Overview of mathematical properties of Echo State Networks

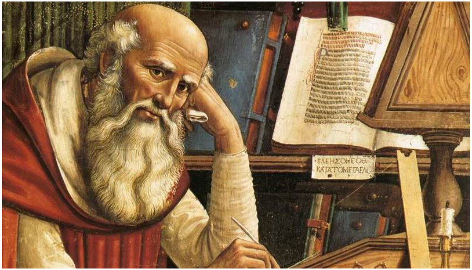
Luca Bonaventura

POLITECNICO MILANO 1863

DIPARTIMENTO DI MATEMATICA

Paris-Saclay, 7.6.22

REFLECTIONS
OF A FORMER
NEURAL NETWORK
SKEPTIC



Overview

Overview

- Review of the main concepts and definitions in **Reservoir Computing** and more specifically **Echo State Networks**

Overview

- Review of the main concepts and definitions in **Reservoir Computing** and more specifically **Echo State Networks**
- Review of some **key mathematical properties** that justify a **wide range of applications**

Overview

- Review of the main concepts and definitions in **Reservoir Computing** and more specifically **Echo State Networks**
- Review of some **key mathematical properties** that justify a **wide range of applications**
- Perspective on the **forthcoming developments** and applications of RC

Overview

- Review of the main concepts and definitions in **Reservoir Computing** and more specifically **Echo State Networks**
- Review of some **key mathematical properties** that justify a **wide range of applications**
- Perspective on the **forthcoming developments** and applications of RC
- Some **motivations** for the choice of ESN as **ML tool for particle beam design**

Acknowledgements and references

Acknowledgements and references

- **Ongoing** collaboration with **Barbara Dalena** (CEA) and more recently discussions with **Massimo Giovanozzi** (CERN)

Acknowledgements and references

- **Ongoing** collaboration with **Barbara Dalena** (CEA) and more recently discussions with **Massimo Giovanozzi** (CERN)
- Help on the statistical side from **Stefano Castruccio** (University of Notre Dame, USA)

Acknowledgements and references

- **Ongoing** collaboration with **Barbara Dalena** (CEA) and more recently discussions with **Massimo Giovanozzi** (CERN)
- Help on the statistical side from **Stefano Castruccio** (University of Notre Dame, USA)

- Some **key references and sources** for this presentation:

I. Daubechies, R. DeVore, S. Foucart et al, Nonlinear Approximation and (Deep) ReLU Networks, Constructive Approximation, 55, 127--172, 2022
M. Lukosevicius and **H. Jaeger**, Reservoir computing approaches to recurrent neural network training. Computer Science Review, 3, 127--149, 2009
A.Hart and J. Hook and J. Dawes, Embedding and approximation theorems for echo state networks, Neural Networks 128, 234--247, 2020

Geophysical Research Letters

RESEARCH LETTER

10.1029/2020GL087776

Key Points:
- A low-resolution, global, reservoir computing-based machine learning (ML) model can forecast the atmospheric state
- The training of the ML model is computationally efficient on a massively parallel computer
- Compared to a numerical

A Machine Learning-Based Global Atmospheric Forecast Model

Troy Arcomano[1], Istvan Szunyogh[1], Jaideep Pathak[2], Alexander Wikner[2], Brian R. Hunt[3], and Edward Ott[4]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Department of Mathematics, University of Maryland, College Park, MD, USA, [4]Department of Physics and Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

Some motivation

Geophysical Research Letters

RESEARCH LETTER
10.1029/2020GL087776

A Machine Learning-Based Global Atmospheric Forecast Model

Troy Arcomano[1], Istvan Szunyogh[1], Jaideep Pathak[2], Alexander Wikner[2], Brian R. Hunt[3], and Edward Ott[4]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Department of Mathematics, University of Maryland, College Park, MD, USA, [4]Department of Physics and Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

Key Points:
- A low-resolution, global, reservoir computing-based machine learning (ML) model can forecast the atmospheric state
- The training of the ML model is computationally efficient on a massively parallel computer
- Compared to a numerical

- RC is starting to solve **large scale, real life problems** that entail the simulation of an underlying **large dynamical system**

Some motivation

Geophysical Research Letters

RESEARCH LETTER

10.1029/2020GL087776

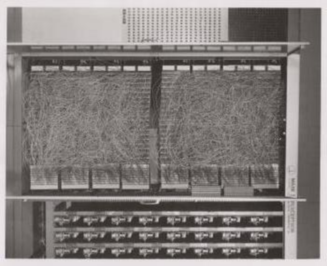
Key Points:
- A low-resolution, global, reservoir computing-based machine learning (ML) model can forecast the atmospheric state
- The training of the ML model is computationally efficient on a massively parallel computer
- Compared to a numerical

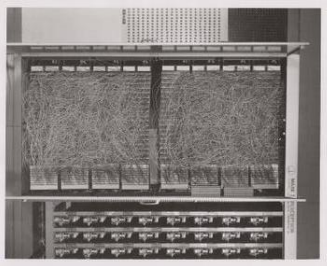
A Machine Learning-Based Global Atmospheric Forecast Model

Troy Arcomano[1], Istvan Szunyogh[1], Jaideep Pathak[2], Alexander Wikner[2], Brian R. Hunt[3], and Edward Ott[4]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Department of Mathematics, University of Maryland, College Park, MD, USA, [4]Department of Physics and Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

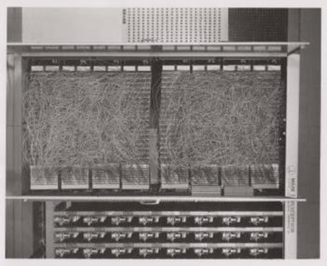
- RC is starting to solve **large scale, real life problems** that entail the simulation of an underlying **large dynamical system**
- RC is **not the only ML approach** that can do so, but it seems to be the **cheapest and easiest to train**





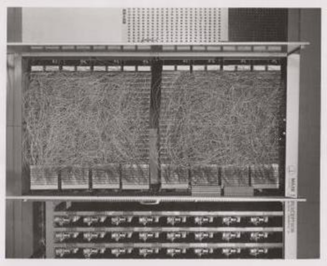
- At the beginning was the **perceptron**: built **as physical device** in the 1950s, analyzed **mathematically** in the 1960s

(Artificial) Neural Networks



- At the beginning was the **perceptron**: built **as physical device** in the 1950s, analyzed **mathematically** in the 1960s
- Example of network **without hidden layers**: unable to approximate **arbitrary input-output** relationships

(Artificial) Neural Networks



- At the beginning was the **perceptron**: built **as physical device** in the 1950s, analyzed **mathematically** in the 1960s
- Example of network **without hidden layers**: unable to approximate **arbitrary input-output** relationships
- Networks with **one or more hidden layers** were first investigated systematically in the 1980s

First basic mathematical property of NN

First basic mathematical property of NN

- Networks with **at least one hidden layer** have the **universal approximation** property: they can approximate **arbitrarily well** continuous and **measurable** functions

First basic mathematical property of NN

- Networks with **at least one hidden layer** have the **universal approximation** property: they can approximate **arbitrarily well** continuous and **measurable** functions
- This should **not be overrated**: also **polynomials, Fourier series, ..., wavelets (affine systems)** have similar properties!

First basic mathematical property of NN

- Networks with **at least one hidden layer** have the **universal approximation** property: they can approximate **arbitrarily well** continuous and **measurable** functions
- This should **not be overrated**: also **polynomials, Fourier series, ..., wavelets (affine systems)** have similar properties! (I. Daubechies also says so...) So what is the **difference**?

First basic mathematical property of NN

- Networks with **at least one hidden layer** have the **universal approximation** property: they can approximate **arbitrarily well** continuous and **measurable** functions
- This should **not be overrated**: also **polynomials, Fourier series, ..., wavelets (affine systems)** have similar properties! (I. Daubechies also says so...) So what is the **difference**?

High degree polynomials are computationally unstable, low degree polynomials are **insufficient** to approximate globally complicated functions (indeed scientific computing uses **local polynomial approximations**)

First basic mathematical property of NN

- Networks with **at least one hidden layer** have the **universal approximation** property: they can approximate **arbitrarily well** continuous and **measurable** functions
- This should **not be overrated**: also **polynomials, Fourier series, ..., wavelets (affine systems)** have similar properties! (I. Daubechies also says so...) So what is the **difference**?

Fourier series suffer from the **Gibbs phenomenon**

First basic mathematical property of NN

- Networks with **at least one hidden layer** have the **universal approximation** property: they can approximate **arbitrarily well** continuous and **measurable** functions
- This should **not be overrated**: also **polynomials, Fourier series, ..., wavelets (affine systems)** have similar properties! (I. Daubechies also says so...) So what is the **difference**?

Deep ANN can be proven to approximate whatever is approximated by **affine systems** at minimal cost: Bölcskei, Grohs, Kutyniok, and Petersen 2019

First basic mathematical property of NN

- Networks with **at least one hidden layer** have the **universal approximation** property: they can approximate **arbitrarily well** continuous and **measurable** functions
- This should **not be overrated**: also **polynomials, Fourier series, ..., wavelets (affine systems)** have similar properties! (I. Daubechies also says so...) So what is the **difference**?

Deep ANN can be proven to approximate whatever is approximated by **affine systems** at minimal cost: Bölcskei, Grohs, Kutyniok, and Petersen 2019 ....and **more**: **I. Daubechies** et al 2022

Feedforward vs Recurrent NN

Feedforward vs Recurrent NN

- Feedforward neural networks are composed by **neurons** linked by **connections** to **other neurons** only: **no cycles**

Feedforward vs Recurrent NN

- Feedforward neural networks are composed by **neurons** linked by **connections** to **other neurons** only: **no cycles**
- Recurrent neural networks are composed by **neurons** linked by **connections to themselves** and to other neurons: **cycles**

Feedforward vs Recurrent NN

- Feedforward neural networks are composed by **neurons** linked by **connections** to **other neurons** only: **no cycles**
- Recurrent neural networks are composed by **neurons** linked by **connections to themselves** and to other neurons: **cycles**
- Feedforward neural networks provide only **input-output** relationships: they are (complicated!) **functions**

Feedforward vs Recurrent NN

- Feedforward neural networks are composed by **neurons** linked by **connections** to **other neurons** only: **no cycles**
- Recurrent neural networks are composed by **neurons** linked by **connections to themselves** and to other neurons: **cycles**
- Feedforward neural networks provide only **input-output** relationships: they are (complicated!) **functions**
- Recurrent neural networks (RNN) preserve an **internal state** that is a **nonlinear transform** of the input signal: they are **dynamical systems**

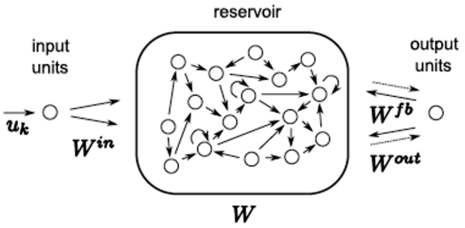
Feedforward vs Recurrent NN

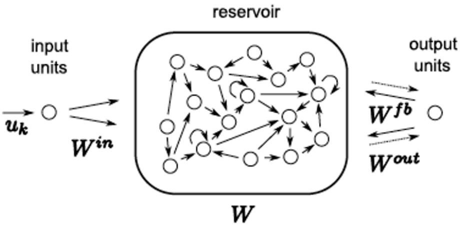
- Feedforward neural networks are composed by **neurons** linked by **connections** to **other neurons** only: **no cycles**
- Recurrent neural networks are composed by **neurons** linked by **connections to themselves** and to other neurons: **cycles**
- Feedforward neural networks provide only **input-output** relationships: they are (complicated!) **functions**
- Recurrent neural networks (RNN) preserve an **internal state** that is a **nonlinear transform** of the input signal: they are **dynamical systems**
- Major difficulties arise when **training** recurrent networks **using standard gradient based approaches**...

Feedforward vs Recurrent NN

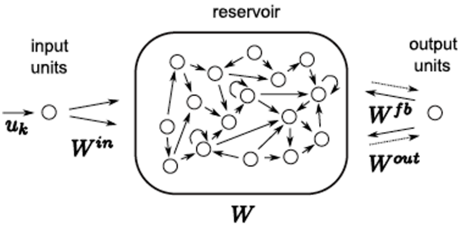
- Feedforward neural networks are composed by **neurons** linked by **connections** to **other neurons** only: **no cycles**
- Recurrent neural networks are composed by **neurons** linked by **connections to themselves** and to other neurons: **cycles**
- Feedforward neural networks provide only **input-output** relationships: they are (complicated!) **functions**
- Recurrent neural networks (RNN) preserve an **internal state** that is a **nonlinear transform** of the input signal: they are **dynamical systems**
- Major difficulties arise when **training** recurrent networks **using standard gradient based approaches**... but **Long Short Term Memory** networks can **mitigate or solve** this problem

Reservoir computing



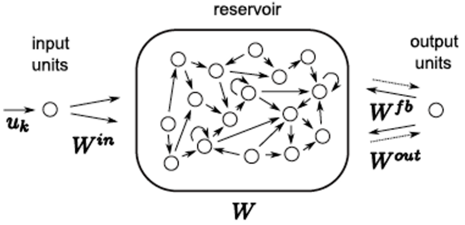


- Reservoir Computing (RC) (**Jaeger, 2001 and Maass et al. 2002**) uses a **random** recurrent network (the dynamical **reservoir**) which holds a nonlinear transform of the input (a.k.a. the **echo**)



- Reservoir Computing (RC) (**Jaeger, 2001 and Maass et al. 2002**) uses a **random** recurrent network (the dynamical **reservoir**) which holds a nonlinear transform of the input (a.k.a. the **echo**)
- No **backpropagation** is necessary

Reservoir computing



- Reservoir Computing (RC) (**Jaeger, 2001 and Maass et al. 2002**) uses a **random** recurrent network (the dynamical **reservoir**) which holds a nonlinear transform of the input (a.k.a. the **echo**)
- No **backpropagation** is necessary
- Training is performed only by (usually) **linear regression** to compute **weights** used to project the reservoir state onto the **output state**

RC brands: LSM vs ESN

RC brands: LSM vs ESN

- **Liquid State Machines** (Maass): use more **realistic neuron models** with complex **spiking** dynamics and **biologically** motivated network connectivity

RC brands: LSM vs ESN

- **Liquid State Machines** (Maass): use more **realistic neuron models** with complex **spiking** dynamics and **biologically** motivated network connectivity
- Reservoir seen as a **liquid** whose **ripples** (excited neurons) propagate information

RC brands: LSM vs ESN

- **Liquid State Machines** (Maass): use more **realistic neuron models** with complex **spiking** dynamics and **biologically** motivated network connectivity
- Reservoir seen as a **liquid** whose **ripples** (excited neurons) propagate information
- **Echo System Networks** (Jaeger): use **simpler neuron models** with randomly generated network connectivity

RC brands: LSM vs ESN

- **Liquid State Machines** (Maass): use more **realistic neuron models** with complex **spiking** dynamics and **biologically** motivated network connectivity
- Reservoir seen as a **liquid** whose **ripples** (excited neurons) propagate information
- **Echo System Networks** (Jaeger): use **simpler neuron models** with randomly generated network connectivity
- ESN found to be easier to implement and **sufficient** for **chaotic system prediction** (Jaeger and Haas, 2004)...but **never say never**...

Some special RC features

Some special RC features

- RC has **super Turing** computational capability (Siegelmann 1995), even with **rational** weights only

Some special RC features

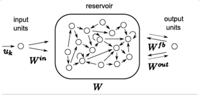
- RC has **super Turing** computational capability (Siegelmann 1995), even with **rational** weights only
- RC approaches based on **directed** networks are **universal approximators** of dynamical systems: Funahashi and Nakamura 1993, Gonon and Ortega, 2020, **Hart, Hook and Dawes, 2020**

Some special RC features

- RC has **super Turing** computational capability (Siegelmann 1995), even with **rational** weights only
- RC approaches based on **directed** networks are **universal approximators** of dynamical systems: Funahashi and Nakamura 1993, Gonon and Ortega, 2020, **Hart, Hook and Dawes, 2020**
- **Physical RC** devices are feasible, often with **photonic** devices:
  K. Nakajima, Physical reservoir computing|an introductory perspective, *Japanese Journal of Applied Physics* 59,060501, 2020
  Van der Sande et al., Advances in photonic reservoir computing, Nanophotonics, 6, 561--576, 2017

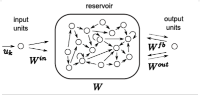
Formal ESN definition in discrete time

$$
\begin{aligned}
x_{k+1} &= f(Wx_k + W^{in}u_{k+1} + W^{fb}x_k^{out}) \\
x_{k+1}^{out} &= g(W^{out}[x_{k+1}; u_{k+1}])
\end{aligned}
$$



Formal ESN definition in discrete time

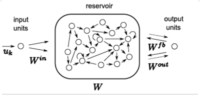
$$\begin{aligned} x_{k+1} &= f(Wx_k + W^{in}u_{k+1} + W^{fb}x_k^{out}) \\ x_{k+1}^{out} &= g(W^{out}[x_{k+1}; u_{k+1}]) \end{aligned}$$



- States $x_k \in \mathbb{R}^N$ **internal state**, $u_k \in \mathbb{R}^K$ **input state**, $x_k^{out} \in \mathbb{R}^L$ **output state**, with $N >> K, L$, at each **discrete time** $k \in \mathbb{Z}$

Formal ESN definition in discrete time

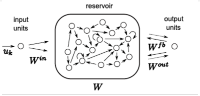
$$\begin{aligned} x_{k+1} &= f(Wx_k + W^{in}u_{k+1} + W^{fb}x_k^{out}) \\ x_{k+1}^{out} &= g(W^{out}[x_{k+1}; u_{k+1}]) \end{aligned}$$



- States $x_k \in \mathbb{R}^N$ **internal state**, $u_k \in \mathbb{R}^K$ **input state**, $x_k^{out} \in \mathbb{R}^L$ **output state**, with $N >> K, L$, at each **discrete time** $k \in \mathbb{Z}$
- Matrices: $W \in \mathcal{M}_N(\mathbb{R})$ **internal weights** (reservoir), $W^{in} \in \mathcal{M}_{N\times K}(\mathbb{R})$ **input**, $W^{fb} \in \mathcal{M}_{N\times L}(\mathbb{R})$ **feedback**, $W^{out} \in \mathcal{M}_{L\times(N+K)}(\mathbb{R})$ **output**

Formal ESN definition in discrete time

$$
\begin{aligned}
x_{k+1} &= f(Wx_k + W^{in}u_{k+1} + W^{fb}x_k^{out}) \\
x_{k+1}^{out} &= g(W^{out}[x_{k+1}; u_{k+1}])
\end{aligned}
$$



- States $x_k \in \mathbb{R}^N$ **internal state**, $u_k \in \mathbb{R}^K$ **input state**, $x_k^{out} \in \mathbb{R}^L$ **output state**, with $N >> K, L$, at each **discrete time** $k \in \mathbb{Z}$
- Matrices: $W \in \mathcal{M}_N(\mathbb{R})$ **internal weights** (reservoir), $W^{in} \in \mathcal{M}_{N\times K}(\mathbb{R})$ **input**, $W^{fb} \in \mathcal{M}_{N\times L}(\mathbb{R})$ **feedback**, $W^{out} \in \mathcal{M}_{L\times(N+K)}(\mathbb{R})$ **output**
- Functions: $f : \mathbb{R}^N \to \mathbb{R}^N$, **componentwise sigmoid**, $g : \mathbb{R}^L \to \mathbb{R}^L$, **componentwise sigmoid or identity**

Echo state property (ESP)

Echo state property (ESP)

- For networks without feedback $W^{fb}=0$ one has $x_{k+1} = f(Wx_k + W^{in}u_{k+1}) = F(x_k, u_{k+1})$

Echo state property (ESP)

- For networks without feedback $W^{fb} = 0$ one has $x_{k+1} = f(Wx_k + W^{in} u_{k+1}) = F(x_k, u_{k+1})$
- For standard sigmoid functions and inputs, x_k, u_k belong to limited, **compact** sets

Echo state property (ESP)

- For networks without feedback $W^{fb} = 0$ one has $x_{k+1} = f(Wx_k + W^{in}u_{k+1}) = F(x_k, u_{k+1})$
- For standard sigmoid functions and inputs, x_k, u_k belong to limited, **compact** sets
- An infinite sequence of states $y_k \quad k \leq 0$ is **compatible** with a given input sequence $u_k \quad k \leq 0$ if $y_k = F(y_{k-1}, u_k) \quad \forall k \leq 0$

Echo state property (ESP)

- For networks without feedback $W^{fb} = 0$ one has $x_{k+1} = f(Wx_k + W^{in}u_{k+1}) = F(x_k, u_{k+1})$
- For standard sigmoid functions and inputs, x_k, u_k belong to limited, **compact** sets
- An infinite sequence of states $y_k \quad k \leq 0$ is **compatible** with a given input sequence $u_k \quad k \leq 0$ if $y_k = F(y_{k-1}, u_k) \quad \forall k \leq 0$
- **ESP**: Given a network such that x_k, u_k belong to compact sets, it has the **Echo State Property** wrt the input sequence $u_k \quad k \leq 0$ if **all sequences compatible with $u_k \quad k \leq 0$ coincide**

Echo state property (ESP)

- For networks without feedback $W^{fb} = 0$ one has $x_{k+1} = f(Wx_k + W^{in}u_{k+1}) = F(x_k, u_{k+1})$
- For standard sigmoid functions and inputs, x_k, u_k belong to limited, **compact** sets
- An infinite sequence of states $y_k \quad k \leq 0$ is **compatible** with a given input sequence $u_k \quad k \leq 0$ if $y_k = F(y_{k-1}, u_k) \quad \forall k \leq 0$
- **ESP**: Given a network such that x_k, u_k belong to compact sets, it has the **Echo State Property** wrt the input sequence $u_k \quad k \leq 0$ if **all sequences compatible with $u_k \quad k \leq 0$ coincide**
- ESP guarantees that the ESN future evolution is **uniquely determined by the input**

Sufficient conditions for ESP

- Original proposal by Jaeger: $\max \sigma(W) < 1$, where $\sigma(W)$ denote singular values, found to be **overly restrictive**

Sufficient conditions for ESP

- Original proposal by Jaeger: $\max \sigma(W) < 1$, where $\sigma(W)$ denote singular values, found to be **overly restrictive**
- Common **sufficient** condition: $\rho(W) < 1$, where $\rho(W)$ denotes **spectral radius** ($\max \sigma(W) \neq \rho(W)$ for **non symmetric** matrices)

Sufficient conditions for ESP

- Original proposal by Jaeger: $\max \sigma(W) < 1$, where $\sigma(W)$ denote singular values, found to be **overly restrictive**
- Common **sufficient** condition: $\rho(W) < 1$, where $\rho(W)$ denotes **spectral radius** ($\max \sigma(W) \neq \rho(W)$ for **non symmetric** matrices)
- More **subtle and complex** conditions on the **internal weight matrix**, discussed in **Yildiz, Jaeger and Kiebel, 2012**

Sufficient conditions for ESP

- Original proposal by Jaeger: $\max \sigma(W) < 1$, where $\sigma(W)$ denote singular values, found to be **overly restrictive**
- Common **sufficient** condition: $\rho(W) < 1$, where $\rho(W)$ denotes **spectral radius** ($\max \sigma(W) \neq \rho(W)$ for **non symmetric** matrices)
- More **subtle and complex** conditions on the **internal weight matrix**, discussed in **Yildiz, Jaeger and Kiebel, 2012**
- In practice, entries of W are **rescaled with maximum eigenvalue** to guarantee ESP

Other important possible features

Other important possible features

- **Leaky neurons**: introduce parameter $a \in [0, 1]$ and define

$$x_{k+1} = (1 - a)x_k + af(Wx_k + W^{in}u_{k+1})$$

Corresponds to **low pass filtering** of the response to the input: **smaller** a implies **slower reservoir response**

- **Leaky neurons**: introduce parameter $a \in [0,1]$ and define

$$x_{k+1} = (1-a)x_k + af(Wx_k + W^{in}u_{k+1})$$

Corresponds to **low pass filtering** of the response to the input: **smaller a** implies **slower reservoir response**

- **Continuous time (leaky)** ESN:

$$\begin{aligned} x'(t) &= f(Wx(t) + W^{in}u(t)) - Dx(t) \\ x^{out}(t) &= g(W^{out}[x(t); u(t)]) \end{aligned}$$

where $D \in \mathcal{M}_N(\mathbb{R})$ is a **diagonal matrix** such that $d_{i,i} = 1/\tau_i$ and τ_i are the time scales for each neuron

Other important possible features

- **Leaky neurons**: introduce parameter $a \in [0,1]$ and define

$$x_{k+1} = (1-a)x_k + af(Wx_k + W^{in}u_{k+1})$$

Corresponds to **low pass filtering** of the response to the input: **smaller a** implies **slower reservoir response**

- **Continuous time (leaky)** ESN:

$$\begin{aligned} x'(t) &= f(Wx(t) + W^{in}u(t)) - Dx(t) \\ x^{out}(t) &= g(W^{out}[x(t); u(t)]) \end{aligned}$$

where $D \in \mathcal{M}_N(\mathbb{R})$ is a **diagonal matrix** such that $d_{i,i} = 1/\tau_i$ and τ_i are the time scales for each neuron

- Combine **multiple** reservoirs with different time scales: Long Short Term Memory (LSTM)

Validation of ESN output

Validation of ESN output

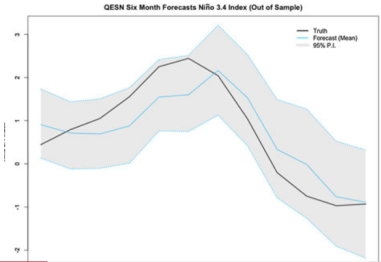
- ESN **validation** has only recently begun to draw attention: in earlier studies, **fixing the reservoir** was considered acceptable...

Validation of ESN output

- ESN **validation** has only recently begun to draw attention: in earlier studies, **fixing the reservoir** was considered acceptable...
- ...but recently a **statistical, ensemble approach** is increasingly common:
  P.L. McDermott, C.K. Wikle, An ensemble quadratic echo state network for nonlinear spatio-temporal forecasting, STAT, 6, 315-330, 2017

Validation of ESN output

- ESN **validation** has only recently begun to draw attention: in earlier studies, **fixing the reservoir** was considered acceptable...
- ...but recently a **statistical, ensemble approach** is increasingly common:
  P.L. McDermott, C.K. Wikle, An ensemble quadratic echo state network for nonlinear spatio-temporal forecasting, STAT, 6, 315-330, 2017



Next generation RC?

Next generation RC?

- RC with **nonlinear readout** is strictly related to **non linear vector autoregression**...

Next generation RC?

- RC with **nonlinear readout** is strictly related to **non linear vector autoregression**... so some people say: **scrap the random network reservoir** and get equivalent quality, but **cheaper** universal approximator for dynamical systems D.J. Gauthier, E. Bollt et al. Next generation reservoir computing, Nature Communications, 12, 5564, 2021

Next generation RC?

- RC with **nonlinear readout** is strictly related to **non linear vector autoregression**... so some people say: **scrap the random network reservoir** and get equivalent quality, but **cheaper** universal approximator for dynamical systems D.J. Gauthier, E. Bollt et al. Next generation reservoir computing, Nature Communications, 12, 5564, 2021
- Personal view on this:

Next generation RC?

- RC with **nonlinear readout** is strictly related to **non linear vector autoregression**... so some people say: **scrap the random network reservoir** and get equivalent quality, but **cheaper** universal approximator for dynamical systems D.J. Gauthier, E. Bollt et al. Next generation reservoir computing, Nature Communications, 12, 5564, 2021
- Personal view on this: they may be right, but **uncertainties** are everywhere and the practical value of **purely deterministic forecasts** is limited

Next generation RC?

- RC with **nonlinear readout** is strictly related to **non linear vector autoregression**... so some people say: **scrap the random network reservoir** and get equivalent quality, but **cheaper** universal approximator for dynamical systems D.J. Gauthier, E. Bollt et al. Next generation reservoir computing, Nature Communications, 12, 5564, 2021
- Personal view on this: they may be right, but **uncertainties** are everywhere and the practical value of **purely deterministic forecasts** is limited
- Having a naturally and **trivially parallellizable** approach to predict **probability distributions** of quantities of interest does not seem such a bad idea

Next generation RC?

- RC with **nonlinear readout** is strictly related to **non linear vector autoregression**... so some people say: **scrap the random network reservoir** and get equivalent quality, but **cheaper** universal approximator for dynamical systems D.J. Gauthier, E. Bollt et al. Next generation reservoir computing, Nature Communications, 12, 5564, 2021
- Personal view on this: they may be right, but **uncertainties** are everywhere and the practical value of **purely deterministic forecasts** is limited
- Having a naturally and **trivially parallellizable** approach to predict **probability distributions** of quantities of interest does not seem such a bad idea
- Furthermore, scrapping the network may imply also losing all the **super-approximation** properties: do we really want that?

The future is hybrid

JAMES | Journal of Advances in Modeling Earth Systems*

RESEARCH ARTICLE
10.1029/2021MS002712

Special Section:
Machine learning application to Earth system modeling

A Hybrid Approach to Atmospheric Modeling That Combines Machine Learning With a Physics-Based Numerical Model

Troy Arcomano[1], Istvan Szunyogh[1], Alexander Wikner[2], Jaideep Pathak[3], Brian R. Hunt[4], and Edward Ott[2,5]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Lawrence Berkeley National Laboratory, Berkeley, CA, USA, [4]Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA, [5]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

Key Points:

- A hybrid model incorporating machine learning produces more

The future is hybrid

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2021MS002712

Special Section:
Machine learning application to Earth system modeling

Key Points:

- A hybrid model incorporating machine learning produces more

A Hybrid Approach to Atmospheric Modeling That Combines Machine Learning With a Physics-Based Numerical Model

Troy Arcomano[1], Istvan Szunyogh[1], Alexander Wikner[2], Jaideep Pathak[3], Brian R. Hunt[4], and Edward Ott[2,5]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Lawrence Berkeley National Laboratory, Berkeley, CA, USA, [4]Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA, [5]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

The future is hybrid

JAMES | Journal of Advances in Modeling Earth Systems*

RESEARCH ARTICLE
10.1029/2021MS002712

A Hybrid Approach to Atmospheric Modeling That Combines Machine Learning With a Physics-Based Numerical Model

Troy Arcomano[1], Istvan Szunyogh[1], Alexander Wikner[2], Jaideep Pathak[3], Brian R. Hunt[4], and Edward Ott[2,5]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Lawrence Berkeley National Laboratory, Berkeley, CA, USA, [4]Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA, [5]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

Special Section:
Machine learning application to Earth system modeling

Key Points:

- A hybrid model incorporating machine learning produces more

- Use RC (and, more generally, ANN) predictions to **complement and integrate** more conventional, equation based models

The future is hybrid

JAMES | Journal of Advances in Modeling Earth Systems®

RESEARCH ARTICLE
10.1029/2021MS002712

A Hybrid Approach to Atmospheric Modeling That Combines Machine Learning With a Physics-Based Numerical Model

Troy Arcomano[1], Istvan Szunyogh[1], Alexander Wikner[2], Jaideep Pathak[3], Brian R. Hunt[4], and Edward Ott[2,5]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Lawrence Berkeley National Laboratory, Berkeley, CA, USA, [4]Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA, [5]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

Special Section:
Machine learning application to Earth system modeling

Key Points:
- A hybrid model incorporating machine learning produces more

- Use RC (and, more generally, ANN) predictions to **complement and integrate** more conventional, equation based models
- No need to waste time **learning what you already know**

The future is hybrid

JAMES | Journal of Advances in Modeling Earth Systems®

RESEARCH ARTICLE
10.1029/2021MS002712

Special Section:
Machine learning application to Earth system modeling

Key Points:
- A hybrid model incorporating machine learning produces more

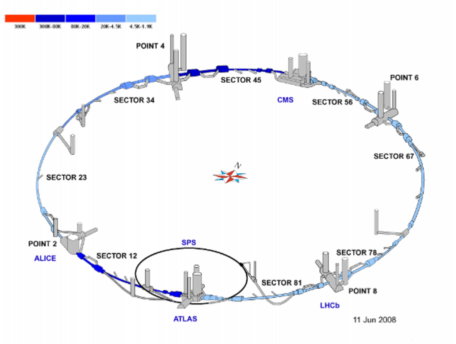
A Hybrid Approach to Atmospheric Modeling That Combines Machine Learning With a Physics-Based Numerical Model

Troy Arcomano[1], Istvan Szunyogh[1], Alexander Wikner[2], Jaideep Pathak[3], Brian R. Hunt[4], and Edward Ott[2,5]

[1]Department of Atmospheric Sciences, Texas A&M University, College Station, TX, USA, [2]Department of Physics, University of Maryland, College Park, MD, USA, [3]Lawrence Berkeley National Laboratory, Berkeley, CA, USA, [4]Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA, [5]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA

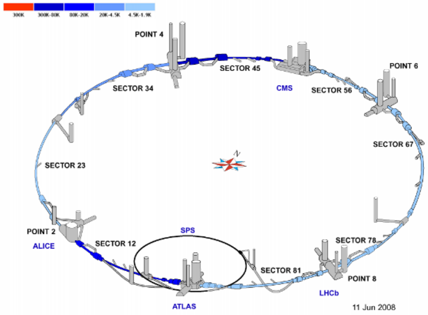
- Use RC (and, more generally, ANN) predictions to **complement and integrate** more conventional, equation based models
- No need to waste time **learning what you already know**
- Possible bottom line: classical approximation methods (polynomials etc.) **do it better** for **smooth functions/larger scales**, while data based methods for **irregular functions/fine, turbulent scales**

What are we doing with RC?



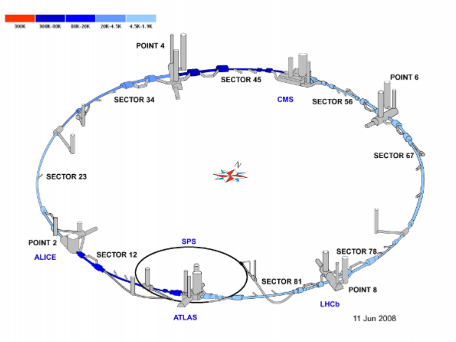
ML emulation of particle beam evolution in **particle accelerators**, such as the **Large Hadron Collider**

What are we doing with RC?



Particle beam modelling mostly relies on simulation of a **small Hamiltonian** system, whose trajectories must be simulated for a **very long time:** $O(10^9)$ **turns or more**: direct simulation is expensive

What are we doing with RC?



An approach based on a combination of **cheaper emulators** and **asymptotic scaling laws** looks promising (see **next talk**!)