Physico-statistical systems
A new challenge for machine learning

Patrick Gallinari

Sorbonne University, Paris, France

patrick.gallinari@lip6.fr

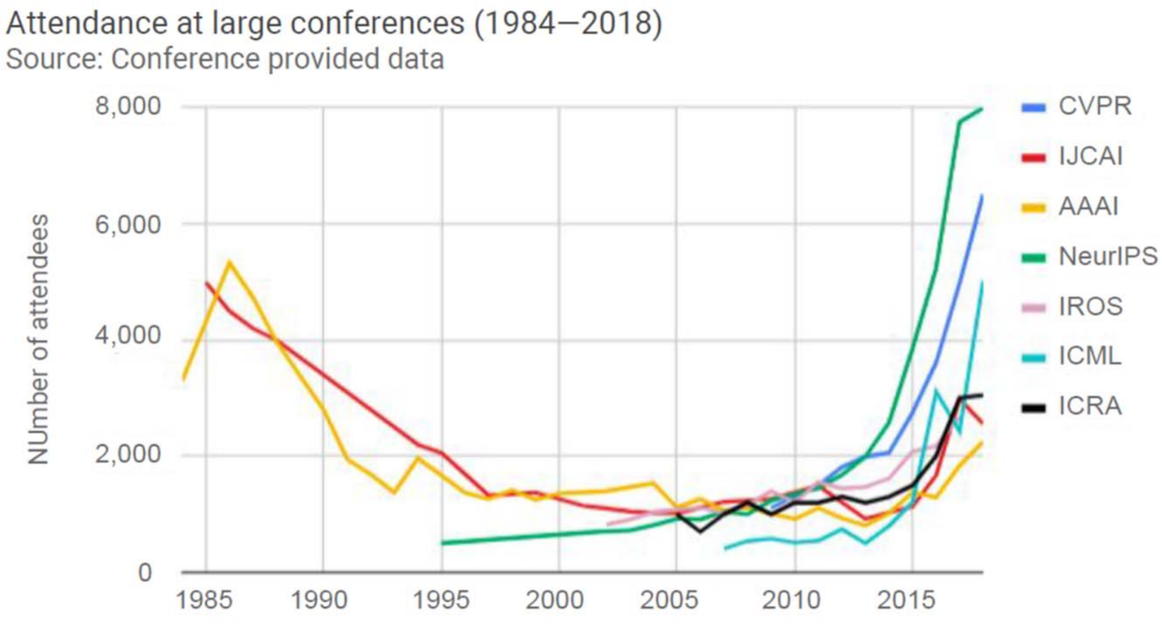
https://mlia.lip6.fr/

Presentation outline

- Deep Learning successes – examples
- Context
  - Statistical vs physical modeling
- Learning differential equations from data
- Learning dynamical systems from noisy observations

Deep Learning successes

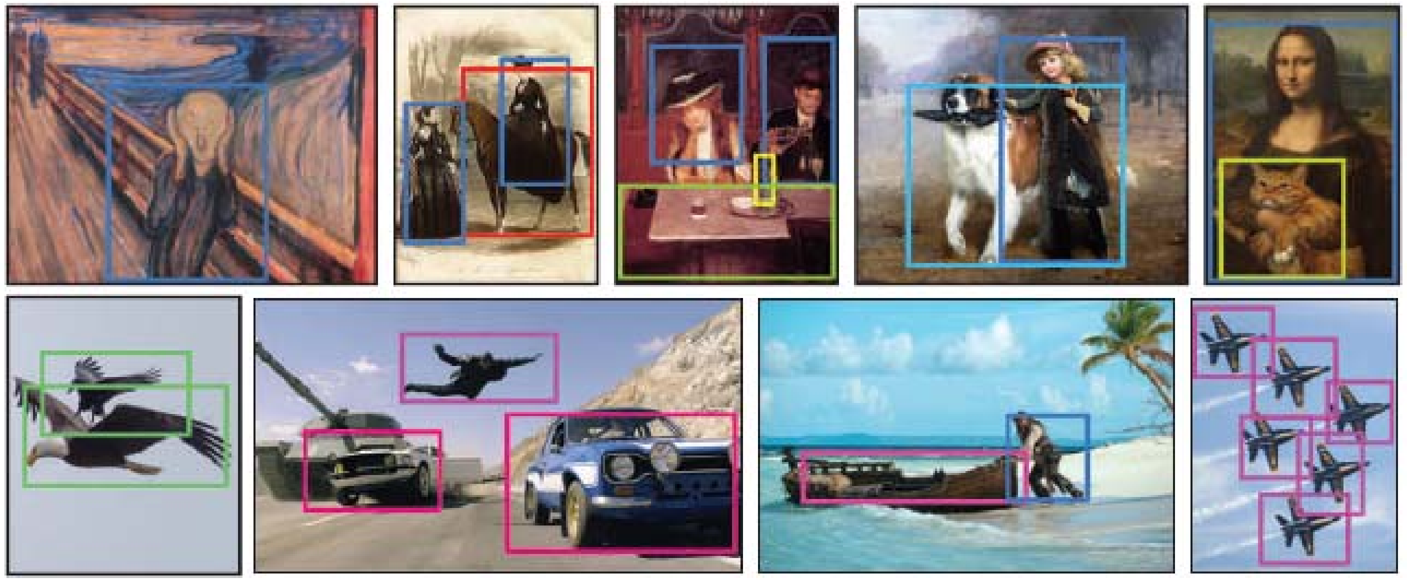
- Machine learning is at the heart of AI revolution
  - Deep Learning is today state of the art technology for data science
  - Unprecedented technological development over the last ten years
  - Important progresses in many engineering domains
  - Research and technological developments are leaded by tech. giants in the US and in China



Source: AI Index 2018 Report

Deep Learning successes - Vision
Object detection -YOLO (Redmon 2016)

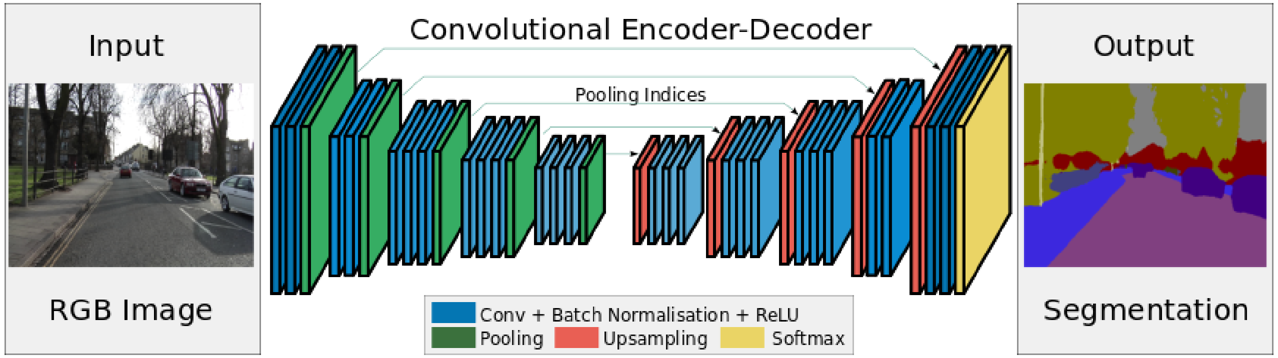
- Objective detect and label objects in images, video



alitative Results. YOLO running on artwork and natural images. It is mostly accurate although it does think one person in airplane.

Deep Learning successes
Vision – Scene segmentation (Segnet, Badrinarayanan 2017)

- Segment objects in image – pixel level



http://mi.eng.cam.ac.uk/projects/segnet/#demo

Deep Learning successes - Translation
Google Neural Machine Translation System

(Wu et al 2016)
https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

- General Architecture

Encoder: 8 stacked LSTM RNN + residual connections

Decoder: 8 stacked LSTM RNN + residual connections + Softmax output layer

Attention mechanism

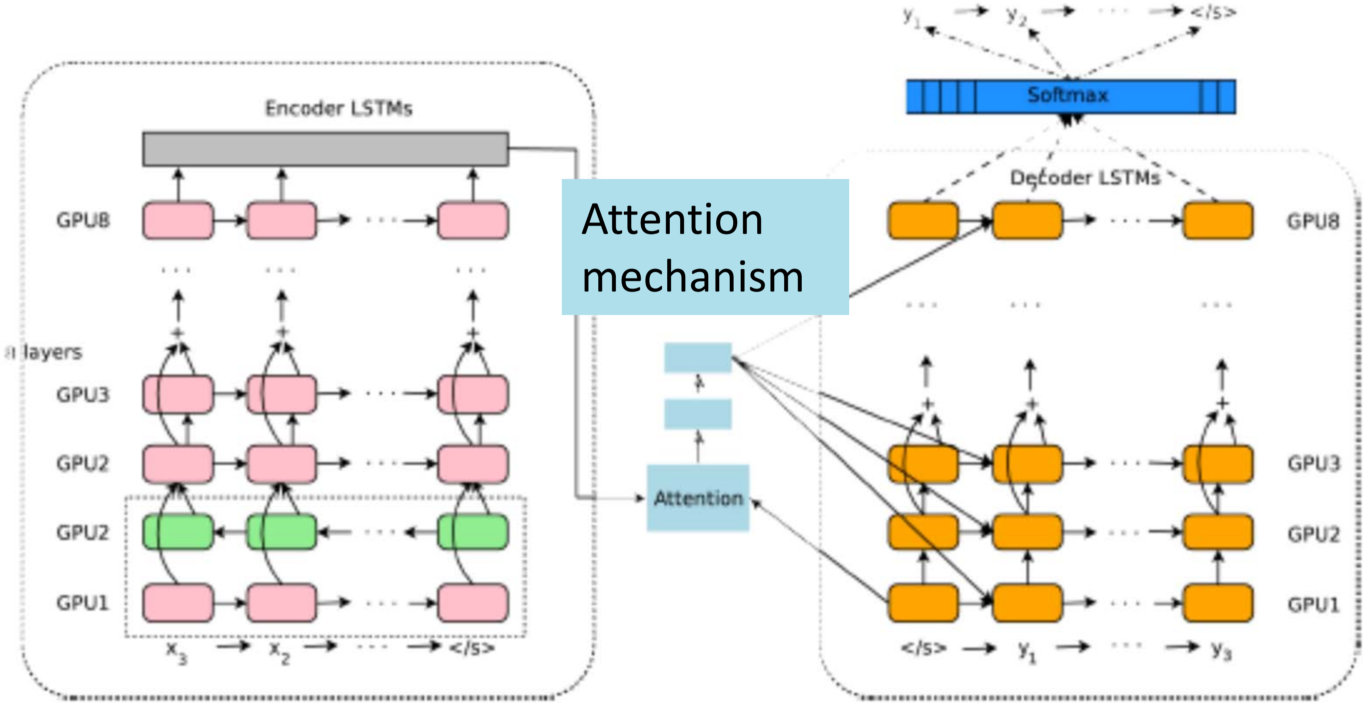


Figure from Wu et al. 2016

Deep Learning successes - Natural Language Processing GPT2 (Radford 2018 - OpenAI)

- Language model trained to predict the next word
  - Probability models of sequences of items $(x^1, x^2, \ldots, x^{\mathrm{t}})$
  - Estimate $p(x^t | x^{t-1}, \ldots, x^1)$
  - Items may be words or characters, or character bigrams, etc
- Model
  - Large transformer- based language model with 1.5 billion parameters, trained on a dataset of 8 million web pages, representing 40 GB of internet text.

"New AI fake text generator may be too dangerous to release, say creators (The Guardian, 2019-02-14)"

https://blog.openai.com/better-language-models/

Deep Learning successes - Natural Language Processing GPT2 (Radford 2018 - OpenAI)

- Context

Some of the most glorious historical attractions in Spain date from the period of Muslim rule, including The Mezquita, built as the Great Mosque of Cordoba and the Medina Azahara, also in Cordoba and now in ruins but still visitable as such and built as the Madinat al-Zahra, the Palace of al-Andalus; and the Alhambra in Granada, a splendid, intact palace. ……..[edit]

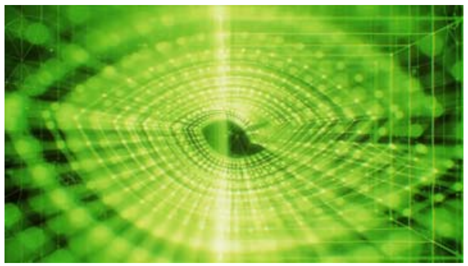
- Text generated

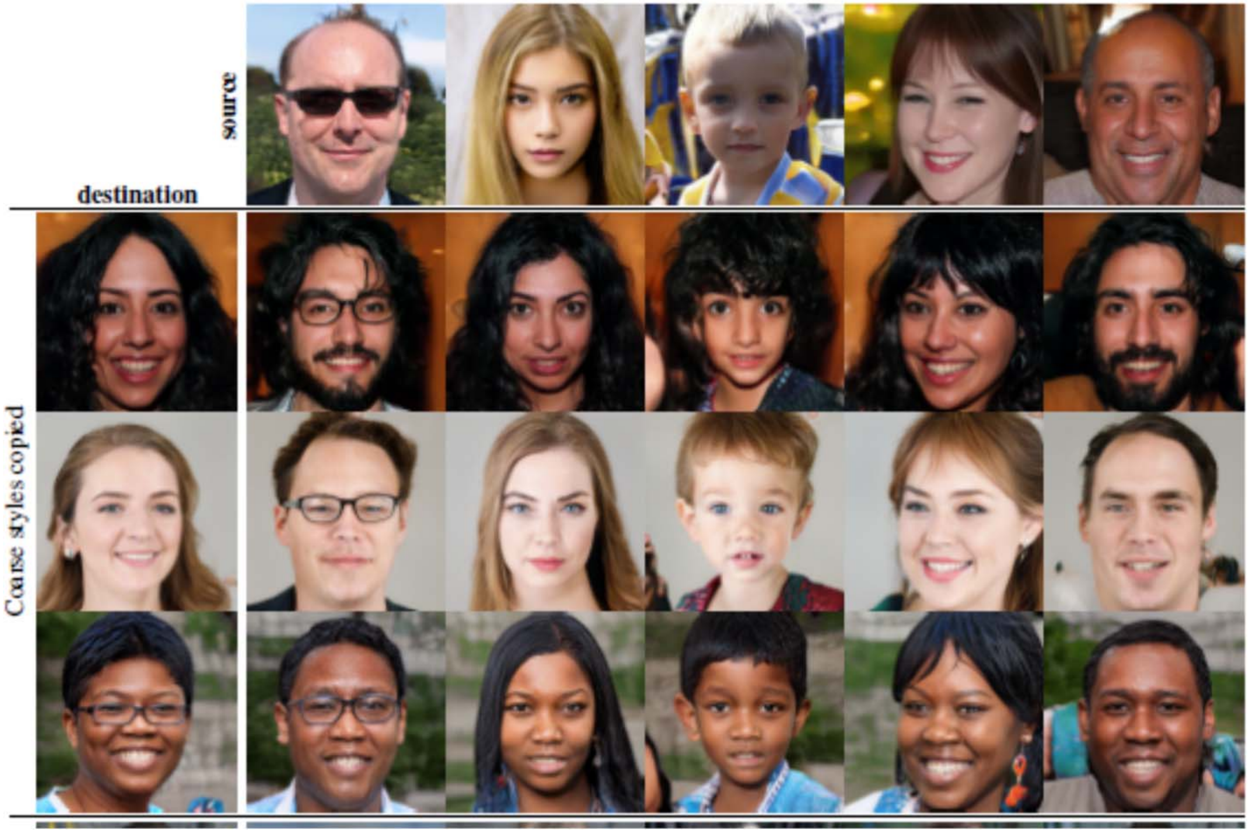
The Reconquista (literally, ”The return to the country”) of 1092 and 1095 saw the establishment in the Iberian peninsula of a united kingdom of Castile and Aragon in which a Muslim dynasty would exist. This was the last time in Europe where Muslim rule would be considered ”native” and the last occasion where Christian-inflamed and organized conflicts would make themselves known in Europe.

The reign of Muslim Moors in Spain is often referred to, especially among non-Muslims, as the ”Conquest of Granada” and Spain is sometimes referred to as Spain in Granada”. The last years of Muslim rule were followed by the Spanish Reconquista where the Muslims were driven or compelled out of the Iberian Peninsula with the victory of Ferdinand IX. in the Battle of Poitiers in March 1189 and the conquest of Jerusalem in 1187 by Saladin.

Deep Learning successes
Learning Complex Distributions
Generative Adversarial Networks -GANs

- About 2.2 k papers on Arxiv at Feb. 2019
  - Example: illustration from (Karras et al. 2019) – Style GAN





Deep Learning successes
Games

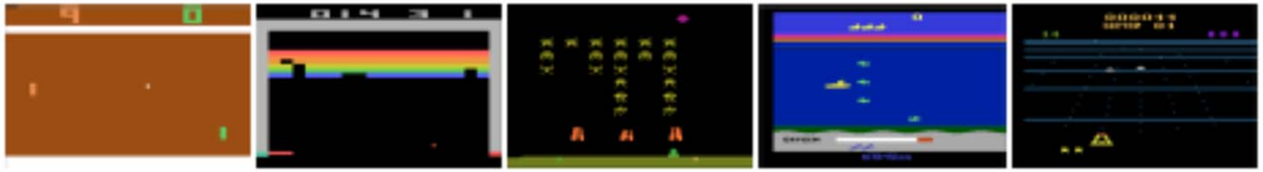


Figure 1: Screen shots from five Atari 2600 Games: (*Left-to-right*) Pong, Breakout, Space Invaders, Seaquest, Beam Rider

Atari games, Self trained on 49 games, (Mnih 2013, 2015)

AlphaGo, AlphaGo Zero, Alpha Zero from Google DeepMind (2015, 2017)

Deep Learning successes
Platforms

- Deep Learning platforms offer
  - Classical DL models
  - Optimization algorithms
  - Automatic differentiation
  - Popular options/ tricks
  - Pretrained models
  - CUDA/ GPU/ CLOUD support
- Contributions by large open source communities: lots of code available
- Easy to build/ train sophisticated models

- Among the most populars platforms:
  - TensorFlow - Google Brain - Python, C/C++
  - PyTorch – Facebook- Python
  - Caffe – UC Berkeley / Caffe2 Facebook, Python, MATLAB
  - Higher level interfaces
    - e.g. Keras for TensorFlow
- And also:
  - PaddlePaddle (Baidu), MXNet (Amazon), Mariana (Tencent), PAI 2.0 (Alibaba), ...

Statistical modeling versus knowledge based approaches

- **AI has a long history on knowledge based vs statistical approaches**
  - Perception, language, diagnosis, reasoning, explainability, learning, knowledge representation, etc
  - e.g. symbolic AI vs statistical learning
- **Elements of answer**
  - **Frederick Jelinek** (1932 – 2010)
    - researcher in information theory, automatic speech recognition, and natural language processing
  - Pioneer of speech recognition technology
    - Apocryphal ? Statement
    - "Every time I fire a linguist, the performance of the speech recognizer goes up"
- Today
  - Availability of data has changed everything
  - All that matters is: Data, Computer Ressources
  - Machine Learning is ubiquituous in many engineering fields
    - Vision, speech, language, web, mobile applications, etc
    - Cars, drones, robots, etc

Physico-statistical systems

Focus on spatio-temporal dynamics

- Learning differential equations from data
- Learning dynamical systems from noisy observations

Motivation - Example
Deep Learning and Physical Sciences - Geoscience

review articles

Communication of the ACM, January 2019

DOI:10.1145/3192335

YOLANDA GIL
University of Southern California
SUZANNE A. PIERCE
The University of Texas Austin
HASSAN BABAIE
Georgia State University
ARINDAM BANERJEE
University of Minnesota
KIRK BORNE
Booz Allen Hamilton
GARY BUST
Johns Hopkins University
MICHELLE CHEATHAM
Wright State University
IMME EBERT-UPHOFF
Colorado State University
CARLA GOMES
Cornell University
MARY HILL
University of Kansas
JOHN HOREL
University of Utah
LESLIE HSU
Columbia University
JIM KINTER
George Mason University
CRAIG KNOBLOCK
University of Southern California
DAVID KRUM
University of Southern California
VIPIN KUMAR
University of Minnesota
PIERRE LERMUSIAUX
Massachusetts Institute of Technology
YAN LIU
University of Southern California
CHRIS NORTH
Virginia Tech
VICTOR PANKRATIUS
Massachusetts Institute of Technology
SHANAN PETERS
University of Wisconsin-Madison
BETH PLALE
Indiana University Bloomington
ALLEN POPE
University of Colorado Boulder
SAI RAVELA
Massachusetts Institute of Technology
JUAN RESTREPO
Oregon State University
AARON RIDLEY
University of Michigan
HANAN SAMET
University of Maryland
SHASHI SHEKHAR
University of Minnesota

A research agenda for intelligent systems that will result in fundamental new capabilities for understanding the Earth system.

Intelligent Systems for Geosciences: An Essential Research Agenda

MANY ASPECTS OF geosciences pose novel problems for intelligent systems research. Geoscience data is challenging because it tends to be uncertain, intermittent, sparse, multiresolution, and multi-scale. Geosciences processes and objects often have amorphous spatiotemporal boundaries. The lack of ground truth makes model evaluation, testing, and comparison difficult. Overcoming these challenges requires breakthroughs that would significantly transform intelligent systems, while greatly benefitting the geosciences in turn. Although there have been significant and beneficial interactions between the intelligent systems and geosciences communities,[4,12] the potential for synergistic research in intelligent

statistical sys

PERSPECTIVE

https://doi.org/10.1038/s41586-019-0912-1

Nature Feb. 2019

Deep learning and process understanding for data-driven Earth system science

Markus Reichstein[1,2*], Gustau Camps-Valls[3], Bjorn Stevens[4], Martin Jung[1], Joachim Denzler[2,5], Nuno Carvalhais[1,6] & Prabhat[7]

Machine learning approaches are increasingly used to extract patterns and insights from the ever-increasing stream of geospatial data, but current approaches may not be optimal when system behaviour is dominated by spatial or temporal context. Here, rather than amending classical machine learning, we argue that these contextual cues should be used as part of deep learning (an approach that is able to extract spatio-temporal features automatically) to gain further process understanding of Earth system science problems, improving the predictive ability of seasonal forecasting and modelling of long-range spatial connections across multiple timescales, for example. The next step will be a hybrid modelling approach, coupling physical process models with the versatility of data-driven machine learning.

Humans have always striven to predict and understand the world, and the ability to make better predictions has given competitive advantages in diverse contexts (such as weather, diseases or financial markets). Yet the tools for prediction have substantially changed over time, from ancient Greek philosophical reasoning to non-scientific medieval methods such as soothsaying, towards modern scientific discourse, which has come to include hypothesis testing, theory development and computer modelling underpinned by statistical and physical relationships, that is, laws[1]. A success story in the geosciences is weather prediction, which has greatly improved through the integration of better theory, increased computational power, and established observational systems, which allow for the assimilation of large amounts of data into the modelling system[2]. Nevertheless, we can accurately predict the evolution of the weather on a timescale of days, not months. Seasonal meteorological predictions, forecasting extreme events such as flooding or fire, and long-term climate projections are still major challenges. This is especially true for predicting dynamics in the biosphere, which is dominated by biologically mediated processes such as growth or reproduction, and is strongly controlled by seemingly stochastic disturbances such as fires and landslides. Such predictive problems have not seen much progress in the past few decades[3].

At the same time, a deluge of Earth system data has become available, with storage volumes already well beyond dozens of petabytes and rapidly increasing transmission rates exceeding hundreds of terabytes per day[4]. These data come from a plethora of sensors measuring states, fluxes and intensive or time/space-integrated variables, representing fifteen or more orders of temporal and spatial magnitude. They include remote sensing from a few metres to hundreds of kilometres above Earth as well as in situ observations (increasingly from autonomous sensors) at and below the surface and in the atmosphere, many of which are further being complemented by citizen science observations. Model simulation output adds to this deluge; the CMIP-5 dataset of the Climate Model Intercomparison Project, used extensively for scientific groundwork towards periodic climate assessments, is over 3 petabytes in size, and the next generation, CMIP-6, is estimated to reach up to 30 petabytes[5]. The data from models share many of the challenges and statistical properties of observational data, including many forms of uncertainty. In summary, Earth system data are exemplary of all four of the 'four Vs' of 'big data': volume, velocity, variety and veracity (see Fig. 1). One key challenge is to extract interpretable information and knowledge from this big data, possibly almost in real time and integrating between disciplines.

Taken together, our ability to collect and create data far outpaces our ability to sensibly assimilate it, let alone understand it. Predictive ability in the last few decades has not increased apace with data availability. To get the most out of the explosive growth and diversity of Earth system data, we face two major tasks in the coming years: (1) extracting knowledge from the data deluge, and (2) deriving models that learn much more from data than traditional data assimilation approaches can, while still respecting our evolving understanding of nature's laws.

The combination of unprecedented data sources, increased computational power, and the recent advances in statistical modelling and machine learning offer exciting new opportunities for expanding our knowledge about the Earth system from data. In particular, many tools are available from the fields of machine learning and artificial intelligence, but they need to be further developed and adapted to geo-scientific analysis. Earth system science offers new opportunities, challenges and methodological demands, in particular for recent research lines focusing on spatio-temporal context and uncertainties (Box 1; see https://developers.google.com/machine-learning/glossary/ and http://www.wildml.com/deep-learning-glossary/ for more complete glossaries).

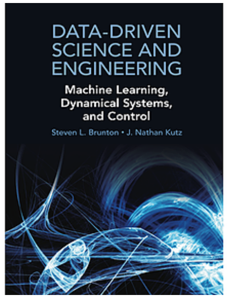
In the following sections we review the development of machine learning in the geoscientific context, and highlight how deep learning—that is, the automatic extraction of abstract (spatio-temporal) features—has the potential to overcome many of the limitations that have, until now, hindered a more wide-spread adoption of machine learning. We further lay out the most promising but also challenging approaches in combining machine learning with physical modelling.

State-of-the-art geoscientific machine learning

Machine learning is now a successful part of several research-driven and operational geoscientific processing schemes, addressing the atmosphere, the land surface and the ocean, and has co-evolved with data availability over the past decade. Early landmarks in classification of land cover and clouds emerged almost 30 years ago through the coincidence of high-resolution satellite data and the first revival of neural networks[6,7]. Most major machine learning methodological

[1]Department of Biogeochemical Integration, Max Planck Institute for Biogeochemistry, Jena, Germany. [2]Michael-Stifel-Center Jena for Data-driven and Simulation Science, Jena, Germany. [3]Image Processing Laboratory (IPL), University of València, Valencia, Spain. [4]Max Planck Institute for Meteorology, Hamburg, Germany. [5]Computer Vision Group, Computer Science, Friedrich Schiller University, Jena, Germany. [6]CENSE, Departamento de Ciências e Engenharia do Ambiente, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, Lisbon, Portugal. [7]National Energy Research Supercomputing Center, Lawrence Berkeley National Laboratory, Berkeley, CA, USA. *e-mail: mreichstein@bgc-jena.mpg.de

Learning Differential Equations from Data



- Data driven discovery of PDEs – (Rudy et al. 2017) - Sparse linear regression
- Deep Hidden Physics models (Raissi 2018) - Neural Networks
- PDE-Nets (Long 2018) - Neural Networks

Objectives

- Uncover the governing equations of dynamical spatio-temporal phenomena from data
- Arguments
  - Data are ubiquitous in many domains
    - Climate, Finance, Epidemiology, ecology, etc
  - Variables or governing dynamics may be partially known or unknown so that using data may help
- Consider non linear equations of the form
  - $u_t = F(u, u_x, u_{xx}, \dots, x, \theta)$
  - Relevant variables and F may be unknown
- Objective (s)
  - Learn the unknown dynamics
  - Learn when possible the explicit form of the underlying PDE
- Current work: tests performed on data generated from PDE

Data driven discovery of PDEs– (Rudy et al. 2017) Sparse regression

- Collect observations
- Consider a library of terms, e.g. system state, derivatives, …
- Use sparse linear regression to fit observations on library terms

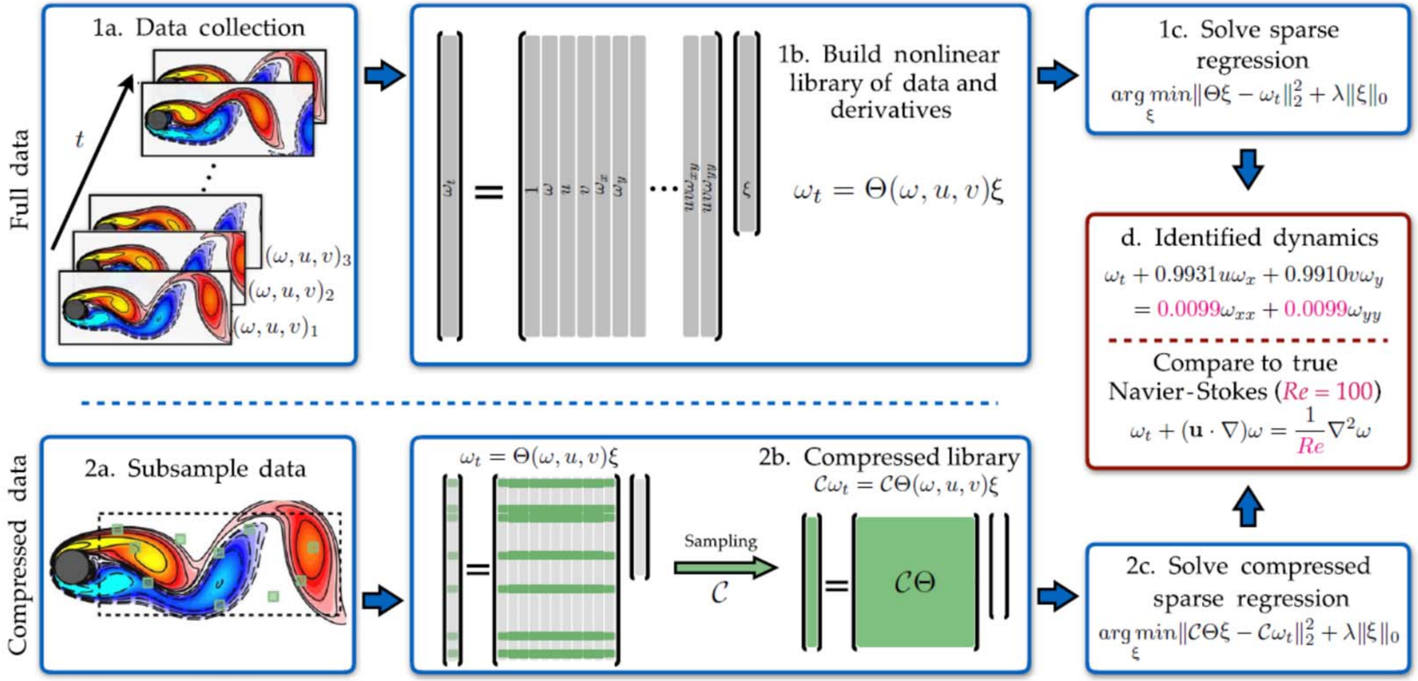
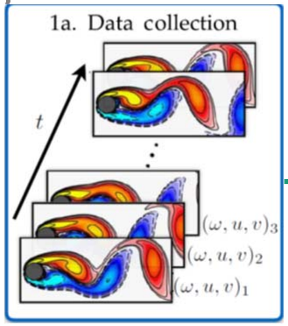


Fig. 1. Steps in the PDE functional identification of nonlinear dynamics (PDE-FIND) algorithm, applied to infer the Navier-Stokes equations from data. (1a) Data are collected as snapshots of a solution to a PDE. (1b) Numerical derivatives are taken, and data are compiled into a large matrix Θ, incorporating candidate terms for the PDE. (1c) Sparse regressions are used to identify active terms in the PDE. (2a) For large data sets, sparse sampling may be used to reduce the size of the problem. (2b) Subsampling the data set is equivalent to taking a subset of rows from the linear system in Eq. 2. (2c) An identical sparse regression problem is formed but with fewer rows. (d) Active terms in ξ are synthesized into a PDE.

Data driven discovery of PDEs– (Rudy et al. 2017) Example

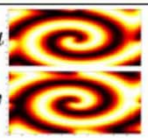
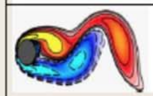


Sample (space and time)
Compute derivatives at the sampled points
Fit the data using sparse regression

- Regression
  - $u_t(x,t) =< u, u_x, u_{xx}, uu_x, \dots ; w >$ sampled at nxm points
  - w parameter vector
- Discrete form for nxm sampled points
  - $U_t = Aw$
    - with U_t : nxm vector, A : $n.m$ x $nb\ library\ terms$
- Regression loss : sparse regression
  - $\hat{w} = argmin_w \|Aw - U_t\|_2^2 + \lambda \|w\|_0$
  - Sparse regression allows finding these terms

Data driven discovery of PDEs– (Rudy et al. 2017) Examples

Table 1. Summary of regression results for a wide range of canonical models of mathematical physics. In each example, the correct model structure is identified using PDE-FIND. The spatial and temporal sampling of the numerical simulation data used for the regression is given along with the error produced in the parameters of the model for both no noise and 1% noise. In the reaction-diffusion system, 0.5% noise is used. For Navier-Stokes and reaction-diffusion, the percent of data used in subsampling is also given. NLS, nonlinear Schrödinger; KS, Kuramoto-Sivashinsky.

PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1 \pm 0.2\%, 7 \pm 5\%$	$x \in [-30, 30], n = 512,\ t \in [0, 20], m = 201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15 \pm 0.06\%, 0.8 \pm 0.6\%$	$x \in [-8, 8], n = 256,\ t \in [0, 10], m = 101$
 Schrödinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25 \pm 0.01\%, 10 \pm 7\%$	$x \in [-7.5, 7.5], n = 512,\ t \in [0, 10], m = 401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + \lvert u \rvert^2 u = 0$	$0.05 \pm 0.01\%, 3 \pm 1\%$	$x \in [-5, 5], n = 512,\ t \in [0, \pi], m = 501$
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3 \pm 1.3\%, 52 \pm 1.4\%$	$x \in [0, 100], n = 1024,\ t \in [0, 100], m = 251$
u v  Reaction Diffusion	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A^2 = u^2 + v^2, \omega = -\beta A^2, \lambda = 1 - A^2$	$0.02 \pm 0.01\%,\ 3.8 \pm 2.4\%$	$x, y \in [-10, 10], n = 256,\ t \in [0, 10], m = 201$ subsample 1.14%
 Navier-Stokes	$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2 \omega$	$1 \pm 0.2\%\ ,\ 7 \pm 6\%$	$x \in [0, 9], n_x = 449,\ y \in [0, 4], n_y = 199,$ $t \in [0, 30], m = 151$, subsample 2.22%

Data driven discovery of PDEs– (Rudy et al. 2017)

- Hypothesis
  - Full state observable
  - An overcomplete dictionary is available
    - All required terms are available
    - Non linearity requires cross product terms
  - Only a few terms will be involved in the dynamics
  - These are the conditions for sparse regression to work
- Comments
  - Interpretability
  - Requires numerical differentiation
    - Pb for large datasets
  - In the experiments noise level does not exceed 1%
  - Requires « enough » data points

Deep Hidden Physics models (Raissi 2018)

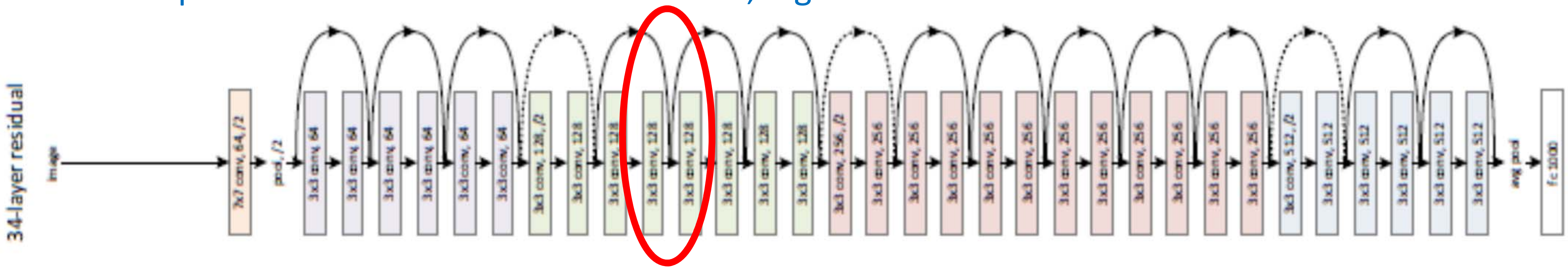
- Same objective and general idea as (Rudy 2017), uncover the dynamics of an underlying PDE
- Novelty
  - Learns a model of the data
    - Allows replacing numerical differentiation by automatic differentiation
  - Replace linear regression by a non linear neural network
- Hypothesis
  - All relevant PDE terms are provided to the model (idem Rudy 2017)
  - Full state observation is assumed (idem Rudy 2017)
  - But no need for cross terms – non linearities are taken into account by the model
- Assesment
  - No more explicit solution – how to assess the quality of the sol. ?

Deep Hidden Physics models (Raissi 2018)

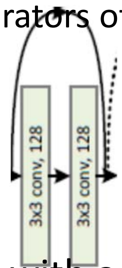
- Dataset of observations: $D = \{(t^i, x^i, u^i)\}$
- Learn model dynamics F
  - $u_t = F(u, u_x, u_{xx}, \ldots, x, \theta)$
- Learn Data model for learning the solution u
  - $u(t, x)$
    - This model will be used to provide derivatives at the sampled points
    - Derivatives computed through automatic differentiation
- Training criterion
  - $L = \sum_{i=1}^{N}(\left|u(t^i, x^i) - u^i\right|^2 + f(t^i, x^i)^2)$
    - with $\mathrm{f(x,t)} = u_t - F(t, x, u, u_x, u_{xx}, \ldots, \theta)$
    - u and f are simple non linear Neural Networks

PDE-NET (Long et al. 2018)

- Inspired from classical NN architectures, e.g. ResNet



- Each block implements
  - Convolution filters
    - Exploit local properties with learned convolutional filters
    - Here used to approximate differential operators
    - Constraints on the filters forces the approximation of differential operators of given order
  - Skip connections
    - $x_{t+1} = x_t + f(x_t)$
    - Or similar schemes used for time discretization

x_t 

$x_{t+1} = x_t + f(x_t)$

- Implement the F dynamics with a specific NN
  - Represent all the polynomials of its inputs (differential operators) with a limited number of operations

PDE-NET (Long et al. 2018)

- Considers equations of the form
  - $u_t = F(u, u_x, u_{xx}, \dots, \mathrm{x}, \theta)$
- Elementary module (δt −block) implements 1 step of forward Euler
  - $u(t + \delta t, .) \approx u(t, .) + \delta t. Net(D_{00}u, D_{01}u, D_{10}u, \dots)$
  - With
    - D_{ij} convolution operator associated to a filter approximating differential operators
      - $D_{ij}u \approx \frac{\partial^{i+j}u}{\partial^i x \partial^j y}$
    - *Net* is a specific « neural network » approximating F

PDE-NET (Long et al. 2018)

- Implementation: δt −block

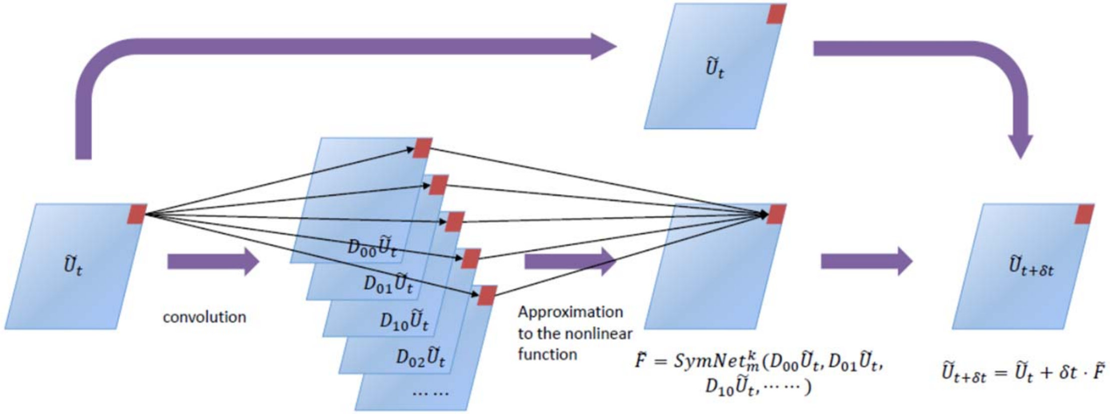


Figure 1: The schematic diagram of a δt-block.

$$u_{t+\delta t} = u_t + \delta t. F$$

- Multiple steps

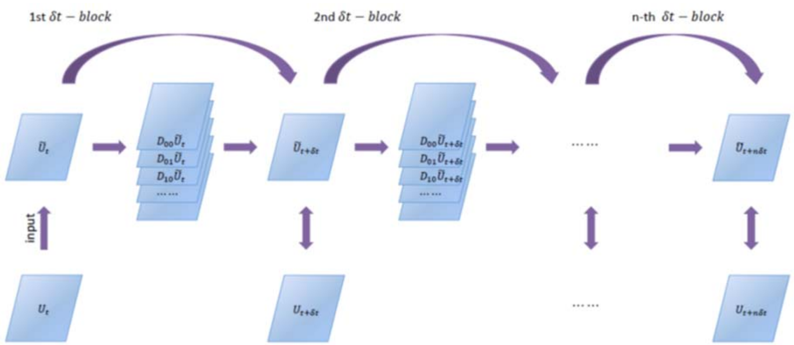
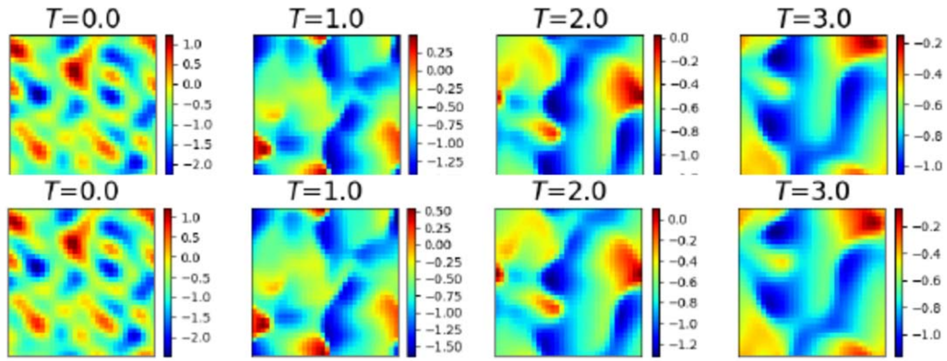


Figure 2: The schematic diagram of the PDE-Net 2.0.

PDE-NET (Long et al. 2018)

- More flexible
  - learned filters
  - Efficient implementation of polynomials
- Provides an explicit form of the underlying PDE
- e.g. 2 D Burgers
  - Only 1st component (u) shown here
    - Correct: $u_t = -uu_x - vu_y + 0.05(u_{xx} + u_{yy})$
    - Identified: $u_t = -0.98uu_x - 0.97vu_y + 0.054u_{xx} + 0.054u_{yy}$

u component
- Top: true dynamics
- Bottom: predicted, $\delta t = 0.01$



Learning from observations

Agnostic Approaches: Spatio-temporal NN for space time series forecasting, 2017

Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge, 2018

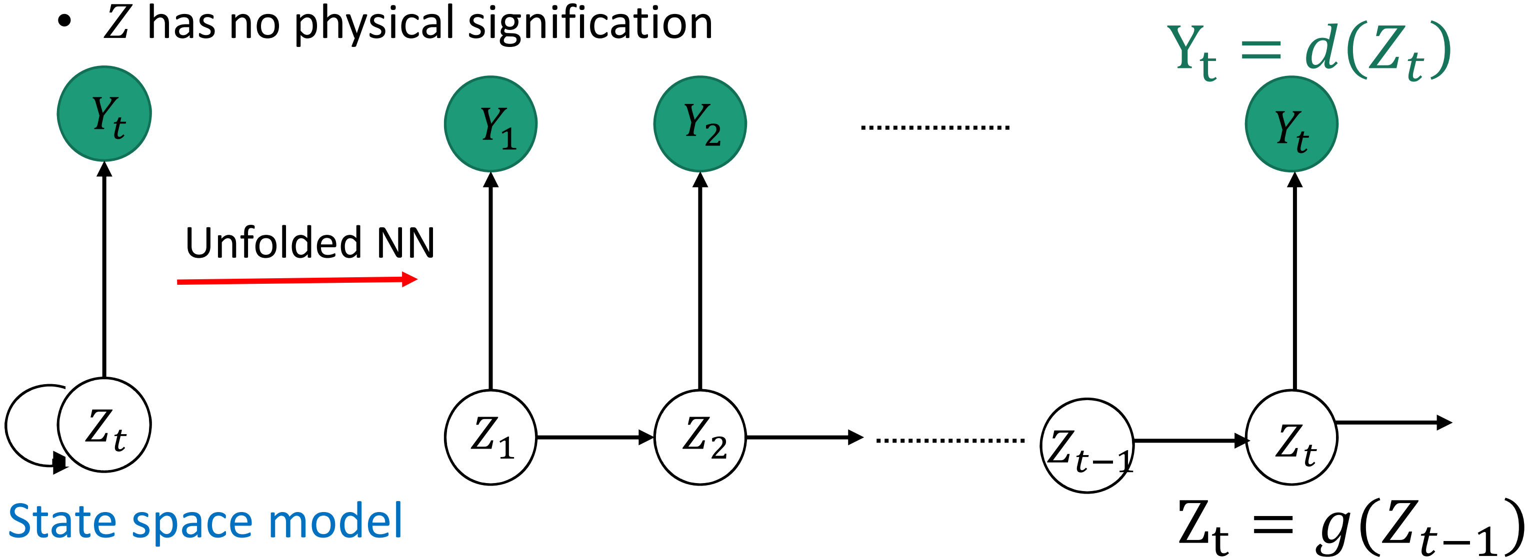
Learning Dynamical Systems from Partial Observations, 2019

Context
Spatio-temporal dynamics

- Dynamical system
  - State variables describe the evolution of the system
    - They may be known, partially known, unknown
  - Observations provide a partial knowledge on the state variables and on the system dynamics
- Objective(s)
  - Model the evolution of the system (dynamics of state variables)
  - Forecast future evolution
    - Observation level
    - State variable level

Agnostic approaches, e.g. (Ziat et al. 2018)

- Data are considered as discrete spatio-temporal time series
  - Observations: $Y_t \in R^m$, $Y_{t,i}$is the i^{th} time series
- An underlying latent dynamical process Z generates the observations:
  - $Z_t \in R^n$ is a latent representation of the process at time t
  - Z has no physical signification

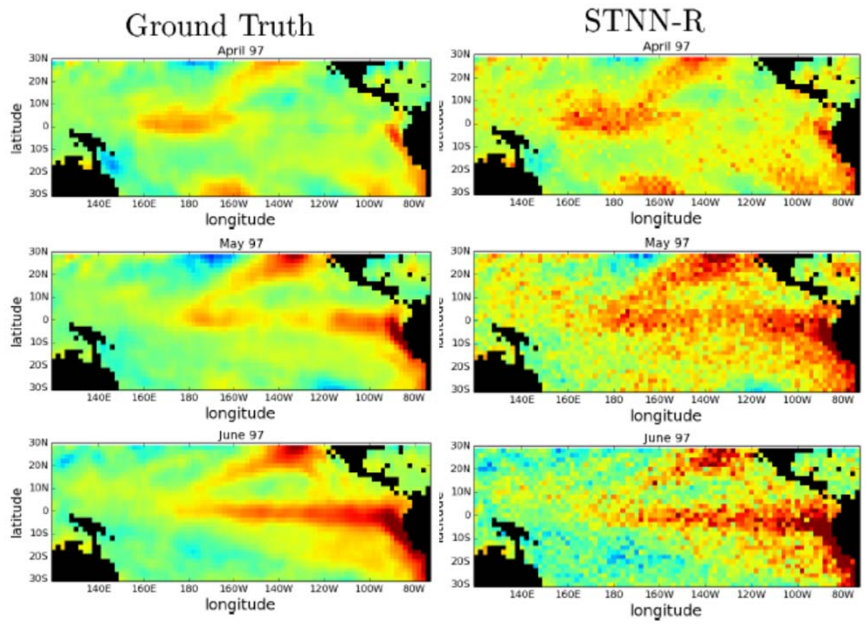


- State space model
  - $Y_t = d(Z_t) + \epsilon_X$
  - $Z_t = g(Z_{t-1}) + \epsilon_Z$

d and g: linear or non linear functions to be learned

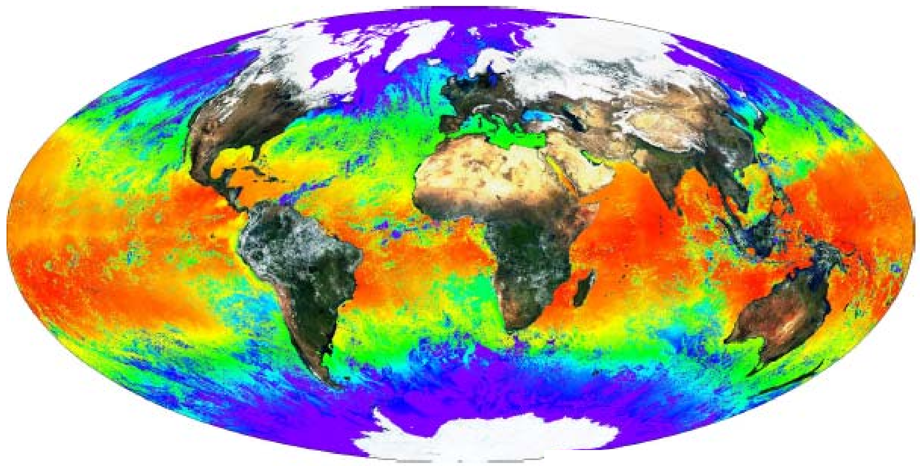
Agnostic approaches, e.g. (Ziat et al. 2018)

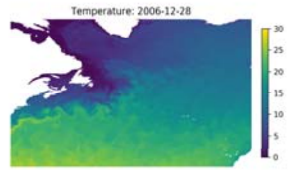
- Example
  - Pacific Sea Temperature dataset (Columbia.edu)
  - Monthly SST in Pacific
  - 2520 spatial locations
  - 395 months (1970-2003)
- Figure
  - Different time steps (3 successive months)
  - Color: actual sea temperature



Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge (de Bezenac 2018)
Example: Sea Surface Temperature Prediction - SST

- Problem: predicting SST (< 1 meter deep) on Atlantic ocean
- Data: satellite imagery (IR)
- Use cases:
  - Weather prediction, anomaly detection, component of climate models
- Classical approach
  - Data assimilation
    - Differential equations
    - Discretization
      - Finite difference
    - Model
    - Assimilation
      - Coupling with SST data
        - Adjust to initial conditions
      - Forward integration for prediction

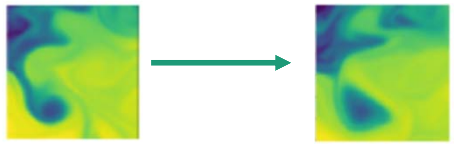




Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge (de Bezenac 2018)
Physical model: Advection – Diffusion

- Describes transport of I through **advection** and **diffusion**

$$\frac{\partial I}{\partial t} + (w.\nabla)I = D\nabla^2 I$$



  - I: quantity of interest (Temperature Image)
  - $w = \frac{\Delta x}{\Delta t}$ motion vector, D diffusion coefficient

- There exists a closed form solution
  - $I_{t+\Delta t}(x) = (k * I_{\mathrm{t}})(x - w(x))$
  - $I_{t+\Delta t}(x)$ can be obtained from I_{t} through a convolution with kernel k (pdf of a Normal distribution: $k(x - w, y) = N(y|x - w, 2D\Delta t)$)

- If we knew the motion vector w and the diffusion coefficient D we could calculate $I_{t+\Delta t}(x)$ from I_{t}
  - **w and D unknown**
  - **-> Learn w and D**

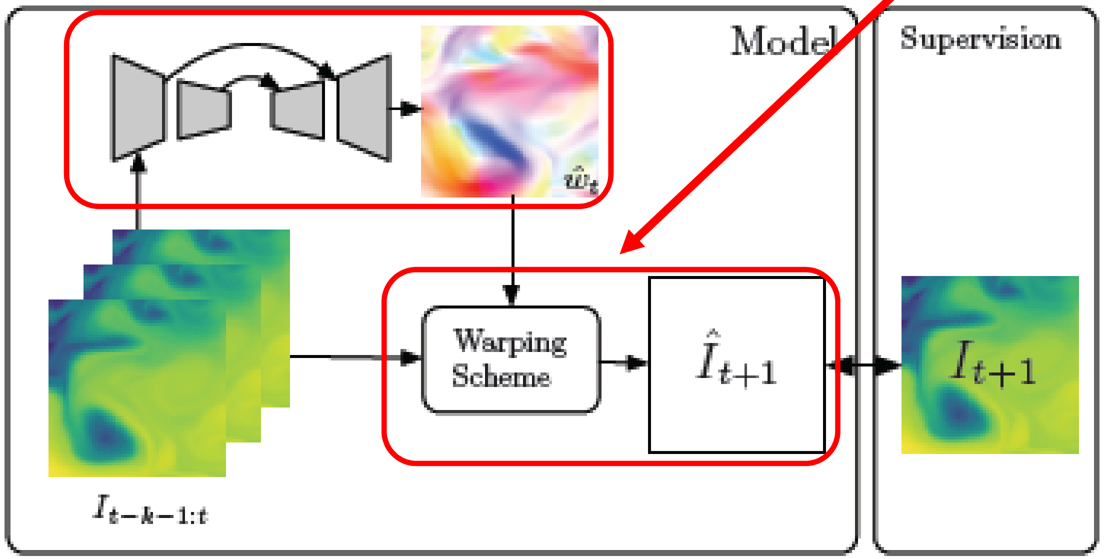
Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge (de Bezenac 2018)
Prediction Model: Objective: predict I_{t+1} from past $I_t, I_{t-1}, \ldots$

- 2 components:

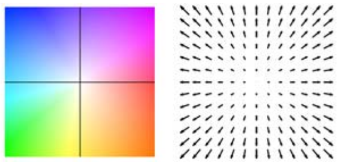
Convolution- Deconvolution NN for estimating motion vector w_t

Warping Scheme
Implements discretized A-D solution

Past Images



Target image

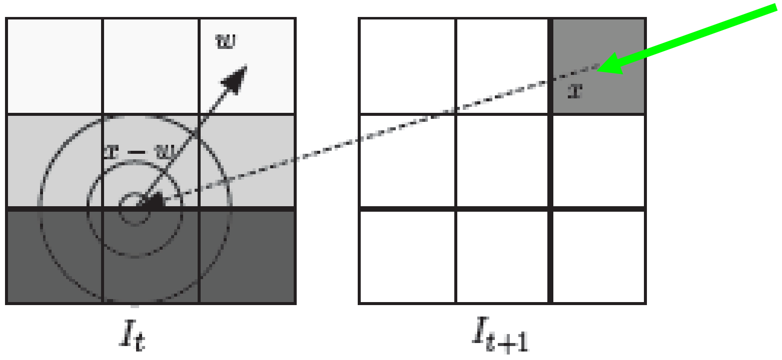


Color: orientation
Intensity: flow intensity

- End to End learning using only I_{t+1} supervision
- Stochastic gradient optimization

Deep Learning for Physical Processes: Incorporating Prior Scientific Knowledge (de Bezenac 2018)
Warping Scheme

- Motion vector $\hat{w}$ is now provided by the Conv-Deconv NN
  - $\hat{w}$ is used to compute $\hat{I}_{t+1}$ from $\hat{I}_t$
    - using the following warping scheme:



$$\hat{I}_{t+1}(x) = \sum_{y \in \Omega} k(x - \hat{w}(x), y) I_t(y)$$

Discretized version of the AD equation solution

Pixel value for time t+1 at position x, $\hat{I}_{t+1}(x)$:

1. Compute its previous position at time t, i.e. $x - w$
2. Center a Gaussian in that position in order to obtain a weight value for each pixel in I_t based on its distance with $x - w$
3. Compute a weighted average of the pixel values of I_t.

- This weighted average will correspond to the new pixel value at x in I_{t+1}

Loss function

- Prediction loss + penalty terms incorporating prior physical knowledge
- $L_t = \sum_{x \in \Omega} \rho\left(\hat{I}_{t+1}(x) - I_{t+1}(x)\right) + penalty\ terms$
  - $\rho(x) = (x + \epsilon)^{1/\alpha}$: Charbonier loss – reduces the influence of outliers compared to L_2 loss – equivalent to L_2 for $\epsilon = 0, \alpha = \frac{1}{2}$
- Penalty
  - $\lambda_{div}\left(\nabla . \widehat{w}_t(x)\right)^2 + \lambda_{magn}\|\widehat{w}_t(x)\|^2 + \lambda_{grad}\|\nabla \widehat{w}_t(x)\|^2$

    divergence magnitude gradient

Experiments

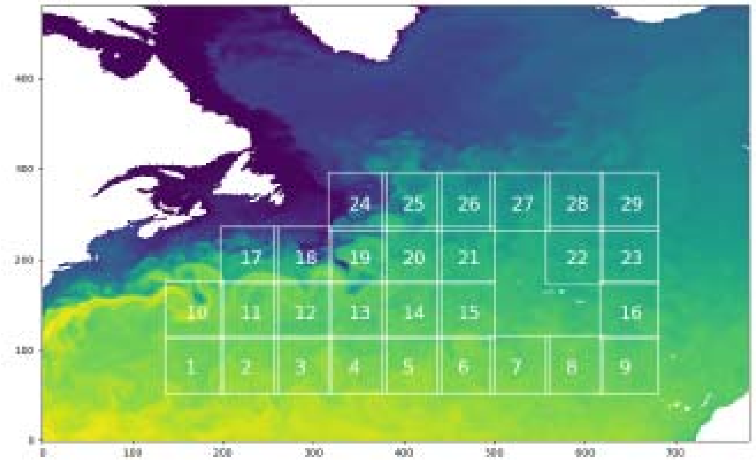


Figure 4: Sub regions extracted for the dataset. Test regions are regions 17 to 20.

- Data
  - Synthetic data from the NEMO simulator (Nucleus for European Modeling of the Ocean)
  - The generated data is based on real SST data (using reanalysis).
  - 3734 daily SST images of 481 x 781 pixels from 2006 to 2017
  - we concentrate of 64 x 64 pixel sub regions
  - we use 2006-2015 for training and validation, and the rest as test

Experiments

- Quantitative results

SOTA Numerical model

Model	Average Score (MSE)	Average Time
Numerical model Béréziat & Herlin (2015)	1.99	4.8 s
ConvLSTM Shi et al. (2015)	5.76	0.018 s
ACNN	15.84	0.54 s
GAN Video Generation (Mathieu et al. (2015))	4.73	0.096 s
Proposed model with regularization	**1.42**	0.040 s
Proposed model without regularization	2.01	0.040 s

Table 1: Average score and average time on test data. Average score is calculated using the *mean square error* metric (MSE), time is in seconds. The regularization coefficients for our model have been set using a validation set with $\lambda_{div} = 1$, $\lambda_{magn} = -0.03$ and $\lambda_{grad} = 0.4$.

SOTA Deep NN models

Proposed model

Learning Dynamical Systems from Partial Observations (Ayed et al. 2019) Problem

- Forecasting non linear dynamical systems from observations only
- Hypothesis
  - the underlying system follows an unknown differential equation
- Objective
  - Learn the evolution of this system (observations and state) with a NN
  - Discover automatically the relation between states (dynamics)
- Results
  - Cast the problem in a general control framework
  - Derive a generic algorithm
    - for solving the optimization problem
    - Correspondance with the training of a NN through an explicit ODE solver for partially observed states
  - Experiments on different spatio temporal data

Learning Dynamical Systems from Partial Observations (Ayed et al. 2019)
Setting

- Dynamical system with initial conditions
  - $$\begin{cases} X_0 \\ \frac{dX_t}{dt} = F(X_t) \\ Y_t = H(X_t) \end{cases}$$
- Variables
  - H: observation operator linking state to observation (known)
  - X : spatio temporal field – system state (unknown)
    - Hyp: X contains the quantities of interest for describing the system
    - Here X is a function of time t and space x (spatio-temporal)
    - e.g. velocity, pressure (Ocean)
  - Y : observation
    - e.g. temperature, salinity, ocean color (Ocean)

Learning Dynamical Systems from Partial Observations (Ayed et al. 2019) Model

- Use a parametrical model
  - $$\begin{cases} X_0 = g_\theta(Y_{-k}, \tilde{X}_0) \\ \frac{dX_t}{dt} = F_\theta(X_t) \\ Y_t = H(X_t) \end{cases}$$
  - Where
    - $Y_{-k} = (Y_{-k+1}, \dots, Y_0)$
    - F_θ is a smooth operator defining the trajectory of X
    - g_θ gives the initialization for X_0
- Learning
  - Learn θ by gradient
  - Approximate the gradient using an adjoint method

Learning Dynamical Systems from Partial Observations (Ayed et al. 2019)
Optimization problem

- Loss function
  - $J(Y, \tilde{Y}) = \int_0^T \left\| Y_t - \tilde{Y}_t \right\|^2 dt$
- Learning problem
  - Minimize $\mathrm{E}_{\mathrm{Y\ observed}}[J(Y, H(X))]$
  - subject to constraints: $\frac{dX_t}{dt} = F_\theta(X_t)$
    $X_0 = g_\theta(Y_{-k}, \tilde{X}_0)$
- Implementation
  - F_θ is implemented as a residual network
  - g_θ is a Unet (for the NEMO experiments)
  - An explicit Euler scheme with a fixed step-size is used to solve the forward equation $\frac{dX_t}{dt} = F_\theta(X_t)$: $X^\theta_{t+\delta t} = X^\theta_{t+\delta t} + \delta t F_\theta(X^\theta_t)$

Datasets

- Euler equations
  - Navier Stokes equations when neglecting the viscosity term
  - Discretised on a spatial 64x64 grid
  - State: fluid particle density + velocity
  - Observations: density only
  - Initial state: true full state X_0
- NEMO – Nucleus for European Modelling of the Ocean Engine
  - Observations: Sea Surface Temperature
    - Daily temperatures 2006-2015
  - State: multiple variables, we make use only of 2 variables corresponding to the velocity field
  - Initial state: interpolated from previous observations
- For all test, data are partitioned into a training and a test set
  - 6 time steps used for the target sequence for training

Euler equations

-

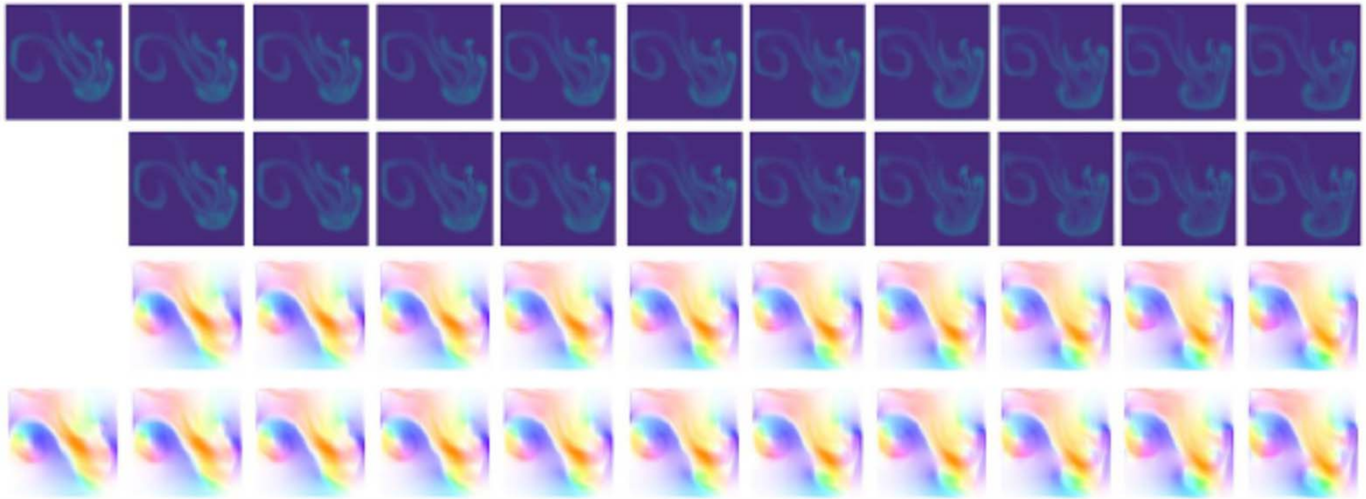
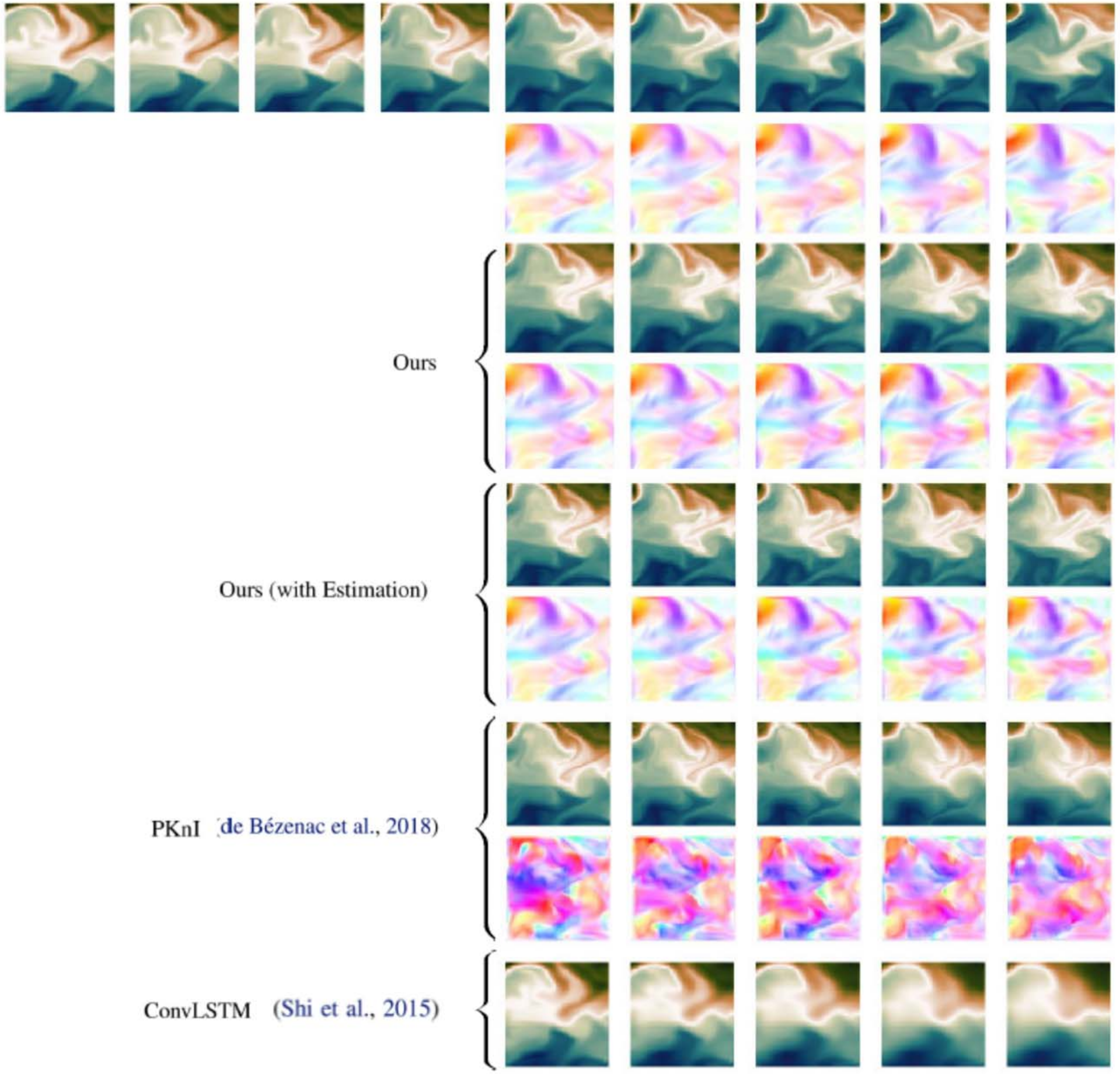


Figure 3. Forecasting the Euler equations on the test set. From top to bottom: input and target observations, model output, model hidden state, hidden state input and ground truth.

NEMO – Global Ocean Physics Reanalysis

-



Targets (Y_t)

Targets (X_t)

Predictions ($\hat{Y}_t$)

Predictions ($\hat{X}_t$)

Conclusion

- Several emerging topics at the crossroad of NNs and dynamical systems
- Open several perspectives both for statistical machine learning and for physical modeling

  - Cross fertilization of model based approaches (physics) and data driven approaches
  - New models for describing complex dynamics exploiting the large amounts of observation data
  - New perspectives for modeling/ training neural networks
    - e.g. as continuous dynamical systems
    - NN models may be interpreted as discretization schemes for dynamical systems

References used in the presentation

- Ayed I., de Bezenac E., Pajot A., Brajard J. , Gallinari P. , (2019), Learning Dynamical Systems from Partial Observations, arXiv:1902.11136
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481–2495.
- de Bezenac, E., Pajot, A., & Gallinari, P. (2018). Deep Learning For Physical Processes: Incorporating Prior Scientific Knowledge. In *ICLR*.
- Karras, T., Laine, S., & Aila, T. (2018). A Style-Based Generator Architecture for Generative Adversarial Networks. Retrieved from http://arxiv.org/abs/1812.04948
- Long, Z., Lu, Y., & Dong, B. (2018). PDE-Net 2.0: Learning PDEs from Data with A Numeric-Symbolic Hybrid Deep Network. https://doi.org/arXiv:1812.04426
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2018). Language Models are Unsupervised Multitask Learners.
- Raissi, M. (2018). Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. *Journal of Machine Learning Research*, *19*, 1–24.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In CVPR (pp. 779–788).
- Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *SCIENCE ADVANCES*, *3* no. *4*(April).
- Sirignano, Justin & Spiliopoulos, Konstantinos. (2017). DGM: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics. 375.
- Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, Technical Report, 2016.
- Ziat, A., Delasalles, E., Denoyer, L., & Gallinari, P. (2017). Spatio-temporal neural networks for space-time series forecasting and relations discovery. *Proceedings - IEEE International Conference on Data Mining, ICDM*, *2017–Nov*, 705–714.