In the last six weeks we In September I developed a framework for auto-differentiation of arbitrary neural networks. This framework enables us to minimize losses of the form $\min g(\bigtriangledown f(x))$. Using this framework we ran a series of experiments aimed at capturing a physically based solution to $\min_\phi \sum_{t=1}^{T} \| \dot{\boldsymbol{r}}_t \cdot \bigtriangledown \phi \left(\boldsymbol{r}_t \right) \|_2^2$. Early results produced trivial solutions e.g. $\bigtriangledown \phi(\boldsymbol{r}) = 0 \implies \phi(\boldsymbol{r}_t) = c$. A gradient normalization term was added to reduce the likelihood of these results, and for numeric stability the cosine distance between $\dot{\boldsymbol{r}}_t$ and $\bigtriangledown \phi(\boldsymbol{r}_t)$ replaced the dot product of the two terms:

$$\min_\phi \sum_{t=1}^{T} \left(\left| \frac{\dot{\boldsymbol{r}}_t \cdot \bigtriangledown \phi \left(\boldsymbol{r}_t \right)}{\| f_t \|_2^2 * \| \bigtriangledown \phi(\boldsymbol{r}_t) \|_2^2} \right| + \left(1 - \| \bigtriangledown \phi \left(\boldsymbol{r}_t \right) \|_2^2 \right)^2 \right)$$

This objective was trained by drawing 4000 samples uniformly from trajectories of each planet resulting in a mini-batch of $n = 32000$ samples. Multiple fully connected base networks were trained with 1-3 layers and 4-256 hidden units per layer. Results were sable, with a small single layer network easily minimizing the objective. When evaluating $\phi(\boldsymbol{r})$ however, the value would drift by a scalar factor resulting in saturation or vanishing $\phi(\boldsymbol{r})$ thus the scale of $\phi(\boldsymbol{r})$ was fixed by adding a second normalization term, $(0.5 - \frac{1}{n} \sum \phi(\boldsymbol{r}))^2$.

This gave a strong distinction between the value of ϕ for each of the planets at the beginning of training, however the constant value of $\phi(\boldsymbol{r})$ along each planet's trajectories converged to the fixed scale, 0.5, during training. In addition the gradient normalization term, $(1 - \| \bigtriangledown \phi \left(\boldsymbol{r}_t \right) \|_2^2)^2$ accounted for a large portion of the final loss leading to a relaxation of this constraint to a modified hinge loss with zero cost for $1 \leq \| \bigtriangledown \phi \left(\boldsymbol{r}_t \right) \| \leq 2$

$$\min_\phi \sum_{n=1}^{N} \left(\left| \frac{\dot{\boldsymbol{r}}_t \cdot \bigtriangledown \phi \left(\boldsymbol{r}_n \right)}{\| f_n \|_2^2 * \| \bigtriangledown \phi(\boldsymbol{r}_n) \|_2^2} \right| + \max \left(\| \bigtriangledown \phi \left(\boldsymbol{r}_n \right) \|_2^2 - 2, 0 \right) + \max \left(1 - \| \bigtriangledown \phi \left(\boldsymbol{r}_n \right) \|_2^2, 0 \right) \right)$$

This objective, while still minimizing the dot-product, $\dot{\boldsymbol{r}}_t \cdot \bigtriangledown \phi \left(\boldsymbol{r}_n \right)$ resulted in stable $\phi(\boldsymbol{r})$ unique to each planet's trajectory that remained separable during training. With large networks (three fully connected layers with 50+ units) these planetary constants would still converge, however for single layer networks (50, 12, 4 hidden units tested) $\phi()$ would remain reasonably stable conditioned on each planet and the variance of a mini-batch was not decreasing as it did with larger networks.

Despite this the constants learned show no obvious relation to the energy function of the orbits they encode. Some issues that I will address in the coming weeks are the dis-continuity between sampled points in trajectories, exploring a smoothness function over $\phi(\boldsymbol{r}))$ and evaluating $\phi(\boldsymbol{r}))$ over the entire domain throughout training.