CS 4485 Second game design for AI Game Opponents

See: <https://youtu.be/N3tRFayqVtk> for a description for the inspiration of how I am designing the neural networks of the opponents.

Changes from the preliminary design (apart from the expanded number of neurons and 64 bit size of the genes, which are increased in number and have a way to save memory) include having the sigmoid function and arctan function replaced with f(x)=max(-1, min(x, 1)), where x is the weighted sum of inputs from source neurons to the present neuron. This function has the characteristics that it is much easier to make neuron structures similar to AND, OR, NOT, and allow a value to be preserved over many iterations (such as the specific analog values that are important in whether the AI agent sees something at a particular range), this function should also be theoretically faster to calculate. The big use of this design change is in the second appendix.

Neurons of AI agent

In this case, we will have a set of available input neurons, a set of available hidden neurons, a set of available output neurons, and 32768 connections. There is also a memory system that is an array that holds 16384 64-bit numbers accessed by the first 14 output neurons giving a memory location (as well as a read-write decision in the 15th output neuron, with the next 64 output neurons being used to hold what would be written to that location), and the previous 64 bit number in that location being given to the first 64 input neurons. The initial state of the memory is that it holds copies of the values of the first 16384 connection descriptions (the memory is instantiated to this state only when the genetic code is changed due to the creation of a new creature due to the genetic algorithm’s technique of pruning the worst AI systems).

Connections

There are 32768 neural connections in each AI (79 connections are not quite randomized). These connections are determined by that many genes in the genetic code of each AI. The genes are ordered for each of the AI systems (after being randomly decided at the start) according to the number represented by bits [15:29] (ties are then decided in the ordering according to bits [0:14]), so that the genetic algorithm can create new AIs to fill the positions of the worst AIs in the simulation with genes spliced from better AIs (a new AI filling position 1000 on the map could have connections 0 to 12458 plus 14 bits of the next connection taken from AI 4; and it could have connection 12459 (from bits 15 to 63) plus connections 12460 to 32767 be taken from AI 3). The genetic algorithm would also randomly have one bit flipped in one of the connections for each of the new AIs (except for the best five, which remain unchanged in the next generation).

The connections use an even-odd iteration system where, during an even iteration, the weighted sum of values sent to the values sent to the even activation of a neuron are used to send weighted values to the odd activation of the destination neuron, and vice versa on an odd iteration. Thus, the number of intervening connections between an input neuron and an output neuron determine the number of refreshes of the game it takes for the AI to react using that output neuron to that input neuron’s stimulus.

The genetic code uses the following system to represent the genes in binary:

[0:2][[1]](#footnote-1) unsigned int less than 5 if the source type is input neuron, otherwise the source type is hidden neuron

[3:17] address mod number of available neurons of that type is the ID of source

[18:20] unsigned int less than 3 if destination is hidden neuron, otherwise destination is output neuron

[21:34] address mod number of available neurons of that type is ID of destination

[35] sign of next integer

[36:63] integer divided by 67108864 to come up with the weight associated with the connection.

The not quite randomized 79 connections are connections chosen after all previous connections have been chosen (taken from the end of the list of connections as needed) where, if output neurons [0:78] have not all been chosen yet, the ones left unchosen shall have connections going from distinct hidden neurons to those neurons starting with the hidden neurons that have the most connections going to them. (the default weight associated with these created connections is 1) This makes sure that the AI systems have the full benefits of being able to use their own memory system, and never forgetting to have it.

Input neurons:

32768 available.

Each neuron has a value from [0, 1], many of which are going to be exactly 0 or 1 (like what is returned from memory).

[0:63] Either consists of 0s, or is the set of bits from memory that was addressed by [0:14] of the output neurons.

[64:4159] Represent the space surrounding 4096 lines from the center of the AI out to 5 feet that are spread out using an epsilon (see notes on correcting the Fibonacci spread) of 500 (with the north pole defined as normal to the top of the creature’s head), each neuron is a separate line. A value of 0 is given when an object is not in the range, and otherwise the value is equal to 1/ceil(1+number of feet from creature).

[4160:8255] Represent the space surrounding 4096 lines out of 8192 (only considering the lines that provide a view in front of or exactly to the side of the creature) spreading out of the creature with an epsilon of 600. The value is 0 when no object is in the range from 5 feet to 15 feet intersecting that line, and otherwise the value of the neuron is 1/(distance to first object in that range on that line – 4).

[8256:12351] Represent the 4096 lines of 12288 (only considering the 1/3rd of lines that are most in front of the creature (imagine the plane traveling vertically from the chest of the creature and bending 60 degrees each way)) lines spreading out of the creature with an epsilon of 700. The value is 0 when no object is in the range from 15 feet to 35 feet of the creature. The value is otherwise equal to 1/(distance to nearest of those objects – 14).

[12352:16447] Represent the 4096 lines of 16384 (only considering the 1/4th of lines that are closest to the line normal to the chest of the creature) lines spreading out of the creature with an epsilon of 750. The value is 0 when no object is in the range from 35 feet to 75 feet of the creature and intersecting the line. The value is otherwise equal to 1/(distance to nearest of those objects – 34).

[16447:18495] Represent the 2048 lines of 16384 (only considering the 1/8th of lines that are closest to the line normal to the chest of the creature) lines spreading out of the creature with an epsilon of 750. The value is 0 when no object is in the range from 75 feet to 155 feet of the creature and intersecting the line. The value is otherwise equal to 1/(distance to nearest of those objects – 74).

[18496:20543] Represent the 2048 lines of 32768 (only considering the 1/16th of the lines that are closest to the line normal to the chest of the creature) lines spreading out of the creature with an epsilon of 800. The value is 0 when no object is in the range from 155 feet to 315 feet of the creature and intersecting the line. The value is otherwise equal to 1/(distance to nearest of those objects – 154).

[20544:24639] Represent the 4096 lines of 131072 (only considering the 1/32nd of the lines that are closest to the line normal to the chest of the creature) lines spreading out of the creature with an epsilon of 800. The value is 1/sqrt(distance to nearest object at least 1 foot away that intersects the line).

[24639:32751] Represents 26 instances of listening for sounds from A0 to E4 (note, frequencies will stack) using 312 lines spreading out from the creature with an epsilon of 100.

[32752:32768] Represents feeling heat (1 for something like firebolt or a fire mephit) or cold (0, like from an ice mephit) from a given direction of 17 directions spread out in a circle. (water is value .25, normal air is value .5, nondamaging flame is from .6 to .8 depending on proximity).

Hidden neurons:

16384 available.

Each has an activation that goes from [-1, 1] with a default of 0; the activation is determined by the following function, s=, then the activation = max(-1, min(s, 1)).

Output neurons:

16384 available.

Each has an activation that goes from [-1, 1] with a default of 0; the activation is determined by the following function, s=, then the activation = max(-1, min(s, 1)).

[0:14] choosing (an activation value of 0.1 or higher is 1, an activation of -0.1 or lower is considered “0”, and in between those values nullifies the operation (meaning the first 64 bits of input neurons receive all 0s)) which address in the memory of 16384 64-bit numbers should be given to the first 64 input neurons.

[15] choosing (an activation of 0.1 or higher is 1, -0.1 or lower is 0, in between those values nullifies the operation (meaning the first 64 bits of input neurons receive all 0s)) whether to write the next 64 bits in the output neurons to memory in the location indexed by [0:14].

[16:79] the bits (an activation greater than .5 is 1, lower than that is 0) to write to the indexed location in memory.

[80:94] the bits (an activation value of 0.1 or higher is 1, an activation of -0.1 or lower is considered “0”, and in between those values nullifies the operation (meaning the neuron structure is unchanged)) choosing which of the connections to change according to the following source and destination IDs (the type (i.e. whether it connects to an input neuron, hidden neuron or output neuron) remains unchanged).

[95:108] the bits (an activation value of 0.1 or higher is 1, an activation of -0.1 or lower is considered “0”, and in between those values means keep the value in the original connection unchanged) choosing what the new source ID for that connection.

[109:122] the bits (an activation value of 0.1 or higher is 1, an activation of -0.1 or lower is considered “0”, and in between those values means keep the value in the original connection unchanged) choosing what the new destination ID for that connection.

[123] The number of connections leading to this “neuron” as a destination is interpreted as the number of “points” the creature is spending on its strength score (see “Character Creation”).

[124] The number of connections leading to this “neuron” as a destination is interpreted as the number of “points” the creature is spending on its dexterity score (see “Character Creation”).

[125] The number of connections leading to this “neuron” as a destination is interpreted as the number of “points” the creature is spending on its constitution score (see “Character Creation”).

[126] The number of connections leading to this “neuron” as a destination is interpreted as the number of “points” the creature is spending on its charisma score (see “Character Creation”).

[127:152] Producing musical notes from A0 to E4 (values 0 or lower are not producing sound, otherwise) multiply the value by 20 to get the decibels of the sound as would be detected 50 feet in front of the creature, or 10 feet behind the creature.

[153: 16384] Controlling each joint of the creature, as per below ideas:

There is a default .1 m/s^2 of friction in the left-right and forward-backward directions on the ground. Gravity is 10 m/s^2, but for winged creatures, the felt acceleration downward is .1 m/s^2 (as that is how fast they fall when not doing anything), but the creature needs to get to more than 10 m/s^2 of upward acceleration in order to flap upward (the limitation caused by dexterity for maximum acceleration only applies to the acceleration reached above 10 for the purpose of relating to the actual upward acceleration).

[153: 191] 39 of these neuron activations added together is acceleration forward as per a piecewise function (the input units is “sum of activations”, the output units is “m/s^2”) forward=lambda x, max\_acc: 0 if x<=1 else x/10 if x<=10 else sqrt(x - 10)\*(max\_acc -1)/sqrt(30) + x/10. (subtract from this result the acceleration calculated backward) However, if the final value of this acceleration is within 50% of the value of the first backwards acceleration value, then this means the creature is rolling instead (the roll speed is forward if this value minus the original calculation of backwards acceleration is greater than 0).

[192:231] 40 of these neuron activations added together is acceleration backward as per a piecewise function (the input units is “sum of activations”, the output units is “m/s^2”) backward=lambda x, max\_acc: 0 if x<=1 else x/10 if x<=10 else sqrt(x - 10)\*(max\_acc -1)/sqrt(30) + x/10. (subtract from this acceleration the acceleration calculated forward) However, if the final value of this acceleration is within 50% of the value of the first forwards acceleration value, then this means the creature is rolling instead (the roll speed is backward if this value minus the original calculation of forwards acceleration is greater than 0).

[232:271] 40 of these neuron activations added together is acceleration left as per a piecewise function (the input units is “sum of activations”, the output units is “m/s^2”) left=lambda x, max\_acc: 0 if x<=1 else x/10 if x<=10 else sqrt(x - 10)\*(max\_acc -1)/sqrt(30) + x/10. (subtract from this acceleration the acceleration calculated right) However, if the final value of this acceleration is within 50% of the value of the first right acceleration value, then this means the creature is rotating (the movement acceleration left (in a rotating frame of reference, so speed is not increasing all that much, and velocity is 0) is left(x)-right(y) (where y is the sum of activations of the next 40 neurons), and the angular acceleration of the creature (counterclockwise in radians/s^2, with a friction component calculated separately of .1+(angular velocity)^2/10) is rotation=lambda x, y, max\_acc, radius: sign(left(x)-right(x))\*sqrt(min(left(x), right(y)))\*max\_acc/(sqrt(40)\*radius^2).

[272: 311] 40 of these neuron activations added together is acceleration right as per a piecewise function (the input units is “sum of activations”, the output units is “m/s^2”) right=lambda x, max\_acc: 0 if x<=1 else x/10 if x<=10 else sqrt(x - 10)\*(max\_acc -1)/sqrt(30) + x/10. (subtract from this acceleration the acceleration calculated left)

[312: 351] 40 of these neurons added together is the amount of work (in units of (strength score + dexterity score)/40 Newton\*meters) being exerted upward (forward if the limb is down) by the creature’s top right limb (the forelimb furthest from the shoulder).

[352] the right hand is opened if 1, closed if -1, stays the same otherwise.

[353] right pointer finger is extended forward if the right hand is closed (thus is pointing)

[354:383] 40 of these neurons added together is the amount of work (in units of (strength score + dexterity score)/40 Newton\*meters) being exerted leftward by the same limb.

[384: 423] 40 of these neurons added together is the work (in units of (strength score + dexterity score)/40 Newton\*meters) being exerted upward (forward if down) of the top right limb closer to the shoulder of the creature.

[424: 463] these neurons added together is the work (in units of (strength score + dexterity score)/40 Newton\*meters) exerted leftward of the top right limb closer to the shoulder of the creature.

[464: 503] added together is the work (in units of (strength score + dexterity score)/40 Newton\*meters) exerted upward by the furthest out top left limb.

[504] left hand is opened if 1, closed -1, stays the same otherwise.

[505] left pointer finger is extended forward if the left hand is closed (pointing)

[506: 545] added together is the work (in units of (strength score + dexterity score)/40 Newton\*meters) exerted rightward by the furthest out top left limb.

[546: 585] added together is the work (in units of (strength score + dexterity score)/40 Newton\*meters) exerted upward (forward if down) by the closer to shoulder top left limb.

[586: 625] added together is the work (in units of (strength score + dexterity score)/40 Newton\*meters) exerted rightward by the closer to shoulder top left limb.

[626: 667] added together is the work (in units of (strength score + dexterity score)/40 Newton\*meters) exerted leftward by a creature’s special limbs (such as a tail, or wings).

[668:700] this is the number (the activations of these neurons added together) of 60ths of degrees by which the chest of the creature should bend forward towards its back limbs vs the current relative position of the chest, this means the creature will bend forward by at most 33 degrees per second (at 60 refreshes per second assumed).

[701:702] both of these values must be 1 in order to start the picking up object animation, and must remain 1 while the animation takes place (unless it is an object that can be held in one hand and the fist of the creature is closed). The object picked up is the nearest object that is smaller than the creature using this action.

[703:704] both of these values must be 1 and the pick up action in [701:702] must be completed and still be 1, then the object held is thrown using an amount of force determined by [705:800], on an angle upward from straight on determined by [801:888], and leftward by [889:900].

[705:800] the number of (sum of these activations)\*(strength score) newton\*meters to use on throwing the object.

[801:888] the sum of these activations is the angle upward from straight from chest to throw the object, in thirds of degrees.

[889:900] the sum of these activations is the angle leftward from straight from chest to throw the object, in degrees.

# Character Creation

Each player character starts with 75 “points” to use on six different ability scores listed below. Increasing a score for the first 13 points of that score only costs 1 point per increase by 1. Increasing a score from 13 to 14, or 14 to 15, and so on up to 16 costs 2 points per increase in the score by 1. Increasing a score from 16 to 17, and so on up to 20 costs 3 points per increase in the score by 1 (fractional increases are not allowed). Increasing a score from 20 to 21, and so on up to 25 costs 4 points per increase in the score by 1. Increasing a score from 25 to 26, and so on up to 30 costs 5 points per increase in the score by 1. The marginal cost, in points, of increasing a score by 1 thus increases by 1 for every 5 increases in the score after 20.

Strength: The damage dealt by a physical attack that hits has a minimum damage of 1+(strength score-10)/2, rounded down. The amount of weight a creature can lift is equal to 1+29\*(strength score + dexterity score) in newtons. The ability of a creature to grapple or resist a grapple is determined by a contest of running the following for each creature, rand(40)+strength score, and seeing which creature gets a higher number.

Dexterity: The acceleration of the furthest point from the fulcrum of rotation of a joint on a creature due to the use of that joint is no higher than 1+sqrt(dexterity score) in m/s^2. This then defines the speed of the creature and jumping height.

Constitution: The hp of the creature is calculated the following way: 10 plus 3\*(each increase in the constitution score up to 5) plus 2\*(each increase in constitution score up to 10) plus each increase in constitution score after 10. Thus a 5 Con score creature has 35 hp; a 10 con score creature has 45 hp; and a 20 con score creature has 55 hp. A creature falls unconscious when brought to 0 hp, and dies when brought to –(max hp)/2 (this causes the creature to disappear from the game). Hp is regained at a rate of 1 hp per (20/(1+(constitution score)\*sqrt(sqrt(constitution score)))) minutes.

Intelligence (not a necessary score for the AI systems since it is already defined by the neural network): This defines the probability that a magical attack will hit a target you are aiming at as rand(40)+intelligence score must be greater than rand(40)+dexterity score of an opponent.

Wisdom (not needed for an AI opponent since this is already defined by the neural network): defines the resolution of how far away you can see before things become blurred as 25 ft plus 5 ft \* (your wisdom score)^3.

Charisma: Defines the amount of magic points you have and the speed at which they are recovered as: magic point max = 60 + 5\*(increases in charisma until 5) + 4\*(increases in charisma score from 6 to 15) + 3\*(increases in charisma score from 16 to 30) + 2\*(increases in charisma score from 31 to 50) + (increases in charisma score from 51 onward); magic point regeneration = 1 point per (400/(1+charisma score^2) minutes). The cost of casting a spell is 1+ceil((level of spell^2.1). The maximum level of a spell is level 7 (61 magic points).

Spellcasting

The spells to be implemented in the game are:

|  |  |  |
| --- | --- | --- |
| Level of spell | Example | Example 2 |
| 0 | Minor Illusion | Firebolt (1d10 (plus another d10 if your charisma is 20, plus another d10 if your charisma is 30, plus a fourth d10 if your charisma is 40)) |
| 1 | Silent Image | Witch Bolt |
| 2 | Mirror Image | Misty Step (teleport exactly 30 feet in the direction you are facing) |
| 3 | Animate Dead (target a creature that is at negative hp, but is not yet at negative ½ that creature’s maximum hp. The creature rises as an undead form of itself (having a number of hit points equal to 10 + abs(number of hp it had while at negative hp), which becomes the new max hp) that is directed to target whatever creature it sees you point at with your pointer finger of your right hand (stopping either when it drops to 0 hp or when it sees you no longer pointing). The undead nature of the creature is such that the creature cannot cast spells, is immune to poison damage and damage from occult weapons, cannot regain any hp, is vaporized instantly if targeted by the Heal spell, and takes double damage from weapons that deal radiant damage.) | Conjure Animals |
| 4 | Stone Shape | Conjure Minor Elementals |
| 5 | Creation | Far Step (play temporarily stops while you choose up to 26 squares in the area (120 ft radius from where you cast the spell) to associate each letter of the alphabet to, and for that 1 minute, each time you press that character capitalized, you teleport to the relevant location. |
| 6 | Heal (restore 70 hp to a creature within 120 feet (or yourself); increases hp maximum to 70 if not already at 70 or higher; removes status conditions except grappled and restrained) | Arcane Gate (has the duration increased to 24 hours, and the range increased to 500 ft + 2\*median distance between locations of interest on the map, and allows you to see the map and the locations to put at each side of the Gate) |
| 7 | Mirage Arcane (enter God-mode (without being able to see creatures) to alter terrain and add/subtract buildings, then return to initial location you were at time of casting) | Teleport (replaces fast travel, simply look at the map and indicate where you would like to go; each creature within 30 feet of you is brought with) |
| 8 (these are implemented as part of debugging the system and making sure that AIs can be uploaded without the game crashing) | New Heart (upload binary file of size 524288 bytes, this file is interpreted as the 2\*\*15 genes (64 bits each) of the connections, the 2\*\*14 entries in the maintained memory, and the 2\*\*14 hidden neuron activation values. The nearest creature excluding the caster (including a potential opponent player character) or 3D model file to the caster at time of casting becomes a “permanent” (until killed) aspect of the game as a creature using that mind and that form. A player character has the effect of being insta-killed from their own point of view.) | Read Mind (doesn’t affect a player character, and can’t be cast by an AI. This spell downloads a 917504 byte representation of the current state of the mind of the creature (the last 393216 bytes of the file represent the current activations of the input neurons and output neurons (as double values) closest to the caster (that must be controlled by an AI)). |

The order the spells are implemented is:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Spell name | Verbal pattern | Left hand pattern | Right hand pattern | Movement | Upcasting |
| Firebolt |  | Point up, rotate hand across chest to point right, rotate hand down | Point down, rotate hand across chest to point left, rotate hand up | Step forward once within 5 seconds of starting the hand movements, step backward once within 5 seconds of ending the hand movements | (Nothing, doesn’t spend any more magic points) |
| Witch Bolt | Note A4 (440 Hz ± 15 Hz) played within 5 seconds after left hand points forward | Point up, rotate to point forward | Point down, rotate hand across chest to point left | Step left once before starting hand movements (do so within 5 seconds of starting hand movements) | Increase initial damage by 1d12 on a hit |
| Heal | Note C followed by note H, followed by note E followed by note A followed by L | Point at creature to be healed before making note C (do so within 5 seconds of making note C) | Point at creature to be healed after making note L (do so within 5 seconds of making note L, and within 8 seconds of making note C) |  | Nothing, doesn’t spend any more magic points. |
| Misty Step | (Do the following over the course of 1 second or less after pointing):  frequency 1700±200 Hz for .03 seconds minimum, 2500±300 Hz for .03 seconds minimum, 3200±300 Hz for .03 seconds minimum. The first of those frequencies must start at or before the next two frequencies, the second frequency has to start at or before the third frequency. This is the only spell where the sound component can have overlapping parts. | Point in a direction (you will travel exactly 30 feet in that direction at .03 seconds of the third sound running). |  |  | Nothing, doesn’t spend any more magic points. |
| Mirror Image | (Do the following over the course of 6 seconds or less):  Note C, followed by note M, followed by note I, followed by note R for at least 1 second, followed by note O, followed by note R |  |  |  | Double the duration for each level. |
| Teleport | (Do the following over the course of 8 seconds or less):  Note C, followed by note T, followed by note E, followed by note L, followed by E, followed by P, followed by O, followed by R, followed by T |  |  |  | Increase radius of circle of creatures brought with to new location by 5 feet for each level. |
| New Heart | C, N, E, W, <bring both hands apart, then bring both hands (open) across chest>, H, E, A, R, T | Hand to the left, then (within 1 second of right hand moving left) bring this hand across chest by moving it right. Hand must be open. | Hand to the right, then bring this hand across chest to left. Hand must be open. |  |  |
| Read Mind | (11 seconds total or less)  C, R, E, A, D, <bring both hands apart, then bring both hands (open) across chest>, M, I, N, D | Hand to the left, then (within 1 second of right hand moving left) bring this hand across chest by moving it right. Hand must be open. | Hand to the right, then bring this hand across chest to left. Hand must be open. |  |  |
| Arcane Gate | (5 seconds total or less) C, G, A, T, E |  |  |  | Double the range and increase radius of portal by 5 feet per level. |
| Far Step | (4 seconds total or less) C, F, A, R |  |  |  | Double the duration per level |
| Mirage Arcane | (Do the following over the course of 13 seconds or less):  M, I, R, A, G, E, A, R, C, A, N, E | 3 minutes’ (± 20 seconds) worth of arm movements following (by an amount of time from 0 seconds to 5 seconds) the last verbal component E1: Move arm up, move arm down, rotate pointer finger out to the right, rotate pointer finger in front of chest to the left, rotate fist closed to the right, rotate hand open to the left, bring arm straight up, strike arm down. | 3 minutes’ (± 20 seconds) worth of arm movements following (by an amount of time from 0 seconds to 5 seconds) the last left hand movement: Move arm up, move arm down, rotate pointer finger out to the right, rotate pointer finger in front of chest to the left, rotate fist closed to the right, rotate hand open to the left, rotate hand open to the right, bring arm straight up, strike arm down. | 3 minutes’ worth of movement following the last right hand movement: | Nothing, doesn’t cost any more magic points. |
| Stone Shape |  | To activate stone shape, move this hand up, then use a closed fist downward, then move this hand left, then move this hand up with a closed fist, then move this hand to center with a closed fist.  Point at rock to shape prior to taking the right hand actions (start pointing within 5 seconds of starting movements with right hand). Continue pointing (doesn’t have to be at the same place) until done with right hand actions. | Raise hand up, then use the following pattern (will involve a combination of right hand movements and position movements) to gradually build up whatever you are making from the stone chosen:  (A stone object is created at the end (and a 5 ft cube (less as a percentage of the number of slices not taken out of 60) of the previous piece of stonework pointed at is destroyed) using up to 60 of 60 x 60 bit maps stacked on top of each other where each “1” in a bit is a 1 inch cube of stone, and “0” is a 1 inch cube of empty space):  a “1” in a bit map is represented as pointing this hand outward (straight out from chest), a “0” in the bit map is represented as pointing downward, pointing right means going to the next bit of the map (ordered left-to-right and forward-to-backward then bottom level-to-top-level height), moving this hand left puts the work in neutral. The work created (at whatever step you stop drawing this stack of bit maps) is saved for use in Minor Illusion, Silent Image and Creation. | After raising your right hand up, you can move your right hand left (neutral for the 3D printing model), and proceed to move backwards, drawing a 1 inch by 1 inch by 5 ft long shaft from the middle of the area pointed to, reset to do this again by moving right, then forward, then right, then left twice such that the hole in the stone is slightly off your center (thus drawing the next shaft from the adjacent stone. | Nothing (doesn’t cost more magic points) until level 7, at which point you can choose a 3D model file from your computer that is scaled to fit in a 5 ft by 5 ft cube, and the cube of rock targeted disappears. |
| Minor Illusion |  | To activate quick Minor Illusion, move this hand up, then use a closed fist to the right, then move this hand left, then move this hand to center with a closed fist. If you have previously used Stone Shape, Minor Illusion, Silent Image, Creation, then whatever image was used the latest time of those is immediately replicated and scaled to fit within a 5 ft cube. If you have not previously used one of those spells, then the form this illusion takes is a 5 ft cube of black. The illusion will last for 1 minute when cast this way.  To activate slow Minor Illusion, move this hand up, then use a closed fist to the right, then open the hand and move it up, then move this hand left, then move this hand to center with a closed fist.  Slow minor illusion then proceeds by presenting a drawing board on the ground that only the caster can see, which takes up a 60 ft by 60 ft square area centered where the caster is standing. The drawing panel drops away as soon as the caster opens its left hand. The illusion that is created this way then lasts for 10 minutes. | Point at the location the illusion is to form from.  If using slow casting, the pointing to where the illusion is to appear need only last for one second; after that, you can choose the color of different areas of the board to be something other than black by pointing right to have the current square increase its blue value by 1 per 1/120th of a second you stand there (and increase the blue value (wrapping around mod 256) of the 121 surrounding squares (in an 11 ft square surrounding the square you are standing on) by 1 per 1/10th of a second in their area. The same is done by pointing to the center to get green value increases, and pointing left to get red value increases. Pointing up (for human player characters) results in seeing the previous illusions or structures the player has created, and the player stands in front of that illusion and points to center to choose the preferred one. | (Movement only required if using slow minor illusion, and only while drawing): The drawing board initially starts all white (255 red, 255 green, 255 blue), then changes in a given 1ft square centered on the player by (-1 red, -1 green, -1 blue) (rgb color scale) for each 1/60th of a second the player character stands in that location. Once the left fist is opened, the illusion is created starting from the base of the illusion in the following way: the illusion is base is scaled to a 5 ft square based on the 60 ft by 60 ft drawing board; the height from that base at any 1-inch-square point is a function of the rgb value of the corresponding 1 ft square of the drawing board f(red, green, blue)=(255\*sqrt(3) - sqrt(red\*red +blue\*blue +green\*green))/51 feet. The color of each 1 inch square base column of the resulting illusion is equal to the color of the corresponding 1 ft square of the drawing board. | Nothing, doesn’t cost any more magic points. |
| Silent Image |  | To activate quick Silent Image, move this hand up, then use a closed fist to the right, then move an open hand down, then move a closed fist right, then move this hand left, then move this hand to center with a closed fist. If you have previously used Stone Shape, Minor Illusion, Silent Image, Creation, then whatever image was used the latest time of those is immediately replicated and scaled to fit within a 15 ft radius sphere. If you have not previously used one of those spells, then the form this illusion takes is a 5 ft cube of black. The illusion will last for 10 minutes when cast this way.  (For player characters) To activate Slow Silent Image, move this hand up, then use a closed fist to the right, then move an open hand down, then move an opened hand left, then move a closed fist right, then move this hand left, then move this hand to center with a closed fist. The game then opens a panel to choose from previous created structures and illusions and allows for files to be imported which represent 3D models or 2D images. These images or models then exist for 10 minutes at the location chosen.  (For AI characters) To activate Slow Silent Image, move this hand up, then use a closed fist to the right, then move an open hand down, then move an opened hand left, then move a closed fist right, then move this hand left, then move this hand to center with a closed fist. The illusion created is a representation of the current activations of the hidden neurons of the neural network: the 16384 hidden neuron activations represent 2 bits each (lambda activation: “0b11” if activation>.5 else “0b10” if activation > 0 else “0b01” if activation > -.5 else “0b00”) where the bits are separated into 512 64-bit blocks, where the first 8 bits represent transparency (0 is fully transparent), the next 24 bits are the rgb color value, and the last 32 bits represent the additive (non-positional) distance from the center of the 15 ft sphere, such that 32 ones equals 32 times the distance of one 1, so each 1 in those 32 bits represents 0.46875 feet). The points can be considered as having their directions be as evenly distributed about a sphere as possible given 512 points, shifted inward or outward according to the 32-bit distance measure. The object created via those points uses those points as the corners of triangles. It is as though the triangles are drawn while all the points are the same distance from the center to start, with each point making triangles to form a complete hexagon with the six adjacent points, then the triangles are elongated on weird lines as the points move to their respective distances. | Point at the direction the illusion will form for the duration from one second before you move your left hand up to one second after your left hand moved up. Define the distance to where the illusion will form from by closing this hand, moving it to center, then moving this hand right to indicate 1 (in binary, as the first digit), left to indicate 0 (in binary as the first digit), then move to center to indicate making the next digit of the binary number. Open this hand to finish casting the spell, at which point, the illusion will form a number of feet away equal to that binary representation.  The illusion will remain that many feet away in that direction when you move. You can point at the illusion and then move your hand towards your chest to move the illusion 20% closer to you. |  | Double the duration per level. |
| Creation | To activate Creation: C, R, E, A, T, I, O, N  Then proceed in one of the following ways:  (if caster is an AI) build a piece of matter made the same way as in the slow form of silent image. The illusion made this way lasts 24 hours, and is physical. The material is unbreakable except with a sword (which can be made of the same material); the breaks from the material are always clean along the alignment of the blade (if the sword doesn’t align its edge to the material within 1 milliradian, then the material does not cut).  (if caster is a human) show a panel of example objects including the five most recent illusions created by AIs, and have an option to upload a 3D model to make into a real form in the game. If the user uploads a unique object the user has a 3D model of, then the object is permanent (if the user previously uploaded an object, then the new object only lasts 24 hours), otherwise it lasts 24 hours (example objects made of metal only last 1 hour). |  |  |  | Increase the height, width and depth of the illusion by 5 feet per level. |
| Animate Dead | C, D, E, A, D | Point at creature before saying C1 (and within 5 seconds of saying C1) | Point at creature after saying last D1 (and within 9 seconds of pointing with left hand. |  | Casting at level 5 allows you to give the notes R, E, T, U, R, N to call all creatures made this way to you (ending their combats), ad saying the notes S, T, A, Y causes the nearest of these creatures to not move until you say R, E, T, U, R, N. Casting at level 4 only spends level 3’s magic points, casting at level 6 or higher only spends level 5’s magic points. |
| Conjure Animals (lasts for 1 hour, or until you cast it again) | (say the following using no more than 14 seconds for “Conjure Animals” in the equivalent notes using the above notation, then use no more than 5 seconds to say the equivalent of “snake”, “bear”, “wolf” using the same notation of music notes for each letter) the first C must last for 1 second minimum.  Saying “snake” means that 1 Giant Constrictor Snake is called forth and attacks the next creature you point at.  Saying “bear” calls forth 2 Brown Bears, which attack the next thing you point at.  Saying “wolf” calls forth 8 wolves, which attack all creatures in front of you (distributing themselves evenly) when you point at a creature. | Point at a creature to have the animals attack that creature. | Point at a creature to have the animals attack that creature. |  | Every 2 levels you increase this spell, the number of creatures summoned increases by the base number of creatures summoned. |
| Conjure Minor Elementals (lasts 1 hour, or until you cast it again) | C for at least 1 second, then do the equivalent of each letter in “onjure minor elementals” using the notes with no more than 1 second per letter on average. This creates a dust Mephit (will create a cloud of dust around an AI enemy making it think it is surrounded), a small fire elemental (20 hp, deals 1d10 fire damage per 6 seconds it is near (within 5 feet) of a creature excluding an ally), an Ice Mephit (will deal 1d10 cold damage and 1d10 piercing damage each time it hits an enemy, has 10 hp, and blows up with 10 hp of cold damage against enemies within 5 feet when killed), and an Earth elemental (100 hp, 30 strength, will only try to restrain the creature). | Point at the creature you want the creatures to attack. | Point at the creature you want the creatures to attack. |  | Every 2 levels you increase this spell, the number of creatures summoned increases by the base number of creatures summoned. |

The rules for upcasting a spell are the following: Ai’s can’t upcast spells; Players on PCs can upcast spells by typing a number higher than the level of a spell within 5 seconds before casting the spell.

Appendix 1

Based on research done here: <http://extremelearning.com.au/how-to-evenly-distribute-points-on-a-sphere-more-effectively-than-the-canonical-fibonacci-lattice/> we can see that we can more evenly distribute points around a sphere using epsilon=3.33, phi=(1+sqrt(5))/2, (x0, y0)= , theta (longitude in radians) = i/phi, psi (latitude measured 0 radians at poles, pi radians at equator) = (i+epsilon)/(n-1+2\*epsilon). Towards the end of the research, it is advised that most engineering models will be better off with epsilon=.36, as that gives the optimal average neighbor distance, but it ends up clustering points at the poles. In our project, we want to avoid having points close to poles entirely (the points at the bottom of an object or very top of an object are generally unseen, and the vision of AI creatures which are bound by lines equally spaced around a sphere should not be pointing upward or downward, since those are hardly going to be places from which they are attacked.

The code is:

from numpy import arange, pi, sin, cos, arccos

n = 50

if n >= 600000:

 epsilon = 214

elif n>= 400000:

 epsilon = 75

elif n>= 11000:

 epsilon = 27

elif n>= 890:

 epsilon = 10

elif n>= 177:

 epsilon = 3.33

elif n>= 24:

 epsilon = 1.33

else:

 epsilon = 0.33

goldenRatio = (1 + 5\*\*0.5)/2

i = arange(0, n)

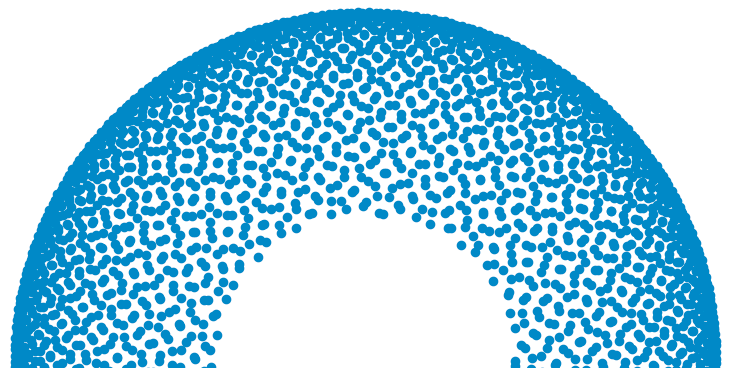
theta = 2 \*pi \* i / goldenRatio

phi = arccos(1 - 2\*(i+epsilon)/(n-1+2\*epsilon))

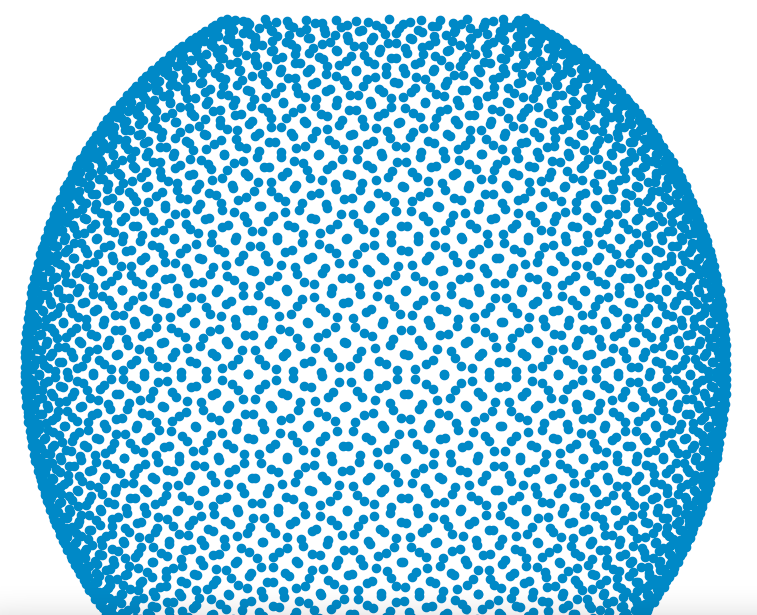
x, y, z = cos(theta) \* sin(phi), sin(theta) \* sin(phi), cos(phi);

Based on human eye ratings here: <https://www.quora.com/What-is-the-minimum-distance-between-two-objects-that-the-human-eye-can-possibly-discern> we see that a human eye can differentiate between two objects 1.75 mm apart at 6 meters, so that is .00175/6 ratio between distance between objects vs distance to the eye.

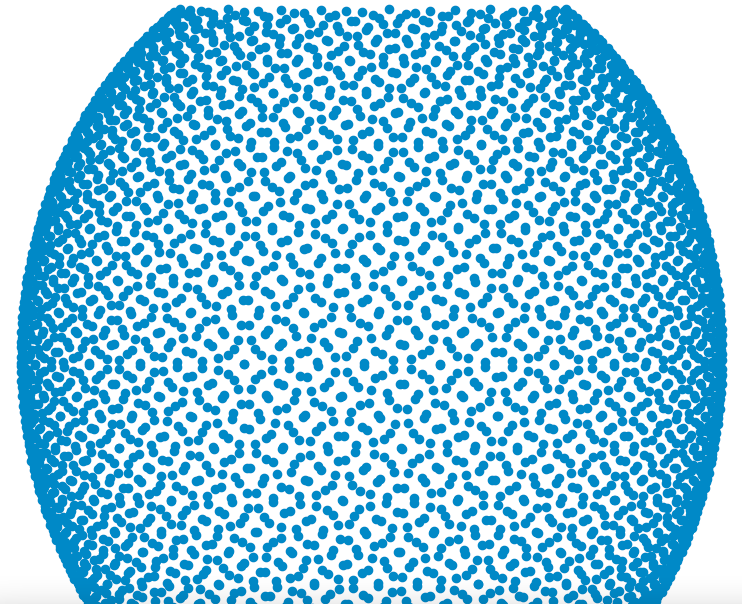
There is a way we can cheat with the AI’s eyes to get to a higher precision while only using 4096 neurons for each range, we can massively increase the value of epsilon. Here is a top view of a sphere of 4096 points (of which half of the sphere is shown) where epsilon is set to 214:



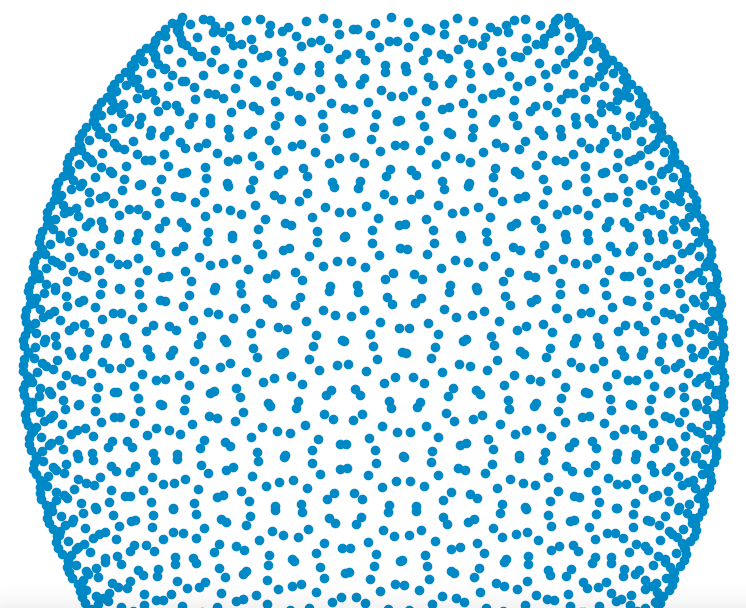
Here is the side view of the sphere:



This is the side view after increasing epsilon to 410:



This is the same sphere after removing the points from the backside of the sphere:



The notation for the verbal component of spells uses the following chart of musical note frequencies, with A0 being “A” and “a”, B0 being “B” and “b”, C1 being “C” and “c” (and being the initial note made when using this notation), and going up by 1 full note per letter in the 26 letters of the alphabet. The lower limit and upper limit of what frequiencies are considered to satisfy the constraints of a given letter are defined as the standard frequency of the letter in the chart, plus and minus min(difference in frequency to different note-.01)/2.

|  |  |  |  |
| --- | --- | --- | --- |
| Alphabetic letter | Note | Frequency (Hz) | Wavelength (cm) |
| A | A3 | 220.00 | 156.82 |
|  | A#3/Bb3 | 233.08 | 148.02 |
| B | B3 | 246.94 | 139.71 |
| C | C4 | 261.63 | 131.87 |
|  | C#4/Db4 | 277.18 | 124.47 |
| D | D4 | 293.66 | 117.48 |
|  | D#4/Eb4 | 311.13 | 110.89 |
| E | E4 | 329.63 | 104.66 |
| F | F4 | 349.23 | 98.79 |
|  | F#4/Gb4 | 369.99 | 93.24 |
| G | G4 | 392.00 | 88.01 |
|  | G#4/Ab4 | 415.30 | 83.07 |
| H | A4 | 440.00 | 78.41 |
|  | A#4/Bb4 | 466.16 | 74.01 |
| I | B4 | 493.88 | 69.85 |
| J | C5 | 523.25 | 65.93 |
|  | C#5/Db5 | 554.37 | 62.23 |
| K | D5 | 587.33 | 58.74 |
|  | D#5/Eb5 | 622.25 | 55.44 |
| L | E5 | 659.25 | 52.33 |
| M | F5 | 698.46 | 49.39 |
|  | F#5/Gb5 | 739.99 | 46.62 |
| N | G5 | 783.99 | 44.01 |
|  | G#5/Ab5 | 830.61 | 41.54 |
| O | A5 | 880.00 | 39.20 |
|  | A#5/Bb5 | 932.33 | 37.00 |
| P | B5 | 987.77 | 34.93 |
| Q | C6 | 1046.50 | 32.97 |
|  | C#6/Db6 | 1108.73 | 31.12 |
| R | D6 | 1174.66 | 29.37 |
|  | D#6/Eb6 | 1244.51 | 27.72 |
| S | E6 | 1318.51 | 26.17 |
| T | F6 | 1396.91 | 24.70 |
|  | F#6/Gb6 | 1479.98 | 23.31 |
| U | G6 | 1567.98 | 22.00 |
|  | G#6/Ab6 | 1661.22 | 20.77 |
| V | A6 | 1760.00 | 19.60 |
|  | A#6/Bb6 | 1864.66 | 18.50 |
| W | B6 | 1975.53 | 17.46 |
| X | C7 | 2093.00 | 16.48 |
|  | C#7/Db7 | 2217.46 | 15.56 |
| Y | D7 | 2349.32 | 14.69 |
|  | D#7/Eb7 | 2489.02 | 13.86 |
| Z | E7 | 2637.02 | 13.08 |

This is a section of the table published by mtu.edu <https://pages.mtu.edu/~suits/notefreqs.html> Called “Physics of Music”

I found on this page: <https://blog.accusonus.com/pro-audio-production/human-voice-frequency-range/> That the frequency range of the human vocal cords goes from 125 Hz to 8000 Hz, with the fundamental frequency of a female voice being 200 Hz, so I shifted up the sounds used in the verbal components of spells.

Appendix 2

In order to implement logic gates, you need a consistent source of the value 1 in one of the input neurons, thus I will set the least significant bit of each gene (the last bit that is determining the weight of a connection) to be 1 for the first half of all the genes (that get written into memory), and I will make sure that the output connections that determine which address in memory to find will always be consisting of 1s and 0s.

The internal representation system of logic for one of these AI creatures does not have to depend on the representation of 1s and 0s inside the output neurons (which has a 1 be an activation in [0.1, 1], and 0 be an activation in [-1, -0.1], and is designed that way to accommodate having a randomly generated AI being able to use its abilities under a wide variety of circumstances), because the values can be scaled to those ranges as soon as useful calculations are done with the internal representation. Therefore I will choose to hand-write one of these AI systems using an internal logical representation of having 1 be an activation in [0.5, 1], and 0 be an activation in [-1, -0.5] because that substantially reduces the number of genes that need to be used on amplifying a signal to 1.

An AND gate can be manifested using the equation equivalent to

neuron\_and=lambda A, B, C=1: max(-1, min( max(-1, min( A\*.51+B\*.51 + C \*(–.3825) , 1))\*4 , 1))

Which is equivalent to the following diagram:

.3825

.51

.51

4

And is equivalent to the following genetic code, where A is hidden neuron 10011100111001, B is hidden neuron 10001100110001, and the result is stored in hidden neuron 10000111100001. The input neuron for the least significant bit of the memory input is 000000000111111 (the constant 1, made constant by having the least significant bit of each gene be 1).

source dest sign weight

101010011100111001 00010000111100000 0 00 10000010100011110101110001

101010001100110001 00010000111100000 0 00 10000010100011110101110001

000000000000111111 00010000111100000 1 00 01100001111010111000010101

101010000111100000 00010000111100001 0 11 11111111111111111111111111

The OR gate can be manifested using the equation equivalent to

neuron\_or=lambda A, B, C=1: max(-1, min(max(-1, min(max(-1, min(A\*2, 1))\*1 + max(-1, min(B\*2, 1))\*1 + C\*.5, 1))\*4, 1))

Which is equivalent to the following diagram:

2

2

0.5

1

1

4

OR is equivalent to the following genetic code, where A is hidden neuron 10011100111001, B is hidden neuron 10001100110001, and the result is stored in hidden neuron 10000111100001.

source dest sign weight

101010011100111001 00010000111100100 0 10 00000000000000000000000001

101010001100110001 00010000111101000 0 10 00000000000000000000000001

000000000000111111 00010000111100000 0 00 10000000000000000000000001

101010000111100100 00010000111100000 0 01 00000000000000000000000001

101010000111101000 00010000111100000 0 01 00000000000000000000000001

101010000111100000 00010000111100001 0 11 11111111111111111111111111

The NOT operation is simply having the weight from one hidden neuron to another hidden neuron be -1.

1. This is designed such that 5/8ths of the time, the source of a connection (when decided randomly) will be an input neuron. {18:20] is similarly designed so that 5/8ths of the time, the destination of a connection is an output neuron. This system mostly guarantees that a completely randomly decided neural network will have many connections that lead directly to useful outputs that can be observed by players. [↑](#footnote-ref-1)