Neural Assimilation

Rossella Arcucci(✉), Lamya Moutiq, and Yi-Ke Guo

Data Science Institute, Imperial College London, London, UK
r.arcucci@imperial.ac.uk

Abstract. We introduce a new neural network for Data Assimilation (DA). DA is the approximation of the true state of some physical system at a given time obtained combining time-distributed observations with a dynamic model in an optimal way. The typical assimilation scheme is made up of two major steps: a *prediction* and a *correction* of the prediction by including information provided by observed data. This is the so called *prediction-correction* cycle. Classical methods for DA include Kalman filter (KF). KF can provide a rich information structure about the solution but it is often complex and time-consuming. In operational forecasting there is insufficient time to restart a run from the beginning with new data. Therefore, data assimilation should enable real-time utilization of data to improve predictions. This mandates the choice of an efficient data assimilation algorithm. Due to this necessity, we introduce, in this paper, the Neural Assimilation (NA), a coupled neural network made of two Recurrent Neural Networks trained on forecasting data and observed data respectively. We prove that the solution of NA is the same of KF. As NA is trained on both forecasting and observed data, after the phase of training NA is used for the *prediction* without the necessity of a correction given by the observations. This allows to avoid the *prediction-correction* cycle making the whole process very fast. Experimental results are provided and NA is tested to improve the prediction of oxygen diffusion across the Blood-Brain Barrier (BBB).

Keywords: Data Assimilation · Machine learning · Neural network

1 Introduction and Motivations

The current approach to forecasting modelling consists of simulating explicitly only the largest-scale phenomena, while taking into account the smaller-scale ones by means of "physical parameterisations". All numerical models introduce uncertainty through the selection of scales and parameters. Additionally, any computational methodology contributes to uncertainty due to discretization, finite precision and accumulation of round-off errors. Finally the ever growing size of the computational domains leads to increasing sources of uncertainties. Taking into account these uncertainties is essential for the acceptance of any numerical simulation. Numerical forecasting models often use Data Assimilation methods for the uncertainty quantification in the medium to long-term analysis.

© Springer Nature Switzerland AG 2020
V. V. Krzhizhanovskaya et al. (Eds.): ICCS 2020, LNCS 12142, pp. 155–168, 2020.
https://doi.org/10.1007/978-3-030-50433-5_13

Data Assimilation (DA) is the approximation of the true state of some physical system at a given time by combining time-distributed observations with a dynamic model in an optimal way. DA can be classically approached in two ways: as variational DA [16] and as filtering [5]. In both cases we seek an optimal solution. The most popular filtering approach for data assimilation is the Kalman Filter (KF) [15]. Statistically, KF seeks a solution with minimum variance. Variational methods seek a solution that minimizes a suitable cost function. In certain cases, the two approaches are identical and provide exactly the same solution [16]. However, the statistical approach, though often complex and time-consuming, can provide a richer information structure, i.e. an average and some characteristics of its variability (probability distribution). During the last 20 years hybrid approaches [11,18] have become very popular as they combine the two approaches into a single taking advantage of the relative rapidity and robustness of variational approaches, and at the same time, obtaining an accurate solution [2] thanks to the statistical approach. In this paper, in order to achieve the accuracy of the KF solution and reduce the execution time, we use Recurrent Neural Networks (RNN). Today the computational power of RNN is exploited for several application in different fields. Any non-linear dynamical system can be approximated to any accuracy by a Recurrent Neural Network, with no restrictions on the compactness of the state space, provided that the network has enough sigmoidal hidden units. This is what the Universal Approximation Theorem [12,20] claims. Only during the last few years, the DA community is starting to approach machine learning models to improve the efficiency of DA models. In [17], the authors combined Deep Learning and Data Assimilation to predict the production of gas from mature gas wells. They used a modified deep LSTM model as their prediction model in the EnKF framework for parameter estimation. Even if the *prediction* phase is speed up due to the introduction of Deep Learning, this only partially affects the whole *prediction-correction* cycle which is still time-consuming. In [9], the authors presented an approach for employing artificial neural networks (NNs) to emulate the local ensemble transform Kalman filter (LETKF) as a method of data assimilation. Even if the Feed Forward NN they implemented is able to emulate the DA process for the time window they fixed, when they need to assimilate observations in new time steps, it still needs the *prediction-correction* cycle and this affects the execution time which is just 90 times faster than the reference DA model. To further speed up the process, in [8] the authors combined the power of Neural Networks and High Performance Computing to assimilate meteorological data. These studies, alongside others discussed in conferences and still under publication, highlight the necessity to avoid the *prediction-correction* cycle by developing a Neural Network able to completely emulate the whole Data Assimilation process. In this context, we developed a Neural Assimilation (NA) as a Coupled Neural Network made of two RNNs. NA captures the features of a Data Assimilation process by interleaving the training of the two component RNNs on the forecasting data and the observed data. That is, the two component RNNs are trained on forecasting and observed data respectively with additional inputs provided by the interaction of these two. This NA network emulates the KF and runs much faster than the KF *prediction-correction* cycle for data assimilation. In this paper we develop the NA architecture and proved its equivalence to the KF. The equivalence between NA and KF is independent from the structure on the RNNs. In this paper we show results we obtained employing two Long short-term memory (LSTM) architectures for the two RNNs. Then we employ the NA model to a practical problem in predicting of oxygen (and drugs) diffusion across the Blood-Brain Barrier (BBB) [1] to justify its correctness and efficiency.

This paper is structured as follows. In Sect. 2 the Data Assimilation problem is described. The Neural Assimilation is introduced in Sect. 3, where we investigate the accuracy of the introduced method and we present a theorem demonstrating that the novel model is consistent with the KF result. Experimental results are provided in Sect. 4. Conclusions and future works are summarised in Sect. 5.

2 Data Assimilation

Data Assimilation (DA) is the approximation of the true state of some physical system at a given time by combining time-distributed observations $o(t)$ with a dynamic model $\dot{x} = \mathcal{M}(x, t)$ in an optimal way. DA can be classically approached in two ways: as variational DA [3] and as filtering. One of the best known tools for filtering approach is the Kalman filter (KF) [15]. We seek to estimate the state $x(t)$ of a discrete-time dynamic process that is governed by the linear difference equation

$$x(t) = M \; x(t-1) + w_t \tag{1}$$

with an observation $o(t)$:

$$o(t) = H \; x(t) + v_t \tag{2}$$

Note that M and H are discrete operators. The random vectors w_t and v_t represent the modeling and the observation errors respectively. They are assumed to be independent, white-noise processes with normal probability distributions

$$w_t \sim \mathcal{N}(0, B_t), \qquad v_t \sim \mathcal{N}(0, R_t) \tag{3}$$

where B_t and R_t are covariance matrices of the modeling and observation errors respectively. All these assumptions about unbiased and uncorrelated errors (in time and between each other) are not limiting, since extensions of the standard KF can be developed should any of these not be valid [5]. The KF problem can be summarised as follows: given a background estimate $x(t)$, of the system state at time t, what is the best analysis $z(t)$ based on the current available observation $o(t)$?

The typical assimilation scheme is made up of two major steps: a *prediction* step and a *correction* step. At time t we have the result of the previous forecast, $x(t)$ and the result of an ensemble of observations $o(t)$. Based on these two vectors, we perform an analysis that produces $z(t)$. We then use the evolution model to obtain a prediction of the state at time $t+1$. The result of the forecast at the *prediction* step is denoted with $x(t+1)$

$$x(t+1) = Mz(t), \tag{4}$$

$$B_{t+1} = M\left((1 - K_t H)B_t\right)M^T, \tag{5}$$

and becomes the background for the next *correction* time step:

$$K_{t+1} = B_{t+1}H^T(HB_{t+1}H^T + R_{t+1})^{-1}, \tag{6}$$

$$z(t+1) = x(t+1) + K_{t+1}\left(o(t+1) - Hx(t+1)\right), \tag{7}$$

We observe that, in case the observed data are defined in the same space of the state variable, the operator H_t in (2) is the identity matrix and the Eqs. (6)–(7) can be simplified becoming:

$$K_{t+1} = B_{t+1}(B_{t+1} + R_{t+1})^{-1}, \tag{8}$$

$$z(t+1) = x(t+1) + K_{t+1}\left(o(t+1) - x(t+1)\right), \tag{9}$$

Due to the high computational cost in updating the covariance matrices B_t by Eq. (5), it in operational DA, is often used to assume $B_t = B_{t+1}$ $\forall t$. This assumption leads to a model which is also called Optimal Interpolation [16].

Statistically, KF seeks a solution with minimum variance. This approach, though often complex and time-consuming, can provide a rich information structure (often richer than information provided by variational DA), such as an average and some characteristics of its variability (probability distribution). In order to maintain the accuracy of the KF solution and reduce the execution time, we introduce, in the next section, a Neural Assimilation (NA) which is a network representing KF but much faster than a KF *prediction-correction* cycle.

3 Neural Assimilation

For a fixed time window $[t_0, t_1]$ and a fixed discretization time step Δt, let $x(t)$ still denote the forecasting result at each time step $t \in [t_0, t_1]$. Let $o(t)$ denotes an observation of the state value (Fig. 1). As it does not affect the generality of our study, we are assuming here the observed data defined in the same space of the state variable, i.e. the operator H_t in (2) is the identity matrix.

Given the data sets $\{x(t)\}_{t\in[t_0,t_1]}$ and $\{o(t)\}_{t\in[t_0,t_1]}$, the Neural Assimilation (NA) is a Coupled Neural Network (for temporal processing) as shown in Fig. 2, where:

- a first forecasting network NN_F is a Recurrent Neural Network trained on forecasting data $x(t)$ with an additional input provided by a second forecasting network NN_O trained on observed data $o(t)$;
- a second forecasting network NN_O is a Recurrent Neural Network trained on observed data $o(t)$ with an additional input provided by a first forecasting network NN_F.

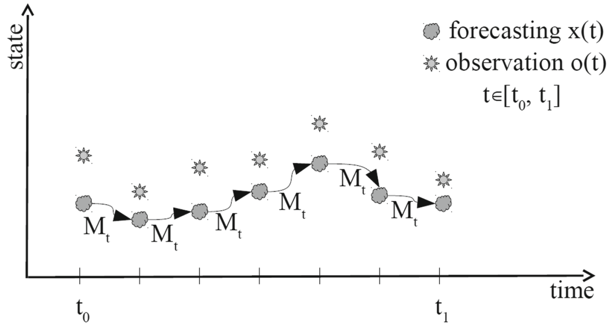


Fig. 1. Available data in the fixed time window.

A fundamental feature of each network is that it contains a feedback connection, so the activations can flow round in a loop. That enables the networks to do temporal processing and learn sequences with temporal prediction. The form of NA is a RNN with the previous set of hidden unit activations feeding back into the network along with the inputs.

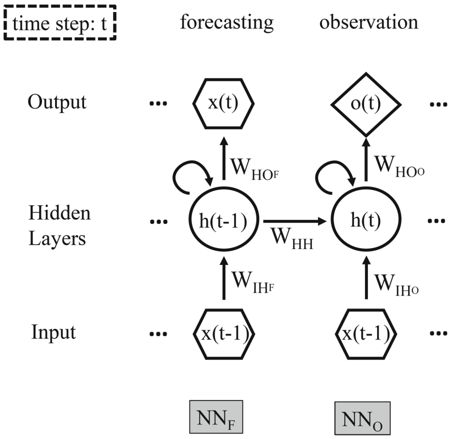


Fig. 2. Neural Assimilation

Note that the time t is discretized, with the activations updated at each time step. The time scale might correspond to any time step of size appropriate for the given problem. A delay unit given by the network NN_F needs to be introduced to hold activations in NN_O until they are processed at the next time step and vice versa. As for simple architectures and deterministic activation functions, learning will be achieved using similar gradient descent procedures to those leading to the back-propagation algorithm for feed forward networks.

The NA scheme is made up of two major steps: a *pre-processing* step and a *training* step. During the *pre-processing* step, the data set is normalized considering the information we have about the error estimations and the error covariance matrices introduced in (3). We consider, to normalise, the inverse of the error covariance matrices so that, data with big covariance/variance are assumed with a small weight [5,16]. We pose

$$\bar{x}(t) = B_t^{-1}x(t) \quad \text{and} \quad \bar{o}(t) = R_t^{-1}o(t). \tag{10}$$

The computed vectors $\bar{x}(t)$ and $\bar{o}(t)$ are the data used in the *training* step:

$$\bar{o}(t) = f_{O_O}\left(W_{HO_O}h(t-1)\right) \tag{11}$$

$$h(t) = f_H\left(W_{IH}\bar{x}(t-1) + W_{HH}h(t-1)\right) \tag{12}$$

$$\bar{x}(t) = f_{O_F}\left(W_{HO_F}h(t)\right) \tag{13}$$

where the vectors $\bar{x}(t-1)$ are the inputs, the matrices W_{IH}, W_{HH}, W_{HO_F} and W_{HO_O} are the four connection weight matrices, and f_H, f_{O_F} and f_{O_O} are the hidden and outputs unit activation functions. The state of the dynamical system is a set of values that summarizes all the information about the past behaviour of the system that is necessary to provide a unique description of its future behaviour, apart from the effect of any external factors. In this case the state is defined by the set of hidden unit activations $h(t)$. The Back propagation Through Time for this algorithm is a natural extension of standard back propagation that performs gradient descent on a complete unfolded network ([21], Chapter 5 of [6]). If the NA training sequence starts at time t_0 and ends at time t_1, the total cost function is simply the sum over time of the standard error function $C(t)$ at each time-step:

$$C_{total} = \sum_{t=t_0}^{t_1} C(t) \tag{14}$$

where

$$C(t) = \frac{1}{2}\sum_{k=1}^{n}\left((\bar{o}_k(t-1) - h_k(t-1))^2 + (\bar{x}_k(t) - h_k(t))^2\right) \tag{15}$$

and n is the total number of training samples. The gradient descent weight updates have contributions from each time-step [19]:

$$\Delta w_{ij} = -\eta\frac{\partial C_{total}(t_0, t_1)}{\partial w_{ij}} = -\eta\sum_{t=t_0}^{t_1}\frac{\partial C(t)}{\partial w_{ij}} \tag{16}$$

where η is the learning rate [14]. The constituent partial derivatives $\frac{\partial C(t)}{\partial w_{ij}}$ have contributions from the multiple instances of each weight

$$w_{ij} \in \{W_{IH}, W_{HH}, W_{HO_O}, W_{HO_F}\}$$

and depend on the inputs and hidden unit activations at previous time steps. The errors now have to be back-propagated through time as well as through the network [23].

We prove that the output function $h(t)$ of the NA model corresponds to the solution of Kalman filter with fixed covariance matrices, i.e. in its Optimal Interpolation version [16]. The following result held.

Theorem 1. *Let $h(t)$ be the solution of NA given by Eqs. (10)–(16) and let $z(t)$ denote the solution of the KF algorithm as defined in (9). We have*

$$h(t) = z(t), \quad \forall t \in [t_0, t_1] \tag{17}$$

Proof: *Due to the definition of the L^2 norm, the loss function in (15) can be written as*

$$C(t) = \|\bar{o}(t-1) - h(t-1)\|_2^2 + \|\bar{x}(t) - h(t)\|_2^2 \tag{18}$$

then, from Eq. (1), and except for the numerical errors that will be introduced later as already included in the data sets, the (18) can be written as:

$$C(t) = \|\bar{o}(t-1) - h(t-1)\|_2^2 + \|M\ \bar{x}(t-1) - M\ h(t-1)\|_2^2 \tag{19}$$

From the properties of the L^2 norm, the (19) can be written as

$$\begin{aligned} C(t) = &(\bar{o}(t-1) - h(t-1))^T(\bar{o}(t-1) - h(t-1)) \\ &+ (M\bar{x}(t-1) - Mh(t-1))^T(M\bar{x}(t-1) - Mh(t-1)). \end{aligned} \tag{20}$$

To minimise this loss function, we compute the gradient

$$\nabla_{h(t-1)}C(t) = 2(\bar{o}(t-1) - h(t-1)) + 2M^T(M\bar{x}(t-1) - M\ h(t-1)) \tag{21}$$

where M^T denotes the Adjoint operator of the linear operator M [7] and we pose $\nabla_{h(t-1)}C(t) = 0$, then we have:

$$2h(t-1) = \bar{o}(t-1) + \bar{x}(t-1) \tag{22}$$

From the definition of $\bar{x}$ and $\bar{o}$ in (10), the (22) gives:

$$h(t-1)\,(B_{t-1} + R_{t-1}) = R_{t-1}x(t-1) + B_{t-1}o(t-1) \tag{23}$$

Then, adding and subtracting the quantity $B_{t-1}x(t-1)$ and merging the common factors, the (23) become

$$h(t-1)\,(B_{t-1} + R_{t-1}) = x(t-1)\,(B_{t-1} + R_{t-1}) + B_{t-1}\,(o(t-1) - x(t-1)) \tag{24}$$

Finally, posed $Q_{t-1} = B_{t-1}\,(B_{t-1} + R_{t-1})^{-1}$, the (24) gives:

$$h(t-1) = x(t-1) + Q_{t-1}\,(o(t-1) - x(t-1)) \tag{25}$$

which is the expression of the KF solution $z(t-1)$ in (9) for the time step $t-1$ and for the case of observed data defined in the same space of the state variable (i.e. $H = I$ and I is the identity matrix). Q_{t-1} is the Kalman gain matrix in (8).

The Eq. (25) in Theorem 1 represents a condition to assume that NA is consistent with KF.

In Sect. 4, we validate the results provided in this section. We also show that the employment of NA alleviates the computational cost making the running less expensive.

4 Experimental Results

In this section we provide experimental results that demonstrate the applicability and efficiency of NA. In our experiment, the NA is implemented by adopting Long short-term memory (LSTM) architecture for the two RNNs. The reason we use LSTMs is that they are suitable to contain information outside the normal flow of the recurrent network so it is easier to plug two networks together. Also, LSTMs allow to preserve the error that can be backpropagated through time and layers which is a very important point for discrete forecasting models. A description of the NA we implemented is provided in Fig. 3.

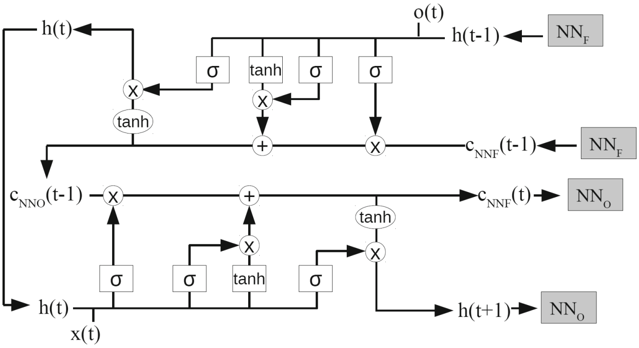


Fig. 3. Implementation of Neural Assimilation

The test case we consider is a numerical model to predict the oxygen diffusion across the Blood-Brain Barrier (BBB). Nevertheless the model can be used for any drugs by replacing the diffusion constant and the initial and boundary conditions [1]. The Blood-Brain Barrier protects the central nervous system, controls the entry of compounds into the brain by restricting access for blood borne compounds and facilitates access for nutrients. This protection makes it difficult to provide therapeutic compounds to brain cells when they are affected by brain diseases as Alzheimer, Autism [13]. The BBB is composed of endothelial cells connected by tight junctions. The main mechanisms allowing the transport of drugs across the membrane are passive transport, carrier-mediated transport, receptor-mediated transcytosis, and adsorption-mediated transcytosis [22]. The passive transport mechanism is the easiest method of drug transport for lipophilic and low molecular size molecules. It means a simple diffusion across any membrane without application of energy and carrier proteins. Opioids and steroids are examples of drugs which can be passively diffused [4]. Assuming that the main transport mechanism is through passive diffusion, the initial three-dimensional space problem can be reduced to a one-dimensional space problem. In fact, passive diffusion involves many simplifications as no reaction term, uniform movement in all directions and an overall diffusion constant. Therefore, a 1D partial differential equation (PDE) as (26) with one initial condition and two boundary conditions is an accurate model for this problem [22] where 0 corresponds to the location at which the blood meets the Blood-Brain Barrier and $L = 400\,\mathrm{nm}$ is the real average thickness of the Blood-Brain Barrier.

$$\begin{cases} \frac{\partial x}{\partial t} = D\frac{\partial^2 x}{\partial y^2} \\ x(0,y) = x_{0,y} \\ x(t,0) = x_{t,0} \\ x(t,L) = x_{t,L} \end{cases} \tag{26}$$

where $t \in [0, 10\,\mathrm{ms}]$ (ms denotes microsecond) and $y \in [0, L]$. We consider that, at time 0 there is no oxygen, then $x_{0,y} = 0$. Moreover, for our boundary conditions, we consider that we have a constant concentration of oxygen in the bloodstream and that at the interface of the barrier and the brain tissue all oxygen will be consumed $x_{t,0} = 0.02945$ L/L blood and $x_{t,L} = 0$. We assume the diffusivity of oxygen through the Blood-Brain Barrier to be $3.24 * 10^{-5}$ cm^2/s [1].

Equation (26) is discretised by a second order central finite difference in space with $\Delta y = 8\,\mathrm{nm}$ and a backward Euler method in time with $\Delta t = 0.1\,\mathrm{ms}$:

$$-Fx_{i-1}^n + (1+2F)x_i^n - x_{i+1}^n = x_{i-1}^{n-1}$$

where $F = D\frac{\Delta t}{\Delta y^2}$, $i = 1, \ldots, 50$ and $n = 1, \ldots, 100$. As we know that it does not affect the generality of our study, in this paper we show results of NA using observed data $o(t)$ provided in [1] by the analytical solution of (26) for the oxygen diffusion. The model can be used for any drugs by replacing the diffusion constant and the initial and boundary conditions. Data sets for observed data can be found in http://cheminformatics.org/datasets/. The NA code and the pre-processed data can be downloaded using the link https://drive.google.com/drive/folders/1C_O-rk5wyqFsG5U-T7_vugBOddTPmOlY?usp=sharing.

The NA network has been trained using the 85% of the data and tested on the remaining 15%. Figure 4 shows the value of the Loos function for training and testing the forecasting network.

NA has been compiled as a sequential neural network with just one LSTM layer of 48 units using as loss function the mean squared error one and as optimiser the Adam one. Weights are automatically initialised by Keras using:

- Glorot uniform for the kernel weights matrix for the linear transformation of the inputs;
- Orthogonal for the linear transformation of the recurrent state.

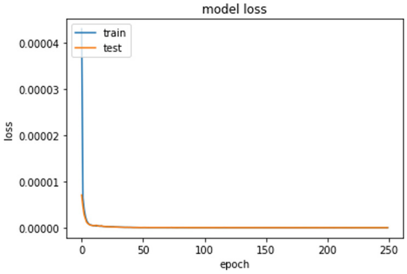


Fig. 4. Values of the loss function.

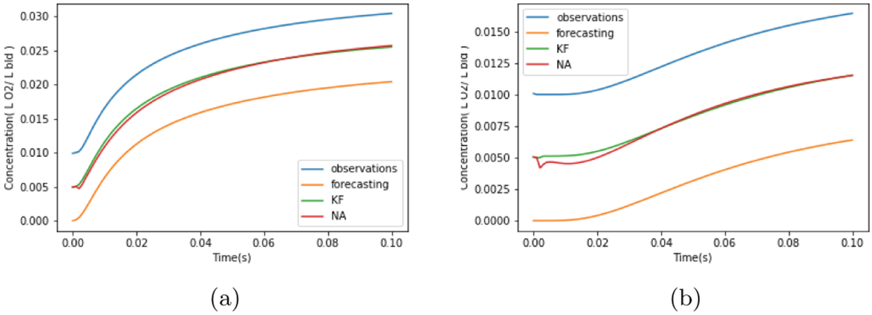


Fig. 5. Temporal evolution of the concentration at (a) $y = 12\,\text{nm}$ and (b) $y = 35\,\text{nm}$

Figure 5 shows the temporal evolution of the concentration at $y = 12\,\text{nm}$ (Fig. 5a) and $y = 35\,\text{nm}$ (Fig. 5b). The accuracy of the NA results is evaluated by the absolute error

$$e_{NA}(t, y) = |z(t, y) - h(t, y)| \tag{27}$$

and the mean squared error

$$MSE(h(t, y)) = \frac{\|z(t, y) - h(t, y)\|_{L^2}}{\|z(t, y)\|_{L^2}} \tag{28}$$

where $z(t, y)$ is the solution of KF performed at each time step. Table 1 shows values of absolute error computed every 10 time steps. We can see that the order of magnitude of the error is between $e{-}07$ and $e{-}04$. The corresponding values of mean squared error are $MSE(h(t, y)) = 1.31e{-}07$ for $y = 12\,\text{nm}$ and $MSE(h(t, y)) = 8.16e{-}08$ for $y = 35\,\text{nm}$ where $t \in [0, 0.10\,\text{ms}]$. Figure 6 shows the comparison of the KF result and the NA result for the temporal evolution of the concentration at each point of the BBB we are modelling. Values of execution time are provided in Table 2. The values are computed as mean of execution times from 100 runnings. We can observe that the time for *prediction* in NA is 1000 faster than the *prediction* with KF.

Table 1. Error computed every 10 time steps at (a) $y = 12\,\text{nm}$ and (b) $y = 35\,\text{nm}$

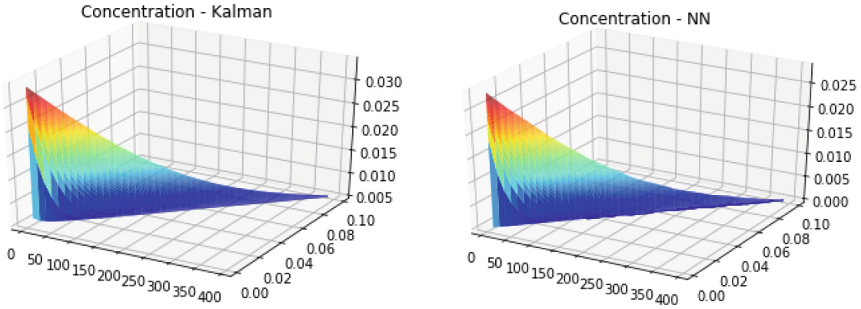
Time step t	$e_{NA}(t,y)$, $y = 12\,\text{nm}$	$e_{NA}(t,y)$, $y = 35\,\text{nm}$
0	0	0
10	7.05e−04	6.11e−04
20	4.17e−04	4.88e−04
30	4.29e−04	1.91e−04
40	1.52e−04	6.05e−07
50	2.51e−04	9.11e−05
60	3.40e−05	1.11e−04
70	4.13e−05	1.05e−04
80	4.72e−05	7.35e−05
90	1.11e−04	1.89e−05
100	1.60e−04	3.18e−05

Table 2. Execution time for 100 time steps and all the distances

	Executing time (s)
Neural Assimilation (training)	121.47
Neural Assimilation (prediction)	0.117
Kalman filter (prediction)	138

Finally, Table 3 shows the values of mean square forecasting error:

$$MSE^F(x(t,y)) = \frac{\|x(t,y) - o(t,y)\|_{L^2}}{\|o(t,y)\|_{L^2}} \tag{29}$$



(a) KF

(b) NN

Fig. 6. Comparison between Data Assimilation (KF) and the Neural Network version for $t \in [0, 10\,\text{ms}]$ (ms denotes microsecond) and $y \in [0, 400\,\text{nm}]$.

and mean square assimilation error:

$$MSE^{NA}(h(t,y)) = \frac{\|h(t,y) - o(t,y)\|_{L^2}}{\|o(t,y)\|_{L^2}} \tag{30}$$

computed with respect observations $o(t,y)$. The values of the errors in the assimilation results present a reduction of approximately one order of magnitude with respect to the error in forecasting data.

Table 3. Mean square error forecasting error MSE^F and mean square assimilation error MSE^{NA} computed every 10 time steps at $y = 12\,\text{nm}$

Time step t	$MSE^{NA}(h(t,y))$, $y = 12\,\text{nm}$	$MSE^F(x(t,y))$, $y = 12\,\text{nm}$
0	5.03e−03	1.00e−02
10	5.48e−03	1.00e−02
20	5.34e−03	9.99e−03
30	5.06e−03	9.95e−03
40	4.88e−03	1.00e−02
50	4.80e−03	1.00e−02
60	4.78e−03	1.00e−02
70	4.80e−03	1.00e−02
80	4.83e−03	1.00e−02
90	4.87e−03	1.00e−02
100	4.90e−03	1.00e−02

5 Conclusions and Future Works

We introduced a new neural network for Data Assimilation (DA) that we named Neural Assimilation (NA). We proved that the solution of NA is the same of KF. We tested the validity of the provided theoretical results showing values of misfit between the solution of NA and the solution of KF for the same test case. We provided experimental results on a realistic test case studying oxygen diffusion across the Blood-Brain Barrier. NA is trained on both forecasting and observed data and it is used for *predictions* without needing a correction given by the information provided by observations. This allows to avoid the *prediction-correction* cycle of a Kalman filter, and it makes the assimilation process very fast. We show that the time for *prediction* in NA is 1000 faster than the *prediction* with KF. An implementation of NA to emulate variational DA [10] will be developed as future work. In particular, we will focus on a 4D variational (4DVar) method [5]. 4DVar is a computational expensive method as it is developed to assimilate several observations (distributed in time) for each time step of the forecasting model. We will develop an extended version of NA able to assimilate set of distributed observations for each time step and, then, able to emulate 4DVar.

Acknowledgments. This work is supported by the EPSRC Centre for Mathematics of Precision Healthcare EP/N0145291/1 and the EP/T003189/1 Health assessment across biological length scales for personal pollution exposure and its mitigation (INHALE).

References

1. Agrawal, G., Bhullar, I., Madsen, J., Ng, R.: Modeling of oxygen diffusion across the blood-brain barrier for potential application in drug delivery (2013)
2. Arcucci, R., D'Amore, L., Pistoia, J., Toumi, R., Murli, A.: On the variational data assimilation problem solving and sensitivity analysis. J. Comput. Phys. **335**, 311–326 (2017)
3. Arcucci, R., Mottet, L., Pain, C., Guo, Y.K.: Optimal reduced space for variational data assimilation. J. Comput. Phys. **379**, 51–69 (2019)
4. Arya, M., Kumar, M.K.M., Sabitha, M., Menon, K.K., Nair, S.C.: Nanotechnology approaches for enhanced CNS delivery in treating Alzheimer's disease. J. Drug Deliv. Sci. Technol. **51**, 297–309 (2019)
5. Asch, M., Bocquet, M., Nodet, M.: Data Assimilation: Methods, Algorithms, and Applications, vol. 11. SIAM, Philadelphia (2016)
6. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Boston (2006). https://doi.org/10.1007/978-1-4615-7566-5
7. Cacuci, D.G.: Sensitivity & Uncertainty Analysis, Volume 1: Theory. Chapman and Hall/CRC, Boca Raton (2003)
8. de Campos Velho, H., Stephany, S., Preto, A., Vijaykumar, N., Nowosad, A.: A neural network implementation for data assimilation using MPI. WIT Trans. Inf. Commun. Technol. **27** (2002)
9. Cintra, R.S., de Campos Velho, H.F.: Data assimilation by artificial neural networks for an atmospheric general circulation model. In: Advanced Applications for Artificial Neural Networks. IntechOpen (2018)
10. D'Amore, L., Arcucci, R., Marcellino, L., Murli, A.: A parallel three-dimensional variational data assimilation scheme. In: AIP Conference Proceedings, vol. 1389, pp. 1829–1831. American Institute of Physics (2011)
11. Desroziers, G., Camino, J.T., Berre, L.: 4DEnVar: link with 4D state formulation of variational assimilation and different possible implementations. Q. J. R. Meteorol. Soc. **140**(684), 2097–2110 (2014)
12. Funahashi, K., Nakamura, Y.: Approximation of dynamical systems by continuous time recurrent neural networks. Neural Netw. **6**(6), 801–806 (1993)
13. Gabathuler, R.: Approaches to transport therapeutic drugs across the blood-brain barrier to treat brain diseases. Neurobiol. Dis. **37**(1), 48–57 (2010)
14. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
15. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960)
16. Kalnay, E.: Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press, Cambridge (2003)
17. Loh, K., Omrani, P.S., van der Linden, R.: Deep learning and data assimilation for real-time production prediction in natural gas wells. arXiv preprint arXiv:1802.05141 (2018)

18. Lorenc, A.C., Bowler, N.E., Clayton, A.M., Pring, S.R., Fairbairn, D.: Comparison of hybrid-4DEnVar and hybrid-4DVar data assimilation methods for global NWP. Mon. Weather Rev. **143**(1), 212–229 (2015)
19. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S.: Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078 (2017)
20. Schäfer, A.M., Zimmermann, H.G.: Recurrent neural networks are universal approximators. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 632–640. Springer, Heidelberg (2006). https://doi.org/10.1007/11840817_66
21. Schuster, M.: On supervised learning from sequential data with applications for speech recognition. Daktaro disertacija, Nara Institute of Science and Technology (1999)
22. Simmons, J.M.: Effects of febuxostat on autistic behaviors and computational investigations of diffusion and pharmacokinetics. Ph.D. thesis, Virginia Tech (2019)
23. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. Neural Netw. **1**(4), 339–356 (1988)