Projectile Position Prediction Report

February 2, 2017

1 Assumption

This problem is a free falling problem in physics. So actually we can just figure out the initial speed on x direction and y direction to do the simulation. If we are not told what scenario it is, we may still plot x and y versus t, then we will find that we can just fit a linear function for (t,x) and a quadratic function for y using lesat square error. So in order to make this problem more like machine learning problem than physic problem, we assume that we don't know this is a free falling problem, and the function relation cannot be gained from graph.

2 Model and Evaluation

According to our assumption, we cannot just fit the linear and quadratic model, but we still know there should be some functional relation between the projectile's current position and its two previous position, i.e. we can build a function x(i),y(i)=f(x(i-1),y(i-1),x(i-2),y(i-2)), where i is an arbitary time point. Due to neural network's high strength of approximating arbotary functions, we built a 4-layer neural network to predict positions. The input layer contains 4 units x(i-2),y(i-2),x(i-1),y(i-1) and the output layer contains the prediction x(i),y(i). In order to train the model, we split our data into training set and test set with 20% of whole data in the test set. The evaluation function is the L2 error

$$error = \frac{1}{2m}\sum_{i=1}^{m}((x_i - xpredict_i)^2 + (y_i - ypredict_i)^2) \tag{1}$$

where m is the number of instances. After fine-tune parameters, we finally got a model with test error 0.312. We then use this model to continuely predict positions for projectile in A1. The predict result looks not good, that may be because we are lack of data in some areas.