Andrei Ivanov, PhD
05x.andrey@gmail.com

A Brief Introduction to Lie Transform and Polynomial Neural Networks.

Combining modern deep learning tools with a more classical theory of differential equation allows us integrating the strength of both neural networks and traditional model-based approaches for physical phenomena modeling.

Mathematical Background

Let's consider a system of nonlinear differential equation $\frac{d}{dt}X=F(t,X)$, where variable t is an independent variable, $X\in R^n$ is state vector. The dynamics of vector X can be presented in form of Lie transform [1]

$$M(t|t_0)=Texp\int_{t_0}^{t} L_F(\tau)\,d\tau,$$

where L_F is Lie operator associated with vector function F. Transformation M is presented in form of time-ordered exponential operator and can be identified with the dynamical system itself.

On the assumption that the function F allows its expansion in Taylor series $F=\sum_{k=0}^{\infty} P^k X^{[k]}$, the required solution of the equation in its convergence region can be presented [2] in form of series

$$X(t)=M\circ X_0=\sum_{k=0}^{\infty} W^k X_0^{[k]}.$$

Here $X^{[k]}$ means k -th Kroneker power of vector X. Map M implements polynomial transformation and can be presented as polynomial neural network

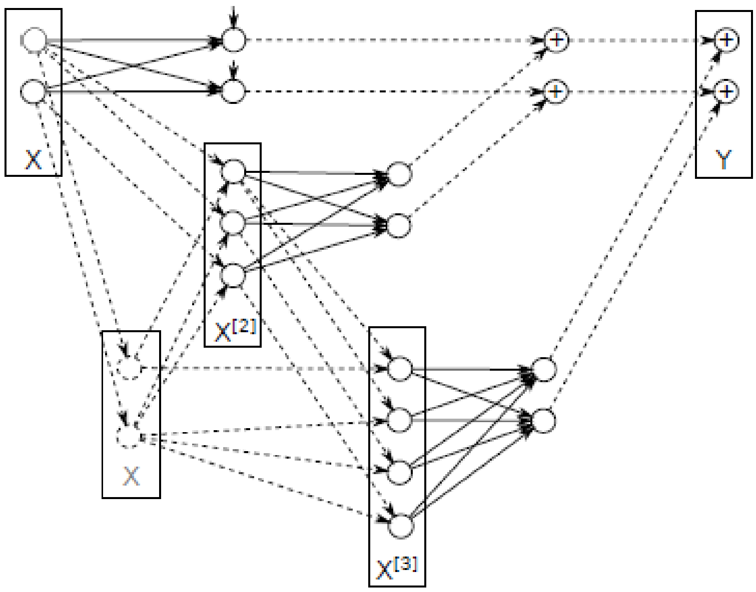

Figure 1: Neural network representation of matrix Lie transform up to the 3rd order of nonlinearities.

In the traditional model-based approach with known function F (right-hand part of the system of differential equations), one can precisely calculate map M up to the necessary order of nonlinearity [2]. In the data-driven approach (machine learning) with unknown function F, one can fit measured vectors $X(t)$ and find out weight matrices W^k that describe the dynamics of the process. This data-driven approach is important for real system identification and optimization.

Example 1: Learning Physical Models

The classical neural networks have no close connection with the mathematical theory of differential equation and physical modeling. Since current deep learning approaches can only learn specific situation, while Lie based neural networks can learn dynamical patterns from data.

As an example let's consider simple predator-prey model (nonlinear system).

$$\frac{d}{dt}x = xy - 2x, \quad \frac{d}{dt}y = y - xy,$$

In the model system, the predators thrive when there are plentiful preys but, ultimately, outstrip their food supply and decline. As the predator population is low the prey population will increase again. These dynamics continue in a cycle of growth and decline.

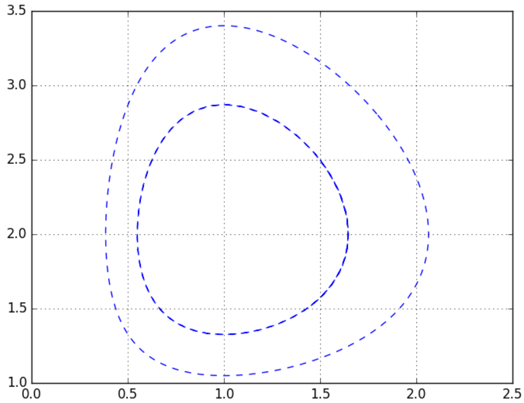

Figure 2: Phase space of predator-prey model.

Let's try to restore system dynamics using knowledge about single situation.

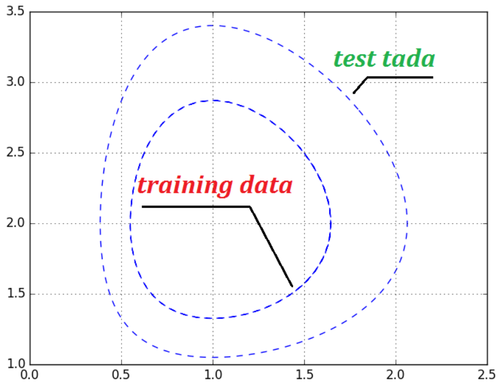

Figure 3: Training and test data in phase space

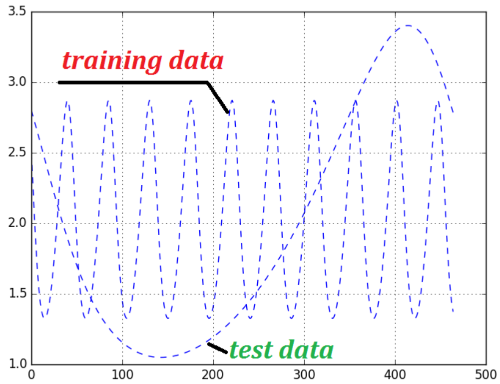

Figure 4: Training and test data in time-space space

Note that training data and test data are generated by the same dynamical processes but starting from different initial points. Figure 5 and 6 demonstrate generalization property and ability to learn system behavior from small data.

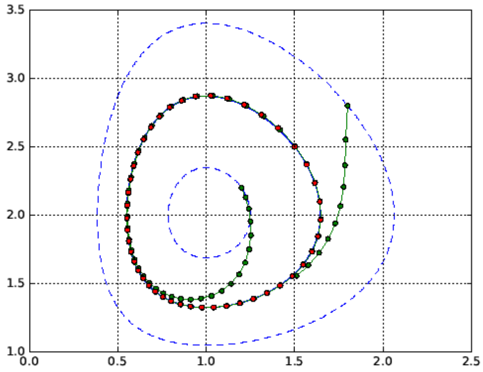

Figure 5: Classical neural networks (MLP, LTSM)

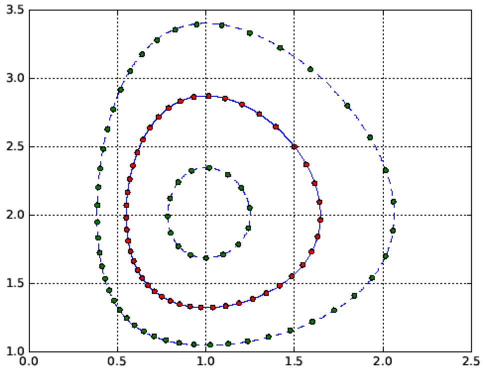

Figure 6: Lie transform based neural network

Classical neural networks (Fig. 5) just learn the current behavior of training data set and generate the wrong prediction in test data. They need more data to learn oscillation pattern. While Lie transform based neural network (Fig. 6) can generalize data dynamics and recognize oscillation pattern.

Table 1. Learning physical models

Classical neural networks	**Lie transform based neural network**
learning the *specific situation*	learning the specific *dynamical pattern*

Example 2: Simulation and Control of Charged Particle Accelerators

Charged particle accelerator consists of the number of physical equipment (e.g. quadrupoles, sextupoles, bending magnets and others, see Fig. 7). The particles move into it affecting forces of electromagnetic fields. Design of accelerators and investigation of charged particles nonlinear dynamics require an accurate computer model of such complicated system.

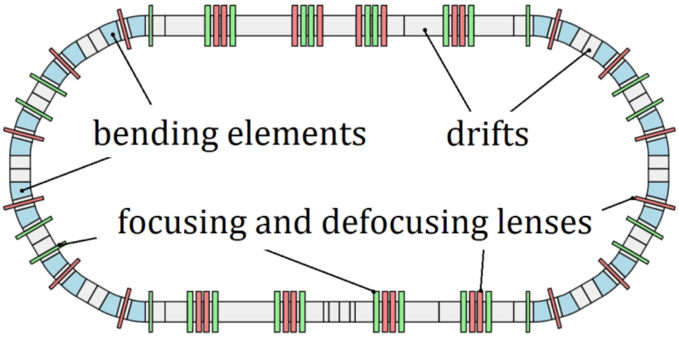

Figure 7: Schematic representation of charged particle accelerator

Each of the physical equipment can be described by the nonlinear differential equationa that has nonlinear form. For instance, for radial motion the equation looks like

$$x'' = \frac{q}{m_0 v}\left(\left(1-\frac{v^2}{c^2}\right)\left(1+x'^2+y'^2\right)\right)^{\frac{1}{2}}$$

$$\left(\frac{\left(1+x'^2+y'^2\right)^{\frac{1}{2}}\left(E_x - x' E_z\right)}{v} - \left(1+x'^2\right)B_y + y'\left(x' B_x + B_z\right)\right),$$

where control parameters (electromagnetic fields) and particle state vector are incorporated. Commonly the state vector may have 6 dimensions for coordinates, momentums, particle energy and additional 3 ones for spin dynamics.

The traditional approach for examining the dynamics of the particles is numerical solving of these equations by step-by-step integration methods (e.g. Runge – Kutta methods). For problems where long-term evolution modeling is required such approaches are not suitable due to the performance limitation.

Instead of solving differential equation directly one can estimate matrix Lie map for each control element and build a neural network (see Fig. 8) of whole accelerator ring [5, 6].

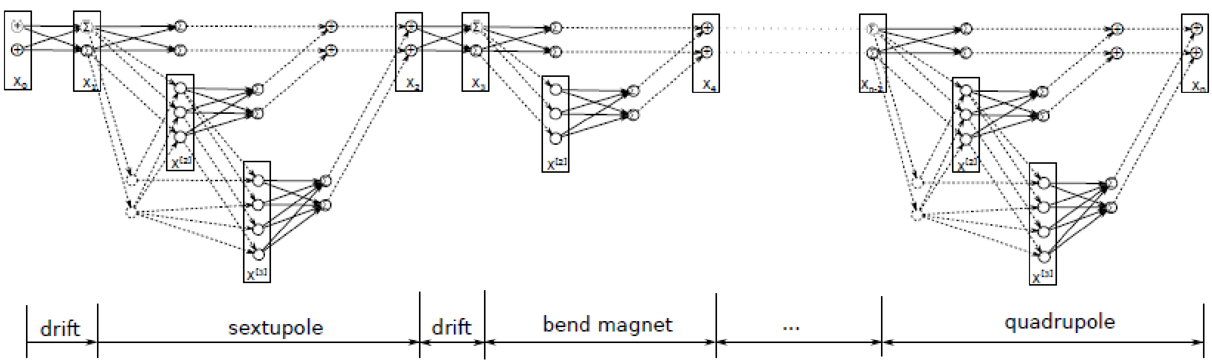

Figure 8: Schematic representation of charged particle accelerator

The process of simulation of nonlinear charged particle dynamics represents an evolution of initial state vectors along the neural network. The neural network architecture is calculated at once and then is used for particle dynamics simulation.

Table 2. Simulation of nonlinear systems

	Classical numerical method	**Lie transform based NN**
operations	implicit equations	sum and multiplication
approach	step-by-step calculation, scheme recalculation for each new initial point	constant map

In EDM search project [6] we have achieved 1500 times performance increasing with certain accuracy compare to traditional step-by-step methods.

References

[1] Dragt A. Lie Methods for Nonlinear Dynamics with Applications to Accelerator Physics. 2011.

[2] Andrianov S. A role of symbolic computations in beam physics. Computer Algebra in Scientific Computing, Lecture Notes in Computer Science, 6244/2010:19-30, 2010.

[3] Ivanov A., Kuznetcov P. Dynamical Systems Identification Based on Nonlinear Matrix Lie presentation // The XIII Conference on High-Performance Computing on Cluster Systems, Nizhnii Novgorod, Russia, 2013.

[4] Ivanov A., Andrianov S., Senichev Yu. . Simulation of spin-orbit dynamics in storage rings Dynamical // Journal of Physics: Conference Series, 747(1), 2016.

[5] Ivanov A. and Senichev Yu. Matrix integration of ODEs for spin-orbit dynamics simulation. Proc. of the Particle Accelerator Conference, page 400-402, 2014.

[6] Senichev Yu., Lehrach A., Maier R., Zyuzin D., Berz M., Makino K., Andrianov S., Ivanov. A. Storage ring EDM simulation: methods and results. Proceedings of ICAP2012, Rostock, Germany, page 99103, 2012.