Part 3 – Reinforcement Learning with Backpropagation

New Attempt

Due Wednesday by 23:59 **Points** 35 **Submitting** a file upload
Available 1 Nov at 0:00 - 31 Dec at 23:59

Part 3 – Reinforcement Learning with Backpropagation: Instructions

In the third and last part you are to replace the LUT used by the Reinforcement Learning component with a neural net. This will be easy to do in your software if your class headers for the LUT and the neural net match. However this is a little tricky to get working since there are many parameters that can be adjusted. Furthermore, the training data is dynamic as it is generated by the Robocode environment. I.e. it is not an a-priori static set of training data. This means that it is not possible to compute a “total error” during training, which also means that there is no conventional way to judge if learning is happening or not.

Some hints:

- It will help if you capture your look up table from part 2. You now have a static set of data, which you could use to train a neural net. Why is this useful? Well, you will be able to use your neural net software from part 1 and instead of training the XOR problem, apply your robocode tank data. This will allow you to adjust the various learning parameters to get the best training for this “kind" of data upon a static training set. (e.g. learning rate, momentum term, number of hidden neurons). Once you’ve identified a good set of parameters, they should work once your neural net is used on live data.
- It will be helpful for you to be able to load and save the weights of your tank. That way you can save your tank’s behaviour and recall it later as needed.
- In part 2, you should have found that it is absolutely necessary to reduce dimensionality of the state space to prevent having to deal with impractically large look up table. Perhaps with a neural net, this is not necessary?

Now answer the following questions:

Questions

(4) The use of a neural network to replace the look-up table and approximate the Q-function has some disadvantages and advantages.

a) There are 3 options for the architecture of your neural network. Describe and draw all three options and state which you selected and why. (3pts)

b) Show (as a graph) the results of training your neural network using the contents of the LUT from Part 2. Your answer should describe how you found the hyper-parameters which worked best for you (i.e. momentum, learning rate, number of hidden neurons). Provide graphs to backup your selection process. Compute the RMS error for your best results. (5 pts)

c) Comment on why theoretically a neural network (or any other approach to Q-function approximation) would not necessarily need the same level of state space reduction as a look up table. (2 pts)

(5) Hopefully you were able to train your robot to find at least one movement pattern that results in defeat of your chosen enemy tank, most of the time.

a) Identify two metrics and use them to measure the performance of your robot with online training. I.e. during battle. Describe how the results were obtained, particularly with regard to exploration? Your answer should provide graphs to support your results. (5 pts)

b) The discount factor γ can be used to modify influence of future reward. Measure the performance of your robot for different values of γ and plot your results. Would you expect higher or lower values to be better and why? (3 pts)

c) Theory question: With a look-up table, the TD learning algorithm is proven to converge – i.e. will arrive at a stable set of Q-values for all visited states. This is not so when the Q-function is approximated. Explain this convergence in terms of the Bellman equation and also why when using approximation, convergence is no longer guaranteed. (3 pts)

d) When using a neural network for supervised learning, performance of training is typically measured by computing a total error over the training set. When using the NN for online learning of the Q-function in robocode this is not possible since there is no a-priori training set to work with. Suggest how you might monitor learning performance of the neural net now. (3 pts)

e) At each time step, the neural net in your robot performs a back propagation using a single training vector provided by the RL agent. Modify your code so that it keeps an array of the last say *n* training vectors and at each time step performs *n* back propagations. Using graphs compare the performance of your robot for different values of *n*. (4 pts)

(6) Overall Conclusions

a) This question is open-ended and offers you an opportunity to reflect on what you have learned overall through this project. For example, what insights are you able to offer with regard to the practical issues surrounding the application of RL & BP to your problem? E.g. What could you do to improve the performance of your robot? How would you suggest convergence problems be addressed? What advice would you give when applying RL with neural network based function approximation to other practical applications? (4 pts)

b) Theory question: Imagine a closed-loop control system for automatically delivering anesthetic to a patient under going surgery. You intend to train the controller using the approach used in this project. Discuss any concerns with this and identify one potential variation that could alleviate those concerns. (3 pts)

For this last and final part, you should submit a written report that includes answers for each of the questions above.

Your report should be well structured, written clearly and demonstrate an understanding of the backpropagation and reinforcement learning paradigms. For each of these, it should describe the problem being addressed and provide an analysis of how that learning mechanism was applied. It is important that you describe how your solution was evaluated and offer a conclusion. Pay attention to your results and be scientific in your approach.

Try to be as thorough and clear as possible with your answers, which may not be unique. When providing explanations, supporting your statements with practical results helps. You'll be judged based on what you can deduce from your experiments and how well you understand the theory. Try to make your report as neat and presentable as possible. It takes many hours (days in fact) to read and mark all material and there is no opportunity to re-read the work. It really helps if your work is presented neatly and is easy to read. Don't be surprised if you lose marks for poorly organized material. You should use tables, graphs and diagrams to help in this respect.

Please also format your report such that you precede each answer with a copy of the question in *italics*.