ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 303-308

Composition methods for the integration of dynamical neural networks

Yves Moreau, Joos Vandewalle

Katholieke Universiteit Leuven, Elektrotechniek-ESAT,
Kardinaal Mercierlaan 94, B-3001 Heverlee, Belgium

Abstract. We apply the symmetric composition method for the integration of ordinary differential equations to dynamical neural networks. In this method, we split the vector field, which is parameterized by a neural network, into the contribution of each of its neurons. We then solve the elementary differential equation associated to each neuron separately, and recombine these contributions in a sequence of compositions. This gives rise to simple integration rules for dynamical neural networks, which we present for dynamical single-hidden-layer perceptrons.

1. Introduction

For the time-integration of continuous-time recurrent neural network, researchers most often use some standard scheme, such as the Euler method or the Runge-Kutta method. But physicists [2, 5] have recently introduced a completely different class of methods for the solution of ordinary differential equations (ODEs): the composition methods. The spirit of these methods is that if the vector field of the differential equation is the superposition of some elementary vector fields, we can approximate its solution by composing the flows of the elementary contributions. Composition methods are particularly useful for numerically integrating ODEs when the equations have some special structure, of which we can take advantage [4].

We will first present the class of dynamical neural networks that we will consider, together with the necessary elements of dynamical system theory and Lie algebra theory. We will then introduce the composition methods, and apply them to dynamical single-hidden-layer perceptrons. This will allow us to formulate simple integration rules for these networks.

This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the framework of a Concerted Action Project of the Flemish Community, entitled *Model-based Information Processing Systems* GOA-MIPS, and within the framework of the Belgian program on inter-university attraction poles (IUAP-17) initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture. The scientific responsibility rests with its authors. Yves Moreau is a research assistant of the N.F.W.O. (Belgian National Fund for Scientific Research).

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 303-308

2. Dynamical neural networks

By the term dynamical neural network, we refer here to a set of ordinary differential equations (ODE) $\dot{x}(t) = A(x(t))$ defined on $\mathbb{R}^n$, where the vector field A is parameterized by an n-input n-output feed-forward neural network. The neural networks we consider here are single-hidden-layer perceptrons (MLP), but we can propose similar integration rules for other single-hidden-layer feed-forward networks.

We write the ODE as $\dot{x} = A(x)$ to follow the conventions of Lie algebra theory. We can write its solution, as a function of the initial condition x_0, in two forms: as a flow $x(t) = \Phi(x_0, t)$ (as is standard in the dynamical system literature), or as an exponential solution $x(t) = e^{t.A}(x_0)$ (as in the Lie algebra literature [1]). The latter notation should read: "$x(t)$ is the image after time t of the initial condition x_0 under the flow of $\dot{x} = A(x)$". It is an extension of the case where the vector field is linear in space and the exponential of the matrix is the solution. But we to stress that the exponential $e^{t.A}$ is a nonlinear map from the state-space into itself.

3. Lie algebra theory

Lie algebra theory is an important tool in physics [1] and an essential part of nonlinear system theory [3]. It provides here the framework for the presentation of composition methods for ODEs. The Lie algebra we consider here the Lie algebra is the vector space of all smooth vector fields, where we define a supplementary operation: the Lie bracket of two vector fields, which is again a vector field. Bracketing is a bilinear, anti-symmetric operation, which also satisfies the Jacobi identity [1]. Recalling that we take the product of exponentials to denote the composition of these maps, we can define the vector field $[A, B]$ as the Lie bracket of the vector fields A, B:

$$[A, B] = \left.\frac{\partial^2}{\partial s.\partial t}\right|_{t=s=0} e^{-s.B}.e^{-t.A}.e^{s.B}.e^{t.A} .$$

As we define our vector fields on $\mathbb{R}^n$, we can further express the bracket as

$$[A, B]_i = \sum_{i=1}^{n} \left(B_i.\frac{\partial A_j}{\partial x_i} - A_i.\frac{\partial B_j}{\partial x_i} \right) .$$

The last mathematical tool we need is the Baker-Campbell-Hausdorff (BCH). This formula gives an expansion for the product of two exponentials of elements of the Lie algebra [1]:

$$e^{tA}.e^{tB} = e^{t(A+B) + \frac{t^2}{2}[A,B] + \frac{t^3}{12}([A,[A,B]] + [B,[B,A]]) + \cdots} \quad (1)$$

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 303-308

4. Integration of ordinary differential equations by compositions

We now look at how to solve ordinary differential equations using compositions. Suppose we want to solve the ODE $\dot{x}(t) = X(x(t))$, for a time-step of t. The problem becomes that of building an approximation to $e^{t.X}$ as we have that $x(t) = e^{t.X}(x_0)$. Suppose, in a first step, that the vector field X is the sum of two vector fields: $X = A + B$, where you can integrate A and B analytically or much more easily than X. Then we can use the BCH formula to produce a first-order approximation to the exponential map:

$$\textbf{BCH:} \quad e^{t.X} = e^{t.A}.e^{t.B} + o(t^2). \tag{2}$$

You can check this relation by multiplying the left- and right-hand sides of Equation 1 by $e^{t.X} (= e^{t.(A+B)})$, and expanding it using the BCH formula itself (1). The relation of first-order approximation (2) between the solution of A and B, and the solution of X is the essence of the method since it shows that we can approximate an exponential map (that is the mapping arising from the solution of an ODE) by composing simpler maps (Fig.1). By using the BCH formula to

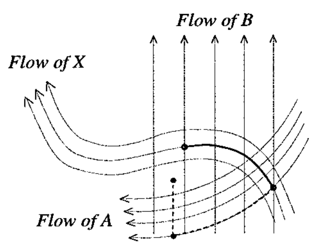

Figure 1: $e^{t.X}(x_0) \approx e^{t.A}.e^{t.B}(x_0)$.

eliminate higher-order terms as we did for the first-order approximation, but on the composition of three terms, we can show that the following symmetric leapfrog scheme is second order:

$$\begin{aligned} \textbf{Leapfrog:} \quad e^{t.X} &= e^{\frac{t}{2}.A}.e^{t.B}.e^{\frac{t}{2}.A} + o(t^3), \\ &= S(t) + o(t^3). \end{aligned}$$

Using this leapfrog scheme as a basis element, we can build a fourth-order scheme:

$$\begin{aligned} \textbf{Fourth order:} \quad e^{t.X} &= S(ct).S(dt).S(ct) + o(t^5), \\ &= SS(t) + o(t^5), \end{aligned} \tag{3}$$

with $c = -2^{1/3}/(2-2^{1/3})$ and $d = 1/(2-2^{1/3})$. Repeating the leapfrog strategy, Yoshida [5] showed that it is possible to produce an approximation to $e^{t.X}$ up

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 303-308

to any order:

$$\textbf{Arbitrary order}: \quad \exists k, \exists w_1, v_1, \dots, w_k, v_k : \\ e^{t.X} = e^{w_1.t.A}.e^{v_1.t.B} \dots e^{w_k.t.A}.e^{v_k.t.B} + o(t^{p+1}) \tag{4}$$

Forest and Ruth [2] also showed that approximations can be built for more than two vector fields:

$$X = \sum_{i=1}^{m} A_i \quad \Rightarrow \quad \exists k, \exists w_{ij} : e^{t.X} = \prod_{j=1}^{k} \prod_{i=1}^{m} e^{w_{ij}.t.A_i} + o(t^{p+1})\,.$$

5. Integration of dynamical single-hidden-layer perceptrons

If we decide to parameterize the vector field of our system by a single-hidden-layer perceptron, we can derive a composition method to integrate the differential equation. The differential equation is $\dot{x}(t) = X(x(t))$, where the vector field is of the following form:

$$X(x) = \sum_{j=1}^{n} \vec{c}_j.\sigma(\vec{b}_j.x + b_{j_0}) = \sum_{j=1}^{n} A^{\vec{c}_j,\vec{b}_j}(x), \tag{5}$$

where $\sigma(x) = \tanh(x)$ and $\vec{c}_j, \vec{b}_j \in \mathbb{R}^{d+1}, j = 1, \dots, n$.
To find a composition method, we first have to solve the differential equation $\dot{x} = A^{c,b}(x)$ associated to a single neuron. To do this, we first solve the one-dimensional problem. Then, we use the solution of the one-dimensional case to derive the solution for the multi-dimensional case. Finally, we derive a composition method where each transformation is the solution of a sigmoïdal ODE.

6. One-dimensional sigmoïdal ODE

We want to solve $\dot{x} = \sigma(x), x \in \mathbb{R}$ with the hyperbolic tangent for the activation function σ. Using the change of variables $y = \tanh x$, solving by separation of variables, and substituting back $x = \text{atanh } y$, we find the solution for positive initial conditions $x_0 > 0$:

$$x(t) = \varphi(x_0, t) = \text{atanh}\Big((1 + \frac{e^{-2t}}{\sinh^2 x_0})^{-1/2}\Big).$$

For $x_0 = 0$, we have $x(t) = 0, \forall t$. For $x_0 < 0$, we can directly use the symmetry of the vector field $\tanh x = -\tanh(-x)$. We thus get

$$x(t) = \Phi(x_0, t) = \begin{cases} \varphi(x_0, t), & \text{if } x_0 > 0 \\ 0, & \text{if } x_0 = 0 \\ -\varphi(-x_0, t), & \text{if } x_0 < 0 \end{cases}$$

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 303-308

7. Multi-dimensional sigmoïdal ODE

We can now use the solution of the one-dimensional case to explicitly integrate the multidimensional system $\dot{x} = A^{\vec{c},\vec{b}}(x) = \vec{c}.\sigma(\vec{b}^T x + b_0)$ for any value of the parameters $\vec{c}, \vec{b}$. The main characteristics of this system is that the velocity at all points is parallel to $\vec{c}$, and that the intensity of the field is constant on any $n-1$ dimensional hyperplane $\vec{b}^T x + b_0 =$ constant. Therefore, the trajectories are straight lines parallel to $\vec{c}$, see Figure 2. To solve this problem, we use a

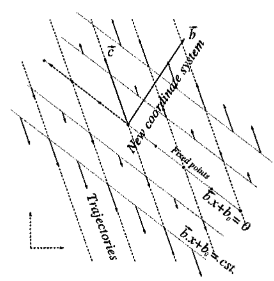

Figure 2: Flow of a multi-dimensional sigmoïd, and change of coordinates.

translation of the origin and a change of coordinates. Let x^* be a solution of $\vec{b}x + b_0 = 0$. Define a translation of the origin $w = x - x^*$, $x = w + x^*$. The system rewrites as $\dot{w} = \vec{c}.\sigma(\vec{b}^T w)$. Now, we look for a change of coordinates $z = P^{-1}w$, $w = Pz$ such that it transforms $\vec{b}$ into the first unit vector and the new coordinate system is orthogonal. Such a change of coordinates can be found using a Gram-Schmidt orthogonalization procedure. If we define $\vec{d} = P.\vec{c}$, the differential equation becomes $\dot{z} = \vec{d}.\sigma(z_1)$. Since the solution for z_1 is independent from the other coordinates, we have

$$z_1(t) = \Phi(z_1^0, d_1.t). \tag{6}$$

Hence, since the trajectories are all parallel to $\vec{d}$, we have

$$z(t) = z^0 + \frac{z_1(t) - z_1^0}{d_1}.\vec{d}. \tag{7}$$

Now, reversing the changes of coordinates, we have

$$x(t) = Pz(t) + x^*. \tag{8}$$

Equations 6, 7, 8, and a change of coordinates back to the old coordinates allow us to compute the solution $x(t) = e^{t.A^{\vec{c},\vec{b}}}$ to $\dot{x} = \vec{c}.\sigma(\vec{b}^T x + b_0)$:

$$\begin{aligned}
d_1 &= \vec{b}^T.\vec{c} \\
z_1^0 &= \vec{b}^T x^0 + b_0 \\
x(t) &= \begin{cases} x^0 + \frac{\Phi(z_1^0, d_1.t) - z_1^0}{d_1}.\vec{c} & \text{if } d_1 \neq 0 \\ x^0 + (t.\sigma(z_1^0)).\vec{c} & \text{if } d_1 = 0 \end{cases}
\end{aligned}$$

8. Composition method for single-hidden-layer perceptron

From the solution of the multidimensional sigmoïdal ODE, we can develop a composition method for the solution of the dynamical single-hidden-layer perceptron. A second-order composition methods would then be:

$$x(t) = S(t)(x_0) + o(t^3) = \prod_{j=1}^{n} e^{\frac{t}{2}A^{\vec{c}_j,\vec{b}_j}} \prod_{k=n}^{1} e^{\frac{t}{2}A^{\vec{c}_k,\vec{b}_k}}(x_0) + o(t^3)\,. \qquad (9)$$

A fourth-order method would be defined as in Equation 3.
The main interest of the composition method is that it takes advantage of the fact that we can compute the solution of the sigmoïdal ODE very efficiently.

9. Conclusions and future work

We have presented new integration rules for the dynamical single-hidden-layer perceptron. To this end, we have used composition methods derived from Lie algebra theory through the use of the Baker-Campbell-Hausdorff formula. The simplicity of these integration rule makes them good candidates for efficient software implementation. This will be the subject of subsequent research.

References

[1] V. Arnold. *Mathematical methods of classical mechanics.* Springer-Verlag, New York, NY, 1989.

[2] E. Forest and R. Ruth. Fourth-order symplectic integration. *Physica D*, 43:105–117, 1990.

[3] A. Isidori. *Nonlinear control theory.* Springer-Verlag, New York, NY, 1989.

[4] R. I. McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM J. Sci. Comput.*, 16(1):151–168, 1995.

[5] H. Yoshida. Construction of higher order symplectic integrators. *Phys. Lett. A.*, 150:262–269, 1990.