1 Problem Definition

We want to model displacements of a projectile launched from origin (0, 0) with unit mass, unknown speed and angle at every 100ms intervals. Air friction will be ignored and gravity will be assumed 9.8 m/s^2 for the simplicity. This model should be learned by a machine learning algorithm to predict displacements of other projectiles at every 100ms intervals.

2 Modeling the Problem

We are trying to model the motion of an object. Motion can be represented by direction and speed which can be calculated from two coordinates on the path with time between these two coordinates. Hence, I will use most recent two coordinates to predict next coordinate. My model generates next coordinate using most recent predictions until predicted y coordinate becomes negative. I will employ linear regression model and multilayer perceptron (MLP) model with one hidden layer containing 10 neurons and *ReLU* activation function.

3 Theoretical Analysis of the Model

Trajectory of a projectile which launched with speed v and angle θ can be described with following equation according to [1]

$$(x_t, y_t) = (v\cos(\theta), v\sin(\theta) - \frac{1}{2}gt^2) \tag{1}$$

where g is gravitational and t is time instance.

Let t_0 be length of interval ($t_0 = 100$ms for our case), and (x_k, y_k) be the coordinate of the projectile at kt_0th moment. Then, we try to teach following equations to machine learning algorithms,

$$f(x_k, y_k, x_{k+1}, y_{k+1}) = (x_{k+2}, y_{k+2}) \tag{2}$$

and these variables satisfies following recurrence relations derived from Eq. 1

$$x_{k+2} = 2x_{k+1} - x_k \tag{3}$$

$$y_{k+2} = 2y_{k+1} - y_k - gt_0^2 \tag{4}$$

Both Eq. 3 and Eq. 4 are linear relations as gt_0^2 is a constant. Linear regression model should capture these relations perfectly, while MLP might struggle to learn correct representation as training set is relatively small.

4 Experiments & Results

Linear regression has no hyper parameters, however MLP has several hyper parameters. I did not fine tune these hyper parameters due to time limitation, instead I used default values and my experience to select reasonable network. Linear regression easily learned the function and generalize it to other samples. This can be seen from both Fig. 1 and Table 1. MLP, also learned and generalized the problem however its performance is worse than Linear regressions. This is most probably result of small number of samples and lack of fine tuning.

Table 1: Absolute prediction error for various v and t

v	t	Linear Regression	MLP
5	15	$1.77e-13 \pm 2.51e-13$	$3.87e-04 \pm 5.48e-04$
20	30	$1.31e-11 \pm 1.39e-11$	$3.72e-02 \pm 3.56e-02$
10	45	$4.72e-12 \pm 5.68e-12$	$1.47e-02 \pm 1.51e-02$
25	60	$3.76e-10 \pm 3.38e-10$	$1.89e+00 \pm 8.03e+00$
10	75	$5.98e-12 \pm 5.72e-12$	$2.38e-02 \pm 2.42e-02$

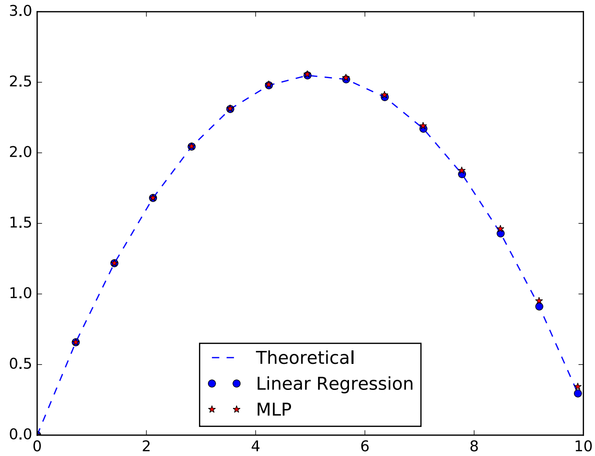

Figure 1: Predicted paths for $v = 10$m/s and $\theta = 45$

5 Discussion

1. **What made you choose this specific model?**
   I have chosen Linear Regression for its simplicity, and I have chosen MLP for its representation power on complex functions.
2. **Is your model well suited to the learning objective?**
   Theoretically suggested model is linear. Hence, both algorithms are suited for this specific task. This can also be seen from the error rates.
3. **How would you evaluate your model?**
   I believe, Linear Regression was perfect fit for this task with proposed features. It is easy to train, and easy to compute. Furthermore, it captures relation between features and outcome perfectly.
4. **Can your model predict projectiles launched at arbitrary angles and velocities equally well (or badly)?**
   Both of the model can predict trajectories of projectiles launched with unknown speed and angle successfully. This can seen from the Table 1. In fact, Linear regression will predict theoretical path perfectly, while MLP may fail at some parameters if parameters are highly unrelated from training samples. This can happen because MLP has enormous representation capacity and small number of samples may not be sufficient to converge global optima.
5. **What assumptions does your model make?**
   I assumed that next coordinate can be computable if previous two coordinates given. Also, I initially assumed that relation was linear. However, this was a basic assumption and I would replace it with more complicated model if it did not hold.
6. **Will your approach/model change if we hadn't told you that the data was from a projectile?**
   If I did not know that the data came from projectiles, I would not select most recent two coordinates as features. I would probably employ a sequence predicting model such as RNN instead of utilizing simpler algorithms to predict next element in the sequence using already predicted ones.
7. **Did you refer to any relevant literature while solving this problem?**
   I refer to physic books to domain specif information. I felt comfortable with my existing machine learning experience to model and implement this particular problem, and so I did not make significant literature review.
8. **If you were given enough time, how would you improve the model?**
   The given problem solved perfectly by the linear regression. Linear regression is a simple and easy to understand method, as such I would not make any improvements on this model even extra time allowed. On the other hand, I can improve MLP with RNN architecture to generate sequence automatically if I had more time and more samples.

References

[1] Douglas C. Giancoli. *Physics for Scientists & Engineers with Modern Physics*. Pearson Prentice Hall, fourth edition, 2009.