

Integral Equation Modelling and Deep Learning

François Lemaire and Louis Roussel

Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRISAL, F-59000 Lille, France

{francois.lemaire,louis.roussel}@univ-lille.fr

Abstract

We present some preliminary results on the integration of integro-differential equations using Deep Learning techniques.

1 Integral equation modelling

Many models (such as the SIR¹ epidemiology model) consist of non-linear differential equations, involving unknown numerical parameters. Recent work [3, 6, 4] investigates the treatment of integro-differential equations i.e. equations involving both derivatives and integration operators. Considering models with integro-differential equations is motivated by the following reason: on some examples, the introduction of integral equations increases the expressiveness of the models, improves the estimation of parameter values from error-prone measurements, and reduces the size of the intermediate equations.

Reducing the order of derivation of a differential equation can sometimes be achieved by integrating it. Algorithm [4, Integrate] was designed for that purpose (see the vertical arrow of Figure 1). However, successfully integrating integro-differential equations is a complex problem. Unfortunately, there are still plenty of differential equations for which algorithm [4, Integrate] does not apply. For example, computing an integrating factor might be required.

In this abstract, we use a hybrid approach between computer algebra and deep learning, and present some preliminary results.

2 Integration of equations with deep learning [12]

In 2019, Lample and Charton published an article [12] that has aroused some interest in the calculus community, and has also inspired a number of articles [1, 13, 11, 14]. Through learning techniques used for the translation of texts from one language to another, the authors of [12] succeed in calculating primitives

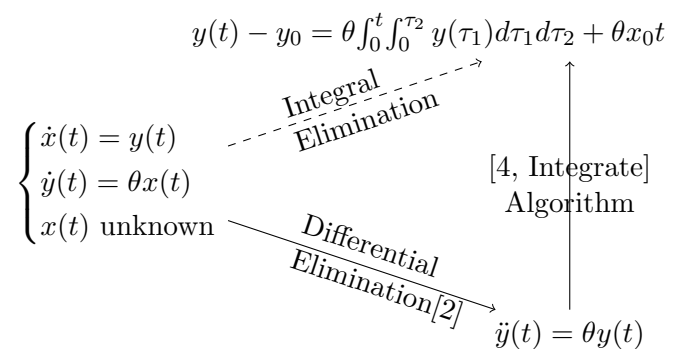


Figure 1: **Modelling, elimination and [4, Integrate] algorithm.** From a differential system, we can obtain an input/output equation [10] which only involves the known function $y(t)$, by using differential elimination techniques [5]. From this equation, we can use the [4, Integrate] algorithm to compute an integral equation. Using numerical methods (e.g. least squares), it is possible to evaluate the value of θ from any of the two input/output equations.

¹Susceptible, Infectious, Recovered

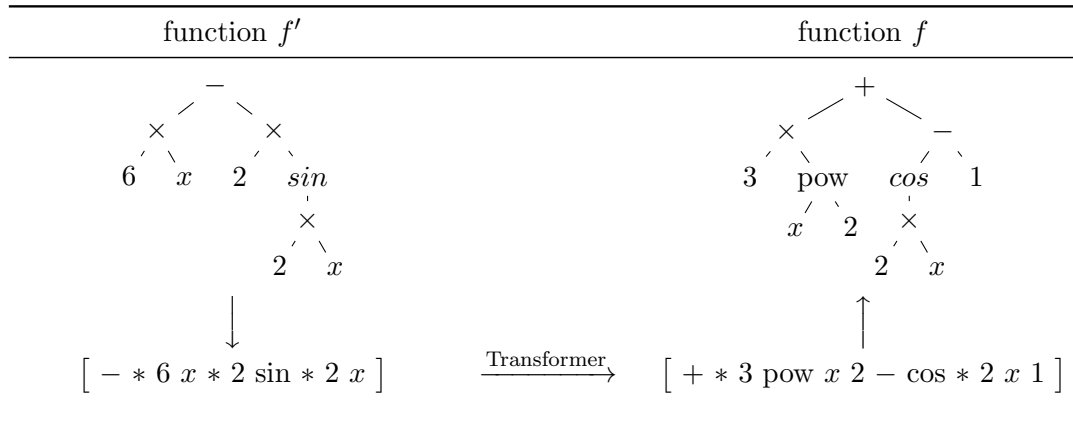


Figure 2: **Usage of the Transformer in [12].** The model takes an expression in Polish notation as input. Here, the model generates each word/token of the output one by one and stops when the end of line token is generated.

of mathematical expressions, and solutions of differential equations. Here, we are talking about formal expressions given in the form of exact mathematical formulas, and not numerical calculations.

The model used is called Transformer and is widely used in natural language processing (e.g. for text translation) since its release in 2017 [14]. It takes a list of words as input and returns the list of translated words as output. Internally, a transformer involves several classical multilayer perceptrons (MLP) and some attention mechanisms [14] organized to form a sequential process.

For their experiments, Lample and Charton have created a data generator to generate the train and test datasets, composed of pairs (f', f) . To obtain a pair (f', f) , they generate a random f (by generating a random tree) and compute the derivative f' (using SymPy). An example of how the transformer works with those kinds of data is given in Figure 2.

3 Adaptation of [12] to integro-differential equations

The results obtained by Lample and Charton in their various works show the effectiveness of machine learning to solve mathematical problems (integration and solving differential equations [12], properties of differential systems [8], operations on matrices [7]).

Our objective is therefore to adapt the techniques used in [12] for integral equation modelling. As we said previously, one way of doing so is to directly integrate the differential equation obtained after differential elimination (Figure 1). Hence, in our work we want to study the capacity of a network to integrate integro-differential functions. Nevertheless, the long-term goal is to successfully perform integral elimination of a system of integro-differential equations.

Previously, we have seen how Lample and Charton generate their data. To adapt their method to our problem, we modified the implementation of [12] (based on PyTorch) to fit our needs. For this, it is necessary to modify the operations and operators that we want to generate in the trees. We then created the unary operators ∂ and \int associated to the differentiation and integration operators. Another important point is the addition of time-dependent functions to the leaves of the tree ($x(t)$ for example). The trees we manipulate are then similar to those of [12], see Figure 3.

Another modification has been made in the training and evaluation steps where we need to check the correctness of an integral. Lample and Charton verify that a prediction \hat{f} is correct by checking that $\partial \hat{f} - f'$ is zero. In our case, a difficulty arises since for any expression F , $\int F$ is a primitive of F . To prevent the network from just learning to add a \int symbol, we adopt the check of Figure 4.

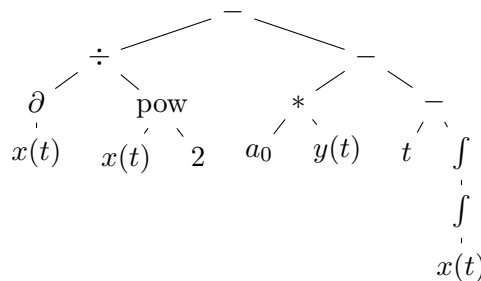


Figure 3: **Tree of a generated integro-differential equation:** $\frac{\dot{x}(t)}{x(t)^2} - a_0 y(t) - t - \int \int x(t)$.

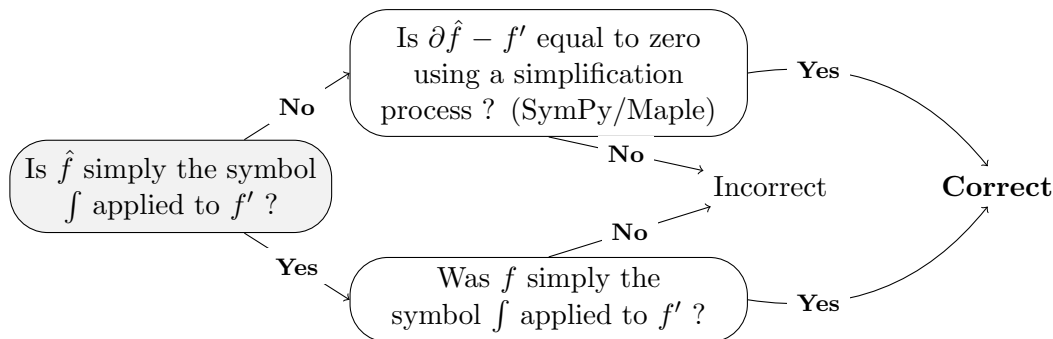


Figure 4: **Test for checking if a predicted integral \hat{f} is correct.**

4 Experiment and results

For our experiment, we generated equations that include the following operations/operators: $+$, $-$, $*$, \div , ∂ , \int . We used two functions: $x(t)$, $y(t)$ and a parameter, a_0 . The functions generated are then differentiated in order to obtain pairs (f', f) . The network is then trained on a training dataset containing more than 600000 pairs (f', f) .

At the end of the training phase, we calculate the accuracy of our model using a test dataset of one thousand equations: the accuracy is defined by the number of correct integrals (see Figure 4) predicted by the model divided by the number of equations of the test dataset.

Table 1 compares the accuracy of our model on our test dataset with the accuracy of the Maple software on the same test dataset. The check of Figure 4 is done using SymPy for computing our model accuracy, and using Maple for computing the Maple one. The Maple accuracy could be enhanced by suggesting Maple to perform integration by parts (IBP) on some parts of the equations. However, this would imply to recognise such parts to integrate by parts, and this process is not straightforward to program. For this reason, the Maple accuracy is obtained using the `int` subroutine. Some examples of functions that Maple fails to integrate while the model succeeds are given in Table 2.

5 Discussion

Our results inherits the same problems as those of Lample and Charton [12], which are listed in [9]. We discuss some of them.

The trained model is specialised on integrating large equations that have a small integral, due to the generation method. As it is the fastest way of generating large datasets, we haven't yet looked to the other generation methods presented in [12].

Another problem occurs during the generation of the training dataset. In order to help the model, the generated pairs (f', f) should be in a quite "simple" form. For example, we should avoid including terms such as $(\dot{x}(t), \frac{x(t)}{1+x(t)-x(t)})$ which are correct mathematically but not very informative for the model. In our process, the previous case does not happen since we use SymPy on generated functions before differentiating it. However, a general simplification would be quite difficult to write, and the SymPy simplification is clearly not powerful enough to correctly handle integro-differential equations. Writing a better simplification process should be investigated.

6 Conclusion

To summarize, we achieve a quite good accuracy on the integration of a specific set of integro-differential equations. Different data generation methods could be investigated (such as [12, IBP and FWD methods]). Our next goal will be focused on the integral elimination of differential system with deep learning.

References

- [1] AGARWAL, V., ADITYA, S., AND GOYAL, N. Analyzing the nuances of transformers' polynomial simplification abilities. *CoRR abs/2104.14095* (2021).
- [2] BOULIER, F. The blad libraries., 2004.
- [3] BOULIER, F., CASTEL, H., CORSON, N., LANZA, V., LEMAIRE, F., POTEAUX, A., QUADRAT, A., AND VERDIÈRE, N. Symbolic-Numeric Methods for Nonlinear Integro-Differential Modeling. working paper or preprint, Apr. 2018.
- [4] BOULIER, F., LALLEMAND, J., LEMAIRE, F., REGENSBURGER, G., AND ROSENKRANZ, M. Additive Normal Forms and Integration of Differential Fractions. *Journal of Symbolic Computation* (2016).
- [5] BOULIER, F., LAZARD, D., OLLIVIER, F., AND PETITOT, M. Computing representations for radicals of finitely generated differential ideals. *Applicable Algebra in Engineering, Communication and Computing* 20, 1 (Mar 2009), 73.
- [6] BOULIER, F., LEMAIRE, F., ROSENKRANZ, M., USHIROBIRA, R., AND VERDIÈRE, N. On Symbolic Approaches to Integro-Differential Equations. In *Algebraic and Symbolic Computation Methods in Dynamical Systems*, A. Quadrat and E. Zerz, Eds., vol. 9 of *Advances in Delays and Dynamics*. Springer, 2020, pp. 161–182.
- [7] CHARTON, F. Linear algebra with transformers, 2021.

Table 1: **Accuracy of the model versus the accuracy of Maple.**

	Accuracy	Comments
Maple	75.8%	We use $\text{int}(f, t)$. Could be enhanced with integration by parts, but it is difficult to automate.
Transformer	96%	Our adapted version of [12].

Table 2: **Functions that the model can integrate, but not Maple.**

Equation f'	Solution f
$\frac{1}{f x} - \frac{t x}{(f x)^2}$	$\frac{t}{f x}$
$\frac{(-a_0 x - 1) \int x y + \dot{y}(t + a_0 \int x) + a_0 x y \int x + x y t - a_0 \dot{y} x - \dot{y}}{(t + a_0 \int x)^2}$	$\frac{\dot{y} + \int x y}{t + a_0 \int x}$

- [8] CHARTON, F., HAYAT, A., AND LAMPLE, G. Learning advanced mathematical computations from examples. In *International Conference on Learning Representations* (2021).
- [9] DAVIS, E. The use of deep learning for symbolic integration: A review of (lample and charton, 2019). Tech. rep., New York University, 2019.
- [10] FLIESS, M. Automatique et corps différentiels. *Forum Mathematicum - FORUM MATH 1* (01 1989), 227–238.
- [11] KUBOTA, H., TOKUOKA, Y., YAMADA, T. G., AND FUNAHASHI, A. Symbolic integration by integrating learning models with different strengths and weaknesses. *CoRR abs/2103.05497* (2021).
- [12] LAMPLE, G., AND CHARTON, F. Deep learning for symbolic mathematics. In *The Ninth International Conference on Learning Representations (ICLR)* (Virtual only, Mai, 6–9 2019), p. 24.
- [13] MISTRY, B., FARRAHI, K., AND HARE, J. S. A primer for neural arithmetic logic modules. *CoRR abs/2101.09530* (2021).
- [14] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need. *CoRR abs/1706.03762* (2017).