École Polytechnique de Thiès

Génie Informatique et Télécommunications

Rapport de Projet

Apprentissage profond bayésien : réseaux bayésiens, méthodes d'inférence approximatives (MC, VI, ELBO), dropout bayésien, incertitude prédictive...

Présenté par **Thierno Daouda LY**

Encadreuse :

Ndèye Fatou NGOM, PhD

Année académique 2024–2025

Résumé

L'apprentissage profond bayésien constitue une approche prometteuse pour intégrer la gestion de l'incertitude dans les réseaux de neurones profonds. Contrairement à l'apprentissage classique, cette méthode considère les paramètres du réseau comme des variables aléatoires, permettant ainsi de modéliser des distributions de probabilité sur les poids. Grâce aux outils d'inférence approximative tels que l'échantillonnage Monte Carlo, l'inférence variationnelle et la borne inférieure de l'évidence (ELBO), les réseaux bayésiens deviennent exploitables même dans des architectures complexes. Des techniques comme le dropout bayésien permettent d'implémenter ces idées de manière simple et efficace. Cette capacité à fournir des prédictions calibrées et à quantifier l'incertitude rend l'apprentissage profond bayésien particulièrement adapté aux domaines critiques tels que la médecine, la robotique ou la finance.

Mots-clés : Apprentissage profond, probabilités bayésiennes, inférence approximative, réseau de neurones bayésien, incertitude, dropout bayésien, ELBO.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

IA Intelligence artificielle
RB Réseau Bayésien
DAG Directed acyclic graph*Graphe acyclique orienté*
DPC Distribution de probabilité conjointe
CPT Conditional Probability Table*Table de probabilités conditionnelles*
MC Monte Carlo
VI Variational Inference*Inférence variationnelle*
MCMC Monte Carlo par chaînes de Markov
HMC Hamiltonian Monte Carlo
CNN Convolutional neural networks*Réseaux de neurones convolutionnels*
RNN Réseaux de neurones récurrents
ELBO Evidence Lower Bound*Borne inférieure de l'évidence*
VAE Variational Autoencoders
OoD Out-of-distribution*Entrées hors distribution*
AUC Area under the curve
NLL Negative Log-Likelihood

I Introduction

I.2 Motivation et contexte

L'apprentissage profond a connu un essor spectaculaire au cours de la dernière décennie, en s'imposant dans des domaines variés comme la vision par ordinateur, le traitement du langage naturel ou les jeux stratégiques. Ces performances s'expliquent par la capacité des réseaux de neurones à apprendre automatiquement des représentations hiérarchiques à partir de grandes quantités de données. Cependant, les réseaux classiques reposent sur des *estimations ponctuelles* de leurs paramètres, ce qui ne permet pas de mesurer la fiabilité des prédictions. Or, dans des contextes sensibles ou incertains, cette limitation peut entraîner des décisions risquées. Un enjeu crucial en apprentissage automatique est donc la **quantification de l'incertitude** : être capable d'exprimer le degré de confiance associé à une prédiction. Cette compétence est essentielle pour concevoir des systèmes robustes, fiables et interprétables. L'**apprentissage bayésien** répond à cet enjeu en proposant un cadre probabiliste rigoureux. En traitant les paramètres du modèle comme des variables aléatoires, il permet de modéliser l'incertitude de manière naturelle et principielle.

I.3 Applications

Ainsi, l'intégration de la modélisation bayésienne dans les réseaux de neurones ouvre la voie à des applications critiques où la quantification de l'incertitude est indispensable. Parmi les domaines les plus concernés, on peut citer :

- **Médecine et santé :** En diagnostic assisté par IA, estimer la confiance d'un modèle permet de détecter les cas incertains et de les référer à des spécialistes humains, réduisant ainsi les erreurs graves.
- **Conduite autonome :** Dans les systèmes embarqués de navigation ou de perception, des décisions doivent être prises dans des environnements dynamiques et partiellement observés. Un réseau bayésien peut éviter des actions risquées en signalant son incertitude.
- **Détection de fraudes et cybersécurité :** Identifier des transactions ou comportements suspects nécessite des modèles capables de détecter les anomalies tout en tenant compte du doute sur les classifications.
- **Robotique :** L'incertitude sur la localisation, les capteurs ou les actions d'un robot peut être intégrée dans un processus décisionnel bayésien pour améliorer la robustesse et la sécurité.
- **Active learning :** Dans un contexte d'apprentissage interactif, les modèles bayésiens sont utilisés pour sélectionner les exemples les plus informatifs à étiqueter, en se basant sur leur incertitude.
- **Modélisation scientifique et physique :** En sciences fondamentales, de nombreux phénomènes sont modélisés à partir d'observations bruitées. Les réseaux bayésiens permettent d'estimer des intervalles de crédibilité pour les paramètres du modèle.

II. L'inférence bayésienne

L'inférence bayésienne constitue le cœur de l'apprentissage probabiliste. Elle vise à actualiser nos croyances sur les paramètres d'un modèle après l'observation de nouvelles données. Contrairement à l'approche fréquentiste, qui considère les paramètres comme des quantités fixes à estimer, l'approche bayésienne les traite comme des variables aléatoires dotées d'une distribution de probabilité. Autrement dit, au lieu de chercher une seule valeur "optimale" des poids d'un réseau (comme en apprentissage classique), on cherche à déterminer la **distribution a posteriori** des poids $P(W \mid D)$, qui reflète notre incertitude après avoir observé les données D.

II.1 Règles de la chaîne et de la somme

Avant de manipuler la fonction de base en probabilités bayésiennes, il est essentiel de maîtriser deux règles fondamentales que sont la **règle de la chaîne** et la **règle de la somme**.

Règle de la chaîne

La règle de la chaîne permet de factoriser une distribution de probabilité conjointe en un produit de distributions conditionnelles :

$$\begin{aligned} P(X_1, X_2) &= P(X_1) \cdot P(X_2 \mid X_1) \\ P(X_1, X_2, X_3) &= P(X_1) \cdot P(X_2 \mid X_1) \cdot P(X_3 \mid X_1, X_2) \\ &\vdots \\ P(X_1, X_2, \ldots, X_n) &= \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1}) \end{aligned}$$

Règle de la somme

La règle de la somme (ou marginalisation) indique que si nous avons une distribution conjointe sur plusieurs variables, nous pouvons obtenir la distribution marginale d'une variable en *intégrant* (ou en sommant, pour les cas discrets) sur les autres variables :

$$\begin{aligned} P(X_1) &= \sum_{x_2} P(X_1, X_2) \quad (\text{ou } \int P(X_1, X_2)\, dx_2) \\ P(X_1) &= \sum_{x_2} \sum_{x_3} P(X_1, X_2, X_3) \\ &\vdots \\ P(X_1) &= \sum_{x_2} \cdots \sum_{x_n} P(X_1, X_2, \ldots, X_n) \end{aligned}$$

Dans le cas continu, on remplace les sommes par des intégrales :

$$P(X_1) = \int \cdots \int P(X_1, X_2, \ldots, X_n)\, dx_2 \cdots dx_n$$

Ces deux règles sont les fondements de nombreux algorithmes d'inférence probabiliste, notamment dans les réseaux bayésiens et l'apprentissage profond bayésien.

II.2 Théorème de Bayes

Le fondement de l'approche bayésienne repose sur le **théorème de Bayes**, qui permet de mettre à jour nos croyances sur des paramètres inconnus à partir de données observées :

$$P(\theta|D) = \frac{P(D|\theta)\,P(\theta)}{P(D)}$$

où :

— $P(\theta)$ est la **distribution a priori** (*prior*), qui représente notre connaissance ou nos croyances sur le paramètre θ **avant** l'observation des données D. Elle peut être subjective (basée sur des connaissances préalables) ou objective (non informative).

— $P(D|\theta)$ est la **vraisemblance** (*likelihood*), qui indique la **probabilité d'observer les données** D sachant que le paramètre est θ. Elle reflète le lien entre le modèle et les observations.

— $P(\theta|D)$ est la **distribution a posteriori** (*posterior*), obtenue après observation des données. Elle combine l'information a priori avec l'information contenue dans les données pour mettre à jour nos croyances sur θ.

— $P(D)$ est l'**évidence** (ou **vraisemblance marginale**), une constante de normalisation donnée par :

$$P(D) = \int P(D|\theta)\,P(\theta)\,d\theta$$

Elle assure que $P(\theta|D)$ est une distribution de probabilité valide (somme ou intégrale égale à 1) et permet aussi de comparer différents modèles.

II.3 Cycle itératif de l'inférence bayésienne

L'inférence bayésienne repose sur un cycle en quatre étapes, qui permet de raffiner progressivement notre connaissance du modèle à mesure que de nouvelles données sont observées :

1. **Prior** ($P(W)$) : Représente nos connaissances initiales (ou croyances) sur les paramètres du modèle, avant toute observation.
2. **Données observées** (D) : Nouvelles informations collectées à partir de l'expérience ou d'un ensemble d'apprentissage.
3. **Posterior** ($P(W \mid D)$) : Mise à jour du prior à l'aide des données, selon la règle de Bayes :

$$P(W \mid D) = \frac{P(D \mid W) \cdot P(W)}{P(D)}$$

où $P(D \mid W)$ est la vraisemblance des données et $P(D)$ est le facteur de normalisation (évidence).

4. **Itération** : Le posterior devient le nouveau prior lorsqu'on incorpore de nouvelles données. Le processus peut alors recommencer.

Ce mécanisme permet d'intégrer progressivement l'information, rendant les modèles bayésiens naturellement adaptatifs et capables de quantifier l'incertitude à chaque étape du processus d'apprentissage.

II.4 Choix du prior

Le prior $P(\theta)$ représente notre croyance initiale sur les paramètres du modèle, avant toute observation. Son choix a une influence directe sur l'inférence, notamment lorsque les données sont limitées ou bruitées.

Types de priors : On distingue généralement trois grandes catégories de priors :

1. **Prior informatif** : encode des connaissances précises sur le paramètre à estimer. Il est utile lorsque des informations fiables sont disponibles (données historiques, expertises, théories établies). Toutefois, un prior mal choisi peut conduire à des inférences biaisées.
2. **Prior non-informatif** (ou vague, objectif) : il cherche à minimiser l'influence a priori pour laisser les données guider l'inférence. Typiquement, on utilise une distribution uniforme ou un prior de Jeffreys. Il est adapté en absence de connaissance préalable.
3. **Prior faiblement informatif** : intermédiaire entre les deux précédents, il intègre une information vague mais raisonnable, afin de régulariser l'apprentissage sans surcontraindre le modèle.

Principe fondamental : L'influence du prior décroît naturellement avec l'augmentation de la quantité de données.

En d'autres termes, plus l'on observe de données, plus le rôle du prior s'estompe au profit de la vraisemblance $P(D \mid \theta)$, et le posterior devient plus concentré autour des valeurs les plus plausibles.

II.5 Familles conjuguées

Dans le cadre bayésien, une famille de distributions est dite **conjuguée** à une distribution de vraisemblance si, lorsqu'on utilise un prior dans cette famille, la distribution *a posteriori* appartient à la même famille. Ce concept permet d'obtenir une mise à jour analytique simple des croyances après observation des données, sans recourir à des méthodes numériques coûteuses. Plus formellement, si la vraisemblance est $P(x \mid \theta)$ et le prior est $P(\theta)$, alors on dit que $P(\theta)$ est une **loi conjuguée** si la posteriori $P(\theta \mid x) \propto P(x \mid \theta)P(\theta)$ est de la même forme fonctionnelle que $P(\theta)$.

Exemples classiques de familles conjuguées :

— **Binomiale – Bêta** : Si les observations suivent une loi binomiale et que le prior est une loi bêta, alors la loi a posteriori est aussi une bêta.

$$x \sim \text{Bin}(n, \theta), \quad \theta \sim \text{Beta}(\alpha, \beta) \Rightarrow \theta \mid x \sim \text{Beta}(\alpha + x, \beta + n - x)$$

— **Poisson – Gamma** : Pour des données discrètes comptées modélisées par une loi de Poisson, un prior gamma est conjugué.

$$x \sim \text{Poisson}(\lambda), \quad \lambda \sim \text{Gamma}(\alpha, \beta) \Rightarrow \lambda \mid x \sim \text{Gamma}(\alpha + x, \beta + 1)$$

— **Normale (connue) – Normale** : Si la variance est connue, un prior normal sur la moyenne reste normal après observation.

$$x \sim \mathcal{N}(\mu, \sigma^2), \quad \mu \sim \mathcal{N}(\mu_0, \tau^2) \Rightarrow \mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2)$$

— **Multinomiale – Dirichlet** : Pour des données catégorielles, la loi de Dirichlet est conjuguée à la loi multinomiale.

$$\mathbf{x} \sim \text{Multinomial}(n, \boldsymbol{\theta}), \quad \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}) \Rightarrow \boldsymbol{\theta} \mid \mathbf{x} \sim \text{Dir}(\boldsymbol{\alpha} + \mathbf{x})$$

Avantages des familles conjuguées : L'utilisation de familles conjuguées présente plusieurs avantages majeurs dans l'inférence bayésienne. Tout d'abord, elles permettent une **mise à jour analytique rapide** des distributions a posteriori, sans nécessiter de calculs numériques complexes. Ensuite, elles offrent une **simplicité computationnelle** appréciable, ce qui les rend particulièrement adaptées à des contextes où les ressources de calcul sont limitées. Enfin, les paramètres des lois conjuguées ont souvent une **interprétation intuitive**, ce qui facilite la spécification des hyperparamètres en fonction des connaissances a priori ou du contexte applicatif.

Cependant, dans des modèles plus complexes, on ne peut souvent pas utiliser de familles conjuguées, ce qui rend nécessaire l'usage d'approximations numériques.

III Réseaux bayésiens

Un **Réseau Bayésien** (RB) est un graphe dirigé acyclique (en anglais *directed acyclic graph* ou DAG) dont les sommets représentent des variables aléatoires, et les arêtes modélisent les dépendances conditionnelles entre ces variables. Il fournit une représentation factorisée et compacte d'une distribution de probabilité conjointe (DPC).

Formellement, une structure d'un réseau bayésien est un graphe $\mathcal{G}$ orienté acyclique dont les sommets sont des variables aléatoires $X_1, \ldots, X_n$. Soit $\mathrm{Pa}_{\mathcal{G}}(X_i)$ l'ensemble des parents de X_i dans $\mathcal{G}$. Le graphe encode les *indépendances conditionnelles locales* suivantes :

$$\boxed{X_i \perp \text{Non-Descendants}(X_i) \mid \mathrm{Pa}_{\mathcal{G}}(X_i)}$$

Autrement dit, chaque variable est indépendante de ses non-descendants, à condition de connaître ses parents dans le graphe. Cela permet de factoriser la distribution conjointe selon la règle de la chaîne :

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}_{\mathcal{G}}(X_i))$$

Exemple illustratif. La note obtenue par un élève ingénieur dépend à la fois de son niveau d'intelligence et de la difficulté du cours suivi. Lorsqu'il sollicite une lettre de recommandation de la part de son professeur, ce dernier s'appuie uniquement sur la note de l'étudiant pour la rédiger — cette lettre peut également être influencée par des facteurs aléatoires tels que le stress, l'humeur du professeur ou d'autres événements stochastiques non observables. D'après cette description, les variables aléatoires suivantes sont en jeu :

— Difficulté (d) : facile ou difficile
— Intelligence (i) : faible ou élevée
— Note (g) : faible, moyenne ou élevée
— SAT (s) : bas ou haut
— Lettre (l) : faible ou forte

La distribution conjointe $P(D, I, G, S, L)$ contient $2 \times 2 \times 3 \times 2 \times 2 = 48$ configurations possibles.
Ce réseau permet la factorisation suivante :

$$P(D, I, G, S, L) = P(D)P(I)P(G|D, I)P(S|I)P(L|G)$$

Un réseau bayésien possible pour cet exemple est :

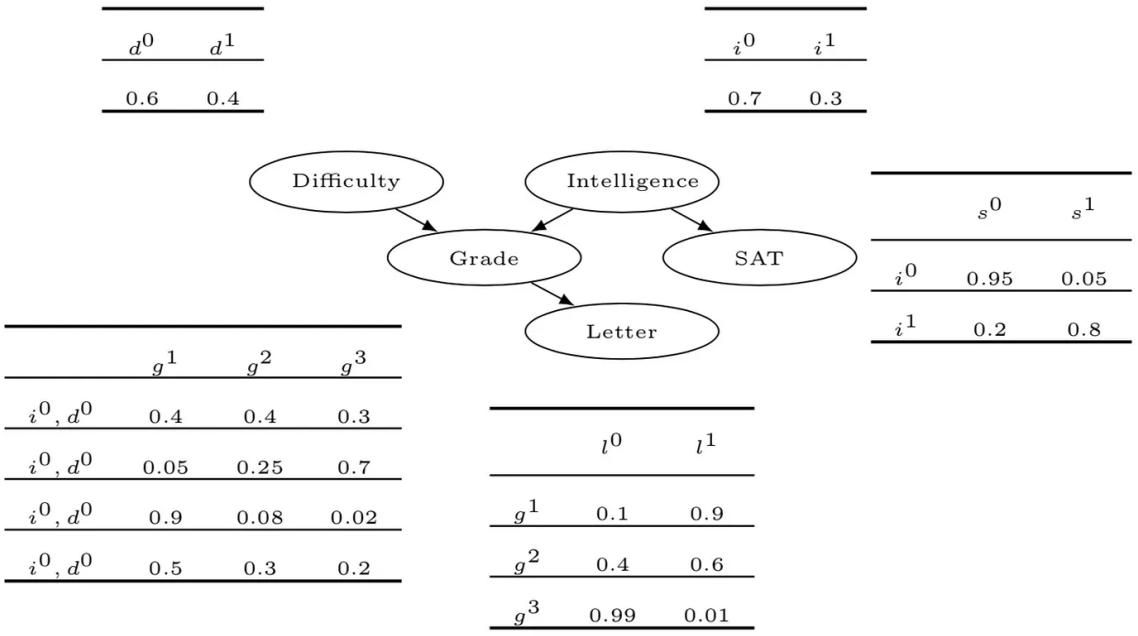

FIGURE II.1 : Réseau bayésien illustratif pour le problème d'admission universitaire incluant les tables de probabilités conditionnelles.

Chaque nœud a une table de probabilité conditionnelle (CPT) :
— $P(D)$ et $P(I)$ sont des distributions marginales.
— $P(G|D, I)$ est spécifitée pour chaque combinaison (d, i).
— $P(S|I)$ est donnée pour chaque valeur de i.
— $P(L|G)$ est donnée pour chaque valeur de g.

Une fois toutes les CPT connues, on peut calculer n'importe quelle probabilité jointe. Par exemple :

$$P(d_0, i_1, g_2, s_1, l_1) = P(d_0)P(i_1)P(g_2|d_0, i_1)P(s_1|i_1)P(l_1|g_2)$$

Lien avec l'apprentissage profond : Dans un réseau de neurones bayésien (BNN), les poids W du réseau sont considérés comme des variables aléatoires. La formulation bayésienne devient alors :
— **a priori** : $P(W)$ exprime nos croyances initiales sur les poids.
— **vraisemblance** : $P(D|W)$ mesure la compatibilité des données avec les poids.
— **a posteriori** : $P(W|D) = \dfrac{P(D|W)P(W)}{P(D)}$ est la distribution des poids données sur les données.
— **évidence** : $P(D) = \int P(D|W)P(W)\,dW$ est une constante de normalisation.

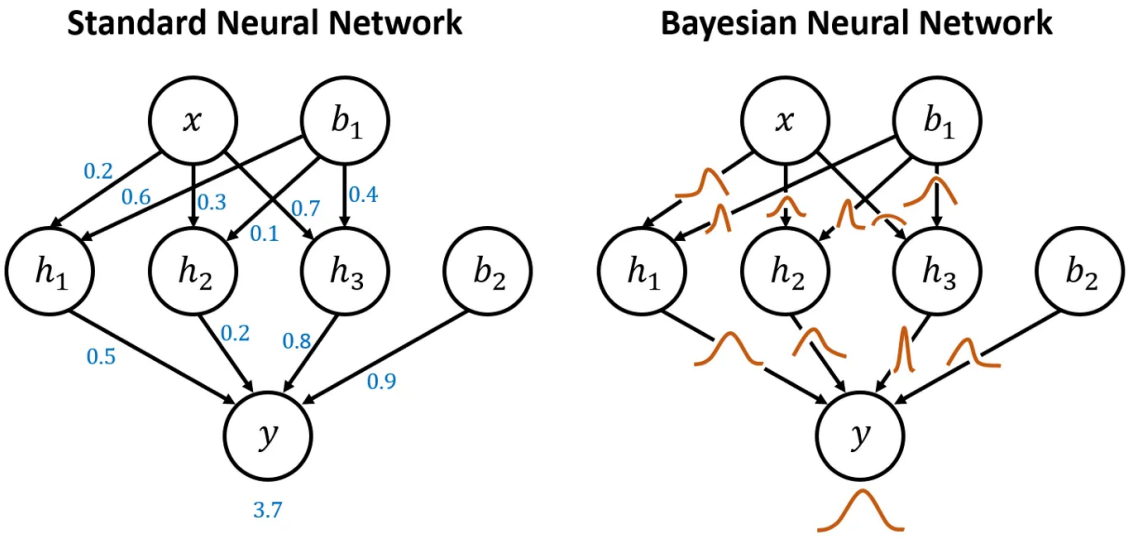

FIGURE II.2 : Structure comparative entre un réseau de neurones classique et un réseau de neurones bayésien. Les poids sont représentées par des distributions dans le modèle bayésien.

IV Méthodes d'inférence approximatives

L'inférence bayésienne dans les réseaux de neurones se fait à travers de techniques d'approximation. Les deux approches principales sont **les méthodes de Monte Carlo (MC)** et **l'inférence variationnelle (VI)**.

IV.1 Méthodes de Monte Carlo (MC)

IV.1.1 Méthodes Monte Carlo par chaînes de Markov (MCMC)

Principe de l'échantillonnage

Les méthodes MCMC visent à approximer une distribution de probabilité intractable — ici, le posterior bayésien $P(W|D)$ — en produisant un ensemble d'échantillons $W^{(1)}, W^{(2)}, \ldots, W^{(T)}$ qui suivent approximativement cette loi. L'idée centrale est de construire une *chaîne de Markov* dont la distribution stationnaire est précisément $P(W|D)$. En générant suffisamment d'échantillons, on peut alors estimer des quantités d'intérêt comme :

$$\mathbb{E}_{P(W|D)}[f(W)] \approx \frac{1}{T}\sum_{t=1}^{T} f(W^{(t)})$$

où f est une fonction (par exemple, la prédiction du réseau pour une entrée donnée).

Algorithme de Metropolis-Hastings

L'un des algorithmes MCMC les plus fondamentaux est Metropolis-Hastings. Il fonctionne comme suit :

— On propose un nouveau point $W^* \sim q(W^*|W^{(t)})$ à partir d'une distribution de proposition q.

— On calcule un taux d'acceptation :

$$\alpha = \min\left(1, \frac{P(D|W^*)P(W^*)q(W^{(t)}|W^*)}{P(D|W^{(t)})P(W^{(t)})q(W^*|W^{(t)})}\right)$$

— On accepte W^* avec probabilité α, sinon on garde $W^{(t+1)} = W^{(t)}$.

Ce processus permet de construire une suite d'échantillons corrélés qui convergent, sous certaines conditions, vers le posterior $P(W|D)$.

Hamiltonian Monte Carlo (HMC)

L'algorithme de Metropolis-Hastings souffre d'un problème majeur dans les espaces de grande dimension : les propositions sont souvent rejetées, ce qui ralentit la convergence. L' **Hamiltonian Monte Carlo (HMC)** est une amélioration qui utilise des notions de mécanique classique pour explorer l'espace plus efficacement.

Chaque poids W est associé à un moment p, et l'on définit une énergie hamiltonienne :

$$\mathcal{H}(W, p) = U(W) + K(p)$$

où :

— $U(W) = -\log P(W|D)$ est l'énergie potentielle (négatif du log-posterior),

— $K(p) = \frac{1}{2}p^\top M^{-1}p$ est l'énergie cinétique, M étant une matrice de masse.

On simule ensuite l'évolution du système selon les équations de Hamilton, typiquement avec le *leapfrog integrator*, sur plusieurs pas de temps, ce qui permet de proposer des mouvements dans l'espace qui conservent l'énergie, et donc sont plus susceptibles d'être acceptés.

HMC est bien plus efficace que Metropolis-Hastings en haute dimension, mais demande de calculer le gradient de $\log P(W|D)$, ce qui est coûteux pour les grands réseaux.

Avantages et inconvénients pour les réseaux profonds

Avantages - Les MCMC :

— fournissent une estimation asymptotiquement exacte du posterior.

— permettent une quantification rigoureuse de l'incertitude.

— bien adaptées aux petits modèles ou aux cas à haute incertitude critique (e.g. santé).

Inconvénients - Les MCMC sont :

— très coûteuses en temps de calcul : la convergence de la chaîne peut nécessiter des milliers ou millions d'itérations.

— peu scalables : difficilement applicables aux grands réseaux (CNN, Transformer...).

— stockage coûteux : nécessite de conserver de nombreux échantillons en mémoire.

En pratique, les méthodes MCMC sont rarement utilisées directement pour les réseaux neuronaux profonds. Elles servent plutôt de référence ("gold standard") pour évaluer d'autres méthodes plus rapides, comme l'inférence variationnelle ou les approches par dropout stochastique.

IV.1.2 Monte Carlo Dropout (MC Dropout)

Interprétation bayésienne du dropout

Le **dropout**, introduit à l'origine comme une technique de régularisation pour les réseaux de neurones, consiste à désactiver aléatoirement certains neurones lors de l'entraînement, selon une probabilité p. Cependant, Gal et Ghahramani (2016) ont montré que cette procédure pouvait être interprétée comme une *approximation bayésienne*.

Plus précisément, ils démontrent que le dropout appliqué à chaque couche est équivalent à une **inférence variationnelle** dans un modèle bayésien profond, où la distribution a posteriori des poids est approximée

par une famille de distributions aléatoires induites par les masques de dropout. Autrement dit, on approxime $P(W|D)$ par une distribution $q(W)$ construite implicitement par la stochasticité des masques binaires appliqués aux couches du réseau.

Échantillonnage des sous-réseaux
Une fois le modèle entraîné avec dropout, on peut le tester de manière stochastique : on **garde actif le dropout à l'inférence** et on effectue plusieurs passes (forward passes) du même exemple x, chaque passe activant un sous-réseau différent. Cela revient à échantillonner des poids $W^{(t)} \sim q(W)$, ce qui nous permet d'estimer la distribution prédictive :

$$P(y|x, D) \approx \frac{1}{T} \sum_{t=1}^{T} P(y|x, W^{(t)})$$

où chaque $W^{(t)}$ correspond à une réalisation stochastique du réseau avec dropout actif.

Cette méthode est appelée **Monte Carlo Dropout (MC Dropout)** car elle utilise une estimation de type Monte Carlo sur les sorties du modèle pour approximer les prédictions bayésiennes.

Estimation de l'incertitude par MC Dropout
L'un des atouts majeurs de MC Dropout est sa capacité à **quantifier l'incertitude prédictive**. En effet, la moyenne et la variance des prédictions $P(y|x, W^{(t)})$ sur les passes aléatoires permettent de mesurer :

— La **moyenne prédictive** :

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} y^{(t)}$$

— L'**incertitude épistémique** (du modèle), estimée par la variance entre les sorties :

$$\mathrm{Var}(y) = \frac{1}{T} \sum_{t=1}^{T} \left(y^{(t)} - \hat{y} \right)^2$$

Cela permet, par exemple, de construire un intervalle de confiance autour de la prédiction moyenne, utile pour prendre des décisions plus prudentes dans des contextes critiques.

Types de dropout bayésien
Le mécanisme du dropout bayésien peut être décliné selon plusieurs variantes, chacune visant à capturer l'incertitude de manière plus fine ou plus adaptée à l'architecture utilisée :

— **Dropout sur les activations** : c'est la forme la plus répandue. Un masque de Bernoulli est appliqué sur les sorties des couches (souvent les couches denses ou convolutionnelles). Cela permet une régularisation efficace tout en simulant des chemins de propagation différents.

— **Dropout sur les poids** : au lieu d'agir sur les activations, cette approche applique directement le masque de Bernoulli sur les poids du réseau. Elle est plus proche de l'interprétation variationnelle formelle et peut être utilisée pour modéliser des distributions directement sur les paramètres du modèle.

— **Dropout structuré** : cette variante supprime des unités structurées (comme des neurones entiers, des canaux de convolution ou même des couches entières). Elle est particulièrement adaptée aux réseaux convolutifs (CNN) ou récurrents (RNN) où la suppression aléatoire d'éléments individuels peut être sous-optimale.

— **Variational dropout** : ici, le taux de dropout lui-même devient un paramètre à apprendre. On introduit une distribution variationnelle sur les poids (ex. log-normale), ce qui permet une estimation plus expressive et personnalisée du niveau d'incertitude pour chaque connexion.

Chaque type de dropout répond à des besoins spécifiques en matière d'architecture, de capacité de généralisation et de précision des incertitudes.

Avantages pratiques de MC Dropout :

- **Implémentation simple et intégration facile** : MC Dropout peut être ajouté à un réseau existant sans refonte majeure du modèle. Il suffit de conserver le masque de dropout actif à l'inférence.
- **Compatibilité avec les frameworks modernes** : cette approche est directement compatible avec les bibliothèques de deep learning comme TensorFlow ou PyTorch.
- **Scalabilité** : elle peut être appliquée à des architectures profondes à grande échelle (avec des millions de paramètres), ce qui la rend adaptée aux environnements industriels.
- **Quantification fiable de l'incertitude épistémique** : contrairement aux approches fréquentistes, MC Dropout permet d'estimer l'incertitude liée à l'ignorance sur les paramètres du modèle, en particulier dans les régions peu couvertes par les données.
- **Efficacité computationnelle relative** : bien que plus coûteux que l'inférence standard, MC Dropout reste beaucoup plus efficace que les méthodes MCMC ou les processus gaussiens pour modéliser l'incertitude.

Limitations :

- **Approximation simplifiée de la postérieure** : la distribution induite par le dropout est très factorisée (indépendance entre les poids), ce qui limite la fidélité à la véritable postérieure bayésienne $P(W|D)$.
- **Sensibilité au taux de dropout** : ce paramètre, souvent choisi de manière empirique, influence directement la variance des prédictions. Un mauvais réglage peut fausser l'estimation de l'incertitude.
- **Coût à l'inférence** : l'estimation de l'incertitude nécessite plusieurs passes forward par entrée. Cela peut ralentir les prédictions, surtout en contexte temps réel.
- **Incapacité à capturer des postérieures complexes** : MC Dropout ne couvre qu'un sous-ensemble restreint des distributions postérieures possibles.

Malgré ces limites, le dropout bayésien représente une méthode *pragmatique et efficace* pour intégrer une forme d'inférence bayésienne dans les réseaux neuronaux profonds, alliant simplicité, performance et interprétabilité des incertitudes.

IV.2 Inférence variationnelle (Variational Inference)

L'inférence variationnelle (VI) est une méthode puissante permettant d'approximer des distributions a posteriori intractables, comme $P(W|D)$, dans le cadre de l'apprentissage profond bayésien. Contrairement aux méthodes de Monte Carlo (comme MCMC), l'approche variationnelle transforme le problème d'inférence en un problème d'optimisation.

IV.2.1 Principe général

Le principe fondamental de l'inférence variationnelle est d'approximer la distribution a posteriori $P(W|D)$ par une distribution paramétrée plus simple $q(W|\phi)$, où ϕ représente les paramètres variationnels. On choisit alors q dans une famille de distributions facile à manipuler (par exemple des distributions gaussiennes factorisées).

L'objectif est de minimiser la **divergence de Kullback-Leibler** (KL) entre $q(W|\phi)$ et la vraie postérieure :

$$\phi^* = \arg\min_{\phi} \mathrm{KL}\big(q(W|\phi) \,\|\, P(W|D)\big)$$

La divergence KL est définie par :

$$\mathrm{KL}(q(W|\phi) \,\|\, P(W|D)) = \int q(W|\phi) \log \frac{q(W|\phi)}{P(W|D)} \, dW$$

Cette quantité mesure à quel point l'approximation q est éloignée de la vraie postérieure. L'objectif est donc de rendre q aussi proche que possible de $P(W|D)$, tout en restant calculable.

IV.2.2 Evidence Lower Bound (ELBO)

Puisque $P(W|D) = \frac{P(D|W)P(W)}{P(D)}$, la divergence KL ci-dessus dépend du terme intractable $P(D)$. On reformule donc le problème d'optimisation à partir d'une borne inférieure du log-vraisemblance marginale $\log P(D)$, appelée **ELBO** (Evidence Lower Bound) :

$$\log P(D) = \text{ELBO} + \text{KL}(q(W|\phi) \,\|\, P(W|D)) \quad \Rightarrow \quad \log P(D) \geq \text{ELBO}$$

Avec :

$$\text{ELBO}(\phi) = \mathbb{E}_{q(W|\phi)}[\log P(D|W)] - \text{KL}(q(W|\phi) \,\|\, P(W))$$

Ce qui nous donne deux termes interprétables :

— **Terme de reconstruction** : $\mathbb{E}_{q(W|\phi)}[\log P(D|W)]$, qui favorise une bonne explication des données.

— **Terme de régularisation** : $\text{KL}(q(W|\phi)\|P(W))$, qui pénalise les écarts entre la distribution approximative et la prior.

Maximiser ELBO revient donc à faire un compromis entre fidélité aux données et respect de la prior. Cette maximisation est réalisée via des méthodes d'optimisation stochastique (SGD, Adam), en échantillonnant des poids $W \sim q(W|\phi)$ à chaque itération.

Reparamétrisation : on utilise souvent le **trick de reparamétrisation** pour rendre l'échantillonnage différentiable :

$$W = \mu + \sigma \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Cela permet de faire passer le gradient à travers l'échantillon et d'optimiser la ELBO par backpropagation.

IV.2.3 Familles de distributions variationnelles

Le choix de la famille $q(W|\phi)$ est crucial. Il détermine à la fois la capacité d'approximation et la complexité computationnelle. On distingue plusieurs cas classiques :

— **Approximation mean-field** : on suppose que $q(W)$ se factorise indépendamment sur chaque poids :

$$q(W) = \prod_i q(w_i) \quad \text{avec } q(w_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

Simple et efficace, mais ne capture pas les dépendances entre poids.

— **Distributions gaussiennes non factorisées** : permettent de modéliser les corrélations entre paramètres, mais sont plus coûteuses.

— **Normalizing Flows** : on commence par une distribution simple (ex. gaussienne), puis on la transforme par une suite de fonctions bijectives différentiables f_k, apprises par le réseau :

$$W = f_K \circ \cdots \circ f_1(z), \quad z \sim \mathcal{N}(0, I)$$

Cela permet de modéliser des distributions complexes tout en gardant un calcul tractable du Jacobien.

En résumé, l'inférence variationnelle permet une approximation scalable du posterior bayésien, bien adaptée aux réseaux de neurones. Elle est aussi utilisée dans de nombreux modèles modernes comme les Variational Autoencoders (VAE), ou encore les Bayesian Transformers.

V. Prédictions bayésiennes

La prédiction bayésienne consiste alors à intégrer toutes les prédictions possibles du modèle $P(y \mid x, W)$, pondérées par cette distribution a posteriori sur les poids. On obtient ainsi une distribution prédictive $P(y \mid x, D)$, plus informative et robuste, qui reflète non seulement la variabilité des données mais aussi celle des paramètres du modèle.

V.1 Principe de la prédiction bayésienne

Une fois la distribution a posteriori $P(W \mid D)$ obtenue, elle peut être utilisée pour faire des prédictions sur de nouvelles données. Contrairement aux méthodes fréquentistes classiques qui prédisent la sortie y pour une entrée x en utilisant un seul jeu de poids (souvent appris par descente de gradient) :

$$\hat{y} = \arg\max_{y} P(y \mid x, \hat{W})$$

où $\hat{W}$ est l'estimation ponctuelle des poids (par exemple celle qui maximise la vraisemblance),

l'approche bayésienne agrège les prédictions de tous les jeux de poids possibles, en les pondérant selon leur probabilité :

$$P(y \mid x, D) = \int P(y \mid x, W)\, P(W \mid D)\, dW$$

On ne fixe pas les poids à une valeur optimale, mais on les considère comme des variables aléatoires, et on les marginalise. Ce changement de paradigme permet de rendre le modèle plus prudent et mieux calibré dans ses décisions.

La prédiction bayésienne est donc définie comme une moyenne marginale. Cette formule signifie que la probabilité de la sortie y, conditionnellement à l'entrée x et aux données d'entraînement D, est obtenue en intégrant toutes les prédictions possibles $P(y \mid x, W)$ sur l'ensemble des poids W, pondérées par la probabilité $P(W \mid D)$ que ces poids soient corrects.

V.2 Interprétation de la moyenne marginale

Intuitivement, on peut voir cette intégrale comme la sortie moyenne d'un ensemble de réseaux de neurones, chacun étant défini par un jeu de poids différent échantillonné depuis la distribution $P(W \mid D)$. Chacun de ces réseaux fournit une prédiction pour x, et on agrège leurs réponses pour obtenir la prédiction finale.

Cette approche présente plusieurs avantages :

— Elle permet de capturer l'incertitude sur les paramètres du modèle, en ne s'engageant pas sur une seule configuration de poids.

— Elle rend les prédictions plus **stables et robustes**, notamment dans les zones de l'espace des données où l'on dispose de peu d'exemples d'entraînement.

— Elle réduit les risques de **sur-apprentissage** (overfitting), car les poids fortement incertains ont moins d'impact sur la sortie finale.

La prédiction bayésienne constitue un pilier fondamental de l'apprentissage profond bayésien. En marginalisant sur les poids plutôt qu'en les fixant, elle permet de produire des sorties plus informatives, plus robustes, et mieux adaptées à la variabilité des données. Bien que son calcul exact soit souvent intractable, son principe guide le développement de nombreuses méthodes d'inférence approximative utilisées en pratique.

VI. Incertitude prédictive

L'un des objectifs majeurs de l'apprentissage bayésien profond est la capacité à **quantifier l'incertitude** associée aux prédictions. Contrairement aux réseaux de neurones classiques, qui produisent des prédictions ponctuelles sans information sur leur fiabilité, les modèles bayésiens génèrent des distributions prédictives permettant d'estimer le niveau de confiance.

VI.1 Types d'incertitude

VI.1.1 Incertitude aléatoire (aleatoric uncertainty)

L'incertitude aléatoire (ou *aleatoric uncertainty*) provient du **bruit inhérent aux données observées**. Elle reflète une variabilité irréductible dans le processus de génération des données, indépendante du modèle utilisé, et persiste même en présence d'un modèle parfait entraîné sur une quantité infinie d'exemples.

On distingue principalement deux formes d'incertitude aléatoire :

— **Homoscédastique** : l'intensité du bruit reste constante pour toutes les valeurs de l'entrée. Exemple typique : une erreur de mesure uniforme induite par un capteur calibré.

— **Hétéroscédastique** : le bruit dépend de l'entrée, c'est-à-dire qu'il varie selon les conditions d'observation. Exemple : une caméra générant plus de flou dans des conditions de faible luminosité.

Pour modéliser ce type d'incertitude dans un cadre probabiliste, on suppose que la sortie suit une distribution gaussienne dont la moyenne $\hat{y}(x)$ et la variance $\hat{\sigma}^2(x)$ sont apprises par le réseau :

$$y \sim \mathcal{N}(\hat{y}(x), \hat{\sigma}^2(x))$$

La fonction de perte correspond alors à la log-vraisemblance négative de cette distribution normale, ce qui permet d'estimer directement la variance en tant que paramètre d'incertitude :

$$\mathcal{L}(x, y) = \frac{1}{2\hat{\sigma}^2(x)}(y - \hat{y}(x))^2 + \frac{1}{2}\log \hat{\sigma}^2(x)$$

Ce terme pénalise à la fois les écarts au carré (comme une MSE pondérée) et les prédictions de variances irréalistes.

VI.1.2 Incertitude épistémique (epistemic uncertainty)

L'incertitude épistémique (ou *epistemic uncertainty*) traduit une **incertitude liée au modèle lui-même**, et plus précisément une connaissance imparfaite de ses paramètres internes. Elle résulte d'un manque de données ou d'une complexité excessive du problème, et peut être réduite en augmentant la quantité ou la diversité des données d'apprentissage.

Dans une approche bayésienne, cette incertitude est représentée par une distribution postérieure sur les poids du réseau neuronal, notée $P(W|D)$, où D désigne l'ensemble d'entraînement. La distribution prédictive obtenue est alors :

$$P(y|x, D) = \int P(y|x, W)\, P(W|D)\, dW$$

Ce formalisme permet de tenir compte de l'incertitude sur les poids dans les prédictions. L'incertitude épistémique est particulièrement pertinente :

— dans les régions de l'espace d'entrée peu couvertes par les données (exemples atypiques ou edge cases),

— pour détecter des entrées hors distribution (OOD),

— ou pour éviter la surconfiance du modèle dans des situations inconnues.

Contrairement à l'incertitude aléatoire, l'incertitude épistémique **diminue avec l'acquisition de nouvelles données informatives**.

VI.1.3 Incertitude globale

L'incertitude prédictive globale peut ainsi être estimée par la variance de $P(y \mid x, D)$, qui se décompose comme suit :

$$\underbrace{\mathrm{Var}_{P(W|D)}\left[\mathbb{E}_{P(y|x,W)}[y]\right]}_{\text{incertitude épistémique}} + \underbrace{\mathbb{E}_{P(W|D)}\left[\mathrm{Var}_{P(y|x,W)}[y]\right]}_{\text{incertitude aléatoire}}$$

Cette formulation permet d'identifier précisément d'où vient l'incertitude sur la prédiction, ce qui est crucial pour les systèmes critiques.

VI.2 Mesures d'incertitude

Une fois la distribution prédictive estimée, différentes métriques peuvent être utilisées pour quantifier l'incertitude. Ces mesures permettent d'analyser la fiabilité du modèle et d'orienter la prise de décision.

— **Variance prédictive** :

$$\mathrm{Var}(y|x, D) = \mathbb{E}_{P(W|D)}[\mathrm{Var}(y|x, W)] + \mathrm{Var}_{P(W|D)}[\mathbb{E}(y|x, W)]$$

Cette décomposition — issue de la loi totale de la variance — distingue deux composantes :
— $\mathbb{E}[\mathrm{Var}]$: l'**incertitude aléatoire**, intégrée dans chaque prédiction conditionnelle.
— $\mathrm{Var}[\mathbb{E}]$: l'**incertitude épistémique**, induite par la variabilité entre prédictions dues aux poids.

— **Entropie prédictive** :

$$H[y|x, D] = -\sum_c P(y = c|x, D) \log P(y = c|x, D)$$

Elle quantifie l'incertitude totale dans une tâche de classification. Une entropie élevée indique une distribution de probabilité plate, donc une prédiction incertaine.

— **Information mutuelle** :

$$\mathrm{MI}[y, W|x, D] = H[y|x, D] - \mathbb{E}_{P(W|D)}[H[y|x, W]]$$

Cette métrique mesure la part d'incertitude due spécifiquement aux paramètres du modèle. Elle constitue un estimateur pur de l'incertitude épistémique.

— **Disagreement entre modèles** : En combinant plusieurs modèles bayésiens (via des techniques comme MC Dropout ou les *deep ensembles*), on peut estimer le désaccord entre leurs prédictions :

$$\mathrm{Disagreement}(x) = \mathrm{Var}\left(\{P(y|x, W^{(t)})\}_{t=1}^{T}\right)$$

Cette variance inter-modèles constitue un indicateur robuste d'incertitude, surtout utile lorsque la postérieure exacte n'est pas accessible.

— **Intervalles de confiance** : Grâce à la distribution prédictive $P(y \mid x, D)$, il devient possible de calculer des intervalles de confiance sur les sorties du modèle. Ces intervalles donnent une estimation de la fiabilité de chaque prédiction, en fournissant une plage dans laquelle la sortie est susceptible de se trouver avec une certaine probabilité (typiquement 95%).

Si $\mu(x)$ et $\sigma(x)$ sont respectivement la moyenne et l'écart-type de la prédiction pour une entrée x, on peut approximer un intervalle de confiance à 95% par :

$$[\mu(x) - 1.96\sigma(x),\ \mu(x) + 1.96\sigma(x)]$$

Ces intervalles sont particulièrement précieux dans les domaines à forte responsabilité (médecine, conduite autonome, finance, etc.), car ils permettent de prendre des décisions non pas seulement en fonction d'une prédiction, mais aussi en tenant compte de sa fiabilité.

En somme, la prédiction bayésienne transforme la sortie d'un modèle déterministe en une **distribution riche en informations**, intégrant à la fois la connaissance extraite des données et les zones d'ignorance du modèle. Ces différentes mesures offrent des perspectives complémentaires pour analyser le comportement du modèle dans des environnements incertains ou non stationnaires.

VII.3 Applications pratiques

La quantification de l'incertitude a de nombreuses applications pratiques en apprentissage profond :

- **Détection d'outliers / OOD (Out-of-distribution)** : les modèles bayésiens tendent à avoir une incertitude épistémique élevée sur les entrées hors distribution, ce qui les rend utiles pour la détection d'anomalies.
- **Active learning** : on sélectionne les échantillons à annoter en priorité parmi ceux pour lesquels le modèle est le plus incertain (par exemple via l'entropie ou l'information mutuelle), ce qui maximise l'efficacité de l'apprentissage supervisé.
- **Prise de décision sous incertitude** : en robotique, médecine ou finance, disposer d'une estimation de l'incertitude permet d'éviter les décisions risquées. On peut par exemple ajuster les seuils de classification selon la confiance prédictive.
- **Rejection sampling / Abstention** : si l'incertitude est trop élevée, le modèle peut refuser de prédire et déléguer à un humain ou un système externe. Cela augmente la robustesse globale du système :

$$\text{Prédire si } H[y|x, D] < \tau$$

Conclusion : la modélisation de l'incertitude n'est pas un luxe, mais une nécessité pour les systèmes intelligents déployés dans des environnements réels. Elle permet de **fiabiliser l'apprentissage automatique**, de renforcer la **transparence** des prédictions, et d'ouvrir la voie à une meilleure interaction homme-machine.

VII. Évaluation et métriques

L'évaluation d'un modèle probabiliste ne se limite pas à sa précision (accuracy) : il est crucial d'évaluer la **qualité des incertitudes** produites. Un bon modèle bayésien doit être à la fois **calibré** (ses prédictions reflètent fidèlement la réalité probabiliste) et **informé** (capable d'indiquer avec justesse les cas incertains ou hors distribution). Cette section explore les métriques couramment utilisées pour quantifier ces aspects.

VII.1 Métriques de calibration

La calibration mesure la correspondance entre les **scores de confiance** fournis par un modèle probabiliste et la **fréquence empirique d'exactitude**. Un modèle bien calibré avec une sortie de confiance de 80% devrait avoir raison environ 8 fois sur 10 lorsque cette confiance est rapportée.

— **Expected Calibration Error (ECE)** : Cette métrique regroupe les prédictions en M intervalles de confiance disjoints (ou bins), puis calcule l'erreur absolue moyenne entre la confiance prédite et la fréquence d'exactitude dans chaque bin :

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$$

où B_m est le sous-ensemble des prédictions dont la confiance appartient au m^{e} bin, $\mathrm{acc}(B_m)$ est l'exactitude empirique de ce bin, et $\mathrm{conf}(B_m)$ la confiance moyenne.

— **Maximum Calibration Error (MCE)** : Variante plus stricte qui retient uniquement la pire erreur de calibration parmi les bins :

$$\mathrm{MCE} = \max_{m \in \{1,\ldots,M\}} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$$

— **Diagrammes de fiabilité (Reliability diagrams)** : Visualisation graphique de la calibration. On y trace $\mathrm{acc}(B_m)$ en fonction de $\mathrm{conf}(B_m)$. Un modèle parfaitement calibré produira une courbe proche de la diagonale $y = x$, tandis que les écarts à cette diagonale signalent une sur- ou sous-confiance.

Les métriques de calibration sont particulièrement utiles dans les contextes décisionnels, où la confiance doit guider les actions (ex. médecine, finance, sécurité).

VII.2 Métriques de qualité d'incertitude

Ces métriques visent à évaluer dans quelle mesure le modèle exprime correctement son incertitude, en lien avec sa capacité à détecter les erreurs, les anomalies ou les données hors distribution (OoD).

— **Area Under the Curve (AUC)** : Lorsqu'un modèle est confronté à des données hors distribution ou des exemples adversariaux, l'idéal est qu'il augmente son incertitude. En traçant la courbe ROC entre les scores d'incertitude (par exemple, l'entropie prédictive) et les étiquettes (in-distribution vs out-of-distribution), l'AUC mesure la capacité de discrimination du modèle :

$$\mathrm{AUC}_{\mathrm{ROC}} = \int_0^1 \mathrm{TPR}(f)\, d\mathrm{FPR}(f)$$

où TPR est le taux de vrais positifs, et FPR celui de faux positifs. Une AUC proche de 1 indique une bonne séparation entre données in et out-of-distribution.

— **Negative Log-Likelihood (NLL)** : Le NLL mesure la vraisemblance moyenne des vraies étiquettes sous la distribution prédictive du modèle. C'est une mesure de calibrage probabiliste rigoureuse :

$$\mathrm{NLL} = -\frac{1}{n} \sum_{i=1}^{n} \log p_\theta(y_i | x_i)$$

Plus le NLL est faible, mieux le modèle représente les vraies probabilités. Il pénalise fortement les prédictions erronées avec forte confiance.

— **Brier Score** : Cette métrique quadratique combine exactitude et calibration :

$$\mathrm{Brier} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(p_\theta(y_i = k | x_i) - \mathbb{1}_{y_i = k}\right)^2$$

où K est le nombre de classes. Contrairement au NLL, le Brier Score est borné et plus robuste aux outliers. Il prend des valeurs dans $[0, 1]$, avec des scores plus faibles indiquant de meilleures performances.

Chaque métrique offre une perspective différente sur la qualité de l'incertitude, et leur utilisation combinée est souvent nécessaire pour une évaluation complète.

VII.3 Tests empiriques

Au-delà des métriques numériques, plusieurs tests empiriques permettent de juger la pertinence des incertitudes produites :

- **Évaluation sur données out-of-distribution (OoD)** : Un bon modèle probabiliste doit détecter les entrées très différentes de la distribution d'entraînement. Par exemple, un classifieur entraîné sur MNIST devrait émettre des incertitudes élevées lorsqu'il est confronté à des images de CIFAR-10 ou SVHN. Cela peut être mesuré via l'entropie prédictive, la variance de MC Dropout, ou des scores de divergence.
- **Robustesse aux exemples adversariaux** : Les attaques adversariales visent à générer des perturbations imperceptibles induisant des erreurs de prédiction. Un modèle bayésien robuste devrait signaler une hausse de l'incertitude sur de telles entrées, au lieu de maintenir une confiance erronée. Cela peut être quantifié par la comparaison des entropies avant et après attaque, ou via des scores comme le *confidence-thresholded accuracy*.
- **Comparaison avec méthodes de référence** : Pour valider l'apport des modèles bayésiens, il est utile de les comparer à des réseaux classiques (non probabilistes), ou à d'autres méthodes d'estimation d'incertitude comme les ensembles de réseaux (*deep ensembles*), les réseaux gaussiens (Bayesian Neural Networks exacts), ou les méthodes post-hoc comme la régression isotone ou la température de calibration.

Ces tests empiriques complètent les métriques formelles en mettant à l'épreuve la robustesse du modèle dans des conditions réalistes et souvent critiques.

Type de métrique	**Nom**	**Interprétation / Objectif**
Calibration	ECE (Expected Calibration Error)	Moyenne pondérée des écarts entre confiance et précision dans des intervalles de prédiction.
	MCE (Maximum Calibration Error)	Plus grand écart entre confiance moyenne et précision réelle parmi tous les intervalles.
	Diagramme de fiabilité	Graphe comparant précision vs confiance ; une diagonale indique une bonne calibration.
Qualité d'incertitude	NLL (Negative Log-Likelihood)	Log-vraisemblance négative des vraies étiquettes sous la distribution prédite ; plus bas est meilleur.
	Brier Score	Erreur quadratique moyenne entre prédictions probabilistes et labels (0/1).
	AUC (Outlier Detection)	Aire sous la courbe ROC pour la détection d'outliers ; reflète la capacité à discriminer.
Tests empiriques	OOD Detection	Performance sur données hors distribution ; mesure de robustesse bayésienne.
	Adversarial Robustness	Réponse du modèle face à des perturbations adversariales ; incertitude devrait augmenter.
	Comparaison aux baselines	Comparaison avec softmax classique, deep ensembles, etc., pour valider l'apport bayésien.

TABLE VIII.1 – Résumé comparatif des principales métriques d'évaluation de l'incertitude en apprentissage profond bayésien.

VIII. Exemples pratiques et cas d'usage

Dans cette section, nous illustrons l'utilisation concrète de l'apprentissage profond bayésien à travers des outils de programmation, des scénarios d'application typiques et des recommandations méthodologiques. Cette partie permet de mieux saisir l'intérêt opérationnel de l'approche bayésienne dans des contextes réels.

VIII.1 Implémentation avec des frameworks

TensorFlow Probability (TFP). TFP est une extension de TensorFlow dédiée à la modélisation probabiliste. Elle permet de construire des couches bayésiennes comme `DenseVariational`, où les poids sont définis comme des distributions plutôt que des scalaires. On peut spécifier les fonctions génératrices de prior/posterior, et la bibliothèque se charge de la régularisation via la divergence KL.

```
model = tf.keras.Sequential([
  tfp.layers.DenseVariational(
    units=128,
    make_prior_fn=prior_fn,
    make_posterior_fn=posterior_fn,
    kl_weight=1./N),
  tf.keras.layers.ReLU(),
  ...
])
```

TFP s'intègre naturellement à Keras, ce qui facilite le prototypage rapide.

PyTorch + Pyro. Pyro est un framework probabiliste construit sur PyTorch. Il repose sur le paradigme du *probabilistic programming*, où l'on déclare explicitement les échantillonnages et les inférences dans des modèles de type fonctionnel.

```
def model(x, y=None):
    w = pyro.sample("w", Normal(0., 1.))
    b = pyro.sample("b", Normal(0., 1.))
    mean = w * x + b
    pyro.sample("obs", Normal(mean, 1.), obs=y)
```

L'entraînement est effectué par SVI (Stochastic Variational Inference), qui optimise une ELBO.

Edward / Edward2. Edward2, aujourd'hui intégré à TensorFlow Probability, fournit une interface plus bas niveau que TFP Layers. Il permet de gérer des modèles probabilistes complexes en définissant des traces stochastiques de manière déclarative.

VIII.2 Cas d'études

Classification d'images avec incertitude. Dans des domaines critiques comme la santé (imagerie médicale), la robotique ou la conduite autonome, il est essentiel de savoir si une prédiction est fiable. L'approche bayésienne permet de quantifier l'incertitude des prédictions, par exemple via l'entropie de la distribution prédictive ou le désaccord entre plusieurs passes du réseau.

Régression avec barres d'erreur. En régression, les modèles bayésiens permettent d'estimer non seulement une valeur prédite mais aussi un intervalle de confiance crédible autour de celle-ci. Cela permet de produire des courbes avec des bandes d'erreur reflétant l'incertitude épistémique et/ou aléatoire.

$$p(y \mid x, \mathcal{D}) = \int p(y \mid x, \theta) p(\theta \mid \mathcal{D}) d\theta \tag{VIII.1}$$

Cette intégrale est approximée via échantillonnage Monte Carlo, en effectuant plusieurs passes avec dropout activé ou en tirant plusieurs θ depuis la distribution postérieure.

Détection d'anomalies. Les données inconnues (out-of-distribution) induisent une incertitude plus forte dans les modèles bayésiens. Ainsi, ces derniers peuvent servir pour la détection d'anomalies ou d'exemples adverses. On évalue souvent cette capacité via des scores comme :

— l'entropie de la prédiction,
— la variance inter-prédictions (via MC Dropout),
— la probabilité maximale.

VIII.3 Bonnes pratiques

Choix des priors. Le choix des distributions a priori est essentiel. Il doit refléter nos connaissances ou incertitudes sur les poids :
- **Normal(0, 1)** : prior classique gaussien (par défaut),
- **Laplace(0, b)** : induit une régularisation L1 (favorise la parcimonie),
- **LogNormal** : utile pour des paramètres strictement positifs (ex. : variance).

Un prior mal choisi peut conduire à un sur-ajustement ou à une mauvaise calibration.

Hyperparamètres pour l'inférence variationnelle. Lorsqu'on utilise l'optimisation variationnelle (VI), plusieurs choix impactent la qualité :
- Le **nombre d'échantillons** Monte Carlo (souvent 10–50),
- Le **coefficient de pondération KL** (ex. : β-VAE), qui régule le compromis entre fidélité aux données et régularité du posterior,
- Le **taille de mini-batch**, influant sur la variance des gradients.

Validation croisée bayésienne. Pour évaluer la qualité prédictive et calibrative d'un modèle bayésien, on utilise :
- La **log-vraisemblance marginale (MLL)**,
- L'**Evidence Lower Bound (ELBO)** en VI,
- Des techniques comme la **Leave-One-Out (LOO) Cross Validation bayésienne**.

Ces métriques sont plus informatives que la simple précision ou RMSE.

IX. Défis computationnels

L'un des principaux défis de l'apprentissage profond bayésien réside dans la complexité computationnelle inhérente au calcul de la distribution a posteriori des paramètres du réseau. Contrairement à l'apprentissage fréquentiste classique, qui estime un point précis (comme le maximum de vraisemblance), l'inférence bayésienne cherche à évaluer toute une distribution $P(W|D)$ sur les poids W du réseau, compte tenu des données D. Cette ambition statistiquement plus riche s'accompagne de nombreux obstacles techniques, notamment :

IX.1 Intractabilité du calcul exact du posterior

L'inférence bayésienne repose sur le théorème de Bayes :

$$P(W \mid D) = \frac{P(D \mid W)P(W)}{P(D)}$$

où :
- $P(W)$ est la distribution a priori sur les poids,
- $P(D \mid W)$ est la vraisemblance des données données les poids,
- $P(D)$ est la probabilité marginale des données, souvent appelée

Ce dernier terme, $P(D) = \int P(D \mid W)P(W)\, dW$, nécessite une intégration sur l'espace complet des paramètres du réseau. Or, dans le cas des réseaux de neurones profonds, cet espace est de très haute dimension et la vraisemblance $P(D|W)$ est souvent non linéaire et multimodale. Il en résulte une intractabilité analytique : le calcul exact est **mathématiquement impossible dans la majorité des cas**, sauf pour des modèles très simples.

IX.2 Complexité exponentielle en dimension

Un réseau de neurones profond peut contenir des millions de poids. L'espace de recherche est donc un espace vectoriel de dimension très élevée. Dans un tel espace, le coût de l'intégration, de l'échantillonnage ou même de l'optimisation **augmente de manière exponentielle avec la dimension**. Ce phénomène est souvent qualifié de *malédiction de la dimensionnalité*. Même les méthodes numériques, comme la quadrature de Gauss ou les grilles régulières, deviennent rapidement inutilisables : si l'on échantillonne seulement 10 points par dimension pour une distribution sur 1000 paramètres, cela implique 10^{1000} points au total, un nombre totalement irréaliste. Ce problème rend la modélisation bayésienne classique impraticable pour des architectures neuronales fondées à grande échelle.

X. Conclusion

L'apprentissage profond bayésien constitue une approche puissante pour intégrer l'incertitude dans les modèles neuronaux. En allant au-delà des prédictions ponctuelles, il offre une vision probabiliste qui permet de :

— quantifier la confiance dans les décisions prises,
— mieux gérer les données rares, bruitées ou hors distribution,
— augmenter la robustesse face aux attaques adversariales.

Nous avons exploré dans cet exposé :

— les fondements théoriques de l'inférence bayésienne et sa transposition aux réseaux de neurones ;
— les méthodes pratiques d'approximation, en particulier l'inférence variationnelle bayésienne ;
— les métriques d'évaluation spécifiques à l'incertitude ;
— des exemples concrets d'implémentation et les bonnes pratiques en usage.

Malgré ses avantages, cette approche reste confrontée à plusieurs défis :

— la scalabilité à grande échelle ;
— le choix de priors pertinents ;
— l'interprétabilité des prédictions incertaines.

Enfin, les liens avec d'autres domaines comme les flows normalisants, les gradients variationnels ou l'apprentissage kernel profond montrent que l'apprentissage profond bayésien est un domaine en forte expansion, au cœur des développements les plus récents de l'IA.

En conclusion, intégrer l'incertitude dans les modèles de deep learning n'est plus une option, mais une nécessité dans les applications critiques. L'approche bayésienne ouvre la voie à une intelligence artificielle plus fiable, plus éthique et plus explicable.

Bibliographie

[1] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, Mohammed Bennamoun.
Hands-on Bayesian Neural Networks – A Tutorial for Deep Learning Users.
University of Western Australia, Murdoch University, Monash University, 2020.

[2] Karim PB.
Understanding Bayesian Networks. https://medium.com/@karimpb/understanding-bayesian-networks-f95ef1372ff0

[3] Gabriel Costa Leirbag. *A First Insight into Bayesian Neural Networks (BNNs).*
https://medium.com/@costaleirbag/a-first-insight-into-bayesian-neural-networks-bnn-c767551e9526

[4] Pinak Datta.
Understanding Bayesian Deep Learning : Probabilistic Models and Uncertainty Estimation.
https://medium.com/@pinakdatta/bayesian-deep-learning-understanding-probabilistic-models-uncertainty-3f0602ee4189