Leveraging knowledge to design machine learning despite the lack of data. Transfer Learning and PINNS models

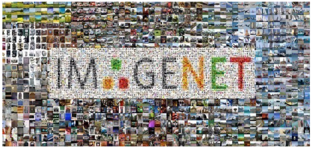
Mathilde Mougeot
ENSIIE & Centre Borelli, ENS Paris-Saclay, France.

MIRES workshop, March 28^{th} 2024

Motivation. In recent years, considerable progress has been made in the implementation of decision support procedures based on machine learning methods through the exploitation of very large databases and the use of learning algorithms. In the industrial environment, the databases available in research and development or in production are rarely so voluminous and the question arises as to whether in this context it is reasonable to want to develop powerful tools based on artificial learning techniques. This talk presents research work around transfer learning and hybrid models that use knowledge from related application domains or physics to implement efficient models with an economy of data. Several achievements in industrial collaborations will be presented that successfully use these learning models to design machine learning for industrial small data regimes and to develop powerful decision support tools even in cases where the initial data volume is limited.

Data Sources & several Successes of "ML/AI" models

- Imagenet is a huge open source database containing more than **14.**10^6 labeled images for 10^3 **categories**, available for object detection and image classification at a large scale, ... "quite expensive" labeling effort.



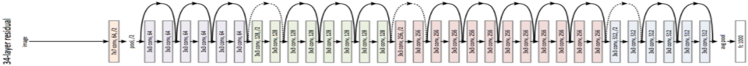
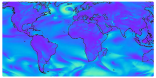


FIGURE – ResNet : a Convolutional Neural Network for image classification (credit :Resnet)

Top-performing deep architectures are trained on massive amounts of labeled data.

- DeepL relied on the hudge French-English Linguee dictionnary.

- GraphCast Weather forecast



Classical Backbone for Supervised Machine learning App.

1. Input/ output (X, Y) (Features, labels set) defined by the operational need.
   Ex : $X \in \mathbb{R}^d$, ... $Y \in \mathbb{R}$, ... $Y \in \{0, 1\}$...
2. Data set. $S = \{(x_i, y_i)\}_{i=1}^m \sim \mathcal{D}^m$ a learning/training sample of m iid pairs.
   with $\mathcal{D}$ an unknown joint probability distribution on the product space $X \otimes Y$
3. Model $\mathcal{H} = \{h_\theta | h_\theta : X \to Y\}$ a hypothesis class, θ parameter
   classifiers or regressors depending on the nature of Y.
4. Loss function $\ell(y, h_\theta(x))$ providing a cost of $h_\theta(x)$ deviating form the true output $y \in Y$.

The best hypothesis is the one that minimizes the true risk, consequently, generalizes well :

$$R_{\mathcal{D}}{}^{\ell}(h_\theta) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(h_\theta(x), y)]$$

The goal of learner consists of finding a good hypothesis function $h_\theta \in \mathcal{H}$ that captures in the best way possible the relationship between X and Y.

$$h_{\theta_{\text{opt}}} = \arg\min_{h_\theta} R_{\mathcal{D}}{}^{\ell}(h_\theta)$$

In pratice : Empirical risk **optimization**, large training sample, regularization, sparsity,...

From Industrial needs

to Transfer Learning

Motivation 1. ML for Automatic Elderly fall detection.

Objective.

The *Tarkett Floor in motion application* tends to detect automatically falls based on sensor information and then trigger an alarm if necessary.

From a first Proof Of Concept (POC) to deployment :

1. Data. As it is not possible to gather large data base with falls of elderly people, a first supervised data base is created with young volunteers containing fall / no fall events.
2. Predictive models. POC to choose and evaluate the performance of a ML model to detect fall on previous data (performances ? true detection, false alarm...).
3. Transfer learning. How to transfer the previous model for elderly care...to a new population given few labeled data ?
4. Budgeted learning. ... and what about in a real environment...

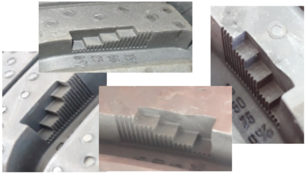
[Minvielle et al., 2017], [Minvielle et al., 2019], [Mounir et al., 2021]

Motivation 2. ML for Automatic tire wear detection

AI IdF 2019 Challenge organized by the IDAML chair in collaboration with Michelin

Industrial objectives :
Design an application for the
1/ detection and localization in an image of a "new generation" wear indicator
2/ Estimation of the wear level

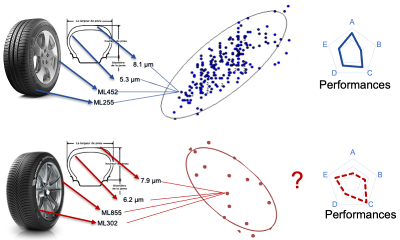
Data base : 1000 tire images with
-various tire views,
-different lighting conditions,
-with and without wear indicator (4 levels).
Learning : 500 labeled examples
(tire images, wear indicator, boundingbox posiiton)
→ Blind Evaluation : 500 labeled images.



Motivation 3. ML for Product Design.

Industrial application in collaboration with Michelin, EDF

- New products are regularly manufactured with a long and costly development.
- Relative small data sets are gathered during the development of products as characteristics (color, shape, weight...) and performances.



Is-it possible to predict the performances of a new tire line given data previously gathered from **other** lines ?

[Richard et al., 2021], [de Mathelin et al., 2021]

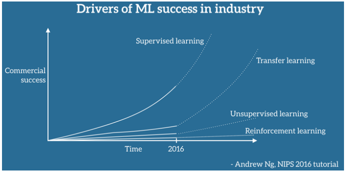
Machine Learning in the industry

Main observations :

- Often small, moderate, evolving database. Ex. manufacturing process.
- Few or not labeled data. Ex. Few production defaults.
- labeled-data is often difficult and time-consuming to acquire.
  Ex. Experimental design to help selecting costly observation outputs.
- In many real-world applications, historical (training) data and newly collected (test) data may often exhibit different statistical characteristics.
- In many ML scenarios, training and test samples are supposed to be generated by the same (unknown) probability distribution.
- Needs for monitoring and diagnosis based on machine learning (ML) .
- Makes sense to re-use knowledge gained from related but distinct datasets.

Need of Transfer Learning, domain adaptation, few shot learning...

Transfer learning : the model can be pre-trained on data from a specific domain and then adapted to meet needs of a given task.



Transfer learning in industry.

The Transfer learning framework

- **Data collections : Source & Target**
  1. **Source data** $\mathcal{S}$.
     $X_{\mathcal{S}} \otimes Y_{\mathcal{S}}$ the source input and output spaces associated with $\mathcal{S}$
     $\mathcal{S}_X$ the marginal distribution of $X_{\mathcal{S}}$, $t_{\mathcal{S}}$ the source learning task
  2. Target data $\mathcal{T}$
     $X_{\mathcal{T}} \otimes Y_{\mathcal{T}}$ the Target input and output spaces associated with $\mathcal{T}$
     $\mathcal{T}_X$ the marginal distribution of $X_{\mathcal{T}}$, $t_{\mathcal{T}}$ the Target learning task

  Δ Source and Target data are not drawn from the same distribution.
- **Focus** on the Target Risk. $R_{\mathcal{T}}{}^{\ell}(h) = \mathbb{E}_{(x,y)\sim\mathcal{T}}[\ell(h(x), y)]$ with ℓ the loss function.
- **Supervised data or calibrated Model** available for the source domain (enough data).

Transfer learning aims to improve the learning of the target predictive function :
$f_{\mathcal{T}} : X_{\mathcal{T}} \rightarrow Y_{\mathcal{T}}$ for $t_{\mathcal{T}}$ using knowledge gained from $\mathcal{S}$ where $\mathcal{S} \neq \mathcal{T}$

$\mathcal{S} \neq \mathcal{T}$ (joint distributions) implies several cases :

- $\mathcal{S}_{\mathcal{X}} \neq \mathcal{T}_{\mathcal{X}}$ i.e. $X_{\mathcal{S}} \neq X_{\mathcal{T}}$ (spaces) or $\mathcal{S}_X(X) \neq \mathcal{T}_X(X)$ (laws) or
- $t_{\mathcal{X}} \neq t_{\mathcal{T}}$ (i.e. $Y_{\mathcal{S}} \neq Y_{\mathcal{T}}$ (target task) or $\mathcal{S}(Y/X) \neq \mathcal{T}(Y/X)$ (conditional law)

... Seems to be a hard problem...
Success stories ?... Theoretical guaranties ? Assumptions ?, Negative transfer ? ....
Answers to the industrial partners.... open source algorithms...

Illustration of the Need of Transfer for Learning Machine

Transfer learning aims at providing ML models with a good generalization capability on a Target domain (same domains, different domains).

Target domain (ex I).
Same Domain $\{\mathcal{X}, P(X)\}$ &
task $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$.

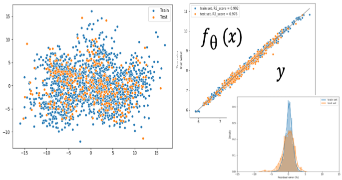


FIGURE – High Prediction capability.

Target domain (ex II).
Different Domain & same task.

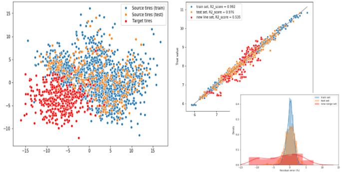


FIGURE – Low Prediction capability (no transfer at this stage)

$P(x, y)$ Joint distribution differences	=	$P(y/x)$ concept shift	*	$P(x)$ covariate-shift

Transfer Learning & Domain adaptation Methods

Several approaches to transfer knowledge from Source to Target domain.

- **Model-based.** Transfer the model parameters learnt on the source data to the target model. - Train model available, not necessary the source data- .
- **Feature-based.** Find a new representation space to bring feature spaces closer. -Source and Target Input data available-.
- **Instance-based.** Re-weight the source samples to bring the distributions closer. -Source and Target Input data available-.

Theoretical guarantees ?

For exemple on the Target Risk given the source risk.

Exemples of Industrial needs and success stories.

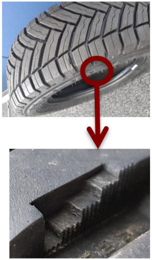
- **Model-based :** Image based tire wear estimation based on Deep architecture (Michelin) (Resnet...), Automatic fall detection based on decision trees/ RF (Tarkett).
- **Feature-based :** Domain adversarial neural networks (EDF, Michelin)
- **Instance-based :** Multi-source domain adaptations for Product design (Michelin) or Electricity prediction (EDF)

Transfer learning
Model-based

Transfer learning
Feature-based

Transfer learning
Instance-based

Model-based Transfer learning. Ex1 : deep NN



Industrial Image classification
Automatic tire wear detection, IdF AI Challenge, 2019.

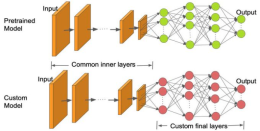
Data base :

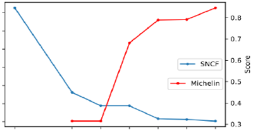
1000 tire images with various tire views, different lighting conditions, with and without wear indicator.
500 images for learning/ 500 images for a blind evaluation.

Two following questions were addressed :
1/ detection and localization in an image of a "new generation" wear indicator
2/ Estimation of the wear level

Development based on Transfer learning

Poor performances obtained with trained model using only the tire data base (20%).
A source pre-trained model (RetinaNet, Yolo...) is used by the candidates (final perf 85%).



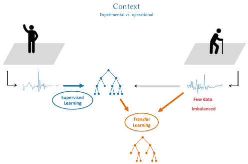


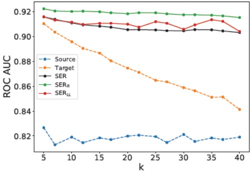
Predicted Class	True Class				
	N/I	25%	50%	75%	100%
N/I	215	0	0	0	2
25%	1	31	2	0	0
50%	0	9	32	1	3
75%	1	0	7	53	8
100%	0	0	0	5	59

FIGURE – Pre-trained model, first frozen weights (credit learnopencv.com)

Model-based Transfer learning. Ex2 : decision trees

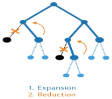
Fall detection. Strong benefits for transfering knowledge from Source to Target :



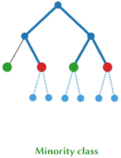


Segev et al. 2017.
SER : Structure Expansion and Reduction

SER has to be adapted to take into account class imbalance (few falls) with conditional reduction [Minvielle et al., 2019]



- Idea : train on source domain, extend on target domain the actives nodes, then cut the inactives edges.



- Expansion left unchanged
- Reduction constrained

SER$_R$	SER$_{LL}$
If node is of minority class, then no pruning	If node is of minority class **and** still significant considering Target **and** $R_L > 0.5$, then no pruning

- Idea : preserve nodes form minority class

Transfer learning
Model-based

Transfer learning
Feature-based

Transfer learning
Instance-based

Feature-based TL.

Deep network to confuse source and target input feature data...
Domain Adversarial Neural Networks. [Ganin and Lempitsky, 2015].

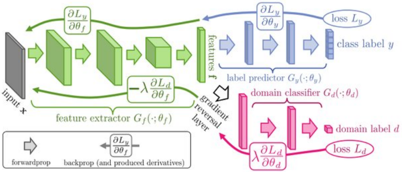


FIGURE – credit [Ganin and Lempitsky, 2015].

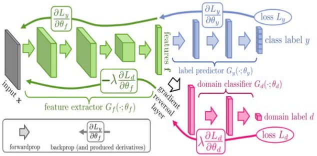
DANN : A neural net architecture and an optimization process to solve both

1. Supervised Task based on Source data to learn the model using an iid sample $\{(x_1, y_1), \ldots, (x_n, y_n)\} \sim (P(X, Y))^n$, $\hat{h} = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \ell(h(x_i), y_i)$
2. Unsupervised Domain Adaptation using Source and Target inputs to minimize a distance characterizing the domain divergence.

Feature-based TL.

Domain Adversarial Neural Networks.

[Ganin and Lempitsky, 2015]



Source obs $i : (x_i, y_i, d_i = 0)$
Label obs $i : (x_i, d_i = 1)$
L_y / L_d : label/ domain loss.
Optimization criteria :

$$E(\theta_f; \theta_y, \theta_d) = \sum_{\substack{i=1\dots N \\ d_i=0}} L_y(G_y(G_f(x_i; \theta_f); \theta_y), y_i) - \lambda \sum_{i=1\dots N} L_d(G_d(G_f(x_i; \theta_f); \theta_d), d_i)$$

The backpropagation optimisation procedure aims to compute the parameters $(\theta_f; \theta_y, \theta_d)$ such that

$$(\hat{\theta}_f; \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f; \theta_y, \hat{\theta}_d)$$

$$(\hat{\theta}_d) = \arg\max_{\theta_d} E(\hat{\theta}_f; \hat{\theta}_y, \theta_d)$$

Stochastic updates with learning rate μ

$$\theta_f \leftarrow \theta_f - \mu\left[\frac{\partial L_y^i}{\partial \theta_f} - \lambda \frac{\partial L_d^i}{\partial \theta_f}\right]$$

$$\theta_y \leftarrow \theta_y - \mu \frac{\partial L_y^i}{\partial \theta_y}$$

$$\theta_d \leftarrow \theta_d - \mu \frac{\partial L_d^i}{\partial \theta_d}$$

Transfer learning
Model-based

Transfer learning
Feature-based

Transfer learning
Instance-based

Instance-based TL.

The risk computed on the Target may be related to the risk on the Source domain.

$$
\begin{aligned}
R^{\ell}_{\mathcal{T}}(h) &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{T}} \ell(h(x),y) = \int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathcal{T}(x,y)\,\ell(h(x),y)dxdy \\
&= \int_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \frac{\mathcal{T}(x,y)}{\mathcal{S}(x,y)}\mathcal{S}(x,y)\ell(h(x),y)dxdy \\
&= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{S}} \left[w(x,y)\ell(h(x),y)\right] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{S}} \left[\frac{\mathcal{T}_X(x)\mathcal{T}(y/x)}{\mathcal{S}_X(x)\mathcal{S}(y/x)}\ell(h(x),y)\right]
\end{aligned}
$$

Rem : The support of $\mathcal{T}_X$ is contained in the support of $\mathcal{S}_X$, $\mathcal{S}(x,y) > 0$.

Instance-based TL.

The covariate shift assumption. The predictive dependency remains unchanged between Source and Target while the marginal distributions change.

Covariate shift assumption $\begin{cases} \mathcal{S}(Y/X) = \mathcal{T}(Y/X) \\ \mathcal{T}_{\mathbf{X}}(\mathbf{X}) \neq \mathcal{S}_{\mathbf{X}}(\mathbf{X}) \end{cases}$

$$
\begin{aligned}
R^{\ell}_{\mathcal{T}}(h) &= \underset{(x,y)\sim\mathcal{T}}{E} \ell(h(x), y) \\
&= \underset{(x,y)\sim\mathcal{S}}{E} \left[\frac{\mathcal{T}_X(x)\mathcal{T}(y/x)}{\mathcal{S}_X(x)\mathcal{S}(y/x)}\right] \ell(h(x), y) \\
&= \underset{(x,y)\sim\mathcal{S}}{E} \left[\frac{\mathcal{T}_X(x)}{\mathcal{S}_X(x)}\right] \ell(h(x), y)
\end{aligned}
$$

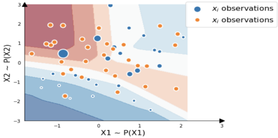


Figure – **Importance Weighting** Source (blue) and target (orange) input samples are drawn according to two different distributions $p_s(x), p_t(x)$. The source samples are reweighted according to the density ratio $w(x) = p_t(x)/p_s(x)$

Mixing strategies (Feature-Instance based)

Unsupervised Multi-source domain adaptation for regression

Application : Non intrusive load monitoring. From the house consumption, estimation of the consumption of an appliance over a period of time.

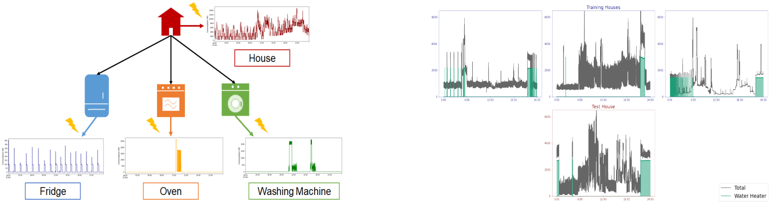


FIGURE – Water heater consumption estimation : input is the whole consumption (gray curve 2s sampling), variable to predict is the whole Water Heater consumption, $y \in \mathbb{R}$ (green area)

Theoretical setup for domain adaptation

BenDavid et al. introduced in 2006 the $\mathcal{H}$-divergence for 01 loss function, in the setting of binary classification ($\ell_{01}(h(x), y) = 1$ if $h(x) = y$; otherwise 0)

- Given two domain distributions $\mathcal{D}_S^X$ and $\mathcal{D}_T^X$ over X, and a hypothesis class $\mathcal{H}$, the $\mathcal{H}$-divergence between $\mathcal{D}_S^X$ and $\mathcal{D}_T^X$ for classification is defined by :

$$\mathrm{d}_{\mathcal{H}}(\mathcal{D}_S^X, \mathcal{D}_T^X) = 2 \sup_{h \in \mathcal{H}} \left| \Pr_{x \sim \mathcal{D}_S^X} [h(x) = 1] - \Pr_{x \sim \mathcal{D}_T^X} [h(x) = 1] \right|$$

- The $\mathcal{H}$-divergence relies on the capacity of the hypothesis class $\mathcal{H}$ to distinguish between examples generated by $\mathcal{D}_S^X$ from examples generated by $\mathcal{D}_T^X$.

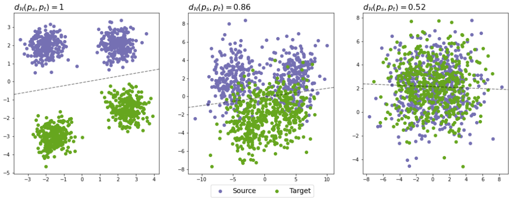


FIGURE – Divergence/discrepancy illustration with linear classifiers. [Richard et al., 2021]

Theoretical setup for domain adaptation

The discrepancy introduced by Ben David et al. 2007, Mansour et al. 2009 measures the availability to discriminate between Source and Target input features distribution.

- Considering two labeling functions f, g and the symmetric loss ℓ over pairs of labels which obeys the triangle inequality.

  The expected loss over any marginal distribution$Q$ is defined by :

$$L_Q(f, g) = \mathbb{E}_Q(\ell(f(X), g(X)))$$

- Consider a hypothesis class $\mathcal{H}$ and
  the marginal distributions S on source domain and T on target domain,
  the discrepancy distance between these two is defined as :

$$\mathrm{disc}_{\mathcal{H},L}(S, T) = \sup_{h,h' \in \mathcal{H}} \left| L_S(h, h') - L_T(h, h') \right|$$

Domain adaptation bound

Mansour et al. 2009 established a bound for the Target risk using the discrepancy :

$$\mathbf{R}_T(h) \leq \mathbf{R}_S(h, h_S^*) + \mathrm{disc}_{\mathcal{H},\ell}(S, T) + \lambda$$

where
$R_Q(h) = \mathbb{E}_Q(\ell(h(X), Y))$,
$R_Q(h, h') = L_Q(h, h') = \mathbb{E}_Q(\ell(h(x), h'(x)))$, $h, h' \in \mathcal{H}$

$h_S^* = \arg\min_{h \in \mathbf{H}} \mathbf{R}_S(h)$, $h_T^* = \arg\min_{h \in \mathcal{H}} \mathbf{R}_T(h)$,
ideal hypothesis for Source and Target domain.

$\lambda = \mathbf{R}_S(h_T^*) + \mathbf{L}_T(h_S^*, h_T^*)$

Comments

- **First term :** source risk, can be minimized with source labels
- **Second term :** discrepancy between domains $\rightarrow$ to minimize!
- **Third term :** risk of the ideal hypothesis on the source and target samples. Assumed to be small and not controlled in unsupervised DA.

Leveraging Knowledge to design Machine Learning.

Knowledge in ML modeling

Data augmentation and tailored knowledge...

Extra Prior knowledge can provide rich information not existing or hard to extract in limited training data and helps improve the data efficiency, the ability to generalize, and the plausibility of resulting models.

- Data augmentation.

Easy for Image classification tasks (symmetry, rotation, texture transformations).

- Tailored knowledge [Features, architecture, function properties]

1. **Feature engineering.** $x_{raw} \rightarrow x \rightarrow f_\theta(x) \sim y$
   Ex. Wavelet based scattering transform, Fourier transform.
   Ex : sounds classification for Delphin challenge classification (frequential data).
2. **Design of specialized NN architecture** associated with a given predictive task. Symmetry groups as rotation, homothety, translation may implement an intrinsic geometry of f_θ
   $x \rightarrow f_\theta(x) \sim y$
   Ex : Convolutional NN [CNN] by craflty respecting invariance along the groups of symmetries
3. **Multi-Task Learning**
   Introduction of knowledge in the cost function, in the optimisation process. Ex : Physical Informed Neural Network

Machine Learning models for physics

Surrogate models

Surrogate models

The supervised data driven approach :

1. Data set provided by experimental design or sampling.
   $\mathcal{D}_n = \{(x_i, y_i),\ 1 \leq i \leq n,\ \text{input } x_i \in \mathcal{X},\ \text{output} y_i \in \mathcal{Y}\}$
2. The Model $f_\theta : f_\theta(x) \to y$
   Parametric model : Gaussian Process....
   Non Parametric models : NN, RF, GradBoost...
3. Calibration by optimisation given the data $\mathcal{D}_n$
   Example : ℓ_2 Loss function $\mathcal{L}_{\text{data}} = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} (f_\theta(x_i) - y_i))^2$

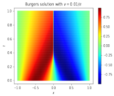
Surrogate/ Meta models approximate the input/output relation.
Several techniques have been developed such as :

- Reduced Order Models (ROM) which reduce the order of the model in an unsupervised manner like Principal Component Analysis (PCA).
- Data fit models which create a fit between input and output models based on simulation data as for example polynomial basis, radial basis function, Gaussian Process, stochastic polynomial chaos expansion.
- Machine learning, Deep Neural Networks models which are known to need large data set.

ML surrogate model

Illustration with Deep Neural Network on a toy example.

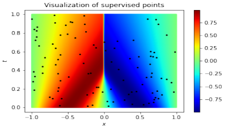
Burger's equation :input (x space , t *time*), output : speed of the fluid given parameter : ν viscosity. Dirichlet boundary condition : $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0$; $u(0,x) = -sin(\pi x)$; $u(t,-1) = u(t,1) = 0$



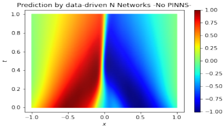
Neural network Surrogate model.

f_θ trained with $n = 100$ supervised observations. NN architecture : 2-4 (x50)-1. Evaluation on a Test data set, regular grid of points (256 (x) , 100 (t)). $E(u,\hat{u}) = ||u-\hat{u}||_2^2/||u||_2^2$ on a grid (256 (x) , 100 (t))

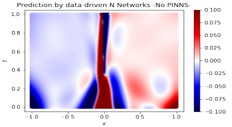
Supervised points



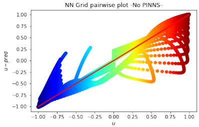
Surrogate model $E(u,\hat{u}) = 0.17$



Error



Pairwise Plot

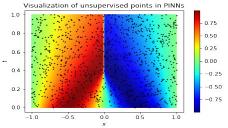


ML Surrogate model

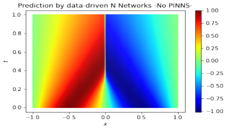
Illustration with Deep Neural Network on a toy example.

$n = 1000$ supervised observations.
Model f_θ : neural network. NN architecture : 2-4 (x50)-1.
Evaluation on a Test data set, regular grid of points (256 (x) , 100 (t)).

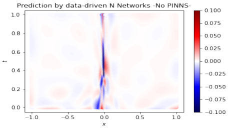
Supervised points



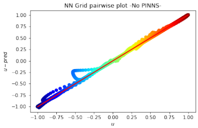
Surrogate model $E(y, \hat{y}) = 0.02$



Error



Pairwise Plot

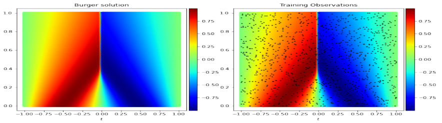


ML Surrogate model

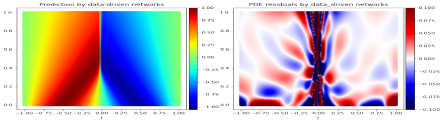
Illustration with Deep Neural Network on a toy example.
Focus on the Pde errors/residuals computed for Burger model (left/bottom figure) :

$N_{\text{data}} = 1000$, $N_{\text{colloc}=0}$, NN architecture : 2-4 (x50)-1. 50 000 epoch, Adam optimizer.

$E(u, \hat{u}) = ||u - \hat{u}||_2^2 / ||u||_2^2$ on a grid (256 (x) , 100 (t))



PINNs model $E(u, \hat{u}) = 0.0163$



Conclusions :

$\rightarrow$The pde constraints are not respected...

$\rightarrow$ The NN model mimics the input/output relation but the underlaying physics is not caught.

ML Surrogate model

First conclusions :

- Especially, in a "small data regime", the vast majority of state-of-the-art machine learning techniques are lacking robustness and fail to "model" the underlaying physics phenomena (pde constraints not respected).
- The cost of data acquisition may be prohibitive.
  Experimental design are proposed to chose the observations.
- We are inevitably faced with the challenge of drawings conclusions and making decision under partial information.

An academic- industrial joint work on Transfer Learning, Hybrid models, Generative models...

thanks to.

Industrial Data Analytics & Machine Learning

with

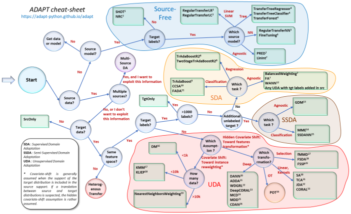
- Antoine de Mathelin, Towards reliable machine learning under domain shift and costly labeling, with applications to engineering design Michelin & IDAML, Centre Borelli
- Khoa Nguyen, Development and assessment of physically informed learning methods : enhancement of multi-physical simulation in industrial contexts, CEA, Michelin & IDAML, Centre Borelli
- Fouad Oubari, Deep Generative design for Industrial Products Michelin & IDAML, Centre Borelli
- Michelin Team since 2018. R. Décatoire, F. Deheeger, T. Dairay, R. Meunier .
- Part of Borelli, ENS-Paris Saclay : Mounir Atiq, Ludovic Minvielle, Transfer learning for fall detection, Tarket, Sergio Peignier, Transfer learning, Guillaume Richard, Transfer learning for Temporal data, EDF , Nicolas Vayatis, Director Centre Borelli, ENS-Paris-Saclay,

Thank you!

From Theory to Practice : The Adapt library

ML Feedbacks in Industry :

- labeled-data is often difficult and time-consuming to acquire
- Makes sense to re-use knowledge gained from related but distinct datasets.
- Transfer learning : the model can be pre-trained on data from a specific domain and then adapted to meet needs of a given task.
- Development of the Adapt library ($\rightarrow$ adapt) :



[de Mathelin et al., 2022],

References

de Mathelin, A., Deheeger, F., Mougeot, M., and Vayatis, N. (2022).
From theoretical to practical transfer learning : The adapt library.
In *Federated and Transfer Learning*, pages 283–306. Springer.

de Mathelin, A., Richard, G., Deheeger, F., Mougeot, M., and Vayatis, N. (2021).
Adversarial weighting for domain adaptation in regression.
In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 49–56. IEEE.

Ganin, Y. and Lempitsky, V. (2015).
Unsupervised domain adaptation by backpropagation.
In *International conference on machine learning*, pages 1180–1189. PMLR.

Minvielle, L., Atiq, M., Peignier, S., and Mougeot, M. (2019).
Transfer learning on decision tree with class imbalance.
In *2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI)*, pages 1003–1010. IEEE.

Minvielle, L., Atiq, M., Serra, R., Mougeot, M., and Vayatis, N. (2017).
Fall detection using smart floor sensor and supervised learning.
In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3445–3448.

Mounir, A., Sergio, P., and Mathilde, M. (2021).
Constrained prediction time random forests using equivalent trees and genetic programming : application to fall detection model embedding.
In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 690–697. IEEE.

Richard, G., Mathelin, A. d., Hébrail, G., Mougeot, M., and Vayatis, N. (2021).
Unsupervised multi-source domain adaptation for regression.
In *Machine Learning and Knowledge Discovery in Databases : European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, pages 395–411. Springer.