PY-599 (Fall 2018): Applied Artificial Intelligence
Homework Assignment
Deadline to Submit: Tuesday, Sept 11th
Submission Method: Please share your Colab notebook
Group Submission is allowed for task 3, training Perceptron using GA; Max group size: 3 people.

But please do not abuse this option! I am allowing group submission because of people that may not feel comfortable with Python or the course yet. Those people can team up with more experienced students. But if you are comfortable with Python and with the GA and can do this on your own, please do it alone – unless you are teaming with those students who need your help!

---

1. Implement a pure random search to find the global minimum point of the given function *f* in the Session 4 Colab notebook, titled Search Mechanisms.

This is a very simple task and somehow it confused many people in the class! You are going to randomly generate 2D inputs, and send it to function *f* and observe the outputs, and take the minimum one as the result! You can do this in a loop, but the better way to do is to go with the vectorized implementation, where you create a NumPy array of size (N,2), where N is the number of your tries, and 2 is the dimension. Initialize the array with random numbers, and send it over to function *f*. And get back the array of the results. And find the minimum one.

The point of this task is to observe in practice that with randomly (and blinding) shooting at a target, you have a very low chance to hit it, especially in a non-convex function with many deep and sharp local and global minimums. Also this method to find the global minimum point does not scale well with the increase of the dimension. You are going to need exponentially many more number of tries as the dimension increases.

2- Find another interesting 2D function *f* that is non-convex, and try the search methods on it.

With this task I am just trying to motivate you to apply the techniques we study in the class to the other tasks, hopefully the ones that is more relevant to your field. Our main mission at PY-599 Applied AI class is to learn AI as a skill-set and apply it to our field of research and study. So always keep thinking about "How can I apply this method that I just learned in PY-599 to the problems that I face in my own field?"

This task (2) of homework assignment is trying (in a very basic way) to follow this mission! Go and find another *f*, preferably something that you have encountered before in your field, and apply the search methods to your own function. In PY-599 class you should be ALWAYS thinking about how I can apply this technique/method/theory/code to my own projects or problems!

3- Train a Perceptron using Genetic Algorithm to classify the two-class data set.

On Thursday Sep 6th lecture, we talked about GA, and we executed a Python implementation of GA that finds the global minimum point of function *f*. In this task we like to use this code to train a perceptron. Training a learning algorithm can be considered as an optimization problem; we are iteratively changing the parameters of the learning algorithm in order to minimize the cost function.

A single perceptron is a simple learning algorithm that can classify two-class data sets that are linearly separable, which is our case in this task. In the class some were commenting that we should use GA to learn the slope and bias value of a line that separates these two classes of data. Well, that is pretty much what a single perceptron is and does. Go and take a look at the slides for Thursday Sep 6th lecture (Session 5), where I explain perception. The perceptron's equation is a linear model with a nonlinear activation function, and in this case we are using a simple step function as the activation function.

We will be discussing perceptron and multilayer perceptron in future sessions in great depth. But for time being let's have a short, quick overview. The beauty of neural networks is that they are universal function approximators. And they learn from data that which function they have to approximate in order to map inputs to the outputs. So we don't have to look in the data and figure out what that function should be. Here in this extremely simple example, the 2D data is so simple that we can eyeball it and say that a straight line should be the classifying function. As we move forward, the data will be more complex, and we will have no way to see or guess what that function should be. That is why neural networks are so useful; they learn and figure out that function on our behalf! Treat this simple example in the same way! Let's all act like we don't know what that function should be, and let the simple perceptron figure it out from the data.

Perceptron is a parametric model, and training perceptron (learning) is nothing more than finding the parameters based on the training data. We have the labeled training data. And we have GA that we are going to use to find the parameters that optimizes perceptron's classification job! Optimality in this context is defined as the fewest number of misclassification.

Last note; the GA we implemented optimizes functions with two parameters, but the two input perceptron has three parameters. You have two options to proceed. 1- Update and upgrade the GA to optimize three-parameter functions. 2- Use the trick I showed in the lecture, which was to get rid of one of the parameters of the perceptron and transform it to a two-parameter function. See the last slide of Sept 6th lecture. Either way you still have to hack the GA implementation to some extend in order to make it to train the perceptron, but option 2 requires minimal changes compared to option 1.

Please start from "Genetic Algorithm to Train a Perceptron" Colab notebook, where you can find the training data set, the perceptron, and the base GA method. All codes and notebooks can be found on the course website. Good luck!