



HAL
open science

Réseaux de neurones informés par la physique pour l'apprentissage de lois de comportement

Antoine Benady, Ludovic Chamoin, Emmanuel Baranger

► **To cite this version:**

Antoine Benady, Ludovic Chamoin, Emmanuel Baranger. Réseaux de neurones informés par la physique pour l'apprentissage de lois de comportement. 25e Congrès Français de Mécanique, Nantes, Aug 2022, Nantes, France. hal-04280059

HAL Id: hal-04280059

<https://hal.science/hal-04280059>

Submitted on 10 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Réseaux de neurones informés par la physique pour l'apprentissage de lois de comportement

A. BENADY^a, L. CHAMOIN^a, E. BARANGER^a

a. Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS - Laboratoire de Mécanique Paris-Saclay, 91190, Gif-sur-Yvette, France.
{antoine.benady, ludovic.chamoin, emmanuel.baranger}@ens-paris-saclay.fr

Mots clés : Endommagement, réseaux de neurones, jumeaux numériques, DDDAS.

Résumé :

Le travail présenté vise à utiliser des réseaux de neurones pour l'apprentissage de lois de comportement. Contrairement aux approches classiques avec identification de paramètres de modèle, les réseaux de neurones permettent de ne pas postuler la forme des lois de comportement et donc de corriger les biais de modèle. Cependant, les réseaux de neurones classiques (au sens où il ne sont pas informés par la physique) présentent les inconvénients de nécessiter des volumes de données importants pour être entraînés, d'avoir des problèmes de généralisation à de nouvelles données et de manquer de garanties en termes d'interprétabilité. Les réseaux de neurones informés par la physique [RAI19] ont permis de lever en partie ces verrous sur diverses applications [KAR21] car l'apprentissage est simplifié (la richesse expérimentale étant ajoutée à une connaissance a priori). Ici, des réseaux de neurones sont utilisés pour l'apprentissage de potentiel thermodynamique. Dans cette approche, l'architecture du réseau respecte les principes thermodynamiques [MAS21] grâce à l'obtention de certaines grandeurs par différentiation automatique ainsi que des propriétés de convexité imposées dans la structure du réseau de neurones. Enfin, l'apprentissage du réseau de neurones est facilité par l'utilisation d'une fonction coût physique, l'erreur en relation de comportement modifiée, déjà utilisée dans le contexte de recalage de paramètres [MARC19]. La méthodologie sera illustrée et analysée sur différents cas tests. Les différents apports de la physique dans les réseaux de neurones seront comparés avec des réseaux de neurones dits classiques.

Abstract :

The presented work aims at using neural networks for constitutive modelling. In contrast to classical approaches with model parameter identification, neural networks does not postulate the shape of the behavioral laws and thus correct for model bias. However, classical neural networks (in the sense that they are not informed by physics) have the disadvantages of requiring large volumes of data to be trained, of having problems of generalization to new data and of lacking guarantees in terms of interpretability. Physics-informed neural networks [RAI19] have been able to partially overcome these drawbacks on various applications [KAR21] because learning is simplified (experimental richness being added to a priori knowledge). Here, neural networks are used for thermodynamic potential learning. In this approach, the architecture of the network respects thermodynamic principles [MAS21] thanks to the obtaining of certain quantities by automatic differentiation as well as convexity properties imposed in the structure of the neural network. Finally, the learning of the neural network is facilitated by the use of a physical cost function, the modified behavioral relation error, already used in the context of parameter updating.

1 Introduction

L'endommagement des structures mécaniques est une préoccupation permanente en ingénierie, liée à des problématiques de durabilité et de sécurité. La thématique est actuellement l'objet de diverses activités de recherche ; un exemple caractéristique est le projet ERC DREAM-ON (2021-2026), dans lequel se place ce travail, qui s'intéresse aux structures mécaniques complexes en matériaux composites et vise à aborder les challenges numériques liés au contrôle santé intégré de structures à grande échelle, afin d'aller des matériaux intelligents vers les structures intelligentes, capables de surveiller leur état de façon autonome et de fonctionner de façon sûre même en mode dégradé. Plus particulièrement, le travail présenté ici s'intéresse à un challenge particulier du projet ERC ; il a pour objectif de construire une procédure numérique efficace pour l'assimilation des données issues de capteurs distribués à base de fibres optiques. L'idée est de créer un jumeau numérique hybride, combinant les modèles physiques (qui représentent une riche histoire des sciences de l'ingénieur, et qui fournissent une forte connaissance a priori) et les techniques d'apprentissage issues de l'IA (ici, les réseaux de neurones). Ces dernières techniques sont donc ici exploitées pour corriger le biais de modèle, et non pas pour le substituer comme dans les approches *data-based*. Deux techniques de couplage entre le *deep learning* et les modèles physiques sont proposées pour intégrer la connaissance *a priori* au sein de l'apprentissage par IA. Une première section vise à présenter l'intégration de contraintes thermodynamiques dans le réseau de neurones. Dans une seconde partie, une fonction coût avec un sens physique fort est proposée pour l'entraînement du réseau.

2 Réseau de neurones thermodynamiquement consistant

2.1 Principe d'un réseau de neurones multicouches

Un réseau de neurones multicouches est une fonction mathématique qui lie un vecteur \mathbf{X} en entrée à une sortie \mathbf{Y} par une succession de couches dans lesquelles la valeur de la couche n notée \mathbf{H}_n s'obtient à partir de la valeur de la couche précédent grâce à la formule suivante :

$$\mathbf{H}_n = \sigma(\mathbf{W}_n \mathbf{H}_{n-1} + \mathbf{B}_n) \quad (1)$$

avec \mathbf{W}_n et \mathbf{B}_n les poids et biais associés à la couche n , et σ une fonction d'activation généralement non-linéaire comme une sigmoïde ou tangente hyperbolique. La sortie finale du réseau s'obtient donc en composant cette opération non-linéaire autant de fois que le nombre de couches, comme on peut le voir sur l'exemple de la figure 1.

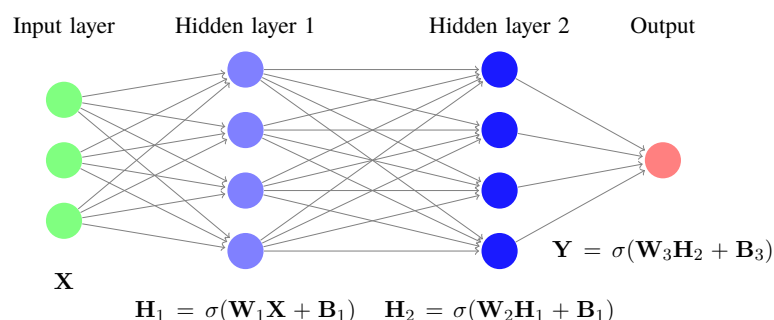
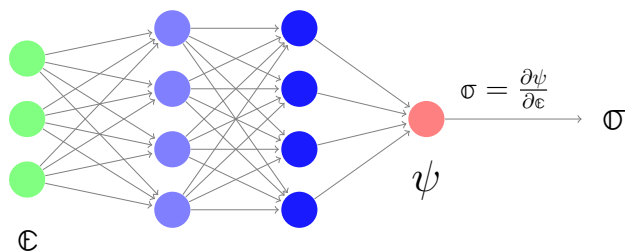


Figure 1 : Schéma d'un réseau de neurones à 2 couches cachées.

Pour trouver les poids et les biais qui permettent de se rapprocher au mieux des données d'entraînement, on minimise une fonction coût qui mesure l'écart entre les sorties du réseau et des observations. Cette fonction est très souvent non-convexe et est minimisée à l'aide de méthodes de descente de gradient adaptées aux réseaux de neurones comme l'algorithme Adam présenté dans [KIN15].

2.2 Architecture du réseau

Les réseaux de neurones classiques (au sens où ils ne sont pas informés par la physique) sont réputés pour ne pas offrir de garanties lors de la phase de prédiction, cela étant lié à la faible interprétabilité des résultats. Ici, une architecture garantissant le respect de la thermodynamique est proposée. Cette architecture est inspirée de [MAS21] mais présente des différences, notamment au niveau de la convexité. Le réseau prend en entrée une déformation et donne en sortie une contrainte. Un potentiel thermodynamique est utilisé comme grandeur intermédiaire. La figure 2 présente un schéma de l'architecture utilisée.

**Figure 2** : Architecture du réseau de neurones thermodynamiquement consistant. Le nombre de neurones n'est pas représentatif de l'architecture réellement utilisée.

L'utilisation d'un potentiel comme grandeur intermédiaire permet d'imposer le respect de la thermodynamique avec :

- la convexité du potentiel par rapport à la déformation ;
- l'obtention de la contrainte par dérivation du potentiel par rapport à la déformation.

$$\sigma = \frac{\partial \psi}{\partial \epsilon} \quad (2)$$

Convexité du réseau

La convexité du réseau par rapport aux variables d'entrée est imposée grâce à une structure particulière de réseau, proposée dans [AMO16]. Cette architecture repose sur la positivité de certains poids, ceci venant d'un résultat mathématique qui stipule que les sommes de fonctions convexes sont également convexes et que la composition d'une fonction croissante convexe avec une fonction convexe est également convexe [BOY04].

Différentiation automatique

L'équation 2 est garantie par le réseau à l'aide de la différentiation automatique [BAY18]. La différentiation automatique est une technique utilisée en *deep learning* pour calculer des dérivées partielles. Elle repose sur la construction d'un graphe qui mémorise toutes les opérations effectuées afin de pouvoir calculer les dérivées de certaines grandeurs par rapport à d'autres lorsqu'elles sont liées par des opérations dans le graphe. Ici, le potentiel s'obtient à partir de la déformation par la succession d'opérations décrites en 2.1 ; il est donc possible de dériver le potentiel par rapport à la déformation

3 Erreur en relation de comportement modifiée comme fonction coût de l'apprentissage

3.1 Formulation

Une technique utilisée pour tenir compte de l'a priori physique en *deep learning* consiste à utiliser une régularisation de la fonction coût qui comporte un sens physique [RAI19]. Dans l'approche proposée ici, la fonctionnelle « Erreur en relation de comportement modifiée » (mCRE) introduite dans [LAD94] dans le contexte de recalage de paramètres est utilisée comme fonction coût de l'apprentissage. Cette fonctionnelle se décompose en 2 termes :

- un terme d'écart aux observations ;
- un terme d'erreur de modèle fondé sur l'erreur en relation de comportement. Ce terme quantifie le niveau de non-vérification des relations de comportement pour un couple déplacement-contrainte admissible.

Pour la large classe des comportements matériau avec formulation standard, [LAD98], la fonctionnelle mCRE s'écrit de la manière suivante pour un couple $(\hat{\mathbf{e}}, \hat{\boldsymbol{\sigma}}) \in \mathcal{A}_d$ où \mathcal{A}_d est l'espace des solutions admissibles :

$$\mathcal{E}_{mCRE}^2(\hat{\mathbf{e}}, \hat{\boldsymbol{\sigma}}) = \alpha \|\Pi \circ \mathcal{M}(\hat{\mathbf{e}}, \mathbf{p}) - \mathbf{Y}_{obs}\|^2 + \int_{\Omega} \psi(\hat{\mathbf{e}}) + \psi^*(\hat{\boldsymbol{\sigma}}) - \hat{\boldsymbol{\sigma}} : \hat{\mathbf{e}} \quad (3)$$

avec :

- $\Pi \circ$ un projecteur des résultats du modèle aux endroits où les observations sont mesurées ;
- $\mathcal{M}(\hat{\mathbf{e}}, \mathbf{p})$ le modèle avec ses paramètres \mathbf{p} ;
- \mathbf{Y}_{obs} les observations ;
- Ω le domaine physique occupé par la structure mécanique ;
- ψ^* le potentiel dual de ψ au sens de la transformation de Legendre-Fenchel :

$$\psi^*(\hat{\boldsymbol{\sigma}}) = \sup_{\mathbf{e}} \{\hat{\boldsymbol{\sigma}} : \hat{\mathbf{e}} - \psi(\hat{\mathbf{e}})\}$$

- α un facteur d'échelle.

3.2 Minimisation de la mCRE

La procédure de minimisation de la mCRE diffère de la minimisation de fonction coût utilisée généralement dans les réseaux de neurones. Cette section vise à présenter la façon dont est minimisée la mCRE.

Par souci de simplicité, on se place dans le cas où les lois de comportement étudiées sont linéaires élastiques, c'est à dire le cas où les potentiels s'écrivent : $\psi(\hat{\epsilon}) = \frac{1}{2}\hat{\epsilon} : \mathbf{K} : \hat{\epsilon}$ et $\psi^*(\hat{\sigma}) = \frac{1}{2}\hat{\sigma} : \mathbf{K}^{-1} : \hat{\sigma}$. La minimisation dans le cas non-linéaire est présentée dans [MARC19, NGU21,22].

On cherche les paramètres optimaux \mathbf{p}_{sol} du modèle tels que :

$$\mathbf{p}_{sol} = \underset{\mathbf{p} \in \mathcal{P}}{\operatorname{argmin}} \left[\min_{(\hat{\mathbf{u}}, \hat{\sigma}) \in (\mathcal{A}_d)} \mathcal{E}_{mCRE}^2(\hat{\epsilon}(\hat{\mathbf{u}}), \hat{\sigma}) \right] \quad (4)$$

On trouve les paramètres optimaux grâce à une minimisation alternée décrite dans l'algorithme 1. Les paragraphes suivant détaillent comment sont effectués les opérations des phases 1 et 2.

Algorithme 1 : Minimisation de la mCRE dans le cas linéaire

Initialisation des paramètres $\mathbf{p}^{(0)}$ et d'un critère d'arrêt ϵ

Pour l'itération n :

Étape 1 : Trouver les champs admissibles optimaux $(\hat{\mathbf{u}}^{(n+1)}, \hat{\sigma}^{(n+1)})$ à paramètres \mathbf{p} fixés tels que :

$$(\hat{\mathbf{u}}^{(n+1)}, \hat{\sigma}^{(n+1)}) = \underset{(\hat{\mathbf{u}}, \hat{\sigma}) \in (\mathcal{A}_d)}{\operatorname{argmin}} \mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\sigma}; \mathbf{p}^{(n)}) \quad (5)$$

Étape 2 : Mise à jour des paramètres avec $(\hat{\mathbf{u}}^{(n+1)}, \hat{\sigma}^{(n+1)})$ fixés :

$$\mathbf{p}^{(n+1)} = \underset{\mathbf{p} \in \mathcal{P}}{\operatorname{argmin}} \mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}^{(n+1)}, \hat{\sigma}^{(n+1)}; \mathbf{p}^{(n)}) \quad (6)$$

Évaluation du critère d'arrêt :

Stop si $\mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}^{(n+1)}, \hat{\sigma}^{(n+1)}; \mathbf{p}^{(n+1)}) > \epsilon$

Étape 1 :

La phase 1 consiste à trouver les champs admissibles qui minimisent la mCRE. Pour cela, on introduit le Lagrangien suivant :

$$\mathcal{L}(\hat{\mathbf{u}}, \hat{\sigma}, \boldsymbol{\lambda}; \mathbf{p}) = \mathcal{E}_{mCRE}^2(\hat{\mathbf{u}}, \hat{\sigma}; \mathbf{p}) - \left[\int_{\Omega} \hat{\sigma} : \boldsymbol{\epsilon}(\boldsymbol{\lambda}) - \int_{\Omega} \mathbf{f}_d^v \cdot \boldsymbol{\lambda} - \int_{\partial_2 \Omega} \mathbf{f}_d^s \cdot \boldsymbol{\lambda} \right] \quad (7)$$

Ce Lagrangien permet d'imposer la contrainte d'admissibilité statique sur $\hat{\sigma}$. La contrainte d'admissibilité cinématique sur $\hat{\mathbf{u}}$ est imposée dans l'espace de recherche (en séparant les degrés libres des degrés imposés une fois le problème discrétisé). Après discrétisation (en négligeant l'erreur de discretisation), le Lagrangien s'écrit :

$$\mathcal{L}^h(\hat{\mathbf{U}}, \hat{\mathbf{V}}, \boldsymbol{\lambda}; \mathbf{p}) = \frac{1}{2}(\hat{\mathbf{U}} - \hat{\mathbf{V}})^T \mathbb{K}(\mathbf{p})(\hat{\mathbf{U}} - \hat{\mathbf{V}}) + \frac{\alpha}{2}(\Pi \hat{\mathbf{U}} - \mathbf{U}_{obs})^T \mathbb{G}_{obs}(\Pi \hat{\mathbf{U}} - \mathbf{U}_{obs}) - \boldsymbol{\Lambda}(\mathbb{K}(\mathbf{p})\hat{\mathbf{V}} - \mathbf{F}) \quad (8)$$

avec $\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$, et $\boldsymbol{\Lambda}$ les vecteurs des valeurs nodales de $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$ (tel que $\hat{\sigma} = \mathbf{K}(\mathbf{p})\hat{\epsilon}(\hat{\mathbf{v}})$) et $\boldsymbol{\lambda}$. \mathbb{K} et \mathbf{F} sont les matrice de rigidité globale et vecteur de force global. Les conditions de stationnarité du Lagrangien permettent d'aboutir à un système linéaire dont la résolution permet de trouver $\hat{\mathbf{U}}$ et $\hat{\mathbf{V}}$.

Étape 2 :

Lors de la phase 2 les paramètres sont mis à jour à l'aide d'une descente de gradient :

$$\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} - \beta_n \nabla_{\mathbf{p}} \mathcal{E}_{mCRE|\mathbf{p}^{(n)}}^{h^2} \quad (9)$$

avec β_n un taux d'apprentissage. Le gradient de la fonctionnelle mCRE peut se calculer à l'aide la méthode de l'état adjoint et s'écrit avec les champs $\hat{\mathbf{U}}$, $\hat{\mathbf{V}}$ issus de la phase 1 :

$$\nabla_{\mathbf{p}} \mathcal{E}_{mCRE|\mathbf{p}^{(n)}}^{h^2}(\hat{\mathbf{U}}, \hat{\mathbf{V}}; \mathbf{p})|_{\mathbf{p}^{(n)}} = \frac{1}{2}(\hat{\mathbf{U}} - \hat{\mathbf{V}})^T \frac{\partial \mathbb{K}(\mathbf{p})}{\partial \mathbf{p}}|_{\mathbf{p}^{(n)}}(\hat{\mathbf{U}} + \hat{\mathbf{V}}) \quad (10)$$

3.3 Utilisation avec les réseaux de neurones : dans le cas linéaire

Dans le cas d'un modèle linéaire à apprendre, une possibilité consiste à utiliser la procédure de minimisation décrite en 3.2 pour trouver les poids du réseau. De la même manière qu'il est possible d'obtenir une contrainte à partir du potentiel dans le réseau par différentiation automatique avec l'équation (2), il est possible d'obtenir un tenseur d'élasticité équivalent $\mathbf{K}_{NN}(\hat{\epsilon})$ de la manière suivante :

$$\mathbf{K}_{NN}(\epsilon) = \frac{\partial^2 \psi(\epsilon)}{\partial \epsilon^2}|_{\epsilon} \quad (11)$$

Ceci a été vérifié dans le cas d'un problème jouet dans lequel une base de données de 300 000 couples (σ, ϵ) générés à partir d'un modèle linéaire orthotrope \mathbf{K}_{ortho} (tel que $\sigma = \mathbf{K}_{ortho} : \epsilon$), a servi à l'entraînement du réseau de neurones. À la fin de l'entraînement, l'application de la relation 11 permet de vérifier que le $\mathbf{K}_{NN}(\epsilon)$ obtenu pour différentes entrées ϵ correspond au \mathbf{K}_{ortho} à apprendre.

4 Conclusion

La méthode proposée vise à faciliter l'apprentissage de potentiels décrivant le comportement matériau par réseau de neurones grâce à l'intégration de certaines connaissances provenant de la physique. L'espace des solutions recherchées est restreint aux solutions qui respectent les principes de la thermodynamique et l'apprentissage est régularisé par l'erreur en relation de comportement qui comporte un sens fortement physique. Des résultats de la méthode sur différents cas tests seront présentés lors de la conférence, en portant l'attention notamment sur les apports de l'IA informée par la physique en comparaison à des méthodes fondées sur de l'IA sans physique. Les différentes lois de comportement envisagées sont l'élasticité isotrope puis orthotrope, la plasticité et l'endommagement orthotrope. Une partie du comportement peut être donnée a priori, l'IA se chargeant alors d'apprendre le complément. Dans un premier temps l'entraînement du réseau de neurones se fera avec des données synthétiques, puis une fois la méthode validée, sur des données issues de fibres optiques.

References :

[AMO16] Amos, Brandon and Xu, Lei and Kolter, J. Zico, Input Convex Neural Networks, <https://arxiv.org/abs/1609.07152>; (2016)

- [BAY18] Baydin, Atilim Gunes, et al. "Automatic differentiation in machine learning : a survey." Journal of Machine Learning Research 18 (2018)
- [BOY04] Boyd, Stephen and Vandenberghe, Lieven. Convex optimization. Cambridge university press, (2004)
- [HOR89] K. Hornik, M. Stinchcombe, and H. White. "Multilayer Feedforward Networks Are Universal Approximators". In : Neural Networks (1989)
- [KAR21] Karniadakis GE, Kevrekidis I, Lu L, Perdikaris P, Wang S, Yang L. Physics-Informed Machine Learning. Nature Reviews Physics 3 :422 40 (2021)
- [KIN15] Kingma, Diederik and Jimmy Ba. "Adam : A Method for Stochastic Optimization." CoRRabs/1412.6980 (2015)
- [LAD94] Ladevèze, P., Nedjar, D., and Reynier, M. Updating of finite element models using vibration tests. AIAA Journal, 32(7) :1485–1491. (1994)
- [LAD98] Ladevèze, P. A modelling error estimator for dynamical structural model updating. In Elsevier, editor, Advances in Adaptive Computational Methods in Mechanics. P. Ladevèze, J.T. Oden eds. (1998)
- [MARC19] Marchand B, Chamoin L, Rey C, Parameter identification and model updating in the context of nonlinear mechanical behaviors using a unified formulation of the modified Constitutive Relation Error concept, Computer Methods in Applied Mechanics and Engineering, 345 :1094–1113 (2019)
- [MAS21] Masi F, Stefanou I, Vannucci P, Maffi-Berthier V, Thermodynamics-Based Artificial Neural Networks for Constitutive Modeling, Journal of the Mechanics and Physics of Solids 147 :104277 (2021)
- [NGU21] <https://tel.archives-ouvertes.fr/tel-03260879> Nguyen, Hai Nam Hai Nam Nguyen LMT - Laboratoire de mécanique et technologie - Université Paris-Saclay - CNRS - Centre National de la Recherche Scientifique - ENS Paris Saclay - Ecole Normale Supérieure Paris-Saclay Nouvelles stratégies numériques pour l'identification robuste, cohérente et efficace de modèles à partir de mesures de champs. (2021)
- [NGU22] H.N. Nguyen, L. Chamoin, C. Ha Minh, mCRE-based parameter identification from full-field measurements : consistent framework, integrated version, and extension to nonlinear material behaviors, CMAME (2022)
- [RAI19] Raissi M, Perdikaris P, Karniadakis G.E., Physics-Informed Neural Networks : A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations, Journal of Computational Physics 378 : 686 707 (2019)

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101002857)