Learning Collaboration Based On the Environment

CS263C: Final Project

Valerie Runge & Tomer Weiss

Hypothesis

Animats will learn to collaborate, using Q-Learning with individual indirect rewards, and through that collaboration harvest together when surrounded by food that requires two or more animats to harvest it. When food requires a single animat to harvest, animats will not collaborate. This collaboration will be measured with two metrics, the number of times the animat harvests with another animat in the same zone and the average number of animats in a zone. If animats cluster together in the same zone reaching an average of over 1 animat in each zone this indicates that animats are collaborating.

Phase 2: Introducing Expiration Dates for Food
Animats will be forced to learn to collaborate faster in order to successfully harvest then eat the food in low expiration date environments. Low expiration dates (ie foods that quickly expire and disappear from the environment) will cause more clustering and more collaboration, while high expiration dates will cause less clustering and less collaboration since there is less pressure from the environment to quickly find and harvest the food.

Phase 3: Introducing the Store Action
Storing would be represented in the environment as decreasing the age of the food by half, thus decreasing the effect of expiration date. With the storing ability, animats will be able to collaborate more by preserving the food and then calling for help to harvest it.

Goal

The goal of this experiment was to build animats that could learn to communicate and work together to harvest food without explicitly being told to collaborate. Animats were given the basis for communication (ability to 'yelp' and to hear other animats yelps), but not told what this communication meant nor rewarded for using this. Any rewards or training needed to reward individual actions (such as eating), not collaborative behaviors. By using reinforcement learning with an indirect reward system, the animats were rewarded based on an individual action (eating) and not collaborating.

Issues/Problems

The first major problem encountered was with the choice of learning, neural networks. Neural networks work well for a variety of tasks and had seemingly less restrictions than other types of learning. The first iteration of phase 1 used neural network learning in Pybrain, however animats were unable to consistently learn with this type of learning. Results were very inconsistent with animats all dying in the environment one turn and then all living the next turn indicating that animats were not learning correctly. At this point, Q-learning was decided upon as the learning method. Q-learning was decided upon because it is an off-policy model free reinforcement learning technique. [1] Q-learning allowed the animat to freely explore and learn, perhaps making incorrect decisions, and then receive a reward or punishment depending on the outcome of their decisions. It also enabled an indirect reward system based on the specific action of eating successfully in the environment. Eating successfully is an individual reward, not related to collaboration. By using an individual reward, animats were not forced to collaborate, but had to learn to collaborate in order to get the reward. Once Q-Learning was introduced as the learning technique, animats successfully learned to collaborate consistently.

Another issue was finding correct values for how much energy should be gained or lost with each action. Since animats are not aware of their own energy level, this metric is used in part to measure how successfully they are at harvesting and then eating food. Energy is only gained when an animat successfully eats harvested food. Animats with an energy level below zero are removed from the environment. Values had to be chosen carefully to enable the animat to survive. For the energy gained by eating carrots and big carrots, the energy values were not an issue as the animat could survive variability in the energy levels. For cucumbers, the energy gained had to be above a certain level significantly higher than either carrot or big carrot. An entire cucumber has 27 energy portioned into 3 equal parts, so eating a cucumber is equal to gaining 9 energy but this can be done three times. However, both a carrot and a big carrot have a total energy of 12 with big carrot being partitioned into 2 equal parts. If the cucumber's energy is less than 9 per eat, then the animats consistently die off before the 1000 turns of the experiment have completed.

Randomization in all aspects of the program posed a major issue throughout all the phases. In order to accurately replicate the environment, animats had to be able to randomly change zones, randomly move towards yelps and randomly take turns. At first, animats had the choice of moving to the next zone in the zone list when they moved and of moving towards a 'yelp'. This led to animats grouping together and moving consistently together from one zone to the next. By restricting the movement to this pattern, the animats were not able to move independently. Since this experiment focused on indirectly encouraging collaboration, this type of movement was not acceptable as it forced the animats to all move in the same pattern. Moving towards a 'yelp' had to also be randomized as at first it was implemented as moving towards the first 'yelp' in the yelp list. This made animats more likely to move towards the first animat that yelped, again biasing animat movement. By having the animat choose a random yelp from the yelp list, this bias was removed. The major issue with randomization was randomizing turn taking. At first, the animats would take their turn based on their sequence number with animats created first going first every turn. This made animats with a lower sequence number more likely to survive as they could take advantage of the fact that they always went first. Different behavior began emerging for animats in the beginning of the list. Randomizing this turn taking prevented this behavior.

[1] Sutton, Richard S. & Andrew G. Barto, "Reinforcement Learning: An Introduction", http://webdocs.cs.ualberta.ca/~sutton/book/ebook/node65.html

Methodology

First, we decided on a basic environment that would allow us to study collaboration. The goal of this environment was to abstract away details that would not matter in measuring collaboration. These details included how the animat moves from space to space and how it moves around obstacles. The next step was to build the neural network of the animat. This encompassed thinking about the necessary capabilities of each animat, eg. the input sensors and actions the animat could perform. This involved thinking about how the animat would communicate, how it would know that another animat was attempting to communicate with it, and how it would take an action based on that communication. After finalizing the capabilities of our animat, we coded the environment and then the animat.

This experiment focused on the effect of different environments with different food types on learning to collaboration and measured collaboration by seeing how many animats were close to each other (i.e. in the same zone) and how many times they harvested together in the same zone. Energy level of the individual animat was used to show how well individual animats were surviving in the environment. Animats who were doing well in the environment would have higher energy levels while animats who did poorly would die off. This was purposely kept separate from collaboration so as to not directly reward collaboration. Energy level also brought realism to the experiment as animals in the real world need to eat to survive and each action needs to cost a certain amount of energy. However, since energy level was not the focus of our experiment, in the results this was abstracted away to if the animat survived.

Since movement was not important to measuring collaboration, most of our abstraction centered around movement. The environment was separated into 10 zones with animats being able to instantly transport themselves from one zone to another. The animat's body's capabilities was also abstracted away. Animat's were restricted to being able to only do one action per turn no matter what the action was. By abstracting away the animat's body's capabilities, we were able to focus on collaboration instead of the capabilities of an individual animat.

Instrumentation

For each experiment, we ran 10 trials with 15 starting animats, and analyzed the average of those results, so that our results will be more robust and accurate. For each trail, we trained the animats for 3000 turns not decrementing energy, then ran the simulation for 1000 turns.

Recorded Data:

Animat Actions: Each animat has either four or five possible actions it can do each turn. Each turn an animat must choose an action to perform. We recorded the index of each action for the last 20 turns in order to observe the animat's actions and see if there were any patterns. This pattern work was not completed.

Animat Energy Level: Each animat has an energy level that increases when it eats food and decreases when it either performs actions other than eating or lives a single turn (cost of living). The animat's energy level measures the animat's success in coping with the environment. We measured the energy level every 20 turns of the simulation.

Food Consumed: Measures the food type and total number of each food type consumed by each animat throughout the simulation, to see what types of food the animat is harvesting. This measurement records how many pieces of food the animat ate and what type of food was that it ate.

Animat Q-learning States: Recorded the state the animat was in the last 20 turns.

Zone Results: For each living animat, we recorded the zone that the animat was in for the last 20 turns.

Collaboration Table: For each animat, we recorded the individual collaboration table, which stores how many times that animat harvested with every other animat.

Environment & Physics

The environment is divided into zones, and each zone can contain at most one food, and any number of animats. We have only one type of animat in the environment, and every experiment has a starting population of 15 animats.

Our environment contains the following: animats and food. Animats can move from zone to zone. Moving from zone to zone is immediate, so every zone is just one step away from other zones. Similarly for sounds – sounds diffuse to all zones at once over one turn, but disappear in the next turn. There is no wind or sounds other than the animats' yelps.

Food appears at random locations on the map, and is generated only in zones that don't contain animats. In phases 2 and 3, food expires after a certain amount of turn, specified in the test, but animats can extend the expatriation date by storing food.

Simulation turn

Each turn we randomize the order in which each of the animats act, so to not give bias to any particular animat. Each animat takes an action in it's turn. If at the end of the turn the animat's energy level is lower then 0, we remove it from the simulation. At the end of the turn, we remove all consumed harvested food.

Animat

Each animat has an individual q-learning neural network. The network receives the current state the animat, which is translated from the animats environment sensors, and outputs a selected action.

Capabilities of each Individual Animat:

1. Sensors:
    a. Is cucumber in zone – 1/0 (Sensor for phase 2,3 only)
    b. Is big carrot in zone – 1/0
    c. Is carrot in zone – 1/0

    d. Are there other animats in the zone –
       For phase 1 – 1/0
       For phase 2- 2/1/0, e.g. at there are at least 2 other animats in the zone.
    e. Is food harvested - 1/0
    f. Has an other animat yelped in previous turn - 1/0

2. Actions:
    a. Eating harvested food.
    b. Attempt to harvest food.
       If animat harvests food that requires collaboration, but there are not enough other other animats harvesting together, the food will remain unharvested.
    c. Move to another random zone.
    d. Yelp for help harvesting.
       Other animats will hear the yelp in the next turn.
    e. Moving towards yelp
       If the animat hears a yelp, it could choose to move towards the zone from which that yelp originated.
    f. Store food (Phase 3 only)
       Extends a food's expatriation date by making the food "younger" – the current age of the food is divided by half, e.g. if the food is 4 turns old, it will be just 2 years old.

Physical Structure of Animat

1. Animats would have hearing capabilities, so they could know the zone from which the sound originated.
2. Animats would have a sense of smell, which allows them to know if food is located in the current zone, and what type of food it is.
3. Animats would be able to move to any zone from any zone.

Action/Energy Changes for each Animat

Action	Energy
CONSUME_FOOD	12
CONSUME_BFOOD	6
CONSUME_CFOOD	9
TRIED_EAT	-0.1
HARVEST	-0.2
MOVE_ZONE	-0.3
YELP_HELP	-0.05
COST_OF_LIVING	-0.05
STORE_FD	-0.2

1. Living
    a. Each turn decreases the animat's energy level by COST_OF_LIVING energy
    b. When the energy level reaches 0, the animat 'dies' and is removed from the environment
2. Harvesting Food
    a. An additional amount of energy would be needed to harvest food during the animats turn, HARVEST energy.
3. Storing Food
    a. An additional amount of energy would be needed to store food during the animats turn, by STORE_FD energy.
4. Eating Food
    a. This would increase the animat's energy level, depending on the food type by CONSUME_FOOD,CONSUME_BFOOD,CONSUME_CFOOD respectivley for carrots, big carrots and cucumbers.
    b. Every time the animat tries to eat food it will decrease it's energy level by TRIED_EAT.
5. Communication with other animats
    a. An additional amount of energy would be needed to yelp for getting help from another animat harvesting, by YELP_HELP energy.
6. Moving
    a. Each time the animat moved the animat will lose MOVE_ZONE energy

Food
There are 3 possible food types: carrot, big carrot and cucumber.

In order to survive, animats eat, but in order to eat, the animats have to harvest food first. The difference between each food is the number of animats required to harvest it: carrot requires only 1 animat, big carrot requires at least 2 animats, and cucumber requires at least 3 animats.

Experiment Results
Phase 1:

Experiment	Food type	Average Number of Animats per Zone
1	Carrot	1.7
2	Mixed, ½ each	1.8
3	Big Carrot	1.73

In experiments 1, 2 & 3, the only difference is the type of food available in the environment. In experiment 1, only carrots are available, experiment 2 has an equal number of carrots and big carrots, and experiment 3 has only big carrots.

For food type carrot, we see the least amount of collaboration, with more collaboration with big carrot, and somewhere in the middle for experiment 2,when the available food type is mixed.

For phase 1, the implementation of the yelping mechanism was not fully perfected and could have some remaining bugs that were fixed in later versions of the code.

Phase 2:

Experiment	Food type	Expiration Date	Average Collaboration Value	Average # Animats per Zone
1	Carrot	10	51.57666667	1.649279762
2	Carrot	50	51.26949495	1.634976189
3	Carrot	150	50.00150794	1.615291666
4	Big Carrot	10	65.26666667	1.279166667
5	Big Carrot	50	97.81363636	1.77248611
6	Big Carrot	150	97.6430303	1.800825397
7	Cucumber	10	109.6217172	1.640515873
8	Cucumber	50	190.2208654	2.513132937
9	Cucumber	150	188.453382	2.419470238
10	Mixed	10	70.90785714	1.582380952
11	Mixed	50	119.7172894	2.169069444
12	Mixed	150	121.3712683	2.135878968

p Values Comparing Average Number of Animats per Zone In Low Expiration Environment (10) to High Expiration Environment (150)

Type of Food	p Value
Carrot	0.321320336
Big Carrot	0.002369563
Cucumber	0.007504555
Mixed	0.000192651

p Values Comparing Average Collaboration Value in Low Expiration Environments (10) to High Expiration Environment (150)

Type of Food	p Value
Carrot	0.14457561
Big Carrot	4.77894E-05
Cucumber	6.29117E-05
Mixed	2.34118E-10

Effect of Expiration Date on Collaboration for each Type of Food

Experiments 1, 2 & 3: All carrots with variation in expiration date

For all three of these experiments, average collaboration level stays at around 50. There is slight variation, but this variation is normal and not statistically significant with a p-value of about .32. Average number of animats per zone is also about equal for all these trials with a p-value of about .14. Animats were able to survive consistently across all trials and experiments.

Experiments 4, 5 & 6: All big carrots with variation in expiration date
For experiment 4, collaboration metrics are significantly lower than for experiment 6, with an average number of animats per zone at 1.28 and an average collaboration value of 65.27. Experiment 6 has on average 1.80 animats per zone and an average collaboration value of 97.64. Animats were not able to successfully survive in all experiments. In low expiration experiment trials, animats died out about 50% of the time. In high expiration experiment trials, animats survived in all trails.

Experiments 7, 8 & 9: All cucumbers with variation in expiration date
For experiment 7, collaboration metrics are significantly lower than experiment 9, with an average of 1.64 animats per zone and an average collaboration value of 109.62. Experiment 9 has on average 2.41 animats per zone and an average collaboration value of 188.45. In low expiration experiment trials, animats died out about 70% of the time. In high expiration experiment trials, animats survived in all trials.

Experiments 10, 11 & 12: Mixed (1/3 of each type of food) with variation in expiration date
Like both big carrots and cucumbers, for experiment 10 with low expiration dates, collaboration metrics are significantly lower than experiment 12 with high expiration dates, with values of 1.58 animats per zone and an average collaboration value of 70.91. In experiment 12, there are 2.14 animats per zone and an average collaboration value of 121.37. In low expiration experiment trials, animats had significantly lower number, and died out about 20% of the time. In high expiration experiment trials, animats survived in all trials and had significantly higher numbers of remaining animats.

Comparing Carrots, Big Carrots and Cucumbers
Using the middle expiration date, collaboration metrics gradually increase from carrots to big carrots to cucumbers. For experiment 2 with only carrots, there are on average 1.63 animats per zone and an average collaboration value of 51.27. This increases in experiment 5 with only big carrots to an average of 1.77 animats per zone and an average collaboration value of 97.81. With cucumbers in experiment 8, this is even greater with on average 2.51 animats per zone and an average collaboration value of 190.22.
For the mixed experiments collaboration metrics were higher than big carrots but lower than cucumbers.

Phase 3:

Experiment	Food type	Expiration Date	Average Collaboration Value	Average Number of Animats per Zone
1	Carrot	50	37.51	1.75
2	Big Carrot	50	78.12	1.72
3	Cucumber	50	142.84	2.01
4	Mixed	50	96.81	2.00

In experiments 1, 2, 3 & 4, all the food types have the same expiration date of 50, and food type available is varied. Experiment 1 has only carrots, experiment 2 has only big carrots, experiment 3 has only cucumbers and experiment 4 has an equal mixture of carrots, big carrots and cucumbers.

The average collaboration value consistently increases from experiment 1 to 3. In experiment 1, the average collaboration value is 37.51, then in experiment 2 it increases significantly to 78.12 and in experiment 3, it increases even more to 142.84. Experiment 4's collaboration value is between experiment 2 and 3 with a value of 96.81. Average number of Animats per Zone stays relatively consistent at around 2 with some variation for carrots and big carrots.

Analysis/Interpretation of Results
Phase 1:

The animats move around in different zones looking for food, so if an animat hears a yelp, it could learn that it has to move, but the eventual zone which the animat reaches might not contain a yelping animat, because the animat only hears the yelp in the next turn.

There is no significant increase in collaboration metrics from the phase 1 data, which goes against our hypothesis that predicts more collaboration with big carrots and less collaboration with carrots. However, this data could result from one of two errors. One is that this data could have been caused by errors not fixed in phase 1, but fixed in phases 2 and 3. Or, since phase 1 only measured the zones that animats were in for the last 20 turns, zones could be an imprecise measure for collaboration which is why the collaboration table was introduced in phases 2 and 3.

Phase 2:
Effect of Expiration Date on Collaboration for each Type of Food

For carrots, expiration date had little to no effect on collaboration. Collaboration levels stayed steady across experiments 1, 2 and 3. These results are consistent with our hypothesis, which states that animats in carrot environments will have little to no collaboration.

For big carrots, expiration date had a significant effect on collaboration. In low expiration environments, collaboration metrics were significantly lower. In addition, animats died out more often. Animats dying out more indicates that that low expiration environment were harsher environments. In harsh environments, in which it is harder for the animat to survive, our hypothesis predicts that animats would learn quicker to work together to harvest food and have higher collaboration metric levels. The data completely contradicts our hypothesis giving significantly lower collaboration metrics for low expiration date environments in comparison to high expiration environments. Significantly lower collaboration metrics means that animats worked together less and collaborated less in harsh environments.

Cucumber environments and mixed environments had similar results to big carrots. Collaboration metrics were significantly lower for low expiration experiments than for high expiration experiments.

Comparing Carrots, Big Carrots and Cucumbers

For the low expiration date environments, the results for average number of animats per zone are higher for carrots than for big carrots and cucumbers. This can be explained by high competition within the animats for harvesting and eating carrots. This high competition in carrot environments means that the animats rapidly move towards food to harvest it causing them to group around food and compete to harvest and eat it first. Further evidence for this, is the animats collaboration values, which increase from carrots to big carrots to cucumbers. The average number of animats per zone would seem to disagree with our hypothesis, but by examining the average collaboration values, the data can be interpreted to agree with our hypothesis and indicate more collaboration in big carrot and cucumber environments.

For both medium and high expiration date environments all of the collaboration metric data agreed with out hypothesis. Big carrots had higher collaboration metrics than carrots indicating more collaboration. Cucumber environments had on average more collaboration than carrot environments and big carrot environments as well as mixed environments. Mixed environments had higher collaboration metrics than big carrots but lower collaboration metrics than cucumbers indicating more collaboration in these environments than in big carrot environments, but less than in cucumber environments. These results indicate that animats collaborated the least in carrot environment, more in big carrot environment, and the most in cucumber environments, which is consistent with our hypothesis. Mixed environments had relatively high collaboration levels indicating that even when given the choice between food types, animats can learn to collaborate and work together to harvest food.

Phase 3:

Our hypothesis predicts that the available food type in the environment will affect collaboration with carrots having the least collaboration and cucumbers have the most collaboration. The data from average collaboration value supports this In environments filled with only carrots, animats have the lowest average collaboration value indicating the least amount of collaboration while in environments filled with only cucumbers, animats have the highest average collaboration value indicating the most collaboration. In environments with equal amounts of carrots, big carrots, and cucumbers, animats have a higher average collaboration value than that in environments filled with only big carrots but less than that in environments filled with only cucumbers. This indicates that animats in mixed environments collaborate and learn to work together to harvest food more than in environments with only big carrots.

The data from average number of animats per zone does not follow the same trend as the average collaboration value. Unlike phase 2, this data does not exhibit an upward trend, which could indicate less collaboration in big carrot environments. The high values and significant upward trend of average collaboration value, dispute this theory, indicating that these results do not accurately represent collaboration in phase 3. This data does suggest that animats did not cluster in the same zone as much as in phase 2, but this could be for a variety of reasons. One possible reason is that with the added ability to store, animats yelped less and instead went around storing food, waiting until other animats were around to harvest and eat food.

Current Status of Work

Phases 2 and 3 are completely coded with expiration date as well as the ability to store food added. Metrics to measure collaboration have been added as well. More metrics were added in the code, such as measuring the amount of each food type the animat harvested, but this data was not recorded. For each experiment there was a significant amount of data collected that could not be properly analyzed, such as the last 20 actions and states. Overall, the coding is complete for phases 2 and 3 with no known bugs, but the experimental data could not be fully analyzed.

Contributions-

We both worked on every part of the project, but each team member contributed different parts. Both of us came up with the vision for the project, hypothesis and learning mechanism.

Tomer built the ground framework of the environment. Valerie built the neural network, Q-learning and animat abilities. Valerie refined the framework of the environment, and came up with ideas on how to make phase 2 and 3 better based on the experience we had with phase 1. Valerie built the initial versions of phase 2 and 3, and Tomer helped refine them. Valerie designed and built methods for recording experiment data, and Tomer helped with coding and further automation of data gathering.

Both Valerie and Tomer wrote the project paper.

Tools/Packages (with URLs)

Python: Main Language

Q-learning class was imported from user studywolf at github.com:
https://github.com/studywolf/blog/blob/master/RL/Cat%20vs%20Mouse%20exploration/qlearn_mod_random.py

All other classes were created specifically for this project in python

Appendix

Key pieces of code:

- Main loop:

```
e = envionrment(FOOD_TYPES)
for x in range (0,1000):
  e.oneTurn(False)
  if x%20==0:
    e.recordResultsEnergy(energyFile)
  if x%3 == 0:
    e.randomlyCreateFood(FOOD_TYPES)
```

- Method for simulating 1 turn passing in environment:

```
  #For each animat still "alive" has the animat perform an action, then if the animat does not
  #have a positive amount of energy removes it from the environment
  #Updates the yelps in the environment
  #Input: train (boolean) True if this should be a training round and not decrement energy, false if
not
  #Return: None
  def oneTurn(self,train):
    anim2remove=[]
    temp=copy.copy(self.animatNum)
    random.shuffle(temp)
    for i in temp:
      self.animats[i].takeAction(train)
      if self.animats[i].energyGet()<0:
        anim2remove.append(self.animats[i])
    for a in anim2remove:
      self.removeAnimat(a)
    self.lastyelps=self.yelps
    self.resetYelpStatus()
```

- Method for getting animat's input sensors:

```
  #Input: None
  #Return: (Integer) to represent the index-1 of the matching state from the states file
  def AgentInputSensorsStateGet(self):
    sensors = [self.isNearCucumber(),self.isNearCarrot(), \
          self.isNearBigCarrot(), self.isOtherAgentInZone(),\
          self.isFoodHarvested(),self.isHearOtherAgentCry()]
    if self.isOtherAgentInZone()>2:
      sensors[3]=2
    res = allStates.index(sensors)
```

```
return res
```

- Methods for simulating an animat performing an action and learning from past actions using Q-Learning table

```
#-------------------------------------------------------------------
#Animat Q-Learning Related Methods - Taking Action based on Q-Learning
#-------------------------------------------------------------------

#Updates the Animat's Q-Learning Table then calls performAction to change then environment
#Decides which action the Animat should take based on its Q-Learning table
#Updates preAction and preState after the action is performed
#Input: train (boolean) representing whether this is a training round or not
#Return: None
def takeAction(self,train):
  inp=self.AgentInputSensorsStateGet()
  reward=self.calReward(self.preAction,self.preState)
  self.nn.learn(self.preState, self.preAction, reward, inp)

  curAction=self.nn.chooseAction(inp)
  if (train):
    self.performAction(curAction,True)
  else:
    self.performAction(curAction,False)
  self.preAction=curAction
  self.preState=inp

#Changes the Environment and Animat to represent the Animat performing the action
#Uses the current environment and the current Animat's status and modifies them
#Can change: if food is in zone by eating, if food is harvested by harvesting,
#what zone Animat is in by moving, if there are yelps in environment by yelping
#Input: actidx (Int) number representing the action the Animat has chosen to take
#      train (boolean) True if the animat is training and not decrementing energy, False otherwise
#Returns: (Boolean) True if the animat should get a reward, False otherwise

def performAction(self,actidx,train):
  #Set local variables at zero, keep at zero if training so do not decrement energy
  #These variables represent energy that animat will lose
  cons_f1=0
  cons_f2=0
  cons_f3=0
  tri_eat=0
  harv=0
  move_z=0
  yelp=0

  #If not training, set the local variables to correct values
  #Decrement the Animat's Energy for living one Turn
  if not train:
    self.turnCostOfLiving()
    cons_f1=CONSUME_FOOD
    cons_f2=CONSUME_BFOOD
    cons_f3=CONSUME_CFOOD
```

```
    tri_eat=TRIED_EAT
    harv=HARVEST
    move_z=MOVE_ZONE
    yelp=YELP_HELP

#Do the action (action indexes are specified above under outputs)
#Eat
if actidx==0:
    if self.isFoodHarvested()==1:
        if self.zone.isFoodCarrot():
            self.energyModify(cons_f1)
            self.eatenFood[0]=self.eatenFood[0]+1
        elif self.zone.isFoodBigCarrot():
            self.energyModify(cons_f2)
            self.eatenFood[1]=self.eatenFood[1]+1
        else:
            self.energyModify(cons_f3)
            self.eatenFood[2]=self.eatenFood[2]+1
        self.zone.eatFood()
        return True
    else:
        self.energyModify(tri_eat)
        return False
#Harvest
elif actidx==1:
    self.energyModify(harv)
    if (self.isFoodHarvested()==0) and (self.zone.food != None):
        if self.zone.isFoodCarrot():
            self.harFood[0]=self.harFood[0]+1
        elif self.zone.isFoodBigCarrot():
            self.harFood[1]=self.harFood[1]+1
        else:
            self.harFood[2]=self.harFood[2]+1
        self.zone.harvestFood()
        return False
    else:
        return False
#Move Randomly
elif actidx==2:
    self.energyModify(move_z)
    newzoneidx=self.findRandomZone(self.zone)
    self.moveZone(self.envir.zones[newzoneidx])
    return False
#Yelp
elif actidx==3:
    self.energyModify(yelp)
    self.yelpforhelp()
    return False
#Move towards closest friend's yelp (or stay in same place if no yelps)
elif actidx==4:
    if len(self.envir.yelpsGET())>0:
        yelps=self.envir.yelpsGET()
        newzone=random.randint(0,len(self.envir.yelpsGET())-1)
        if newzone != self.zoneGET().idxGet():
            self.energyModify(move_z)
```

```
        self.moveZone(self.envir.zoneGET(int(newzone)))
    return False
  else:
    print "Error with actidx",actidx
    return False
```

An example of Data derived:
Animat energy results from single trial phase 2 experiment 1: