
+         total_wins = mod_stats$total_wins[rnd]
+       )
+       break
+       # End of game. Computer went to zero or below, game over, player wins
+     } #end for 0007
+
+     # end of game play code
+   } # end while 0012
+ } #0013
>
```

Environment | History | Connections | Tutorial

Import Dataset | 61 MiB | List

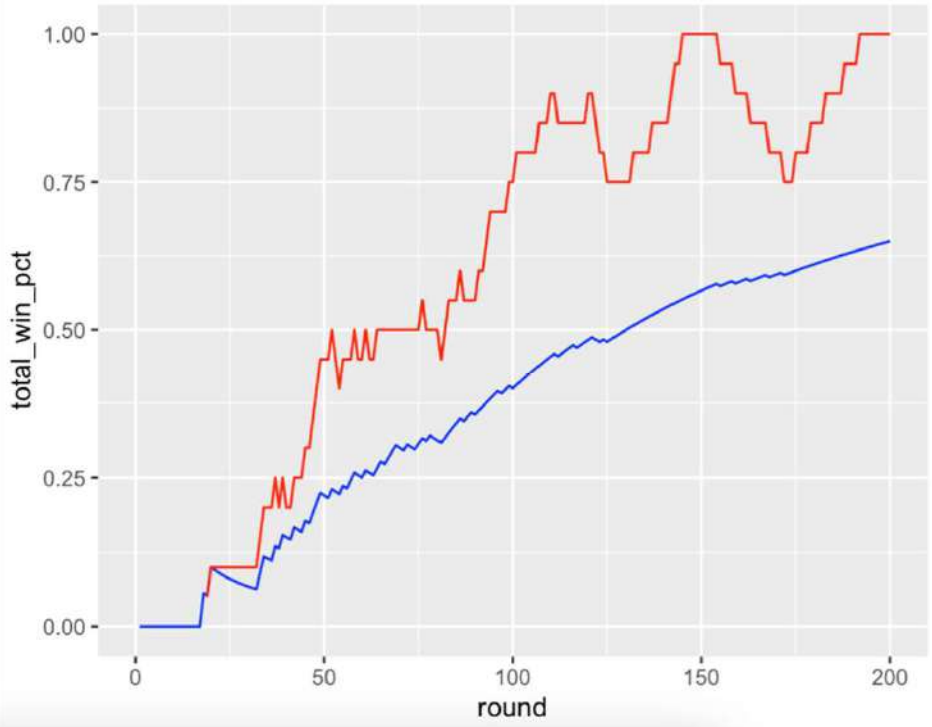
R | Global Environment

Start	13
trials	200

Functions

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



Professor Nimble at the State Fair

Inspiration

- Matt Parker - Standup Maths
- MENACE - Donald Michie (1961)

Tools for Teaching About AI

- Teachable Machine
- Create ML
- 3Blue1Brown
- Desmos Interactive

MIDAS

1. Illustrate the Game
2. Show sample of code
    a. How you build the program

What to present

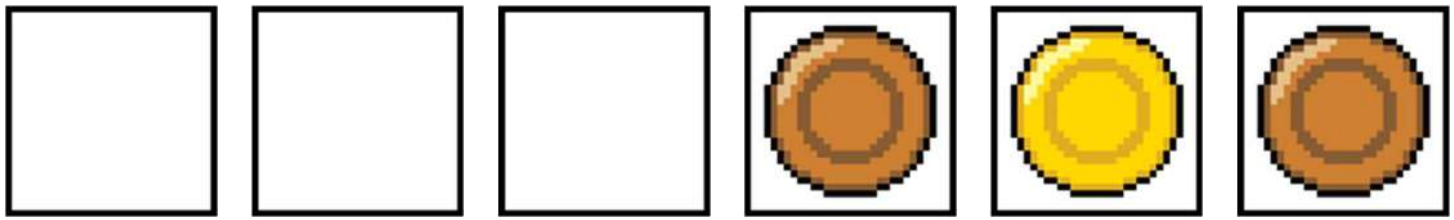
1. Set initial parameters
2. Play game
3. How the neural network decides
4. Gradient descent
5. Calculate derivatives (backpropagation)

MIDAS

How to play the old gold

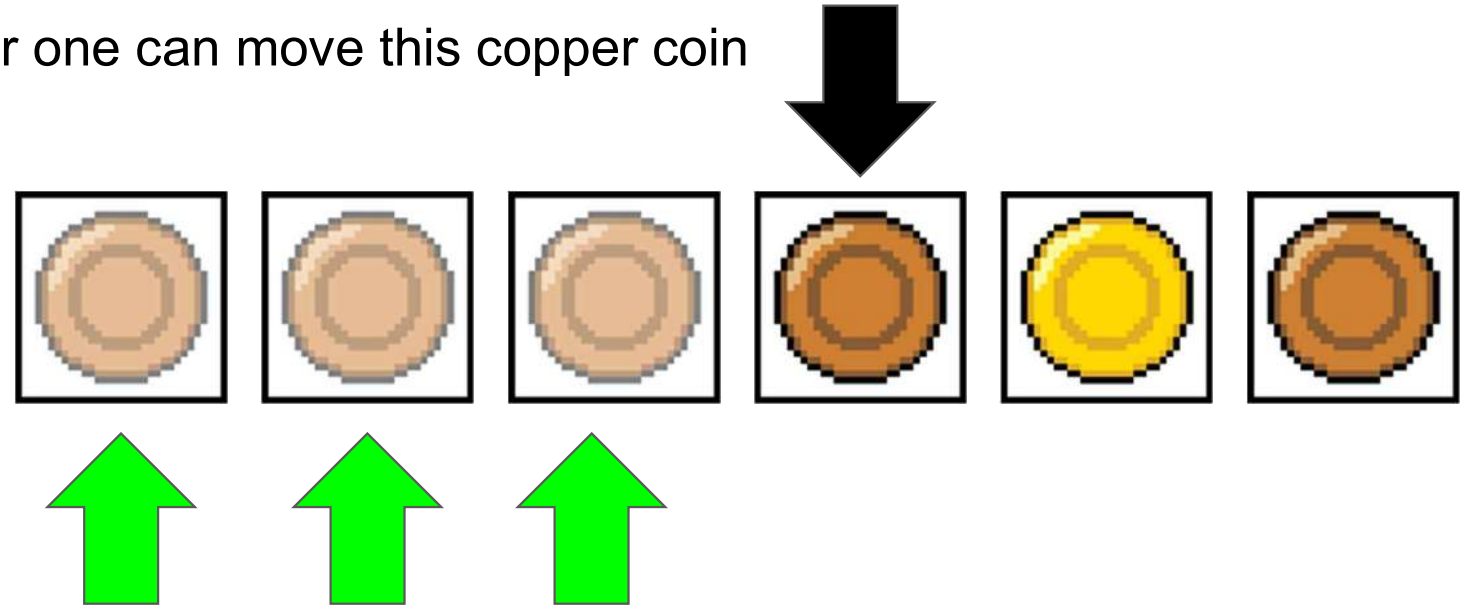
1. You can move any of the coins as far left as you want as long as the coin does not jump over another coin
2. If a coin is on the far left square, you can remove the coin
3. The objective of the game is to remove the gold coin

Board setup



Example

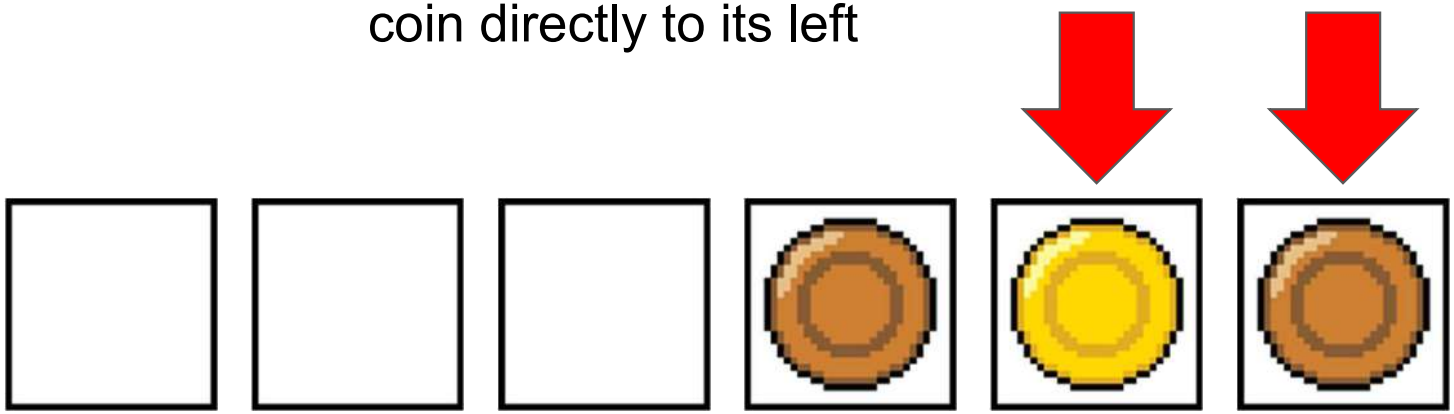
Player one can move this copper coin



To any one of these squares

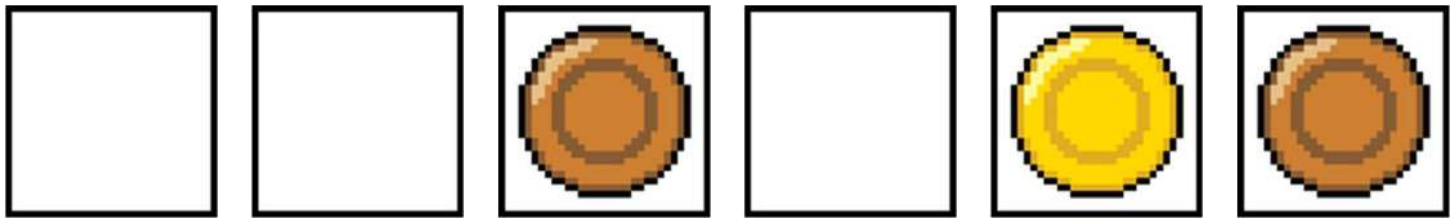
Example

But can't move these coins because there is a coin directly to its left



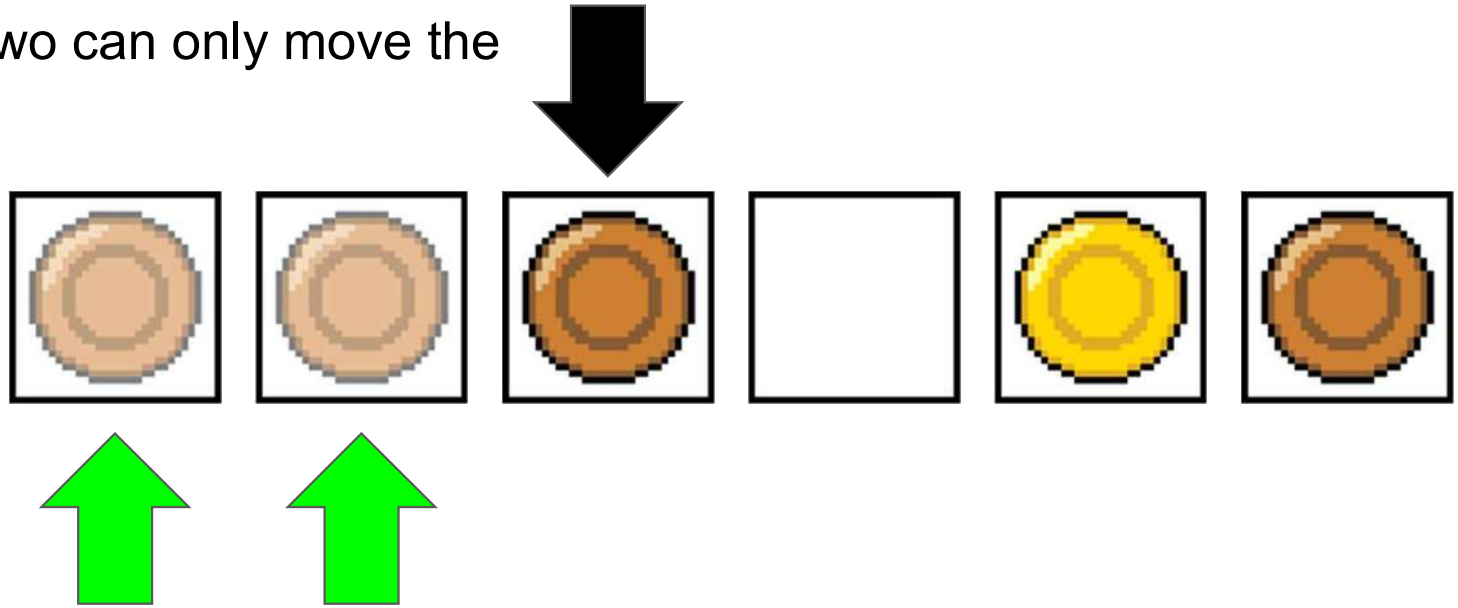
Example

Player one moves the left copper coin left one space



Example

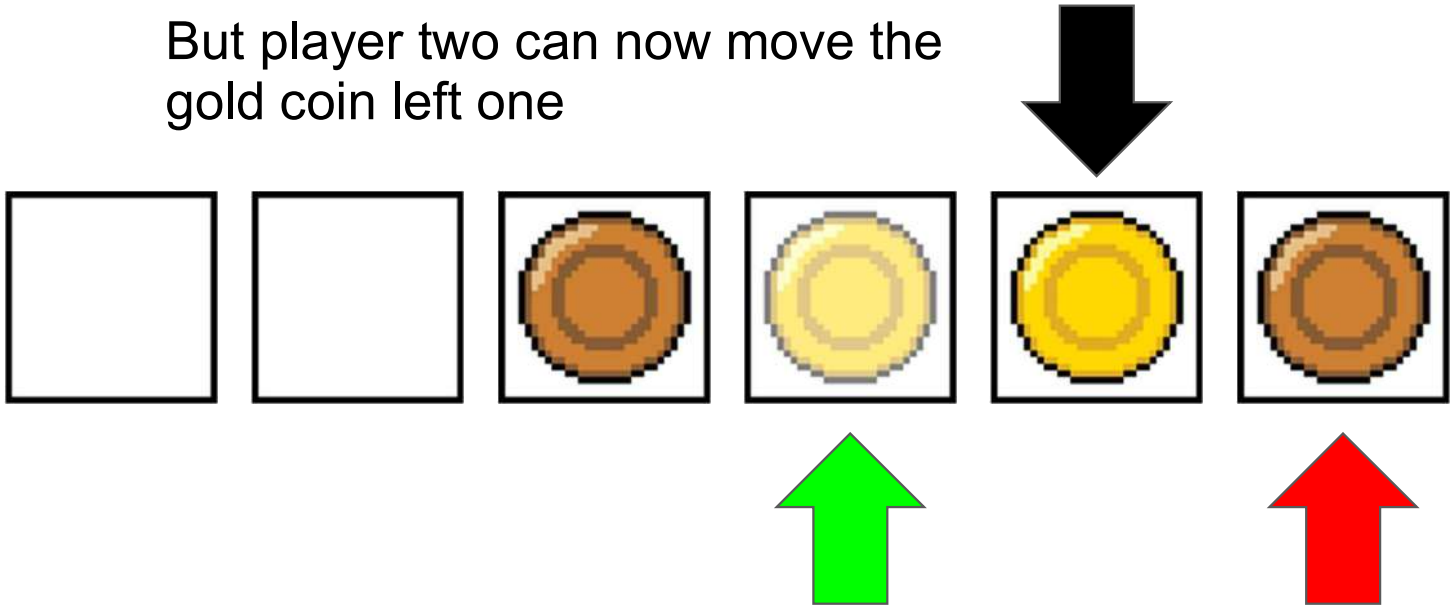
Now player two can only move the copper coin



To one of these two squares

Example

But player two can now move the gold coin left one



And this copper coin still can't mo

Example

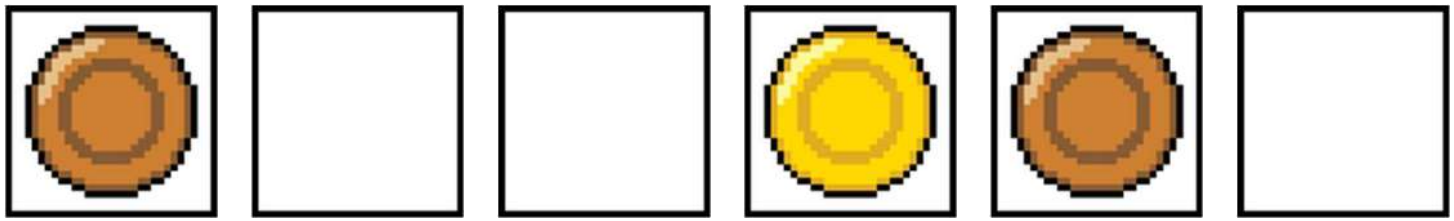
Player two moves the gold coin left one space

Example

Player one moves the right copper coin left one space

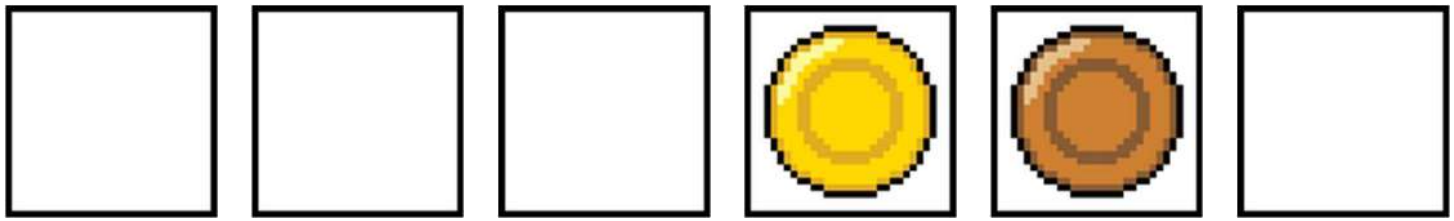
Example

Player two moves the left copper coin left two spaces



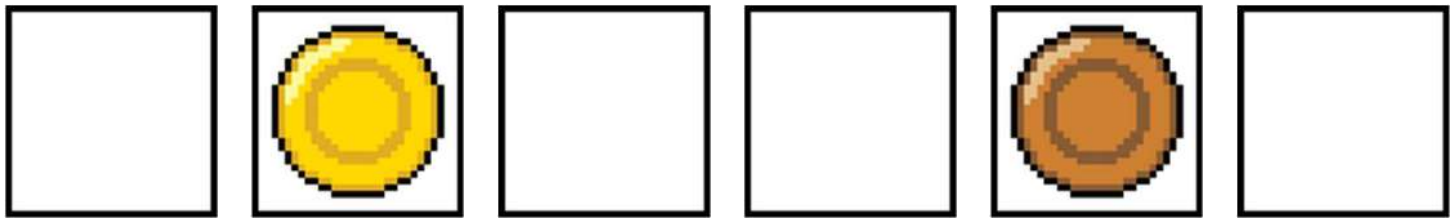
Example

Player one removes the left copper coin



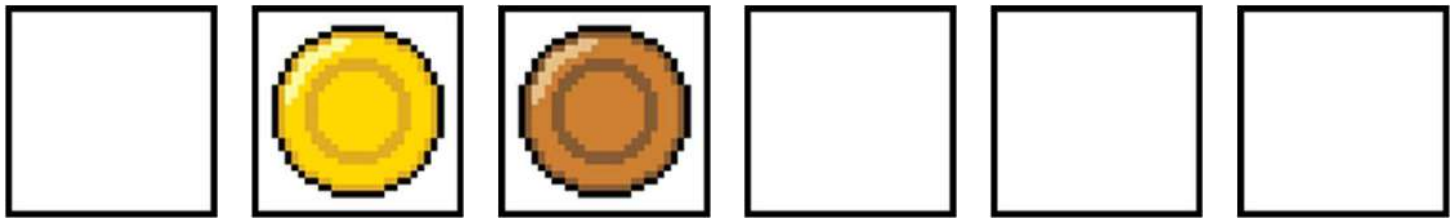
Example

Player two moves the gold coin left two spaces



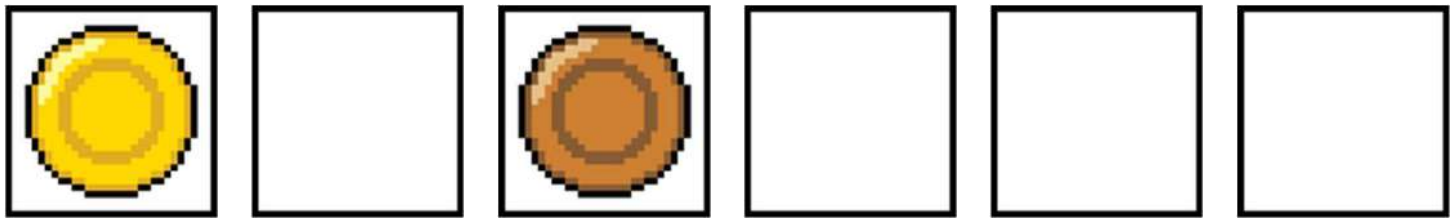
Example

Player one moves the copper coin left two spaces



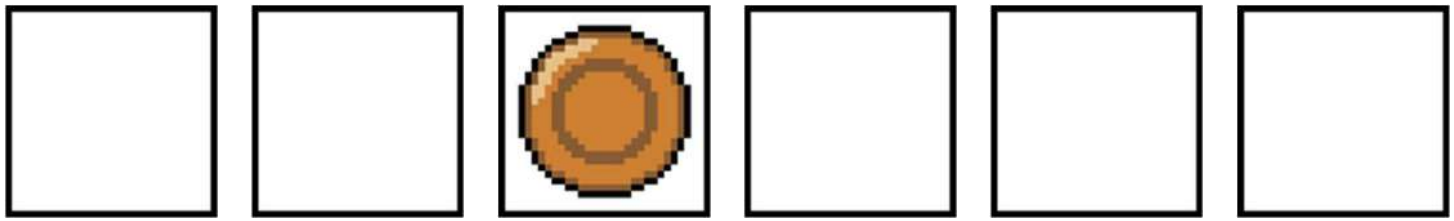
Example

Player two moves the gold coin left one space



Example

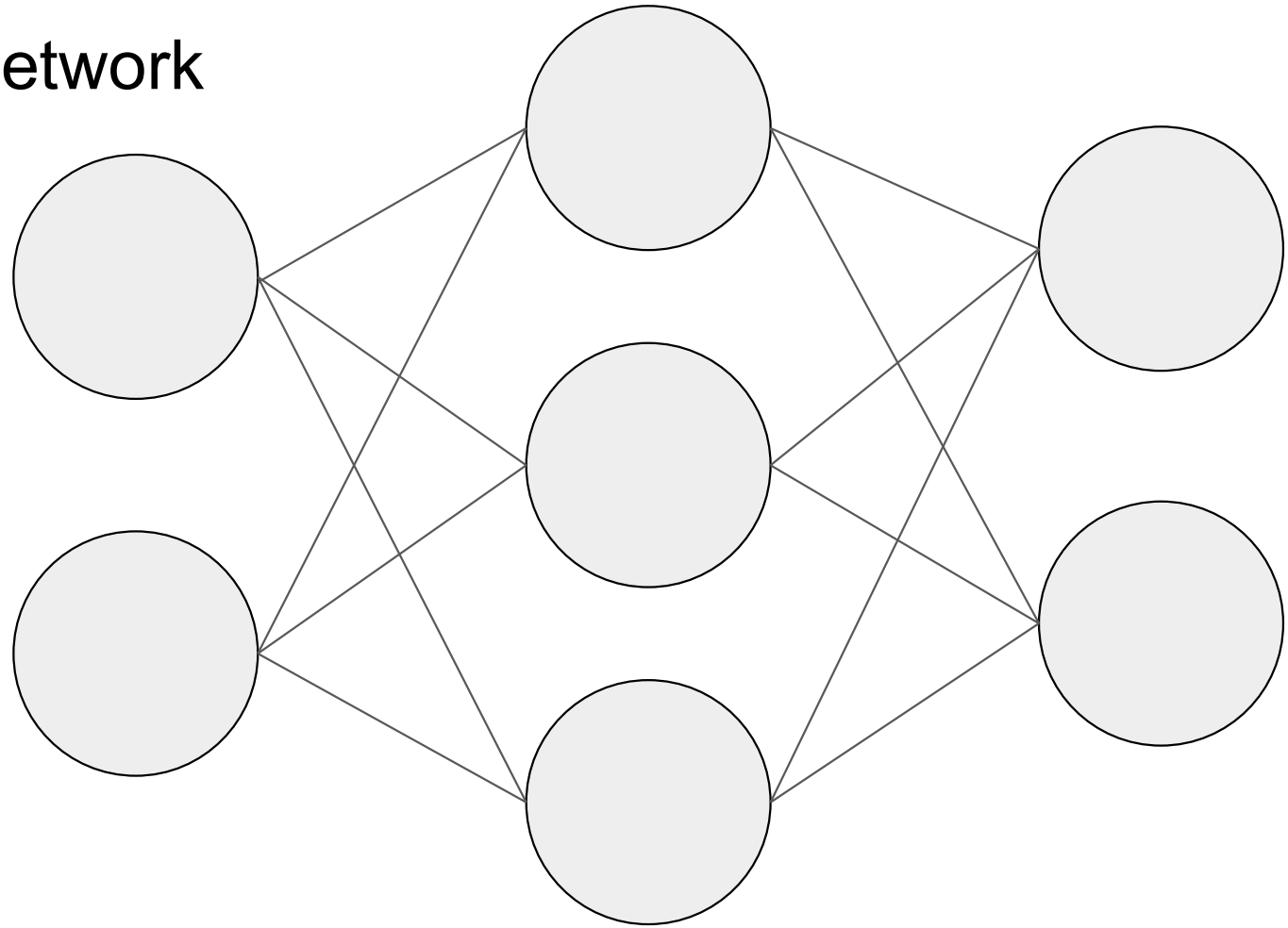
Player one removes the gold coin



Player one wins

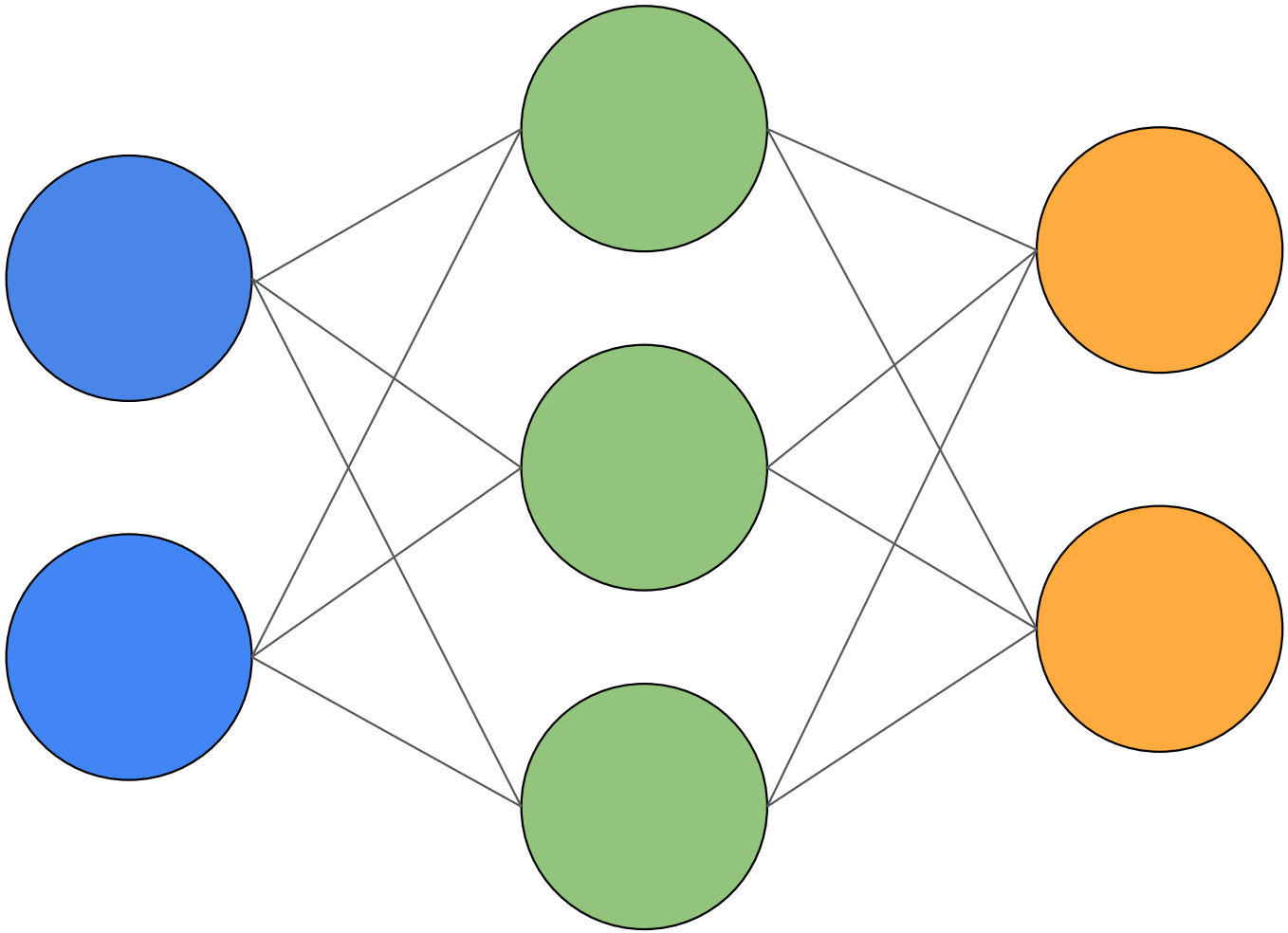
How does MIDAS know how to play?

Neural network



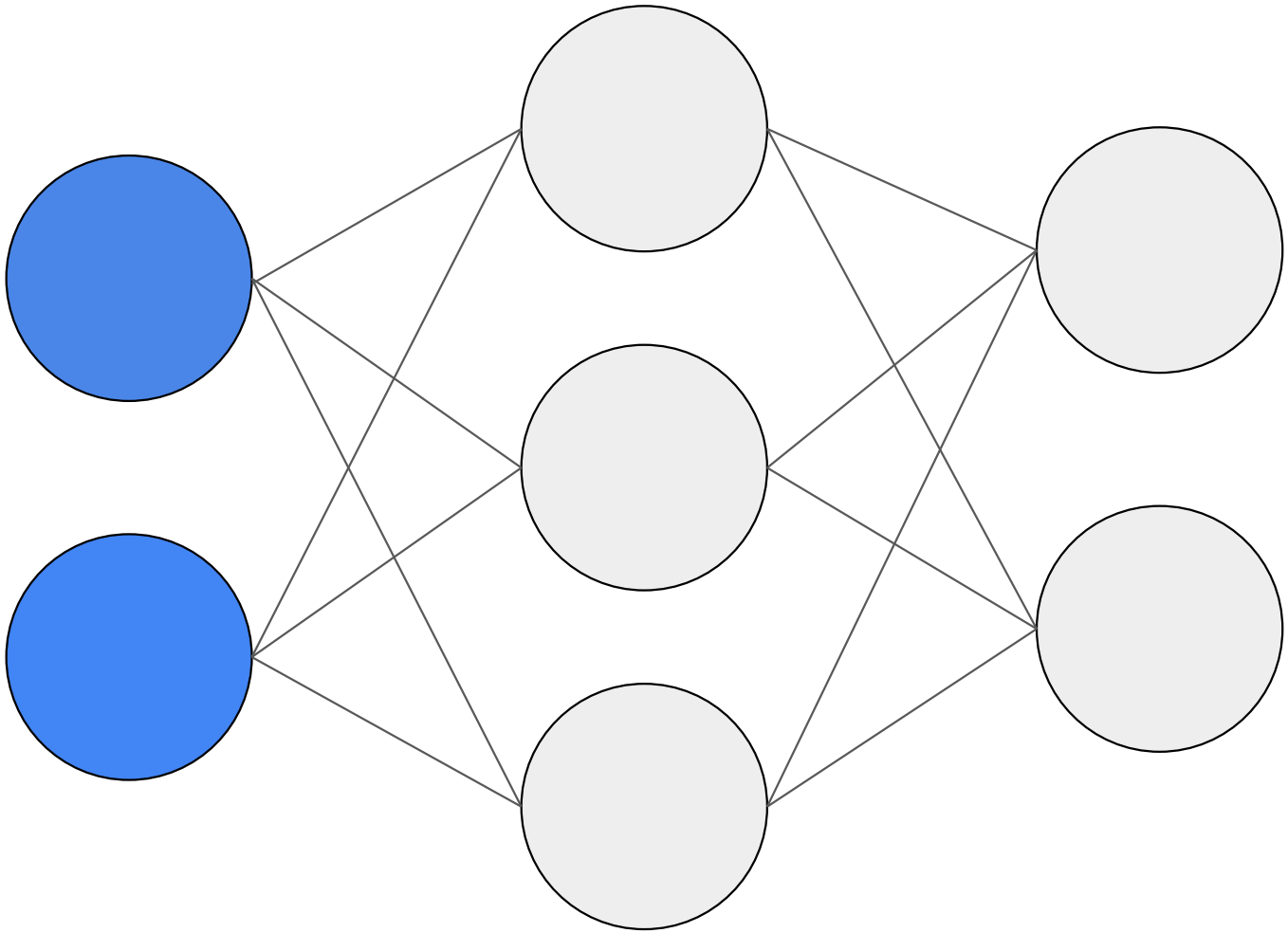
Neural network

Layers:
Neural networks are comprised of layers of neurons in order to do their calculations



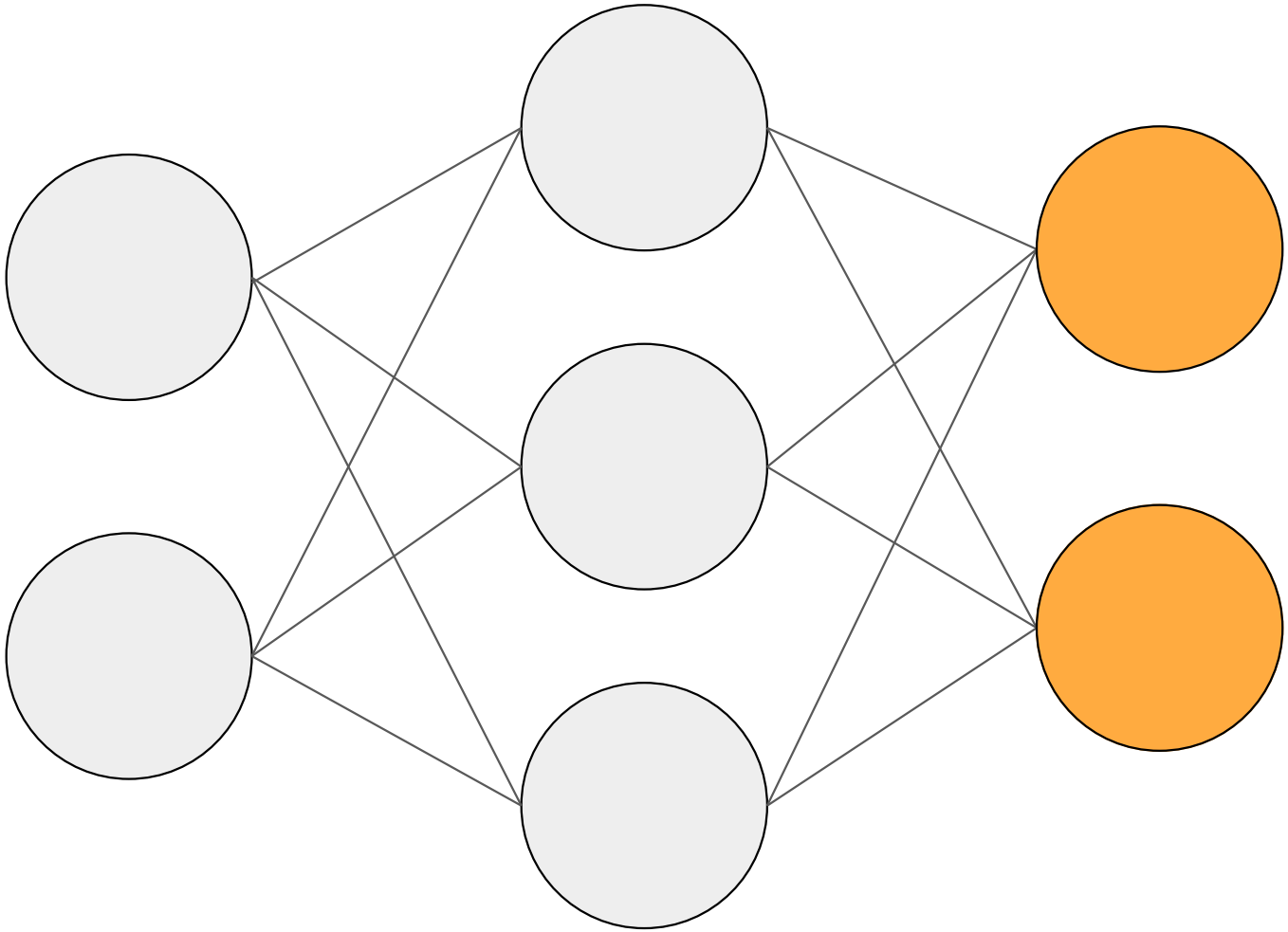
Neural network

Input layer:
This layer is just meant to receive and tell the network the input



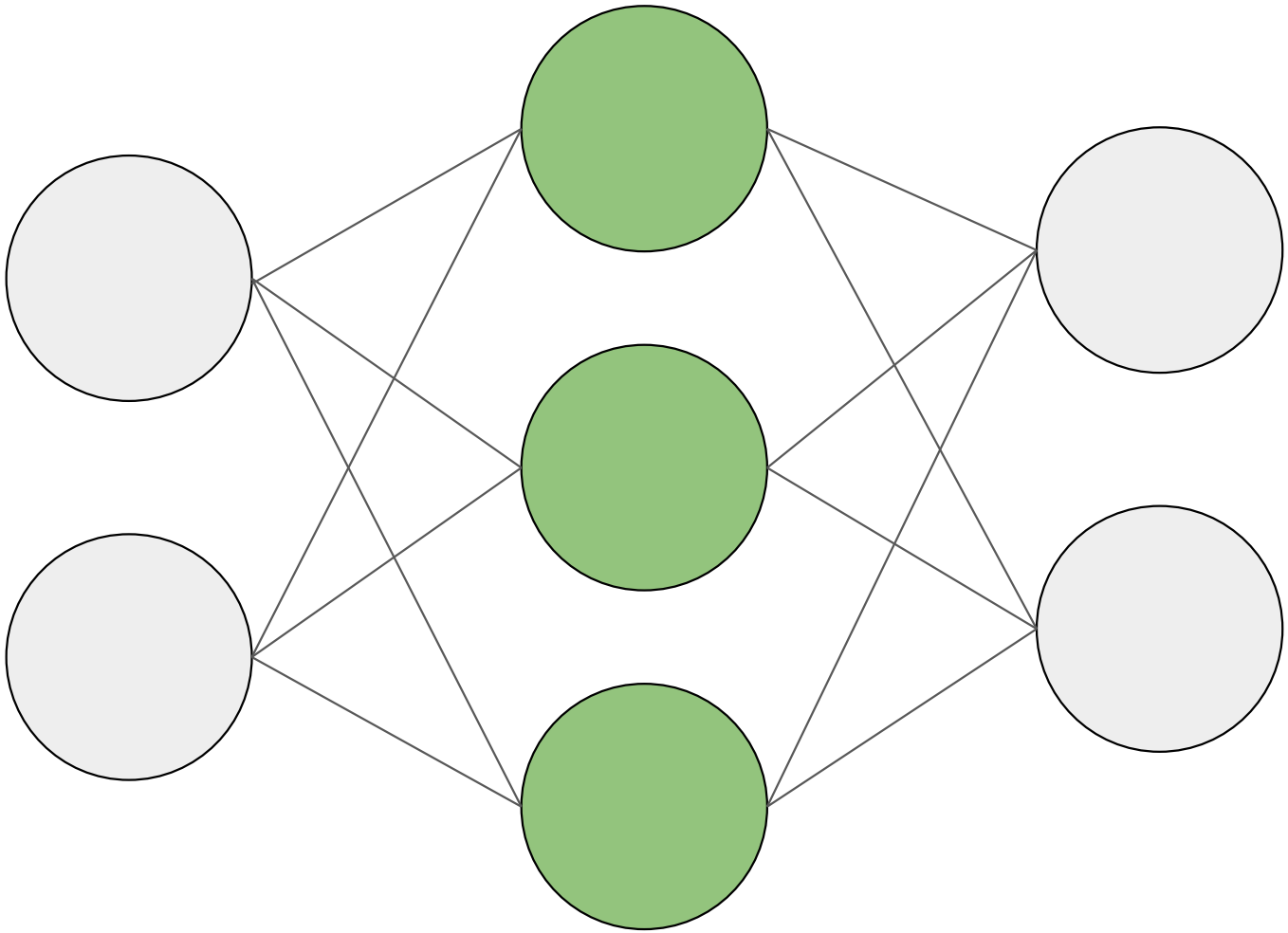
Neural network

output layer:
This layer makes the final calculations and output what the network calculated



Neural network

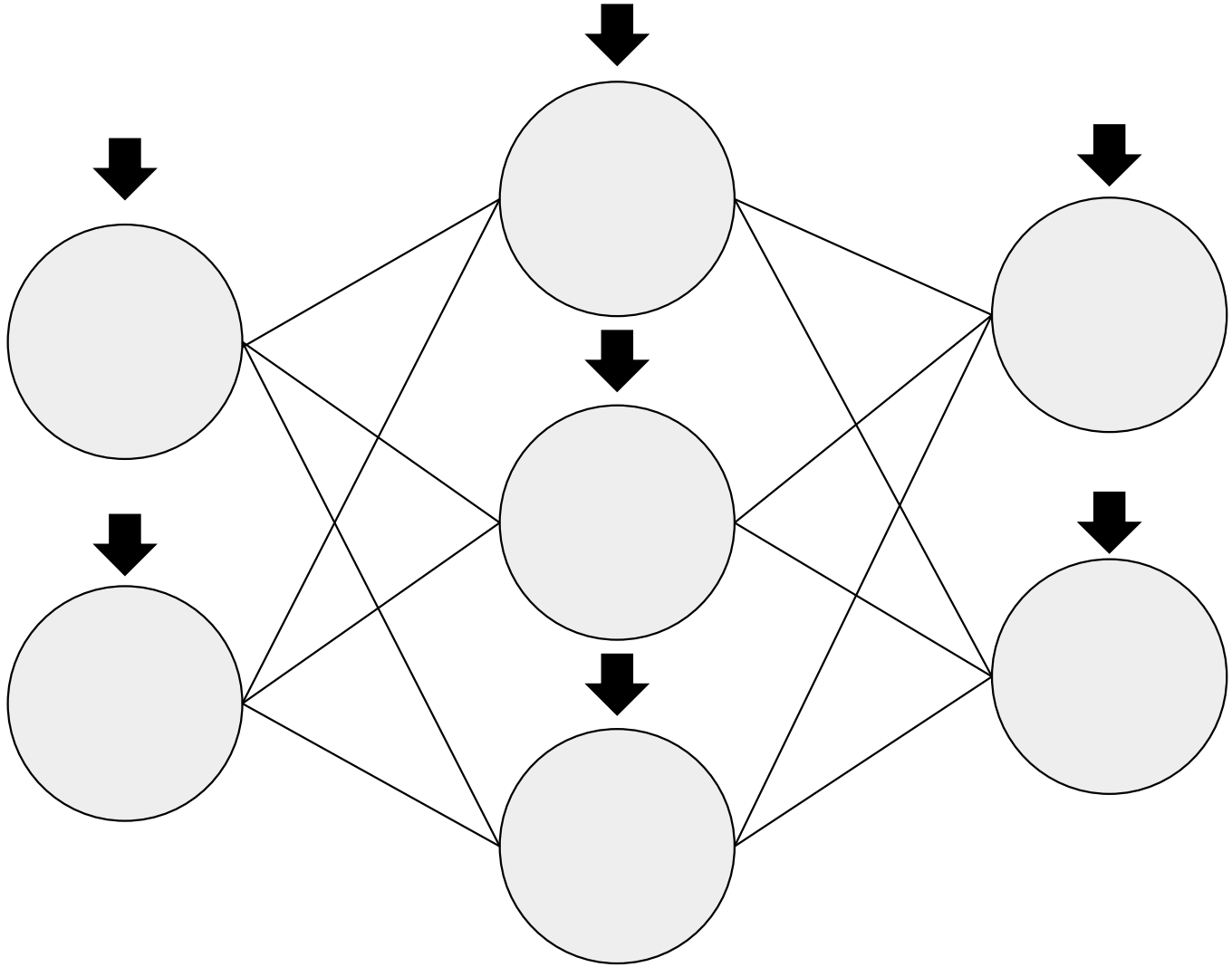
hidden layers:
These layers are responsible for most of the calculations used to determine the output



Neural network

Neurons:
Each layer is comprised of neurons, each neuron intake values and output a value, neurons also have a value called a bias.

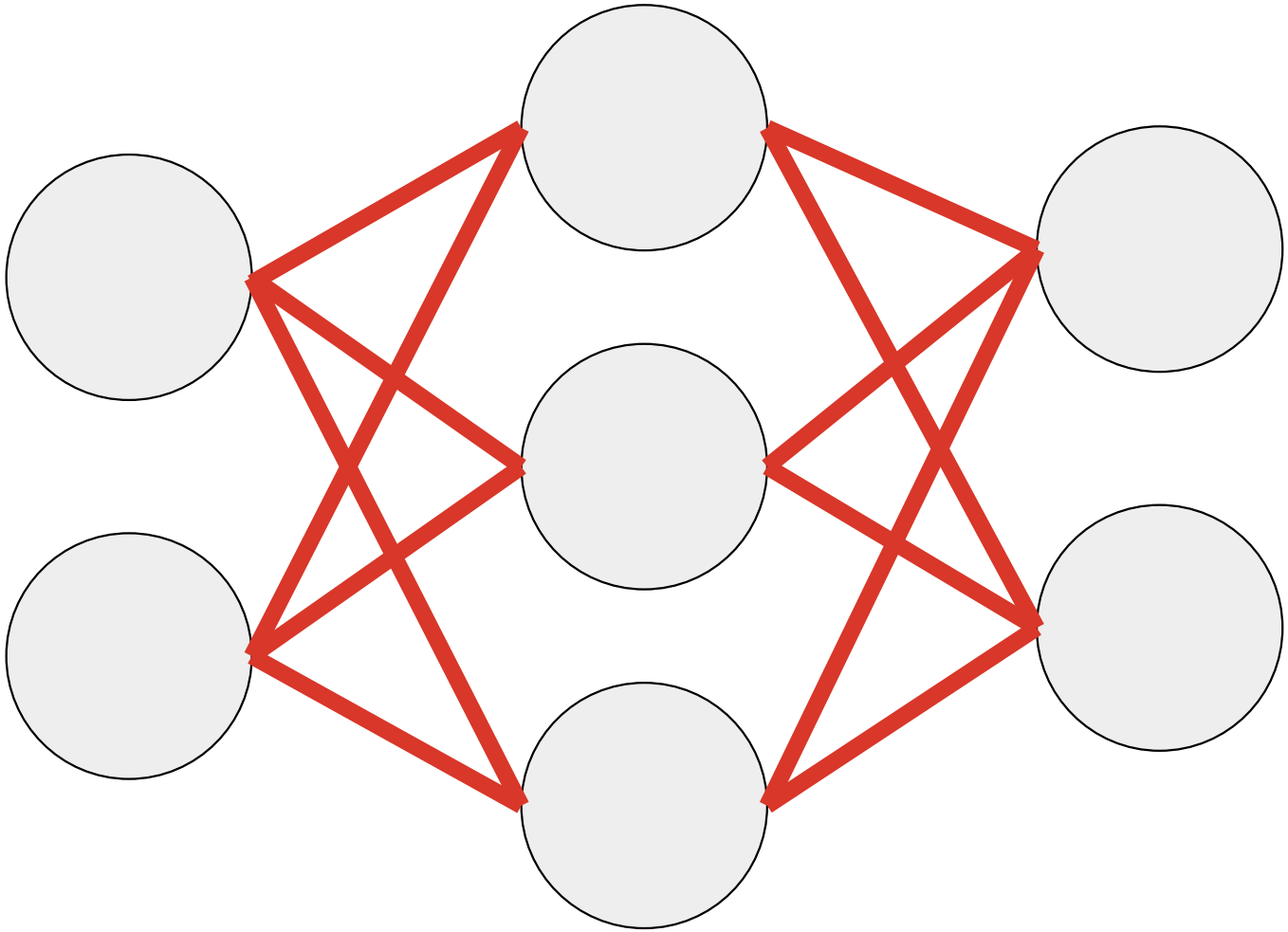
Note: all biases start at zero



Neural network

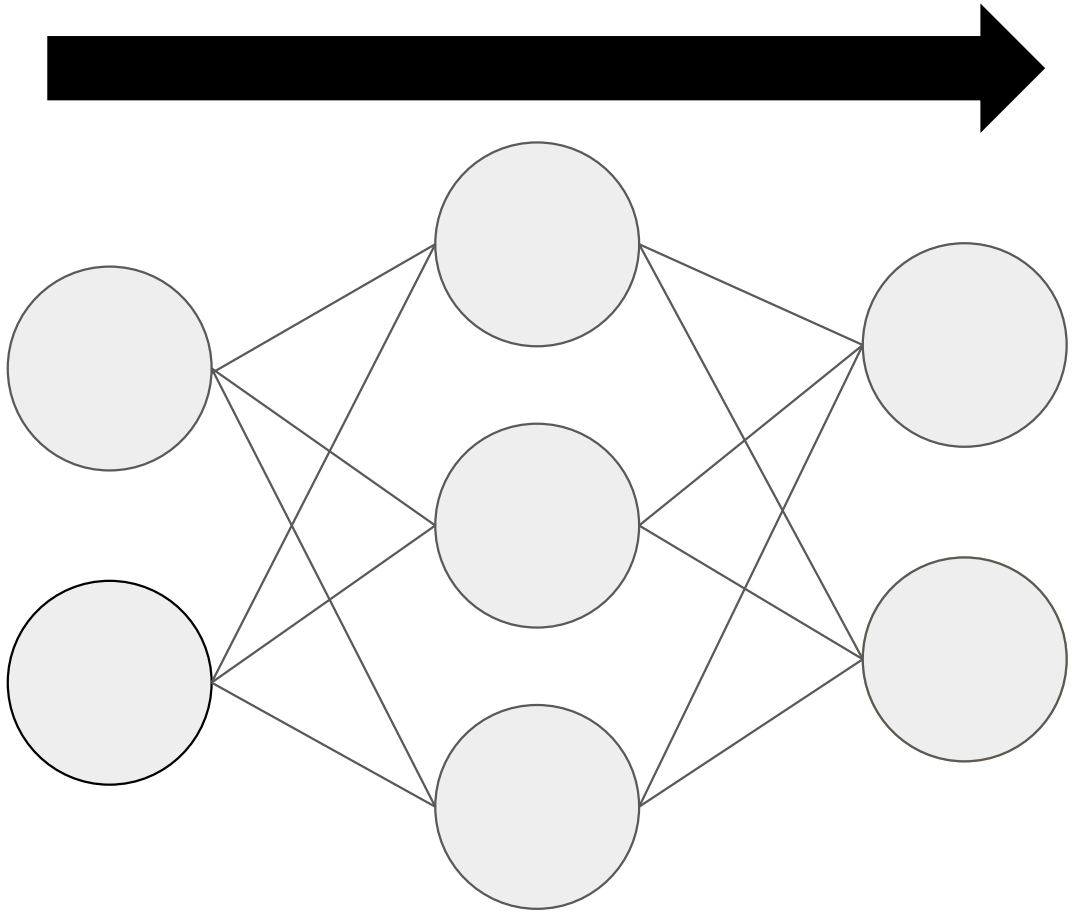
Weights: is how much the input effects the current output

Note: all weights start at a random value



How a neural network make decisions

For each layer except for the input layer, all the neurons take in the inputs of the last layer, plug it into an equation, and output a value for the next layer
This is known as Feedforward



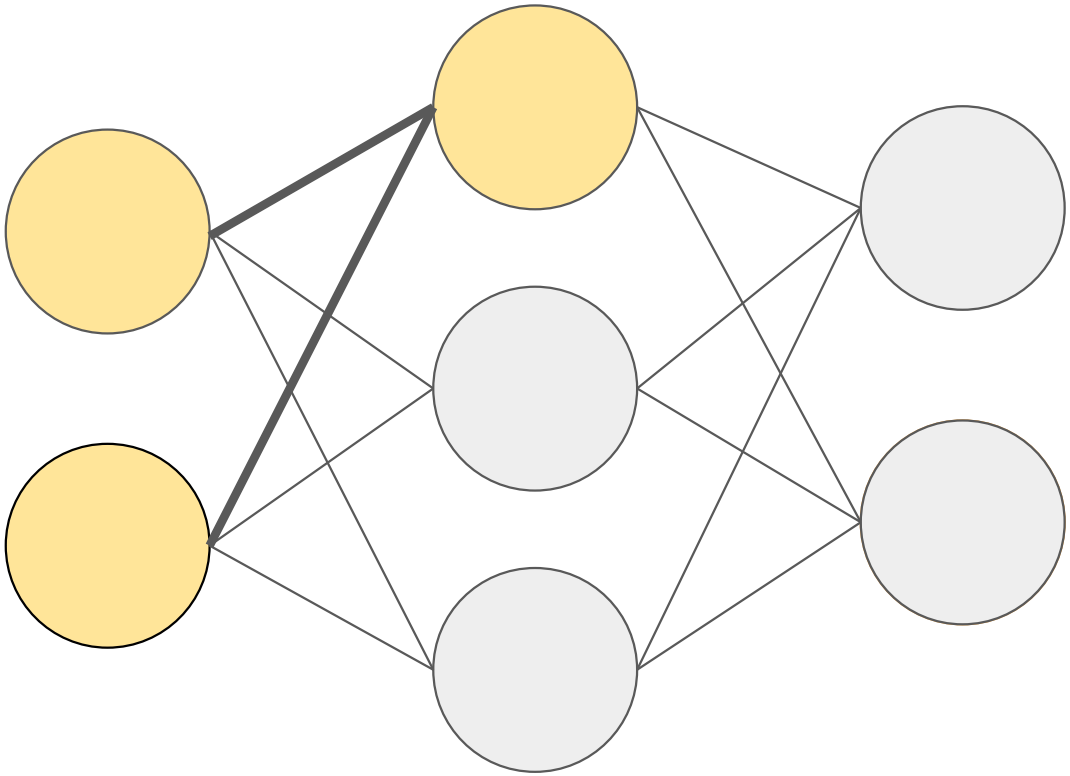
How a neural network make decisions

$$o_j = b_j + \sum_{i=1}^{n} w_i \cdot a_i$$

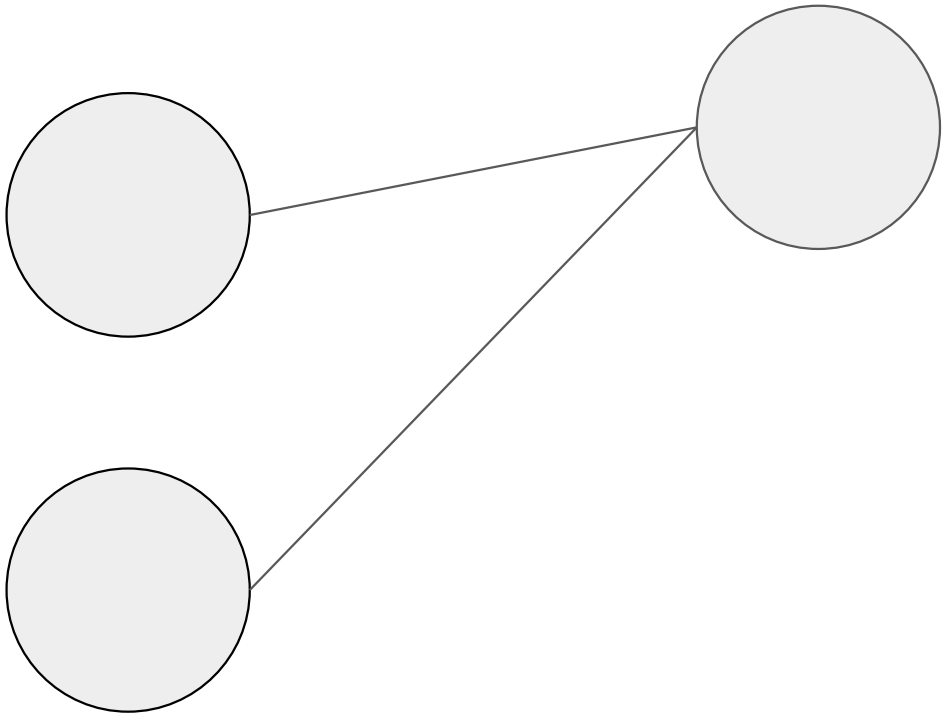
$$a_j = A(o_j)$$

a = activation value
w = weight
b = bias
o = output value
A(o) = activation function

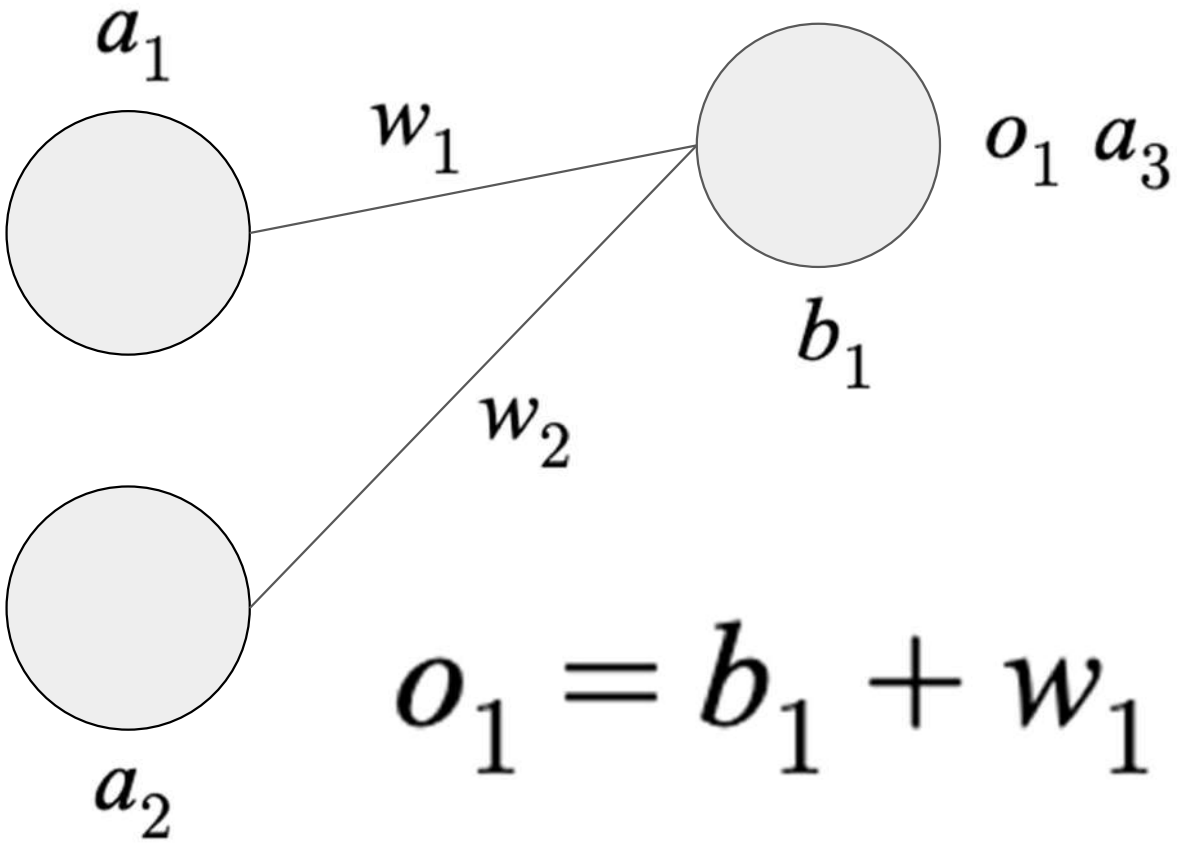
How a neural network make decisions



How a neural network make decisions

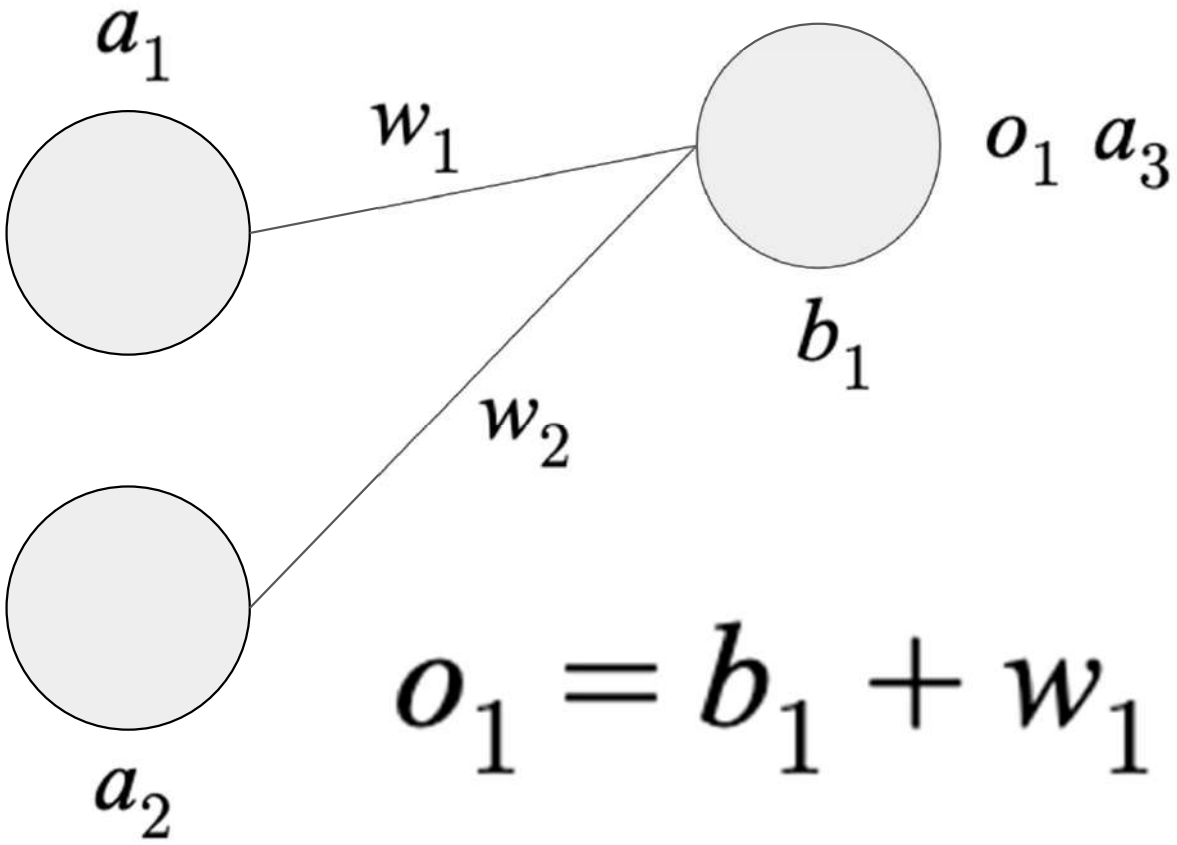


How a neural network make decisions $o_j = b_j + \sum_{i=1}^{n} w_i \cdot a_i$



$$o_1 = b_1 + w_1 \cdot a_1 + w_2 \cdot a_2$$

How a neural network make decisions $o_j = b_j + \sum_{i=1}^{n} w_i \cdot a_i$



$a_1 = 0.25$

$a_2 = 0.75$

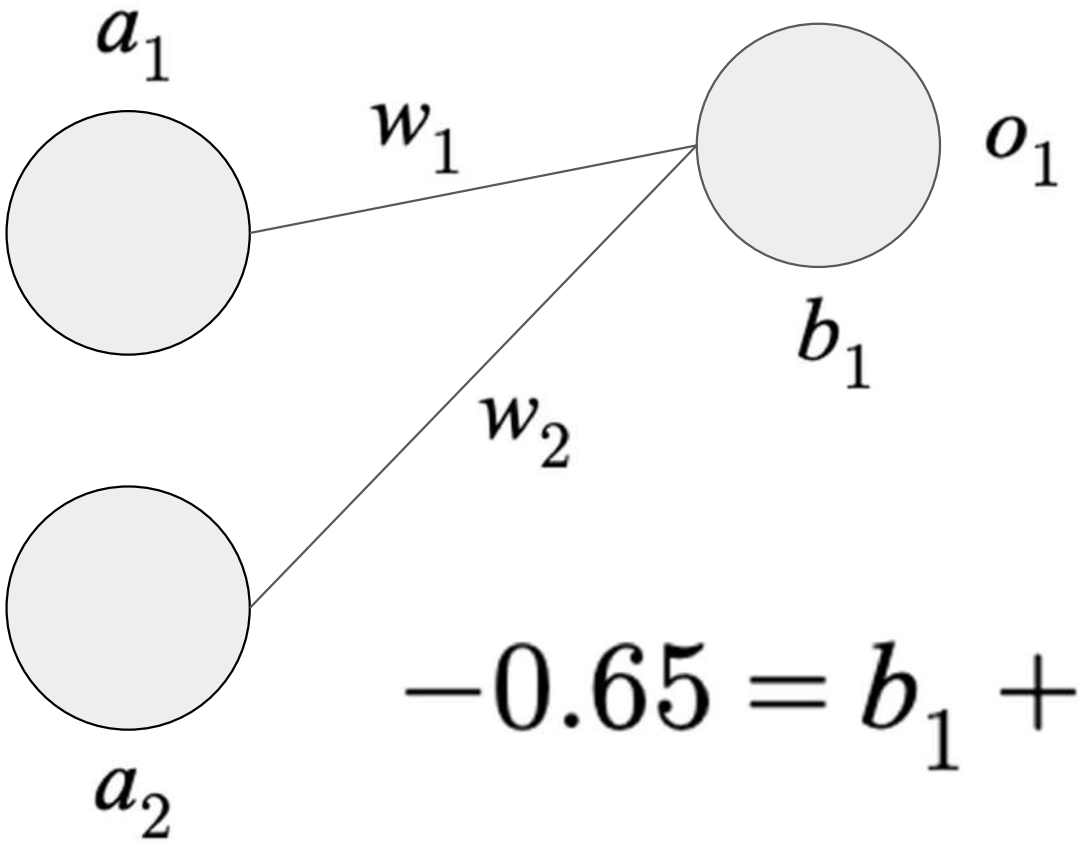
$w_2 = 0.3$

$w_1 = 0.5$

$b_1 = -1$

$$o_1 = b_1 + w_1 \cdot a_1 + w_2 \cdot a_2$$

How a neural network make decisions $o_j = b_j + \sum_{i=1}^{n} w_i \cdot a_i$



$a_1 = 0.25$

$a_2 = 0.75$

$w_2 = 0.3$

$w_1 = 0.5$

$b_1 = -1$

$$-0.65 = b_1 + w_1 \cdot a_1 + w_2 \cdot a_2$$

Activation functions

Activation functions restrict the output

Hidden layers:
Swish relu

$$A(o_1) = \frac{o_1}{1 + e^{-o_1}}$$

Output layer:
Sigmoid

$$A(o_1) = \frac{1}{1 + e^{-o_1}}$$

How a neural network learns

Neural networks use a cost/lost function to determine how accurate the network was.

$$C = \sum_{i=1}^{n} (a_i - p_i)^2$$

They use the gradient of the cost/lost function to adjust the weights and biases. This is Gradient Descent

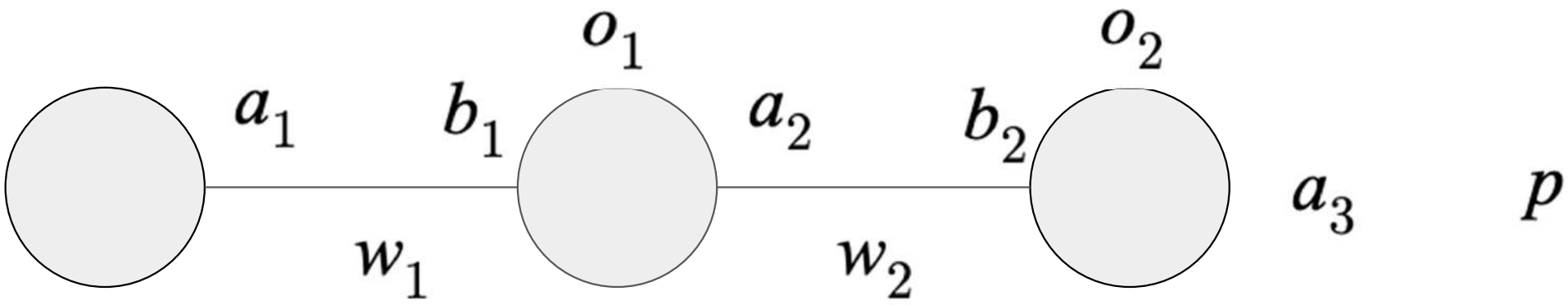
$$\nabla C(w,b)$$

How a neural network learns

We can use backpropagation to calculate all of the partial derivatives of the cost function with respect to the weights and biases

How a neural network learns

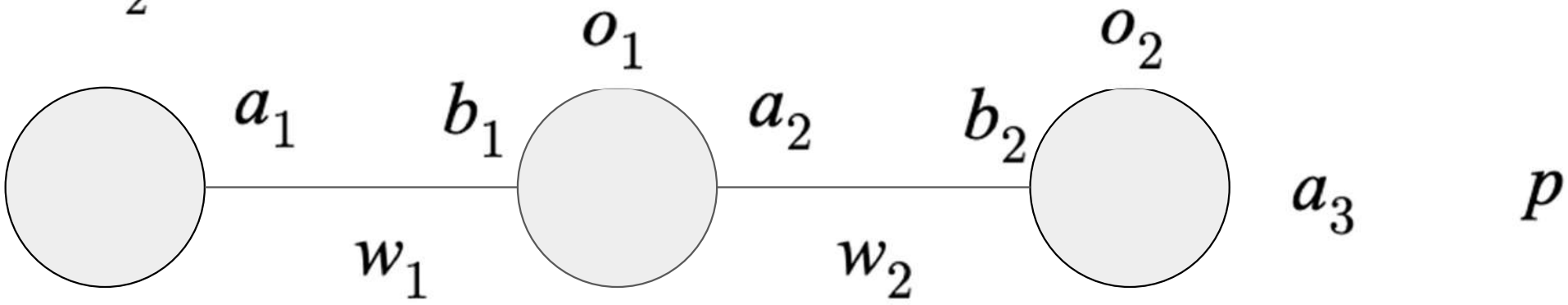
We can use backpropagation to calculate all of the partial derivatives of the cost function with respect to the weights and biases



How a neural network learns

$$\frac{dC}{dw_2}$$

$$\frac{dC}{db_2}$$



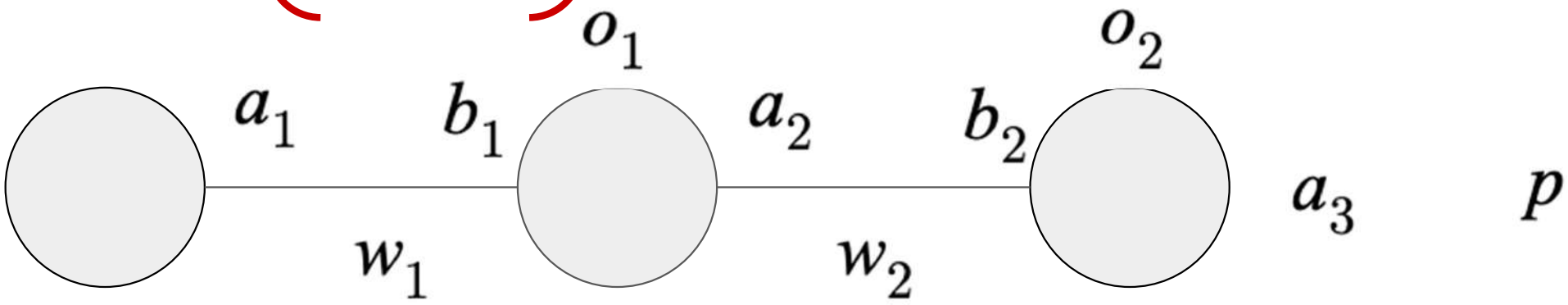
How a neural network learns

$$\frac{dC}{dw_2} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \frac{do_2}{dw_2}$$

$$\frac{dC}{db_2} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \frac{do_2}{db_2}$$

$$\frac{dC}{da_3} = 2\left(a_3 - p\right)$$

$$\frac{da_3}{do_2} = a_2\left(1 - a_2\right)$$

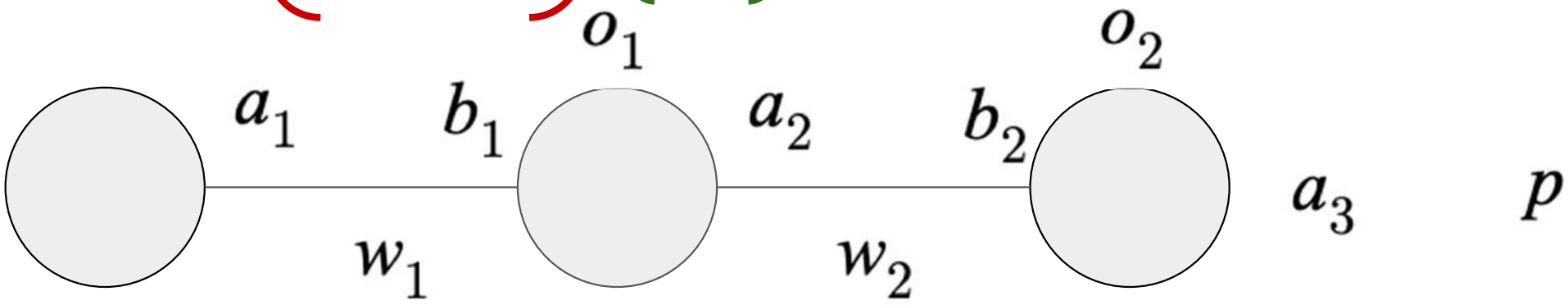


How a neural network learns

$$\frac{dC}{dw_2} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{dw_2}\right)$$

$$\frac{dC}{db_2} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{db_2}\right)$$

$$o_2 = b_2 + a_2 \cdot w_2$$



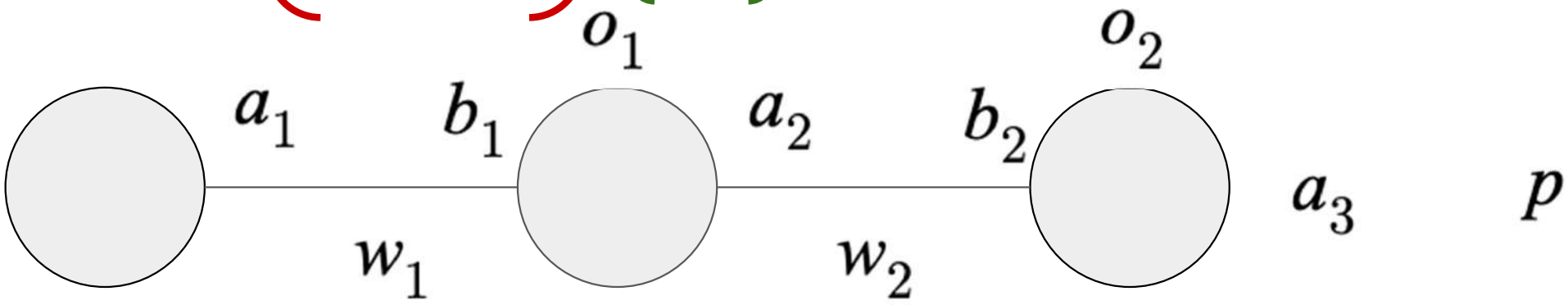
How a neural network learns

$$\frac{dC}{dw_2} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{dw_2}\right)$$

$$\frac{dC}{db_2} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{db_2}\right)$$

$$\frac{do_2}{dw_2} = a_2$$

$$\frac{do_2}{db_2} = 1$$

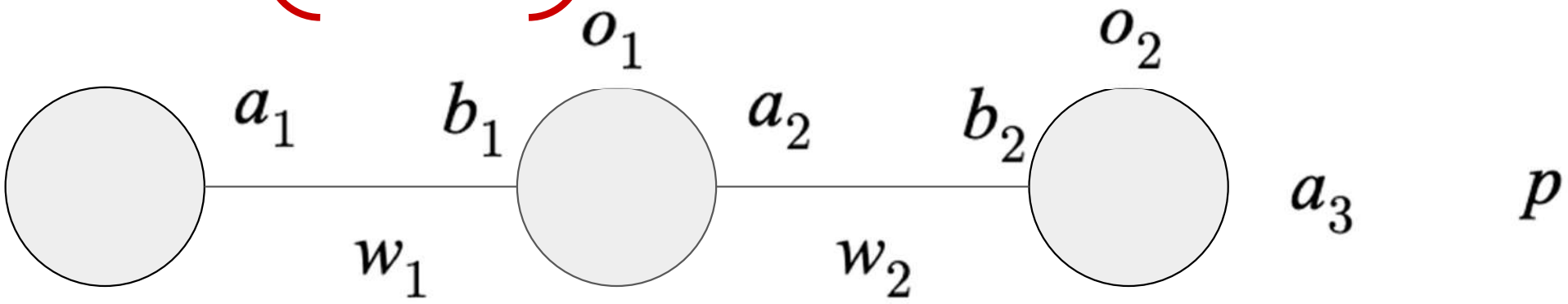


How a neural network learns

$$\frac{dC}{dw_2} = \frac{dC}{da_3} \cdot \frac{da_3}{do_2} \cdot \frac{do_2}{dw_2}$$

$$\frac{do_2}{dw_2} = a_2$$

$$\frac{dC}{db_2} = \frac{dC}{da_3} \cdot \frac{da_3}{do_2}$$

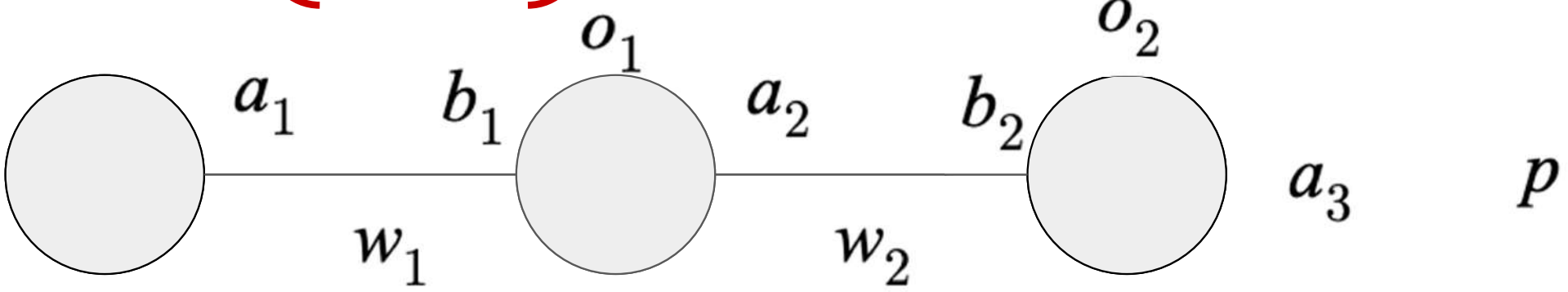


How a neural network learns

$$o_2 = b_2 + a_2 \cdot w_2$$

$$\frac{dC}{dw_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \frac{do_2}{da_2} \cdot \frac{da_2}{do_1} \cdot \frac{do_1}{dw_1}$$

$$\frac{dC}{db_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \frac{do_2}{da_2} \cdot \frac{da_2}{do_1} \cdot \frac{do_1}{db_1}$$

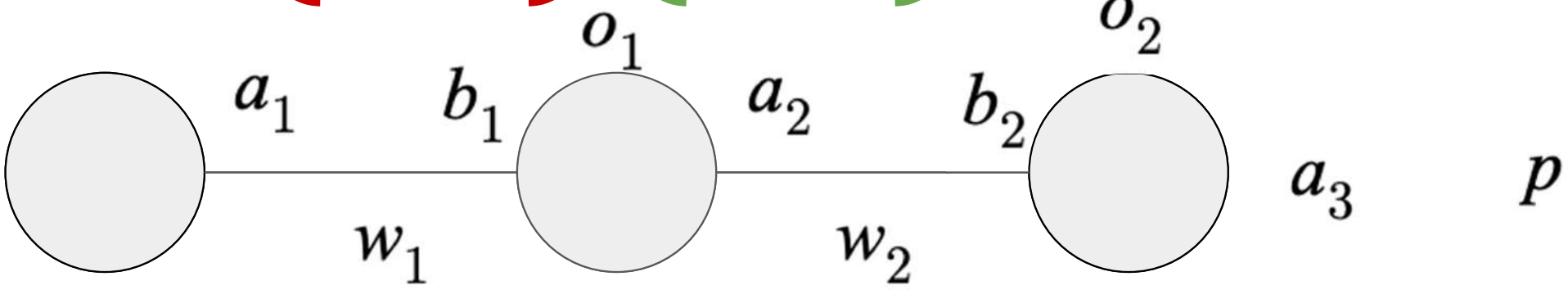


How a neural network learns

$$o_2 = b_2 + a_2 \cdot w_2$$

$$\frac{dC}{dw_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{da_2} \cdot \frac{da_2}{do_1}\right) \cdot \frac{do_1}{dw_1}$$

$$\frac{dC}{db_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{da_2} \cdot \frac{da_2}{do_1}\right) \cdot \frac{do_1}{db_1}$$

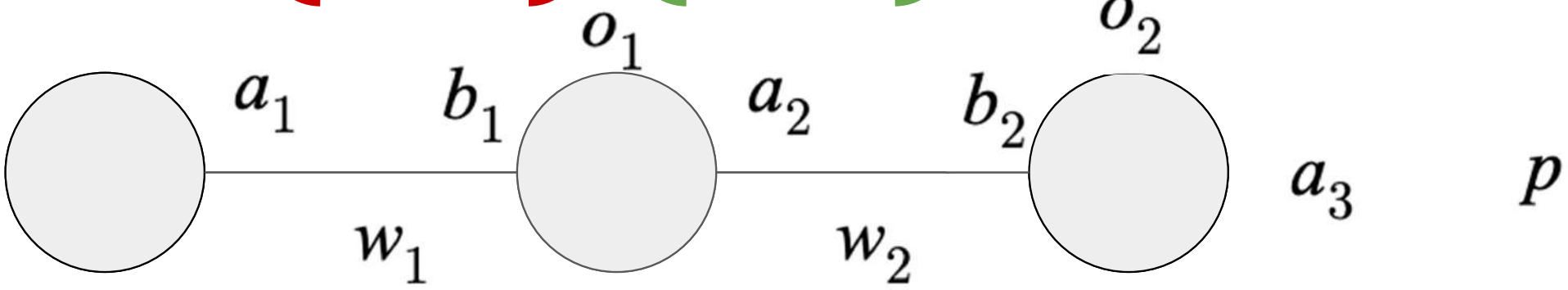


How a neural network learns

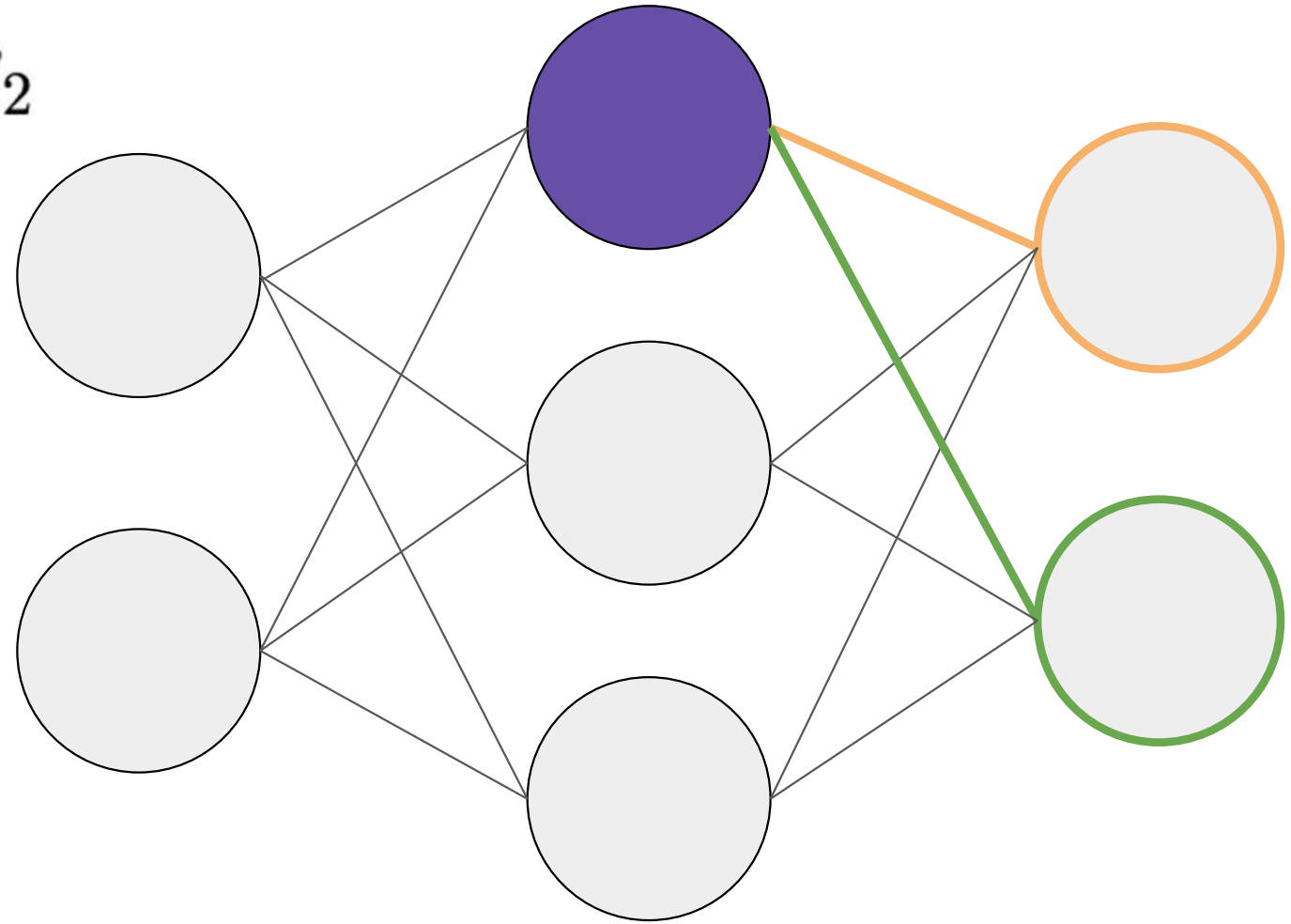
$$\frac{da_2}{do_1} = a_1\left(1 - a_1\right)$$

$$\frac{dC}{dw_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{da_2} \cdot \frac{da_2}{do_1}\right) \cdot \frac{do_1}{dw_1} \qquad \frac{do_2}{da_2} = w_2$$

$$\frac{dC}{db_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{da_2} \cdot \frac{da_2}{do_1}\right) \cdot \frac{do_1}{db_1}$$



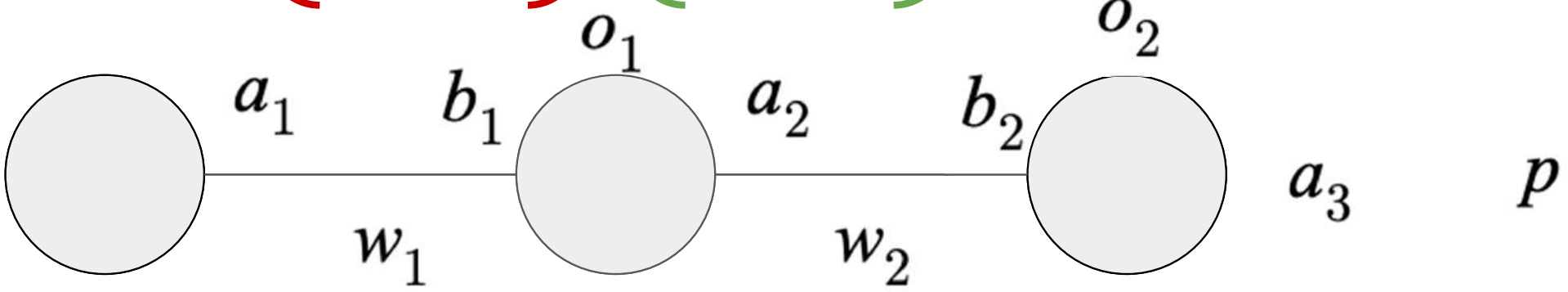
$$\frac{do_2}{da_2} = w_2$$



How a neural network learns

$$\frac{dC}{dw_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{da_2} \cdot \frac{da_2}{do_1}\right) \cdot \frac{do_1}{dw_1} \qquad \frac{do_1}{dw_1} = a_1$$

$$\frac{dC}{db_1} = \left(\frac{dC}{da_3} \cdot \frac{da_3}{do_2}\right) \cdot \left(\frac{do_2}{da_2} \cdot \frac{da_2}{do_1}\right)$$



How a neural network learns

Backpropagation (hidden layer)

$$\frac{dC}{dw_n} = \frac{dC}{db_{n+1}} \cdot \frac{do_{n+1}}{da_{n+1}} \cdot \frac{da_{n+1}}{do_n} \cdot \frac{do_n}{dw_n}$$

$$\frac{dC}{db_n} = \frac{dC}{db_{n+1}} \cdot \frac{do_{n+1}}{da_{n+1}} \cdot \frac{da_{n+1}}{do_n}$$

How a neural network learns

Backpropagation (output layer)

$$\frac{dC}{dw_n} = \frac{dC}{da_{n+1}} \cdot \frac{da_{n+1}}{do_n}$$

$$\frac{dC}{dw_n} = \frac{dC}{da_{n+1}} \cdot \frac{da_{n+1}}{do_n} \cdot \frac{do_n}{dw_n}$$

Code

```
func Calculate_outputs(I:Array,out: bool) -> Array:
    num_activation_b.append([])
    num_weights.append([])
    var outputs = []
    num_activation_a.append(I)
    for i in num_neron_out:
        var output = nerons[i].bias
        for o in num_neron_in:
            output += I[o]*nerons[i].weights[o]
        num_weights[num_weights.size()-1].append(output)
        if(out):
            outputs.append(Activation(output))
            num_activation_b[num_activation_b.size()-1].append(Activation(output))
        else:
            outputs.append(Activation_hidden(output))
            num_activation_b[num_activation_b.size()-1].append(Activation_hidden(output))
    return outputs
```

Code

```
func Calculate_output_layer(I:int,W: bool, data_set_number: int,max: float) -> Array:
    var distance = 1
    var neron_values = []
    var expected = []
    for  n in num_activation_b[data_set_number].size():
        expected.append(num_activation_b[data_set_number][n])
    if(W):
        expected[I] += ((1-expected[I])/(((data_set_number/max)*distance)+1))
    else:
        expected[I] += ((0-expected[I])/(((data_set_number/max)*distance)+1))
    for i in num_activation_b[data_set_number].size():
        var cost_derivitive = Neron_cost_derivitive(num_activation_b[data_set_number][i],expected[i])
        var activation_derivitive = Activation_derivitive(num_weights[data_set_number][i])
        neron_values.append(activation_derivitive*cost_derivitive)
    return neron_values
```

Code

```
func Calculate_Hidden_layers(L: layer,V: Array,data_set_number: int) -> Array:
    var new_neron_values = []
    for n in num_neron_out:
        var new_neron_value = 0.0
        for o in V.size():
            var weighted_input_derivitive = L.nerons[o].weights[n]
            new_neron_value += weighted_input_derivitive * V[o]
        new_neron_values.append(new_neron_value*Activation_hidden_derivitive(num_weights[data_set_number][n]))
    return new_neron_values
```

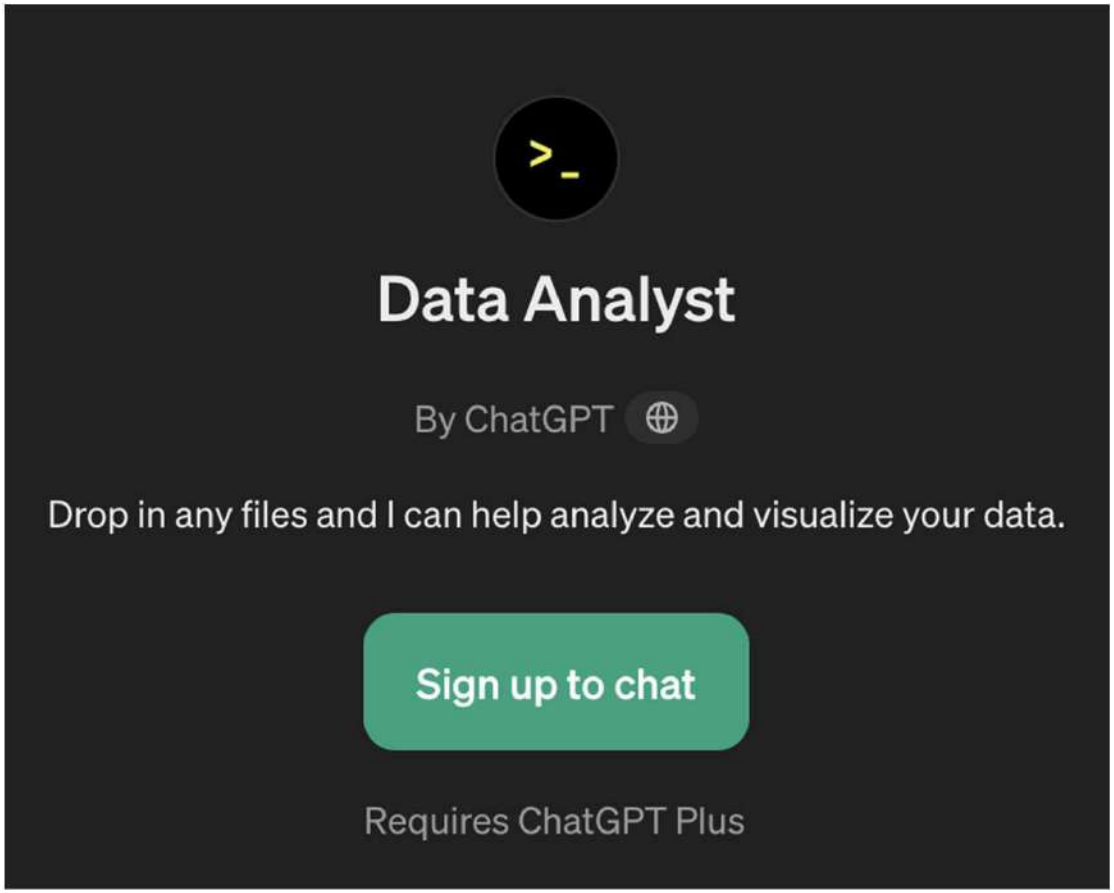
Code

```
func Apply_gradient(I: float) -> void:
    var B = 0.9
    var e = 0.0000000001
    for o in num_neron_out:
        gradiant_bias_velocity[o] = I*(B*gradiant_bias_velocity[o] + (1-B)*pow(cost_gradiant_bias[o],2))
        nerons[o].bias -= I*(cost_gradiant_bias[o]/sqrt(gradiant_bias_velocity[o]+e))
        for i in num_neron_in:
            gradiant_weights_velocity[o][i] = (B*gradiant_weights_velocity[o][i] + (1-B)*pow(cost_gradiant_weights[o][i],2))
            nerons[o].weights[i] -= I*(cost_gradiant_weights[o][i]/sqrt(gradiant_weights_velocity[o][i]+e))
```

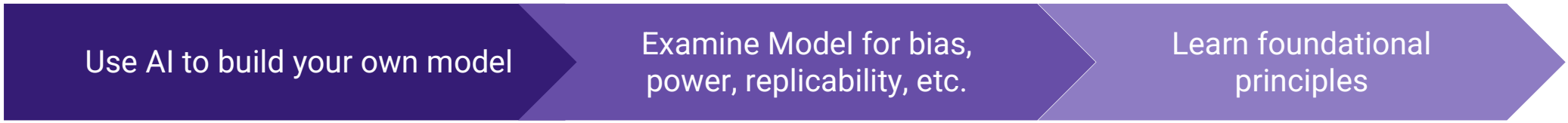
Changing How We Teach

Traditional Data Analysis & Statistics

ChatGPT: Data Analyst



Traditional Data Analysis & Statistics



NCTM Position Statement on Artificial Intelligence

Artificial Intelligence (AI)-driven tools can respond to students' thinking and interests in ways that previous tools could not. By drawing from large language sets, AI has the potential to adjust application-based problems to student interests and identify the sense students have made even in their incorrect answers. Students will continue to need teachers' mathematical, pedagogical, and relational expertise, though teachers are also likely to benefit from AI-driven tools. In some cases, AI may serve as a teaching assistant, but students will need teachers to help them create a bridge between prior knowledge, new knowledge, and shared knowledge. Teachers must tell students to be very skeptical about AI results, especially about the unique challenges of using tools that may have been trained on biased datasets. This skepticism can be woven into existing pedagogical and assessment techniques. Knowing this, educators need to be involved in developing and testing AI tools in math education to stay up to date with current AI trends to best prepare students for an AI future. Contrary to some popular opinions, this effort will require teachers with even deeper knowledge of math instruction and assessment—math teachers with more experience, not less.

https://www.nctm.org/standards-and-positions/Position-Statements/Artificial-Intelligence-and-Mathematics-Teaching/

https://www.appliedlearninginsights.com/blogs/thoughts-on-nctm-ai-position

NCTM Position Statement on Artificial Intelligence

Artificial Intelligence (AI)-driven tools can respond to students' thinking and interests in ways that previous tools could not. By drawing from large language sets, AI has the potential to adjust application-based problems to student interests and identify the sense students have made even in their incorrect answers. Students will continue to need teachers' mathematical, pedagogical, and relational expertise, though teachers are also likely to benefit from AI-driven tools. In some cases, AI may serve as a teaching assistant, but students will need teachers to help them create a bridge between prior knowledge, new knowledge, and shared knowledge. Teachers must tell students to be very skeptical about AI results, especially about the unique challenges of using tools that may have been trained on biased datasets. This skepticism can be woven into existing pedagogical and assessment techniques. Knowing this, educators need to be involved in developing and testing AI tools in math education to stay up to date with current AI trends to best prepare students for an AI future. Contrary to some popular opinions, this effort will require teachers with even deeper knowledge of math instruction and assessment—math teachers with more experience, not less.

https://www.nctm.org/standards-and-positions/Position-Statements/Artificial-Intelligence-and-Mathematics-Teaching/

https://www.appliedlearninginsights.com/blogs/thoughts-on-nctm-ai-position

NCTM Position Statement on AI

- AI Tools Do Not Replace the Need to Teach Math or Problem Solving
- AI Tools Encourage Teachers to Reimagine Teaching and Assessment
- AI Tools Can Personalize Learning

Other AI Tools

https://ai-search.io/search/education

https://www.futurepedia.io/ai-tools/search-engine