

Chapitre X

Reconstruction de la masse d'une résonance grâce au *Machine Learning*

Sommaire

1	Introduction	1
2	Événements utilisés	2
2.1	Génération avec FASTSIM	3
2.2	Sélection des événements	3
2.3	Événements obtenus et pondération	5
3	Arbres de décision améliorés	6
3.1	Arbres de décision	6
3.2	<i>Gradient Boosting</i>	7
3.3	Fonction de coût et <i>Gradient Descent</i>	7
3.4	Sous-entraînement et surentraînement	8
4	Réseaux de neurones profonds	8
4.1	Neurones	8
4.2	Réseaux de neurones	11
4.3	Entraînement	11
5	Sélection d'un modèle	12
5.1	Variables d'entrée	13
5.2	Type de modèle	14
5.3	Fonction de coût	15
5.4	Initialisation des poids et optimisation	15
5.5	Structure	16
5.6	Fonction d'activation	18
6	Discussions	18
6.1	Effets de l'intervalle de masse	18
6.2	Effets de l'empilement	18
6.3	Effets de la reconstruction	18
6.4	Effets des faux taus hadroniques	18
6.5	Effets de la séparation des canaux	18
6.6	Effets de bord	18
6.7	Modèle final	18
7	Utilisation du modèle dans les analyses CMS	19
8	Conclusion	19

1 Introduction

L'utilisation de l'intelligence artificielle (IA) s'est grandement développée au cours des dernières années. L'IA est la capacité qu'ont des programmes à prendre des décisions, selon les informations qui leurs sont données par exemple sur leur environnement, de manière à maximiser leurs chances de réussite. L'entreprise Google DeepMind a par exemple développé AlphaGo [1], un programme destiné à jouer au jeu de Go, qui a battu en 2016 le champion du monde de la discipline 4 à 1.

Le *Machine Learning* (ML) est une branche de l'IA dans laquelle un modèle (algorithme ou programme) s'améliore à réaliser une tâche par accumulation d'expérience sur des jeux de données d'entraînement, sans pour autant être programmé explicitement pour réaliser cette tâche. Pour y parvenir, les jeux de données d'entraînement comprennent les informations $\{\vec{x}_i\}$ à donner au modèle ainsi que les « bonnes réponses » $\{y_{\text{vraie},i}\}$ qu'il doit fournir en sortie. L'objectif du modèle est donc de donner une fonction F approximant celle reliant les entrées $\{\vec{x}_i\}$ aux cibles $\{y_{\text{vraie},i}\}$. Il peut alors donner une prédiction $y_{\text{préd}}$ sur une nouvelle entrée selon $y_{\text{préd}} = F(\vec{x})$. La tâche du modèle est :

une classification lorsque y est discrète, par exemple lorsqu'il s'agit de déterminer si une image représente un chat ou un chien [2];

une régression lorsque y est continue, par exemple estimer le prix d'un bien immobilier [3].

Les applications du ML à la physique des particules sont variées et proposent de nombreux sujets d'étude [4-7]. Dans les chapitres précédents, le ML est déjà activement utilisé pour diverses tâches :

- identification des jets issus de quarks b (b -tagging) avec DEEPCSV [8-10];
- identification des taus hadroniques avec DEEPTAU [11];
- catégorisation des événements comme exposé dans le chapitre 4 [12, 13].

Dans les événements $H \rightarrow \tau\tau$ présentés au chapitre 1, et plus généralement lors de tout processus physique $X \rightarrow \tau\tau$ où une particule X se désintègre en paire de leptons tau, des neutrinos sont émis lors des désintégrations des taus. Or, ils sont invisibles dans les détecteurs tels que CMS ou ATLAS. Il est donc impossible de déterminer la masse invariante totale du système $\tau\tau$ issu de X . Plusieurs méthodes ont été développées afin de reconstruire la masse du système $\tau\tau$ [14-16]. Dans le cadre des analyses $H \rightarrow \tau\tau$, la collaboration CMS utilise SVFIT [17].

La reconstruction la masse de la particule X , ou résonance, se désintégrant en paire de leptons tau grâce au *Machine Learning* a été étudiée par BÄRTSCHI & coll. [18] dans le cas où X est un boson de Higgs avec une masse entre 80 et 300 GeV. Ils ont obtenu une résolution de 8,4 % sur la masse du Higgs, contre 17 % avec SVFIT. Le temps de calcul nécessaire à l'obtention de la masse est de plus bien plus court avec le ML. L'utilisation du ML est donc très prometteuse. Cependant, ces travaux utilisent des événements générés avec une simulation grossière du détecteur CMS basée sur DELPHES [19, 20] et sans empilement, notion introduite dans le chapitre 2.

Les travaux présentés dans ce chapitre vont plus loin. La génération des événements, introduite dans la section 2, utilise FASTSIM [21-24] pour modéliser le détecteur CMS. Bien qu'il ne s'agisse pas de la simulation complète basée sur GEANT4 [25-27], FASTSIM est bien plus proche de la réalité que DELPHES. De plus, l'empilement est pris en compte. Les modèles obtenus sont ainsi directement utilisables dans de réelles analyses, telles que celle présentée dans le chapitre 4.

Deux types de modèle sont étudiés :

- des arbres de décision améliorés, introduits section 3;
- des réseaux de neurones profonds, introduits section 4.

La comparaison des modèles obtenus et la sélection de l'un d'entre eux est présentée section 5. Dans la section 6, divers effets sur les performances des modèles sont discutés, en particulier la prise en compte de l'empilement. Enfin, l'utilisation en conditions réelles du modèle issu de ces travaux dans des analyses de physique est présentée dans la section 7.

2 Événements utilisés

L'objectif des modèles à entraîner est de reconstruire la masse des particules se désintégrant en paire de leptons tau. Il s'agit d'une tâche de régression, il faut donc entraîner les modèles sur le plus de valeurs différentes possible. Dans l'optique d'une utilisation dans les analyses telles que celle présentée dans le chapitre 4, il a été choisi d'utiliser des événements $\mathcal{H} \rightarrow \tau\tau$ où \mathcal{H} est le boson de Higgs du modèle standard h dont la masse est modifiée, à l'instar de ce qu'ont fait BÄRTSCHI & coll. [18]. La cible du modèle est donc la masse $m_{\mathcal{H}}$.

2.1 Génération avec FASTSIM

Nous avons généré nos propres données simulées [28] afin d’obtenir des événements indépendants de ceux utilisés dans les analyses. Dans le contexte de la collaboration CMS, nous avons utilisé FASTSIM [21–24]. Cet outil permet de procéder à l’ensemble de la simulation des événements introduite chapitre 2, de la génération du processus physique à la reconstruction des objets physiques par le détecteur.

Les processus physiques sont générés par PYTHIA 8 [29] avec les réglages CUEP8M1 [30, 31]. L’énergie dans le centre de masse est de 13 TeV. Pour ne pas générer d’événements indésirables, seules les collisions créant un boson de Higgs par fusion de gluons, mode dominant pour le modèle standard, sont autorisées. De plus, le rapport de branchement $\mathcal{BR}(\mathcal{H} \rightarrow \tau\tau)$ est fixé à 1, c’est-à-dire que \mathcal{H} se désintègre forcément en paires de leptons taus.

La masse de \mathcal{H} varie de 50 à 800 GeV par pas de 1 GeV. Il est important d’utiliser l’intervalle le plus étendu possible, il correspond à la gamme utile des modèles obtenus. L’effet de l’étendue de cet intervalle est discuté dans la section 6. Lorsque $m_{\mathcal{H}}$ est supérieure à 800 GeV, les propriétés de \mathcal{H} , basées sur celles de h , ne permettent pas d’obtenir des valeurs de $m_{\mathcal{H}}$ cohérentes avec la méthode de génération utilisée. Nous ne considérerons pas de masse plus haute. Bien qu’il soit possible pour une particule de se désintégrer en deux taus dès que sa masse est plus élevée que $2m_{\tau} = 3,5$ GeV, la sélection des événements présentée dans la section 2.2 rejette plus de 99 % des événements lorsque $m_{\mathcal{H}} < 50$ GeV. Nous ne considérerons pas de masse plus basse. L’efficacité des sélections appliquées est représentée sur la figure X.1. S’il est possible d’appliquer des poids aux événements afin d’équilibrer l’entraînement sur l’ensemble des valeurs de la cible, plus d’événements sont générés à basse masse afin d’obtenir des topologies d’événements variées malgré la faible efficacité de sélection. Ainsi, la quantité d’événements générés pour chaque valeur de $m_{\mathcal{H}}$ est de :

- 60 000 pour $50 \leq m_{\mathcal{H}} < 300$;
- 20 000 pour $300 \leq m_{\mathcal{H}} < 500$;
- 10 000 pour $500 \leq m_{\mathcal{H}} \leq 800$.

L’empilement est modélisé par superposition du signal $\mathcal{H} \rightarrow \tau\tau$ à des événements dits de « biais minimum » [29]. Il s’agit d’événements pouvant contenir des interactions dures, mais n’activant pas de chemin de déclenchement. La quantité d’empilement ajoutée à l’événement $\mathcal{H} \rightarrow \tau\tau$ suit le profil de l’année 2017.

2.2 Sélection des événements

2.2.1 Canaux $\tau_h\tau_h$, $\mu\tau_h$, $e\tau_h$ et $e\mu$

La sélection des événements se fait comme exposé dans le chapitre 4 pour l’année 2017 et les canaux $\tau_h\tau_h$, $\mu\tau_h$, $e\tau_h$ et $e\mu$ y étant exploités, à l’exception des coupures servant à séparer la région de signal des régions de contrôle et de détermination, sur $m_T^{(\mu)}$ dans le canal $\mu\tau_h$, $m_T^{(e)}$ dans le canal $e\tau_h$, D_{ζ} dans le canal $e\mu$. La construction du *dilepton* est inchangée. La correspondance des objets du *dilepton* avec ceux ayant activé le chemin de déclenchement n’est pas vérifiée. Ce choix permet d’obtenir un modèle dont les prédictions auront non seulement un sens dans les régions de contrôle et de détermination, mais aussi plus facilement dans le contexte d’autres analyses dans lesquelles les sélections peuvent différer.

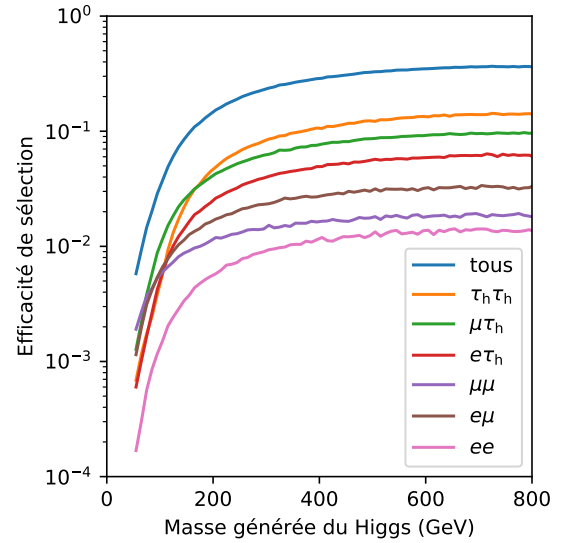


Figure X.1 – Efficacité de sélection des événements pour $m_{\mathcal{H}} \in [50, 800]$ GeV dans les différents canaux et pour tous les canaux.

En plus des canaux listés ci-dessus, nous avons également sélectionné des événements des canaux $\mu\mu$ et ee , selon les procédures présentées ci-après.

2.2.2 Canal $\mu\mu$

Sélection des muons Tout muon respectant les critères listés ci-après est retenu pour jouer le rôle de L_1 ou L_2 dans le *dilepton* :

- $p_T^\mu > 10 \text{ GeV}$;
- $|\eta^\mu| < 2.4$;
- paramètres d'impact $d_z < 0,2 \text{ cm}$ et $d_{xy} < 0,045 \text{ cm}$;
- $I^\mu < 0,15 p_T^\mu$;
- passer le point de fonctionnement *medium* du *muonID*.

Sélection du *dilepton* L'événement est retenu à condition qu'au moins une paire $L_1 L_2 = \mu\mu$ puisse être construite avec L_1 et L_2 de charges électriques opposées. Il est de plus requis que L_1 et L_2 soient séparés dans le plan (η, ϕ) tel que $\Delta R > 0,3$. Si plus d'une paire possible existe dans l'événement, une seule est retenue selon la logique exposée dans le chapitre 4.

Vétos de leptons supplémentaires Les vetos de leptons supplémentaires doivent être respectés, c'est-à-dire que l'événement ne contient pas :

- de second muon tel que $p_T^\mu > 10 \text{ GeV}$, $|\eta^\mu| < 2,4$, passant le point de fonctionnement *medium* du *muonID* et d'isolation $I^\mu < 0,3 p_T^\mu$;
- de second électron tel que $p_T^e > 10 \text{ GeV}$, $|\eta^e| < 2,5$, passant le point de fonctionnement à 90 % d'efficacité de l'*electron ID MVA* et d'isolation $I^e < 0,3 p_T^e$, l'électron devant passer le veto d'électron de conversion et présenter moins de deux points de passage manquants dans le trajectographe.

2.2.3 Canal ee

Sélection des électrons Tout électron respectant les critères listés ci-après est retenu pour jouer le rôle de L_1 ou L_2 dans le *dilepton* :

- $p_T^e > 20 \text{ GeV}$;
- $|\eta^e| < 2.4$;
- paramètres d'impact $d_z < 0,2 \text{ cm}$ et $d_{xy} < 0,045 \text{ cm}$;
- $I^e < 0,1 p_T^e$;
- passer le point de fonctionnement à 90 % d'efficacité de l'*electron ID MVA*.

Sélection du *dilepton* L'événement est retenu à condition qu'au moins une paire $L_1 L_2 = ee$ puisse être construite avec L_1 et L_2 de charges électriques opposées. Il est de plus requis que L_1 et L_2 soient séparés dans le plan (η, ϕ) tel que $\Delta R > 0,5$. Si plus d'une paire possible existe dans l'événement, une seule est retenue selon la logique exposée dans le chapitre 4.

Vétos de leptons supplémentaires Les vetos de leptons supplémentaires doivent être respectés, c'est-à-dire que l'événement ne contient pas :

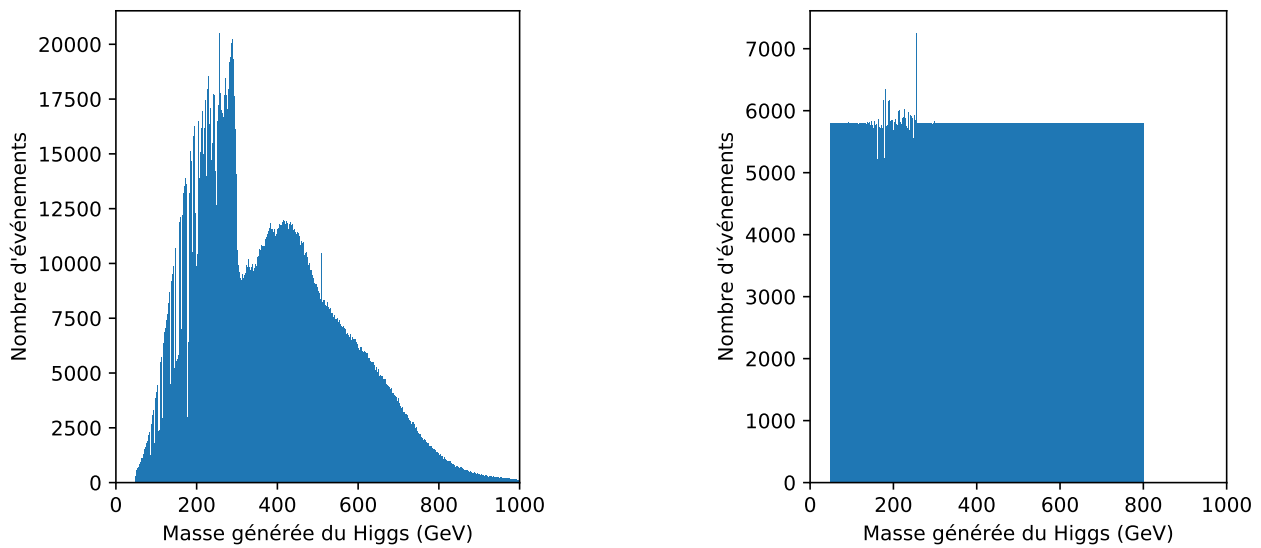
- de second muon tel que $p_T^\mu > 10 \text{ GeV}$, $|\eta^\mu| < 2,4$, passant le point de fonctionnement *medium* du *muonID* et d'isolation $I^\mu < 0,3 p_T^\mu$;
- de second électron tel que $p_T^e > 10 \text{ GeV}$, $|\eta^e| < 2,5$, passant le point de fonctionnement à 90 % d'efficacité de l'*electron ID MVA* et d'isolation $I^e < 0,3 p_T^e$, l'électron devant passer le veto d'électron de conversion et présenter moins de deux points de passage manquants dans le trajectographe.

2.3 Événements obtenus et pondération

Plus de 22 millions d'événements ont été générés. Environ 3 millions sont sélectionnés selon les critères présentés précédemment. La distribution de $m_{\mathcal{H}}$ dans ces événements sélectionnés est représentée sur la figure X.2a. Quelques événements présentent des valeurs de $m_{\mathcal{H}}$ au-delà de 800 GeV, cet effet est dû à la largeur de cette particule, représentée sur la figure X.3 en fonction de sa masse. La largeur à 800 GeV est ainsi d'environ 300 GeV. Le réglage $m_{\mathcal{H}} = 800$ GeV donne donc des événements contenant un boson dont la masse effective se situe entre 500 et 1100 GeV, d'où la queue de la distribution observée à haute masse sur la figure X.2a. À basse masse en revanche, la largeur est inférieure à 100 MeV, cet effet n'est donc pas présent. La cible du modèle est la masse effective du boson. Les événements retenus dans la suite sont ceux où celle-ci se situe bien entre 50 et 800 GeV, d'où la disparition de la queue à haute masse sur la figure X.2b.

Ces événements sont de plus séparés en trois groupes selon les proportions suivantes :

- 70 % pour l'entraînement. Ce sont ces événements que les modèles pourront exploiter afin d'ap-



(a) Distribution brute sur tous les événements.

(b) Distribution pondérée pour les événements d'entraînement.

Figure X.2 – Distributions de la masse générée de \mathcal{H} .

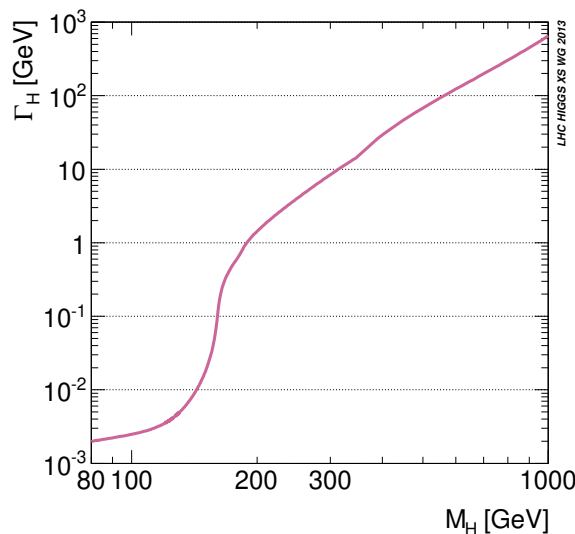


Figure X.3 – Largeur du boson de Higgs du modèle standard [32].

Le gain G obtenu par la création de deux nouvelles branches b_1 et b_2 s'exprime

$$G = S_{b_1} + S_{b_2} - S_{b_1+b_2} \quad (\text{X.2})$$

avec $S_{b_1+b_2}$ la similarité du jeu de donnée non séparé, S_{b_1} (S_{b_2}) la similarité du jeu de donnée se retrouvant dans la branche b_1 (b_2). La condition retenue pour former les deux branches est celle présentant le gain le plus élevé. Ce processus est alors itéré sur chacune des nouvelles branches, jusqu'à ce que :

- le gain soit inférieur à γ ;
- la profondeur de l'arbre (nombre de conditions successives) est supérieure à $N_{\max}^{\text{prof.}}$.
- la quantité d'échantillons dans une branche est inférieure à $N_{\min}^{\text{échant.}}$.

Les paramètres γ , $N_{\max}^{\text{prof.}}$ et $N_{\min}^{\text{échant.}}$, fixés par l'utilisateur, sont nommés « hyper-paramètres ». Ils ne doivent pas être confondus avec les paramètres propres à l'arbre, déterminés lors de la construction des branches.

3.2 Gradient Boosting

La technique du *Gradient Boosting* consiste en l'utilisation de modèles simples, ici des arbres de décision, pour obtenir un modèle global plus robuste. La construction se fait de manière itérative.

À chaque étape $k \geq 1$, un arbre de décision M_k nommé estimateur est construit avec pour objectif de prédire

$$y_{\text{vraie},i} - F_{k-1}(\vec{x}_i) \quad (\text{X.3})$$

pour une entrée \vec{x}_i , avec $y_{\text{vraie},i}$ la valeur que doit prédire le modèle global pour l'entrée \vec{x}_i et F_{k-1} la fonction du modèle global issu de l'étape $k-1$, F_0 étant égale à M_0 , l'arbre de décision obtenu sans *Gradient Boosting*. Le modèle M_k corrige donc l'écart résiduel des prédictions $\{y_{\text{préd},i}\}$ du modèle global à $\{y_{\text{vraie},i}\}$. Les prédictions F_k du modèle global s'expriment alors

$$y_{\text{préd},i} = F_k(\vec{x}_i) = F_{k-1}(\vec{x}_i) + \eta M_k(\vec{x}_i) \quad (\text{X.4})$$

avec η le taux d'apprentissage, inférieur à 1, permettant de corriger progressivement l'écart résiduel. L'itération s'arrête lorsque le nombre maximal d'estimateurs $N_{\max}^{\text{estim.}}$ est atteint. Les grandeurs η et $N_{\max}^{\text{estim.}}$ sont également des hyper-paramètres.

3.3 Fonction de coût et Gradient Descent

Une fonction de coût (*loss function*) compare les prédictions d'un modèle aux valeurs vraies. Elle doit être différentiable et est définie de manière à être minimale lorsque les prédictions sont égales aux valeurs vraies, c'est-à-dire lorsque le modèle est parfait. Les fonctions de coûts les plus répandues sont :

MSE *Mean Squared Error* ou erreur quadratique moyenne,

$$L_{\text{MSE}}(\{y_{\text{vraie},i}\}, \{y_{\text{préd},i}\}) = \frac{1}{2N} \sum_{i=1}^N (y_{\text{préd},i} - y_{\text{vraie},i})^2 ; \quad (\text{X.5})$$

MAE *Mean Absolute Error* ou erreur absolue moyenne,

$$L_{\text{MAE}}(\{y_{\text{vraie},i}\}, \{y_{\text{préd},i}\}) = \frac{1}{N} \sum_{i=1}^N |y_{\text{préd},i} - y_{\text{vraie},i}| ; \quad (\text{X.6})$$

MAPE *Mean Absolute Percentile Error* ou erreur absolue relative moyenne,

$$L_{\text{MAPE}}(\{y_{\text{vraie},i}\}, \{y_{\text{préd},i}\}) = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_{\text{préd},i} - y_{\text{vraie},i}}{y_{\text{vraie},i}} \right|. \quad (\text{X.7})$$

L'objectif du modèle M_k défini dans la section précédente est de prédire, pour \vec{x}_i ,

$$y_{\text{vraie},i} - F_{k-1}(\vec{x}_i) = - \frac{\partial \text{L}_{\text{MSE}}(y_{\text{vraie},i}, F_{k-1}(\vec{x}_i))}{\partial F_{k-1}(\vec{x}_i)}. \quad (\text{X.8})$$

Il est ainsi possible de généraliser le *Gradient Boosting* en considérant que l'objectif de M_k est de prédire

$$- \frac{\partial \text{L}(y_{\text{vraie},i}, F_{k-1}(\vec{x}_i))}{\partial F_{k-1}(\vec{x}_i)} = - \vec{\nabla}_{F_{k-1}(\vec{x}_i)} (\text{L}(y_{\text{vraie},i}, F_{k-1}(\vec{x}_i))) \quad (\text{X.9})$$

avec L une fonction de coût quelconque. Il s'agit du *Gradient Descent*, où l'objectif est de minimiser L . La fonction de coût est un hyper-paramètre du modèle.

3.4 Sous-entraînement et surentraînement

La construction d'un modèle, aussi appelé « entraînement », est un processus itératif visant à minimiser la fonction de coût. Dans le cas des arbres de décision améliorés créés avec XGBOOST, l'entraînement cesse lorsque le nombre d'estimateur maximal est atteint, lorsqu'il est renseigné. Dans tous les cas, il est légitime de se demander si le modèle obtenu à la fin de l'entraînement est optimal.

Il faut que l'entraînement soit suffisamment long pour que le modèle propose les prédictions les plus précises possible. Autrement dit, il faut que le modèle ait le temps d'apprendre. S'il ne l'a pas, les prédictions ne sont pas aussi précises qu'elles pourraient l'être, c'est le sous-entraînement. La valeur de la fonction de coût appliquée au jeu de données d'entraînement diminuant lors de l'apprentissage, un critère pourrait être de l'utiliser afin de déterminer si le modèle apprend encore ou non. Arrivé à un plateau, le modèle ne s'améliore plus et l'entraînement s'arrête.

Cette approche masquerait toutefois une spécialisation du modèle. En effet, un modèle peut apprendre à prédire parfaitement $\{y_{\text{vraie},i}\}$ sur le jeu de données d'entraînement, ce qui correspond à une fonction de coût nulle, mais être moins bon qu'un modèle entraîné moins longtemps lorsqu'il est utilisé sur d'autres données. C'est le surentraînement. Cet effet peut être évité en utilisant un jeu de données dit de « validation », non utilisé pour régler les paramètres du modèle.

L'intérêt du jeu de validation est illustré sur la figure X.5. Un modèle sous-entraîné ou dont l'entraînement est optimal présente des erreurs similaires dans les deux jeux de données. Dans le cas d'un surentraînement, les erreurs continuent à diminuer sur le jeu d'entraînement, mais pas sur le jeu de validation. Une fonction d'évaluation E , éventuellement égale à la fonction de coût L , permet de quantifier ces erreurs et de mettre fin à l'entraînement avant de surentraîner le modèle. Il s'agit de l'arrêt prématuré (*early stopping*).

Dans le cas des arbres de décision améliorés, une itération de l'entraînement consiste en l'ajout d'un estimateur, comme exposé dans la section 3.2. Un arrêt prématuré est réalisé lorsque l'erreur quadratique moyenne ne diminue pas sur le jeu de validation pendant 5 itérations.

4 Réseaux de neurones profonds

Les réseaux de neurones (NN, *Neural Networks*) sont un autre type de modèle permettant d'approximer la fonction reliant les entrées $\{\vec{x}_i\}$ aux cibles $\{y_{\text{vraie},i}\}$ [35]. La section 4.1 introduit le concept de neurone dans le cadre du ML. Puis, les réseaux de neurones sont présentés dans la section 4.2. L'entraînement de ce type de modèle est discuté section 4.3.

4.1 Neurones

4.1.1 Principe

Un neurone est une entité ayant un certain nombre d'entrées $x_j, j \in \{1, \dots, n\}$, auxquelles sont associées des poids w_j , un biais b et une fonction f dite d'« activation », discutée section 4.1.2. Les poids w_j et le biais b sont les paramètres du neurone, la fonction d'activation est un hyper-paramètre. La sortie s du neurone s'exprime

$$s = f \left(\sum_{j=1}^n w_j x_j + b \right). \quad (\text{X.10})$$

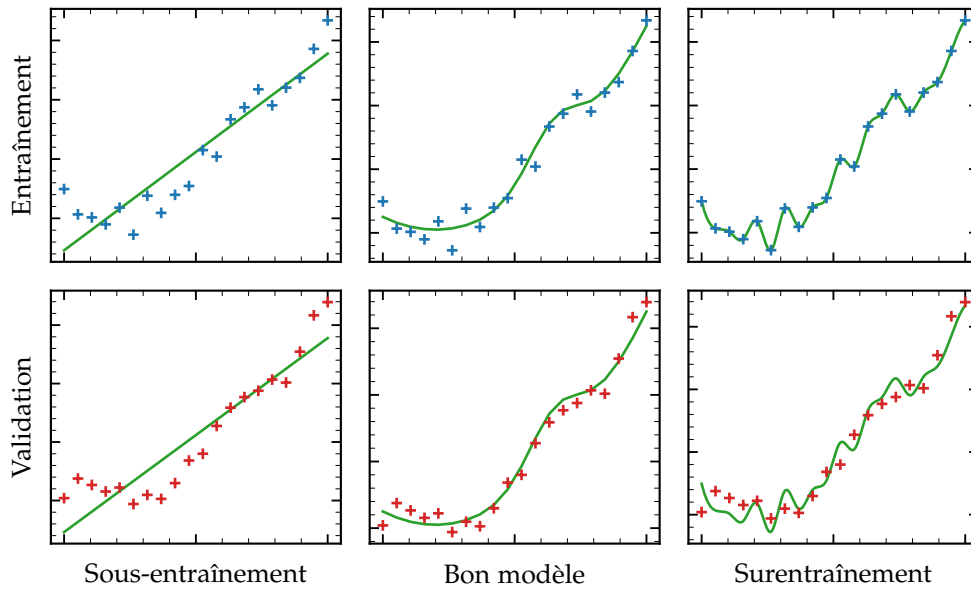


Figure X.5 – Illustrations du sous-entraînement et du surentraînement. Un même modèle est peu (gauche), suffisamment (milieu) ou trop entraîné (droite). Ses prédictions (ordonnées) en fonction de l'entrée (abscisses) sont tracées en vert. Le jeu de données d'entraînement (de validation) est représenté par des croix bleues (rouges) sur la ligne du haut (bas).

Le fonctionnement d'un neurone est résumé sur la figure X.6.

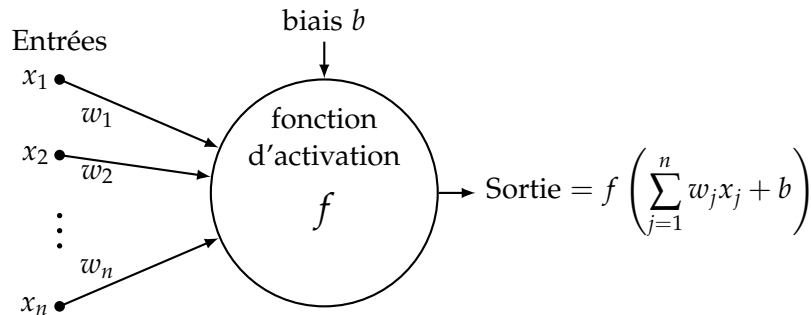


Figure X.6 – Structure d'un neurone. Une fonction f dite d'« activation » est appliquée à la somme des entrées x_j pondérées par les poids w_j et du biais b afin d'obtenir la valeur de sortie.

4.1.2 Fonctions d'activation

En principe, toute fonction définie sur l'ensemble d'existence des entrées x_j peut être utilisée comme fonction d'activation. Celles-ci étant généralement à valeurs réelles et unidimensionnelles, les fonctions sont définies sur \mathbb{R} . Les plus utilisées sont :

tangente hyperbolique notée \tanh , définie par

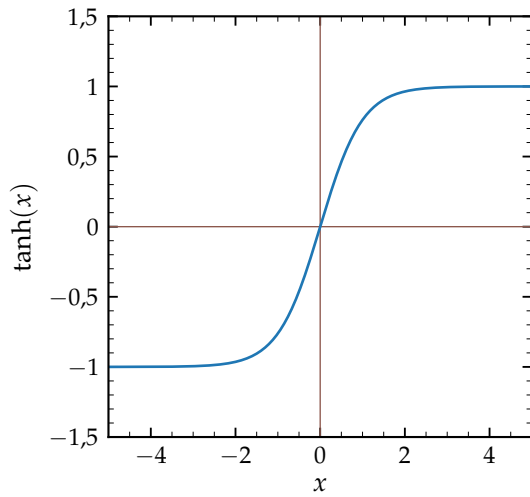
$$\tanh : x \mapsto \frac{e^x - e^{-x}}{e^x + e^{-x}} ; \quad (\text{X.11})$$

sigmoïde notée sig , définie par

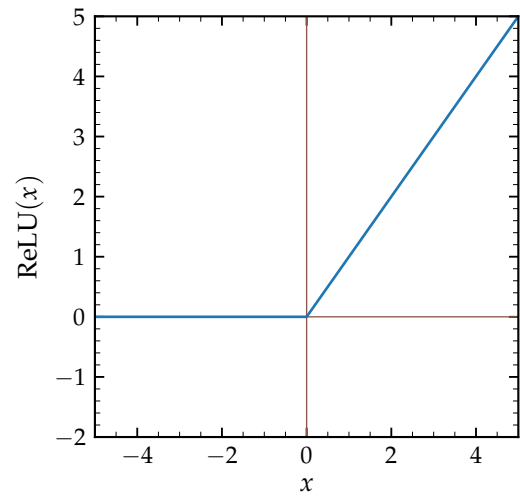
$$\text{sig} : x \mapsto \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}} ; \quad (\text{X.12})$$

Softsign notée Ssg , définie par

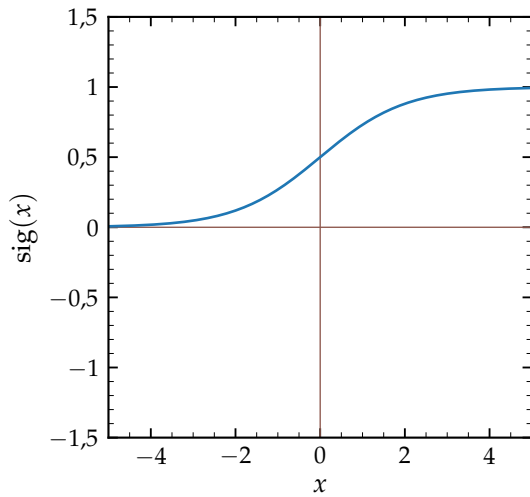
$$\text{Ssg} : x \mapsto \frac{x}{1 + |x|} ; \quad (\text{X.13})$$



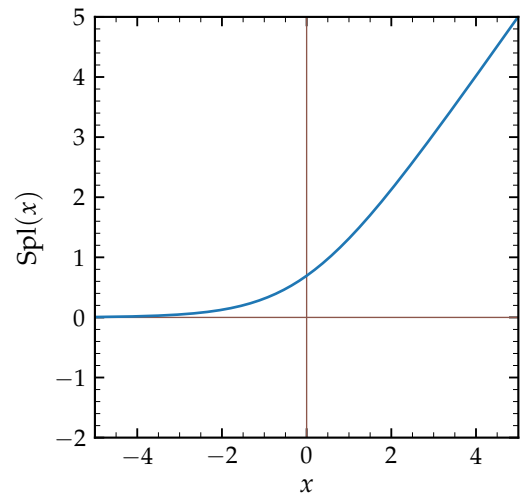
(a) Tangente hyperbolique.



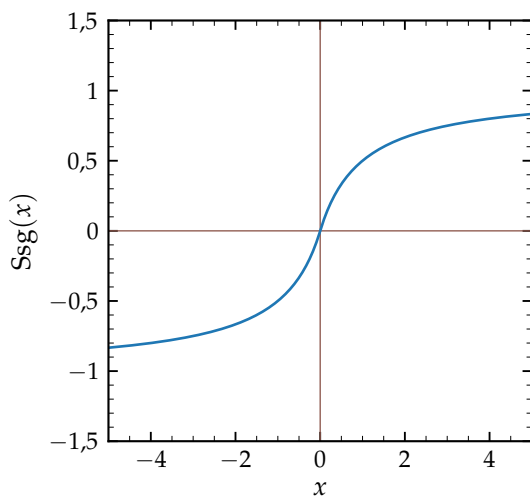
(b) ReLU.



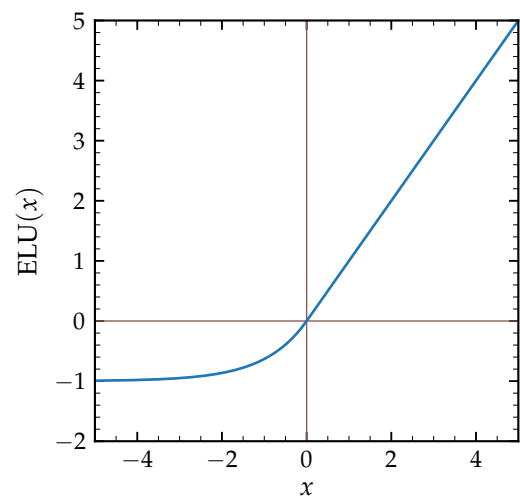
(c) Sigmoïde.



(d) Softplus.



(e) Softsign.



(f) ELU.

Figure X.7 – Exemples de fonctions d'activation. À gauche, des fonctions à valeurs bornées, généralement utilisées en classification. À droite, des fonctions à valeurs non bornées, utilisables pour des tâches de régression.

ReLU (*Rectified Linear Unit*), définie par

$$\text{ReLU} : x \mapsto \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}; \quad (\text{X.14})$$

Softplus notée *Spl*, définie par

$$\text{Spl} : x \mapsto \ln(1 + e^x); \quad (\text{X.15})$$

ELU (*Exponential Linear Unit*), définie par

$$\text{ELU} : x \mapsto \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}, \quad \alpha = 1; \quad (\text{X.16})$$

SELU (*Scaled Exponential Linear Unit*), définie par

$$\text{SELU} : x \mapsto \lambda \times \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases}, \quad \alpha \simeq 1,67, \quad \lambda \simeq 1,05; \quad (\text{X.17})$$

ou encore la fonction linéaire identité $\mathbb{I} : x \mapsto x$. Certaines d'entre elles sont représentées sur la figure X.7.

4.2 Réseaux de neurones

Un NN est obtenu par l'interconnexion de plusieurs neurones entre eux. Ces connexions peuvent se faire selon diverses architectures [35]. Nous utilisons ici, comme dans les travaux de BÄRTSCHI & coll. [18], une architecture normale profonde à propagation avant complètement connectée (*normal deep feedforward fully-connected*), c'est-à-dire avec :

- des neurones répartis en couches (normale);
- plusieurs couches « cachées », situées entre celles d'entrée et de sortie (profonde);
- les entrées des neurones de la couche k :
 - prises parmi les sorties de ceux de la couche $k - 1$ (à propagation avant),
 - étant toutes les sorties de ceux de la couche $k - 1$ (complètement connectée).

Le nombre de neurones par couche cachée N_N est pris constant. Le nombre de couches cachées (*hidden layers*) est noté N_L . Le NN ayant une structure profonde, il s'agit d'un DNN (*Deep Neural Network*).

La tâche du réseau est une régression sur une seule grandeur, $m_{\mathcal{H}}$, à partir de n variables d'entrée $x_j, j \in \{1, \dots, n\}$. La couche de sortie est donc composée d'un seul neurone dont la fonction d'activation est linéaire. La couche d'entrée comporte n neurones, chacun ayant une entrée correspondant à une variable avec un poids de 1 et un biais de zéro. La fonction d'activation des neurones des couches d'entrée et cachées est identique, plusieurs possibilités sont essayées dans la section 5. La structure obtenue est représentée sur la figure X.8.

4.3 Entraînement

L'entraînement d'un NN est le réglage des paramètres des neurones du réseau situés sur les couches cachées et la couche de sortie. Il s'agit des poids w_i et du biais b . Pour un DNN avec $n_{\text{in}} = 27$ variables d'entrée, $N_L = 3$ couches cachées de $N_N = 1000$ neurones, le nombre de paramètres est ainsi de

$$\begin{aligned} N_{\text{params.}} &= \underbrace{N_N \times (n_{\text{in}} + 1)}_{\text{couche cachée 1}} + \underbrace{(N_L - 1) \times N_N \times (N_N + 1)}_{\text{autres couches cachées}} + \underbrace{N_N + 1}_{\text{couche de sortie}} \\ &= 28\,000 + 2 \times 1\,001\,000 + 1001 = 2\,031\,001, \end{aligned} \quad (\text{X.18})$$

soit près de deux millions. Les termes « +1 » correspondent aux biais b à ajouter au nombre d'entrées des neurones.

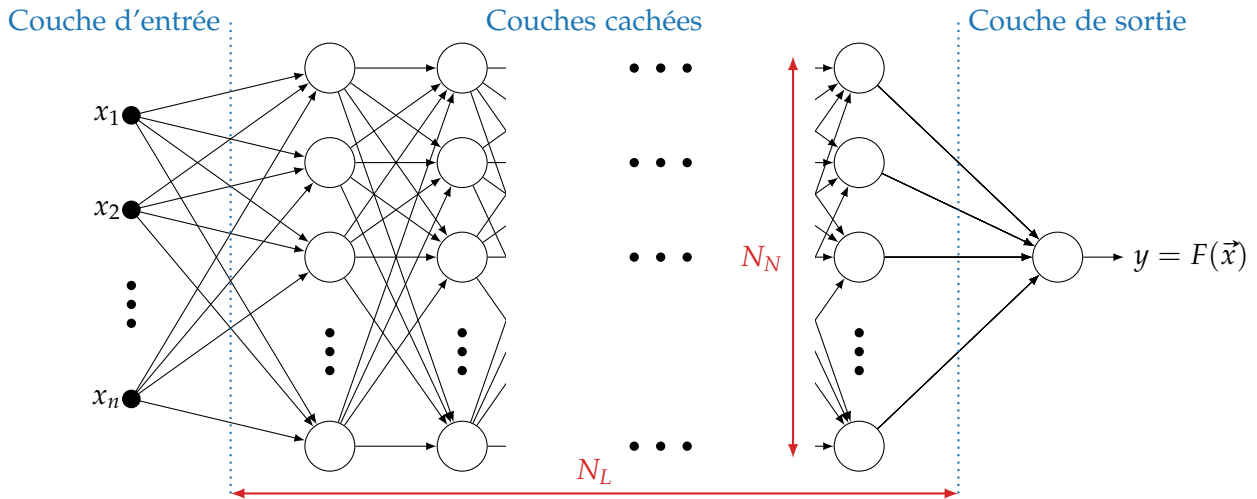


Figure X.8 – Structure d'un réseau de neurones. Une couche d'entrée comporte autant de neurones que de variables x_i . La couche de sortie en comporte autant que de valeurs à donner, c'est-à-dire une. Les fonctions d'activation de ces deux couches sont linéaires. Entre elles se trouvent N_L couches cachées, chacune contenant N_N neurones. Diverses fonctions d'activation peuvent être utilisées dans les couches cachées.

4.3.1 Initiation des paramètres

Les poids w_i sont initialement fixés à une valeur constante donnée ou aléatoirement selon une loi de probabilité. Le mode d'initiation est un hyper-paramètre du modèle. Lors de ces travaux, nous avons testé les lois normale et uniforme. Dans le cas des DNNs, ces modes d'initiation peuvent être améliorés par la méthode de GLOROT & BENGIO [36] afin de faciliter l'entraînement. Il s'agit alors des lois « Glorot uniforme » et « Glorot normale », également testées.

4.3.2 Fonction de coût et optimisation des paramètres

Les modifications apportées aux paramètres ont pour objectif l'amélioration des prédictions du modèle. La qualité de ces prédictions est quantifiée par une fonction de coût L à minimiser, comme exposé section 3.3. Il s'agit donc de trouver le minimum de L dans l'espace à D dimensions formé par les $D = N_{\text{params.}}$ paramètres à régler. Cela peut être fait de manière itérative par *Gradient Descent*.

Le *Gradient Descent* détermine le gradient de L , $\vec{\nabla}(L)$, autour de la « position » du modèle dans l'espace à D dimensions. Chaque paramètre p (w_i et b de chaque neurone) est alors modifié selon

$$p \rightarrow p - \eta \vec{\nabla}(L) \cdot \vec{e}_p = p - \eta \frac{\partial L}{\partial p} \quad (\text{X.19})$$

avec η le taux d'apprentissage.

optimizers and variations of GD Adam, Adadelata, SGD most of them = backpropagation
mini-batch, epoch
local minima?
backpropagation and vanishing grad
early stopping

5 Sélection d'un modèle

Deux types de modèle sont étudiés :

- des arbres de décision améliorés, introduits section 3, notés XGB;
- des réseaux de neurones profonds, introduits section 4, notés DNN.

Les hyperparamètres des XGBs sont :

- la profondeur maximale des arbres $N_{\text{max}}^{\text{prof.}}$;
- la quantité d'échantillons minimale dans une branche $N_{\text{min}}^{\text{échant.}}$;

- le nombre d'arbres $N_{\max}^{\text{estim.}}$;
- le gain minimal γ ;
- le taux d'apprentissage η ;
- la fonction de coût L ;
- la liste des variables d'entrée.

Les hyperparamètres des DNNs sont :

- le nombre de couches cachées N_L ;
- le nombre de neurones par couche cachée N_N ;
- la fonction d'activation des neurones (hormis celui de sortie) ;
- la fonction de coût L ;
- l'optimiseur ;
- le mode d'initiation des poids ;
- la liste des variables d'entrée.

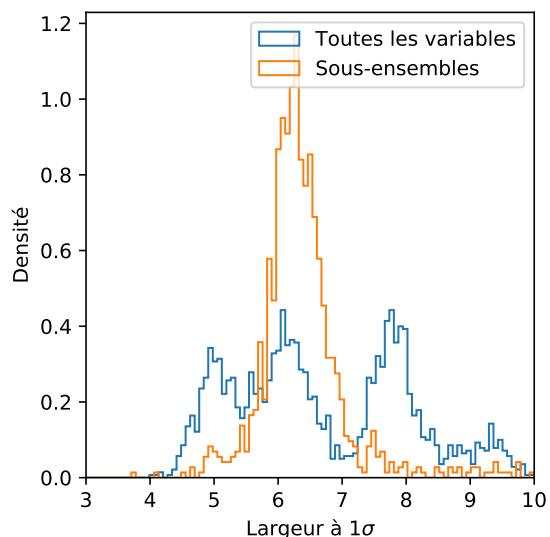
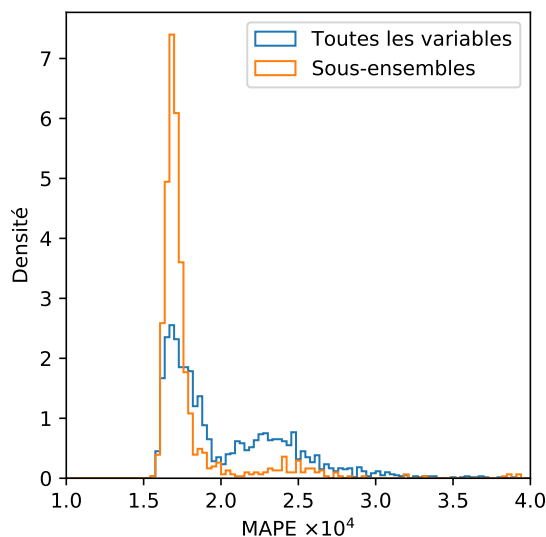
no customized loss yet.

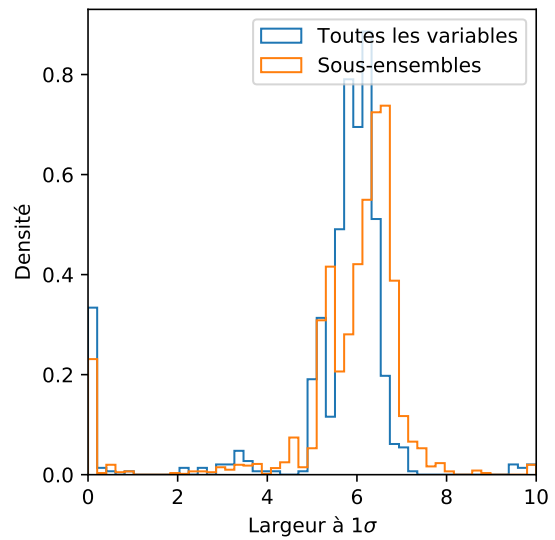
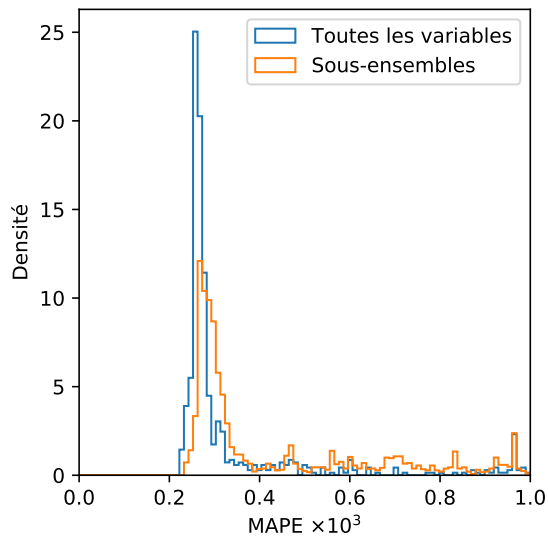
hard to get one single score to determine which model is the best : use mse, mae, mape, median diff, ± 1 or 2σ width ... low, medium, high and full mass regions as well.

5.1 Variables d'entrée

Model inputs : DNN not that sensible but XGB is better when having all of them, then use all inputs (give list) and not a subset of them.

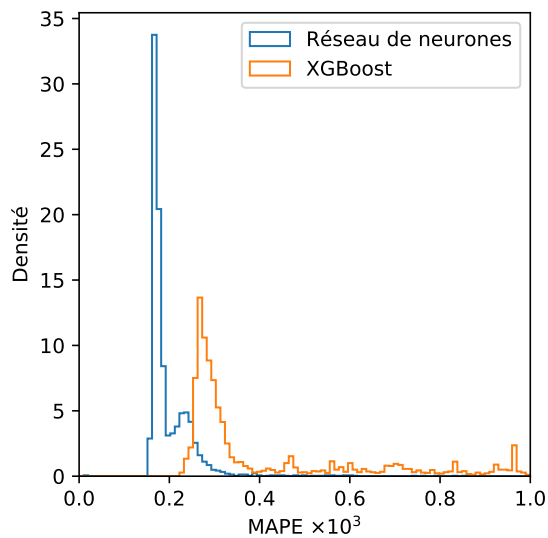
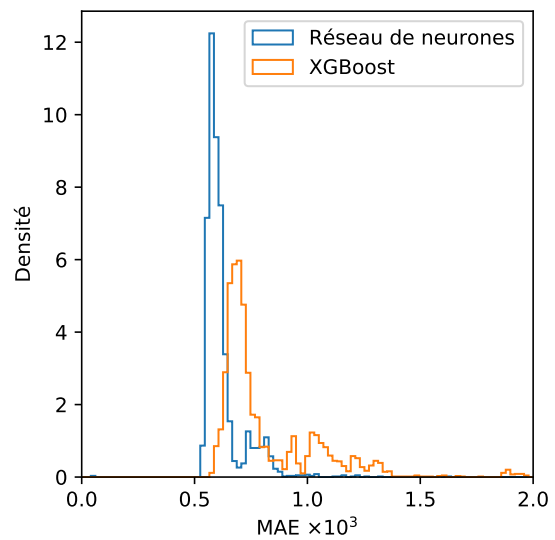
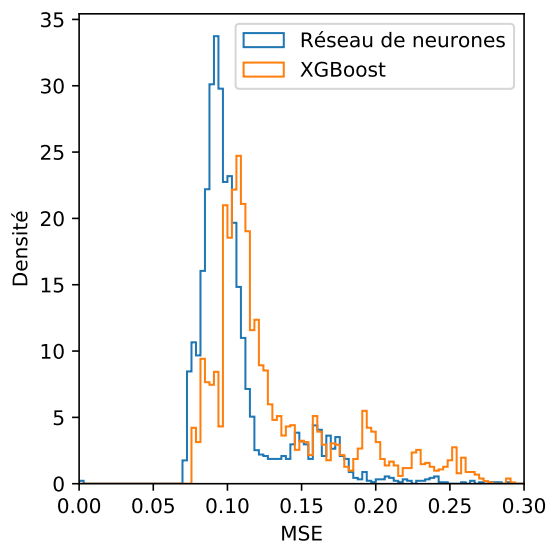
use plots with ref when relevant





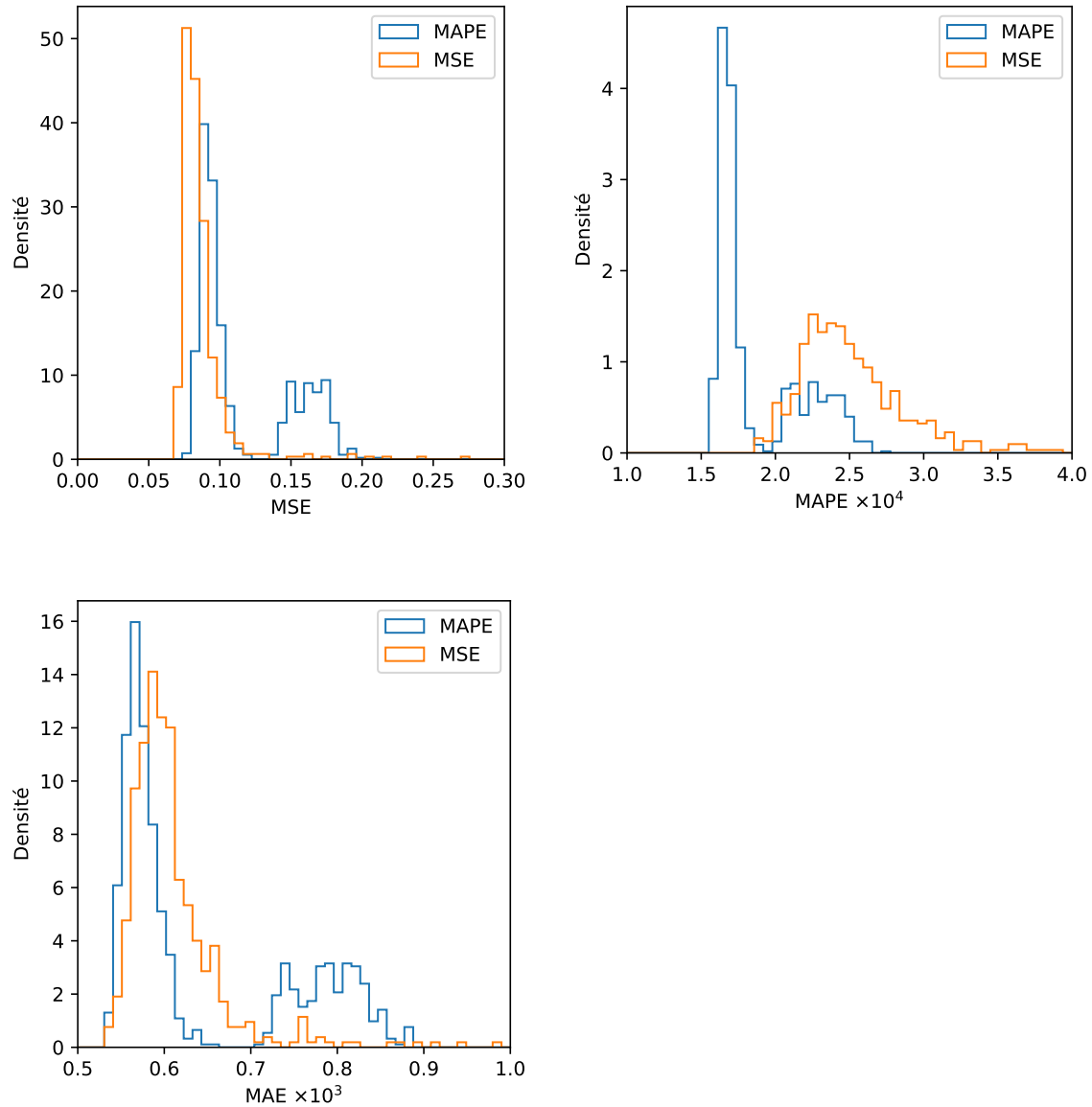
5.2 Type de modèle

DNN vs XGB : use DNN!



5.3 Fonction de coût

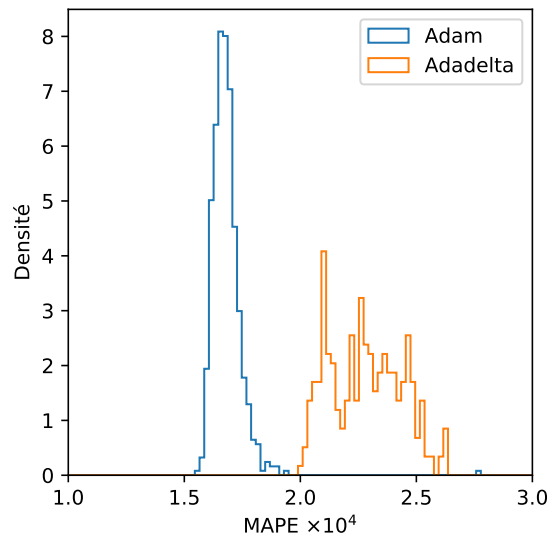
loss : when using a given loss, the corresponding models are of course better when using the loss as score.



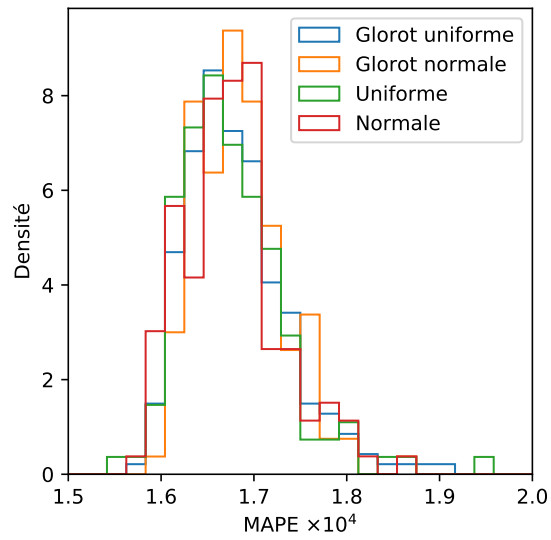
use mape loss, gives the better results

5.4 Initialisation des poids et optimisation

optimizer : Adam

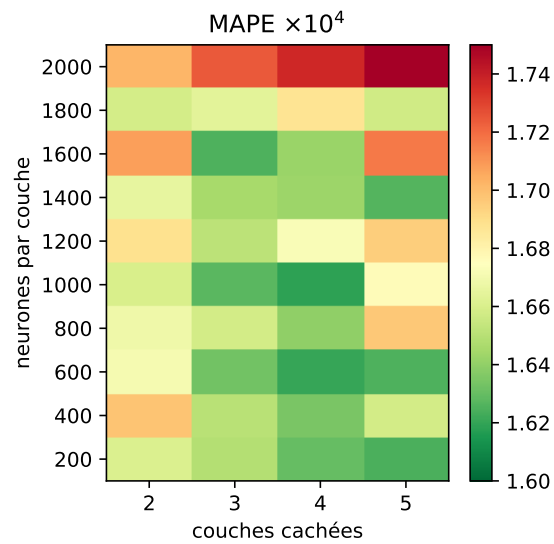


w init mode Glorot Uniform

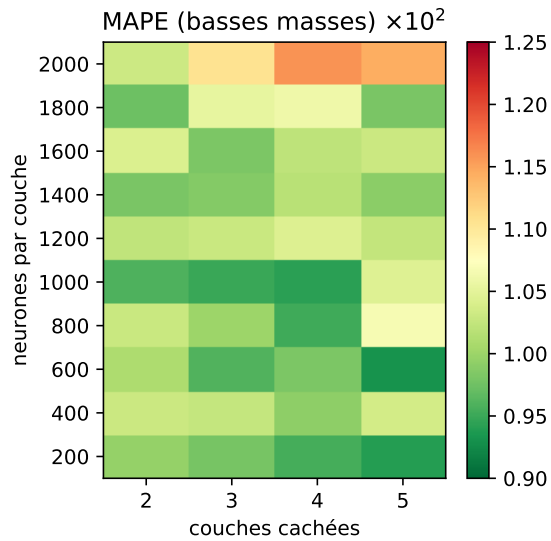


5.5 Structure

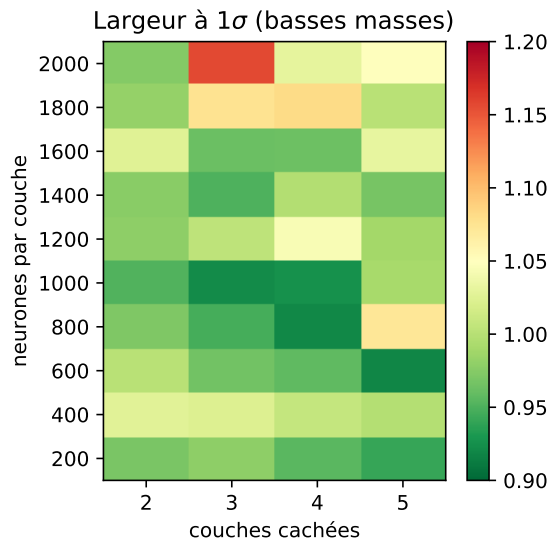
which structure?



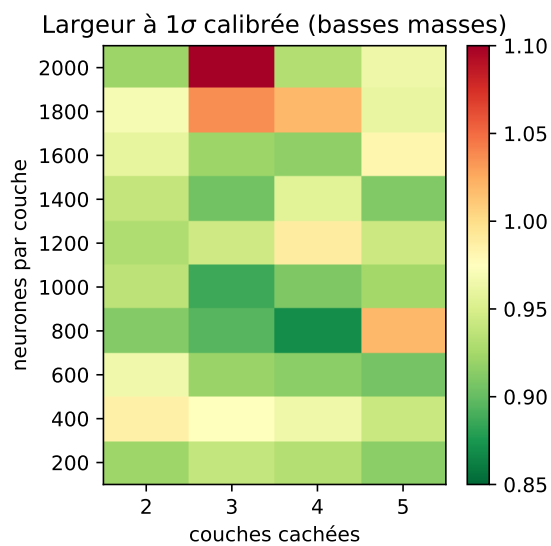
several possibilities, but the loss mass region contains the Z boson and is important



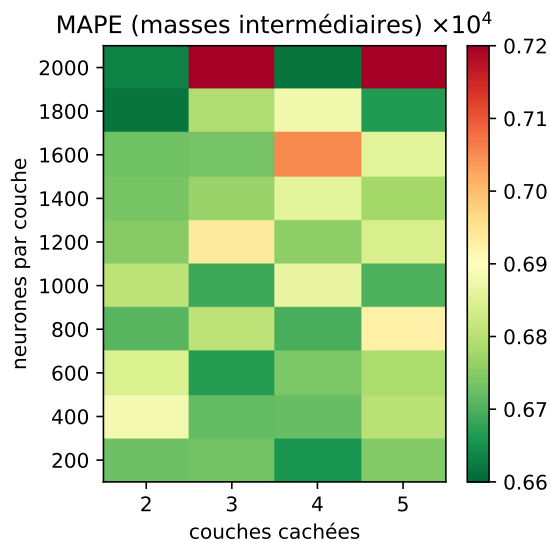
2x900 and 5x600 seem to be the best options, check the low mass resolution



5x600 seems good, check the low mass **calibrated** resolution



and in the medium mass region we have



3x1000 is the best compromise we found

5.6 Fonction d'activation

activation = softplus

6 Discussions

6.1 Effets de l'intervalle de masse

6.2 Effets de l'empilement

also show PU effect (see fig 2 and 3 from report 2020-11-20, update with new models and samples)

6.3 Effets de la reconstruction

show trained/tested on gen tau, gen tau decays, reco tau decays (=real), see fig 3 from report 2021-01-11

the model understand the physics, now it has to deal with the reco resolution and fakes.

6.4 Effets des faux taus hadroniques

6.5 Effets de la séparation des canaux

not relevant (fig3 report 2021-01-21)

6.6 Effets de bord

use the custom loss with boundaries cuts (basically all the report 2021-02-04)

Follow report from 2021-02-04 but for section 3 : We saw that predictions come out too low, which already is a motivation to put larger weights on higher masses, i.e. to weight by truth. Choosing $\sqrt{\text{truth}}$ is of course just a guess then

extend up to 1TeV using the tails

6.7 Modèle final

DEEPTAU

1 TeV

all inputs

activation softplus

```

loss mapesqrt_b
opti Adam
glorot uniform
3 layers of 1000 neurons
show reponses and 2d histo

```


7 Utilisation du modèle dans les analyses CMS

```

show distributions of mTtot and ML predictions
discuss
show limits
discuss

```

8 Conclusion

 To cite :

- DELPHES 3.4.2 [19, 20]?
- CMS Fast Simulation (FASTSIM) [21-24]
- PYTHIA 8.235 [29]
- FASTJET [37, 38]
- KERAS [39]
- TENSORFLOW [40]
- XGBOOST [33]
- W. SARLE. « Neural Networks and Statistical Models ». 1994. URL : https://people.orie.cornell.edu/davidr/or474/nn_sas.pdf
- P. BÄRTSCHI & coll. « Reconstruction of τ lepton pair invariant mass using an artificial neural network ». Nuclear Instruments and Methods in Physics Research **A929** (2019), p. 29-33. DOI : [10.1016/j.nima.2019.03.029](https://doi.org/10.1016/j.nima.2019.03.029). URL : <http://www.sciencedirect.com/science/article/pii/S0168900219303377>
- SVFIT [17]

L. TORTEROTOT, E. AŞILAR & C. BERNET. *Reconstruction of di-tau mass using Machine Learning*. URL : https://github.com/lucastorterotot/DL_for_HTT_mass

Références

- [1] DEEPMIND. *AlphaGo*. URL : <https://www.deepmind.com/research/case-studies/alphago-the-story-so-far>.
- [2] C. BERNET. *The Data Frog – Image Recognition : Dogs vs Cats !* URL : <https://thedatafrog.com/en/articles/dogs-vs-cats/>.
- [3] M. MIR. *House Prices Prediction Using Deep Learning*. URL : <https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154>.
- [4] G. TOUQUET. « Search for an additional neutral MSSM Higgs boson decaying to tau leptons with the CMS experiment ». Thèse de doct. Université Claude Bernard Lyon 1, oct. 2019. URL : <https://hal.archives-ouvertes.fr/tel-02526393>.
- [5] M. SCHAM. « Standard Model $H \rightarrow \tau\tau$ Analysis with a Neural Network Trained on a Mix of Simulation and Data Samples ». Mém. de mast. Fakultät für Physik des Karlsruher Instituts für Technologie (KIT), juin 2020. URL : <https://publish.etp.kit.edu/record/21993>.
- [6] T. KOPF. « Recoil Calibration as a Neural Network Task ». Mém. de mast. Fakultät für Physik des Karlsruher Instituts für Technologie (KIT), fév. 2019. URL : <https://publish.etp.kit.edu/record/21500>.

- [7] P. BALDI, P. SADOWSKI & D. WHITESON. « Enhanced Higgs Boson to $\tau^+\tau^-$ Search with Deep Learning ». *Physical Review Letters* **114**.11 (mar. 2015). DOI : [10.1103/physrevlett.114.111801](https://doi.org/10.1103/physrevlett.114.111801).
- [8] D. GUEST & coll. « Jet flavor classification in high-energy physics with deep neural networks ». *Physical Review* **D94**.11 (déc. 2016). DOI : [10.1103/physrevd.94.112002](https://doi.org/10.1103/physrevd.94.112002).
- [9] The CMS Collaboration. « Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV ». *Journal of Instrumentation* **13**.05 (mai 2018). DOI : [10.1088/1748-0221/13/05/p05011](https://doi.org/10.1088/1748-0221/13/05/p05011).
- [10] The CMS Collaboration. *DeepJet : deep learning based on physics objects for jet reconstruction*. URL : <https://twiki.cern.ch/twiki/bin/viewauth/CMS/DeepFlavour>.
- [11] The CMS Collaboration. « Performance of the DeepTau algorithm for the discrimination of taus against jets, electron, and muons » (oct. 2019). URL : <https://cds.cern.ch/record/2694158>.
- [12] J. ANDREJKOVIC & coll. « Measurement of Higgs(125) boson properties in decays to a pair of tau leptons with full Run II data using Machine-Learning techniques ». *CMS analysis Note* (sept. 2020). URL : https://cms.cern.ch/iCMS/jsp/db_notes/noteInfo.jsp?cmsnoteid=CMS%5C%20AN-2019/177.
- [13] J. ANDREJKOVIC & coll. « Multi-class neural network architecture and training for measurements of Higgs(125) boson decays to two tau leptons on full Run II data ». *CMS analysis Note* (mai 2020). URL : https://cms.cern.ch/iCMS/jsp/db_notes/noteInfo.jsp?cmsnoteid=CMS%5C%20AN-2019/178.
- [14] A. ELAGIN & coll. « A new mass reconstruction technique for resonances decaying to $\tau\tau$ ». *Nuclear Instruments and Methods in Physics Research* **A654**.1 (2011), p. 481-489. DOI : [10.1016/j.nima.2011.07.009](https://doi.org/10.1016/j.nima.2011.07.009).
- [15] A. J. BARR & coll. « Speedy Higgs boson discovery in decays to tau lepton pairs : $h \rightarrow \tau\tau$ ». *Journal of High Energy Physics* **2011**.10 (oct. 2011). DOI : [10.1007/JHEP10\(2011\)080](https://doi.org/10.1007/JHEP10(2011)080).
- [16] B. GRIPAIS & coll. « Reconstruction of Higgs bosons in the di-tau channel via 3-prong decay ». *Journal of High Energy Physics* **2013**.3 (mar. 2013). DOI : [10.1007/JHEP03\(2013\)106](https://doi.org/10.1007/JHEP03(2013)106).
- [17] L. BIANCHINI & coll. « Reconstruction of the Higgs mass in $H \rightarrow \tau\tau$ Events by Dynamical Likelihood techniques ». *Journal of Physics : Conference Series* **513**.2 (juin 2014). DOI : [10.1088/1742-6596/513/2/022035](https://doi.org/10.1088/1742-6596/513/2/022035).
- [18] P. BÄRTSCHI & coll. « Reconstruction of τ lepton pair invariant mass using an artificial neural network ». *Nuclear Instruments and Methods in Physics Research* **A929** (2019), p. 29-33. DOI : [10.1016/j.nima.2019.03.029](https://doi.org/10.1016/j.nima.2019.03.029). URL : <http://www.sciencedirect.com/science/article/pii/S0168900219303377>.
- [19] J. de FAVEREAU & coll. « DELPHES 3 : a modular framework for fast simulation of a generic collider experiment ». *Journal of High Energy Physics* **2** (fév. 2014). DOI : [10.1007/jhep02\(2014\)057](https://doi.org/10.1007/jhep02(2014)057).
- [20] A. MERTENS. « New features in DELPHES 3 ». *Journal of Physics : Conference Series* **608**.1 (2015). Sous la dir. de L. FIALA, M. LOKAJICEK & N. TUMOVA. DOI : [10.1088/1742-6596/608/1/012045](https://doi.org/10.1088/1742-6596/608/1/012045).
- [21] S. ABDULLIN & coll. « The Fast Simulation of the CMS Detector at LHC ». *Journal of Physics : Conference Series* **331**.3 (déc. 2011). DOI : [10.1088/1742-6596/331/3/032049](https://doi.org/10.1088/1742-6596/331/3/032049).
- [22] A. GIAMMANCO. « The Fast Simulation of the CMS Experiment ». *Journal of Physics : Conference Series* **513**.2 (juin 2014). DOI : [10.1088/1742-6596/513/2/022012](https://doi.org/10.1088/1742-6596/513/2/022012).
- [23] M. KOMM. « Fast emulation of track reconstruction in the CMS simulation ». *Journal of Physics : Conference Series* **898** (oct. 2017). DOI : [10.1088/1742-6596/898/4/042034](https://doi.org/10.1088/1742-6596/898/4/042034).
- [24] S. SEKMEN. *Recent Developments in CMS Fast Simulation*. 2017. arXiv : [1701.03850](https://arxiv.org/abs/1701.03850).
- [25] S. AGOSTINELLI & coll. « GEANT4 – A simulation toolkit ». *Nuclear Instruments and Methods in Physics Research* **A506**.3 (2003), p. 250-303. DOI : [10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8). URL : <http://www.sciencedirect.com/science/article/pii/S0168900203013688>.

- [26] J. ALLISON & coll. « GEANT4 developments and applications ». *IEEE Transactions on Nuclear Science* **53.1** (fév. 2006), p. 270-278. DOI : [10.1109/tns.2006.869826](https://doi.org/10.1109/tns.2006.869826).
- [27] J. ALLISON & coll. « Recent developments in GEANT4 ». *Nuclear Instruments and Methods in Physics Research A* **835** (2016), p. 186-225. DOI : [10.1016/j.nima.2016.06.125](https://doi.org/10.1016/j.nima.2016.06.125). URL : <http://www.sciencedirect.com/science/article/pii/S0168900216306957>.
- [28] E. AŞILAR. *How to produce nanoAOD events of $h \rightarrow \tau\tau$ where Higgs has a 130 GeV mass*. URL : https://github.com/easilar/cmssw/blob/from-CMSSW_10_2_22/README.
- [29] T. SJÖSTRAND & coll. « An Introduction to PYTHIA 8.2 ». *Computer Physics Communications* **191** (2015), p. 159-177. DOI : [10.1016/j.cpc.2015.01.024](https://doi.org/10.1016/j.cpc.2015.01.024). arXiv : [1410.3012](https://arxiv.org/abs/1410.3012) [hep-ph].
- [30] The CMS Collaboration. « Event generator tunes obtained from underlying event and multiparton scattering measurements ». *European Physical Journal C* **76.3** (2016). DOI : [10.1140/epjc/s10052-016-3988-x](https://doi.org/10.1140/epjc/s10052-016-3988-x). arXiv : [1512.00815](https://arxiv.org/abs/1512.00815) [hep-ex].
- [31] The CMS Collaboration. « Extraction and validation of a new set of CMS PYTHIA 8 tunes from underlying-event measurements ». *European Physical Journal C* **80** (mar. 2019). DOI : [10.1140/epjc/s10052-019-7499-4](https://doi.org/10.1140/epjc/s10052-019-7499-4). URL : <https://cds.cern.ch/record/2669320>.
- [32] LHC Higgs Cross Section Working Group. « Higgs Properties ». *Handbook of LHC Higgs Cross Sections. 3*. CERN Yellow Reports : Monographs. Geneva : CERN, 2013. DOI : [10.5170/CERN-2013-004](https://doi.org/10.5170/CERN-2013-004). URL : <https://cds.cern.ch/record/1559921>.
- [33] T. CHEN & C. GUESTIN. « XGBOOST : A Scalable Tree Boosting System ». *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (août 2016). DOI : [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [34] *Kaggle Competitions*. URL : <https://www.kaggle.com/competitions>.
- [35] I. GOODFELLOW, Y. BENGIO & A. COURVILLE. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [36] X. GLOROT & Y. BENGIO. « Understanding the difficulty of training deep feedforward neural networks ». *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Sous la dir. d'Y. W. TEH & M. TITTERINGTON. **9**. Proceedings of Machine Learning Research. PMLR, mai 2010, p. 249-256. URL : <http://proceedings.mlr.press/v9/glorot10a.html>.
- [37] M. CACCIARI, G. P. SALAM & G. SOYEZ. « FASTJET user manual ». *European Physical Journal C* **72** (nov. 2012). DOI : [10.1140/epjc/s10052-012-1896-2](https://doi.org/10.1140/epjc/s10052-012-1896-2). arXiv : [1111.6097](https://arxiv.org/abs/1111.6097) [hep-ph].
- [38] M. CACCIARI & G. P. SALAM. « Dispelling the N^3 myth for the k_T jet-finder ». *Physics Letters B* **641.1** (sept. 2006), p. 57-61. DOI : [10.1016/j.physletb.2006.08.037](https://doi.org/10.1016/j.physletb.2006.08.037).
- [39] F. CHOLLET & coll. KERAS. <https://keras.io>. 2015.
- [40] M. ABADI & coll. TENSORFLOW : *Large-scale machine learning on heterogeneous distributed systems*. Software available from tensorflow.org. 2015. URL : <https://www.tensorflow.org/>.
- [41] W. SARLE. « Neural Networks and Statistical Models ». 1994. URL : https://people.orie.cornell.edu/davidr/or474/nn_sas.pdf.
- [42] L. TORTEROTOT, E. AŞILAR & C. BERNET. *Reconstruction of di-tau mass using Machine Learning*. URL : https://github.com/lucastorterotot/DL_for_HTT_mass.

