Sujet 14

Y. Gal
Z. Ghahramani
2016

L. Bresson
K. Iakovlev
E. Roblin

Dropout as a Bayesian Approximation

Projet de Statistique bayésienne

Table des Matières

1 Introduction

L'apprentissage profond (ou deep learning en anglais) a révolutionné le domaine de l'intelligence artificielle: cette technique d'apprentissage, basée sur des réseaux de neurones artificiels, est de plus en plus utilisée, en particulier depuis 2010 et la parution des premières recherches appliquées au N.L.P. ("Natural Language Processing") [1]. Ces modèles peuvent être analysés comme des extensions du domaine du machine learning, tout en présentant l'avantage d'avoir la capacité à analyser des bases de données plus volumineuses. Les domaines de prédilection de l'apprentissage profond sont l'imagerie et le son. Mais, que ce soit dans le cadre de la classification ou de la régression, on mesure trop souvent la qualité de ces modèles en se penchant uniquement sur leur erreur de prédiction. On oublie souvent une composante fondamentale de l'étude de la qualité du modèle, à savoir l'incertitude.

Or l'incertitude permet de déterminer dans quelle mesure la réponse donnée par le modèle a un sens. Prenons l'exemple de la classification d'images d'animaux. Si l'on met en place un modèle de classification d'images de chiens et de chats et qu'on lui présente une image d'autruche, le modèle tentera de classifier l'image présentée. De ce fait, l'autruche sera classifiée chien ou chat, mais avec une très faible certitude. L'idée ici est de rendre compte de cette incertitude. Plus précisement, nous étudions le papier *"Dropout as a Bayesian approximation: representing model uncertainty in deep learning"* [4] dans lequel Y. Gal et Z. Ghahramani développent des méthodes afin d'appréhender l'incertitude inhérente à l'apprentissage profond. Nous nous demanderons dans quelle mesure la technique du dropout, habituellement utilisée pour éviter le sur-apprentissage, permet de mesurer l'incertitude. Nous nous pencherons également sur la question du lien avec l'approximation bayésienne.

Le papier étudié propose d'utiliser des techniques bayésiennes de mesure de l'incertitude et de les appliquer au cadre de l'apprentissage profond. Pour ce faire, les auteurs se concentrent sur la méthode du dropout dans les réseaux de neurones et l'assimilent à un processus gaussien complexe. De plus, ils utilisent une méthode de Monte Carlo pour estimer cette incertitude et qualifient leur méthode de "MC dropout" (nous utiliserons cet anglicisme dans la suite de ce rapport). Ils montrent comment cette méthode peut être appliquée à des problèmes de régression et de classification. Par ailleurs, cette méthode peut être utilisée dans le cadre de l'apprentissage par renforcement ("reinforcment learning" en anglais) et permet, par exemple, une convergence plus rapide de l'algorithme de Thomson Sampling.

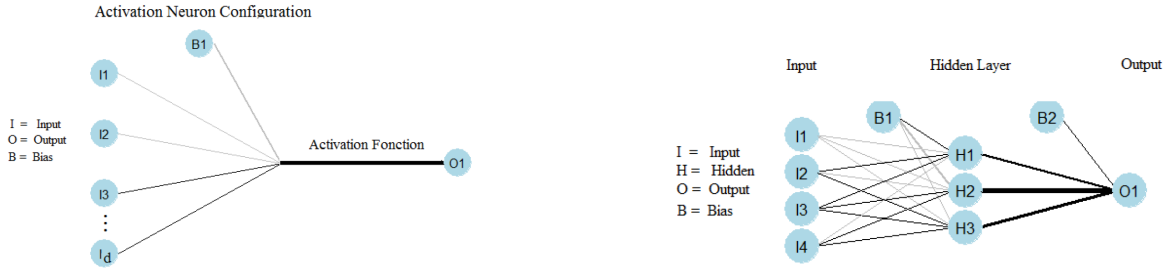
2 Cadre théorique: présentation des notions clés

Nous présentons d'abord le cadre théorique de l'article, à savoir la technique du dropout dans les réseaux de neurones, les processus gaussiens, la méthode d'inférence variationnelle et la mesure de l'incertitude.

2.1 Les réseaux de neurones profonds et la technique du dropout

1) Les réseaux de neurones

Un réseau de neurones vanille correspond à un ensemble de neurones artificiels organisés en couches (plusieurs couches dites "cachées" et une couche de sortie). Les neurones d'une couche donnée sont tous reliés aux neurones de la couche suivante.
Un neurone est caractérisé par une somme pondérée des entrées du modèle. Celles-ci sont transformées par une fonction d'activation (fonction non linéaire - la fonction sigmoïde ou la fonction tangente hyperbolique sont parmi les plus populaires) pour produire sa sortie. Les poids sont estimés en minimisant une fonction de perte grâce à l'algorithme de rétropropagation du gradient. Le gradient de l'erreur est calculé pour chaque neurone, de la dernière couche vers la première.



(a) Neurone artificiel.

(b) Réseau de neurones.

Figure 1: Représentation d'un réseau de neurones.

De nombreuses variantes existent selon le choix de la structure et de la dynamique du réseau (boucles, etc.), du degré de complexité (nombre de neurones, etc.) ou encore du type de neurones (fonction d'activation).

On parle de réseaux de neurones profonds lorsque le nombre de couches est très élevé. Cela nécessite généralement une implémentation plus minutieuse (régularisation des poids, "early stopping", répartition des calculs sur G.P.U., etc.). Toutefois, cette augmentation du nombre de couches a tendance à favoriser le sur-apprentissage : les neurones deviennent capables de détecter de très légères variations du modèle d'apprentissage, entraînant un sur-ajustement. Ceci a pour conséquence que le modèle ne se généralise pas bien à de nouvelles données.

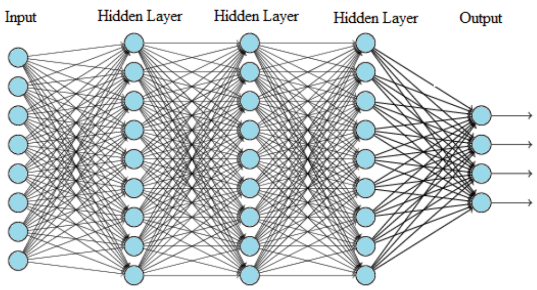
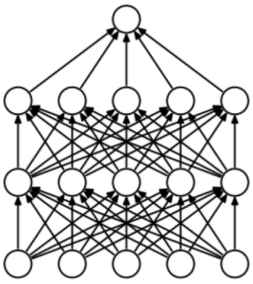


Figure 2: Réseau de neurones profond.

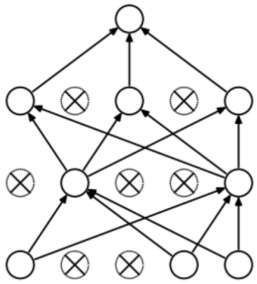
2) La technique du dropout

Le dropout permet de prévenir le sur-apprentissage. En effet, lorsque cette technique n'est pas appliquée, le réseau de neurones profond est entièrement connecté, ce qui entraînent les neurones à créer des dépendances entre eux. Cela réduit le pouvoir de chaque neurone indépendamment et entraîne un sur-apprentissage des données d'entraînement.

Le dropout, ou méthode de désactivation, correspond au fait de retirer des unités dans un réseaux de neurones. Une unité correspond à un neurone, qu'il fasse partie des couches cachées ou d'entrée. Cette technique est appliquée lors de la phase d'entraînement en choisissant des unités de façon aléatoire ; la probabilité pour une unité d'être rejetée est alors de $1 - p$.



(a) Réseau de neurones standard.



(b) Après l'application du dropout.

Figure 3: La méthode du dropout.

Posons les notations suivantes :

- W_1 et W_2 les matrices des poids pour la couche cachée et la couche d'activation respectivement ;
- σ la fonction d'activation (ou de transformation non-linéaire) ;
- b le biais appliqué sur les observations ;

- b_1 et b_2 les vecteurs utilisés pour mettre en place la méthode du dropout. Ils sont de dimension D et chaque ligne vaut 0 avec une probabilité $1 - p1$ pour b_1, $1 - p_2$ pour b_2, 1 sinon. Pour des problèmes de dimensions, ce sont en fait des matrices diagonales.

Si l'on prend l'exemple du dropout dans le cas d'un réseau de neurones à une couche cachée et appliqué à une régression, on a alors:

$$\begin{cases} y_{\text{couche cachée}} & = \sigma(xb_1W_1 + b) \text{ avec } b_1 = 1 \text{ avec proba. } p \text{ et 0 sinon.} \\ y_{\text{couche sortie}} & = \hat{y} = \sigma(xb_1W_1 + b)b_2W_2 \text{ avec } b_2 = 1 \text{ avec proba. } p \text{ et 0 sinon.} \end{cases}$$

On détermine W_1 et W_2 de sorte à minimiser la fonction de perte suivante :

$$L = E + \lambda(||W_1||_2^2 + ||W_2||_2^2 + ||b||_2^2)$$

où E est la fonction de perte quadratique (on préférera une perte "softmax" pour un problème de classification) à laquelle on ajoute un terme de régularisation (L2 ici).

2.2 Les modèles de processus gaussiens

Un processus stochastique est un ensemble de variables aléatoires $\{Y(x)|x\}$ indexées par un ensemble X. Un processus stochastique est caractérisé par sa distribution de probabilité pour tout sous-échantillon fini $Y(x_1, ..., x_k)$. Un processus gaussien peut être caractérisé par sa moyennne $\mu(x) = \mathbb{E}(Y(x))$ et sa covariance $K(x, x^{'}) = \mathbb{E}((Y(x) - \mu(x))(Y(x^{'}) - \mu(x^{'})))$.

Les processus gaussiens sont un outil utilisé en statistique bayésienne. En effet, considérons un jeu de N données, avec en entrée $X = \{x_1, ..., x_N\}$ et en sortie $Y = \{y_1, ..., y_N\}$. L'objectif est d'estimer la fonction f telle que $y = f(x)$. Pour ce faire, on définit une distribution a priori parmi l'espace des fonctions p(f). D'après le théorème de Bayes, on observe ensuite la distribution a posteriori en considérant nos données (X,Y) :

$$p(f|X, Y) \propto p(Y|X, f)p(f)$$

En modélisant notre distribution à l'aide d'un processus gaussien, il est alors possible de déterminer la distribution a posteriori correspondante dans le cas de la régression. En effet, on place une distribution gaussienne jointe sur toutes les valeurs :

$$F|X \sim \mathcal{N}(0, K(X, X)) \text{ et } Y \sim \mathcal{N}(F, \tau^{-1}I_N)$$

avec I_N la matrice identité de dimension $N \times N$. τ représente la précision du modèle. Elle est égale au produit de l'échelle de longueur a priori l de la fonction, de la moitié de la probabilité de ne pas éliminer une unité $\frac{p}{2}$, de l'inverse du nombre de données et de l'inverse du retard λ : $\tau = \frac{l^2 p}{2N\lambda}$.

Il est également possible d'estimer la distribution a posteriori dans le cas de la classification en appliquant au préalable la fonction "softmax" aux sorties Y :

$$F|X \sim \mathcal{N}(0, K(X, X)) \text{ et } Y|F \sim \mathcal{N}(F, 0I_N) \text{ avec } c_n|Y \sim \text{Catégorie}(\exp(y_{nd})/\sum_{d^{'}} \exp(y_{nd^{'}}))$$

pour $n = 1, ..., N$ avec les labels c_n. L'idée ici est de définir une prior comme processus gaussien sur la fonction latente f(x) puis de la transformer à l'aide d'une fonction logistique.

Pour définir le processus gaussien et l'utiliser dans le cas de la régression ou de la classification, il est nécessaire de définir une fonction de covariance $K(X_1, X_2)$. Cette dernière représente la similitude entre chaque pair de points $K(x_i, x_j)$. Avec N points, la matrice de covariance s'écrit $K(X, X)$. Un exemple de fonction de covariance non linéaire est la tangente hyperbolique. Le problème dans le cas d'un processus gaussien est d'abord de trouver les propriétés convenables pour la fonction de covariance. En effet, comme le processus gaussien n'est pas un modèle paramétrique, nous n'avons pas à nous soucier de savoir si le modèle s'ajuste aux données ou non. Pour estimer cette distribution gaussienne, différentes techniques peuvent être utilisées, en particulier l'inférence variationnelle.

2.3 L'inférence variationnelle

Pour estimer le modèle décrit ci-dessus, on pourrait restreindre le modèle à un échantillon fini de variables aléatoires ω. La distribution pour une nouvelle entrée x^* deviendrait alors :

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, \omega)p(\omega|X, Y)$$

avec $y^* \in \mathbb{R}^D$. Mais cette distribution ne peut pas être estimée. On définit alors une distribution variationnelle $q(\omega)$ pour estimer $p(y^*|x^*, X, Y)$. L'objectif est d'obtenir une distribution variationnelle aussi proche que possible de la ditribution a posteriori obtenue avec le processus gaussien. Pour ce faire, on minimise la divergence de Kullback-Leibler, qui est une mesure de similarité entre deux distributions :

$$KL(q(\omega)|p(\omega|X, Y))$$

On a donc: $q(y^*|x^*) = \int p(y^*|x^*, \omega)q(\omega)d\omega$. Minimiser la différence de Kullback-Leibler revient à maximiser :

$$\mathcal{L}_{VI} = \int q(\omega) \log(p(Y|X, \omega))d\omega - KL(q(\omega)||p(\omega))$$

par rapport au paramètre variationel qui définit $q(\omega)$. On obtient alors une distribution variationnelle $q(\omega)$ qui permet d'expliquer les données tout en étant proche de la distribution a priori, empêchant ainsi le sur-apprentissage.

2.4 Mesure de l'incertitude: technique d'évaluation bayésienne

Les réseaux de neurones profonds auxquels on applique la technique du dropout sur les poids peut être assimilé aux processus gaussiens profonds avec inférence variationnelle.

On commance par estimer la fonction de covariance d'un processus gaussien, qui est de la forme :

$$K(x, y) = \int \mathcal{N}(w; 0, l^{-2}I_Q)p(b)\sigma(w^T x + b)\sigma(w^T y + b)dwdb$$

à l'aide d'une intégration par Monte Carlo sur K termes :

$$\hat{K}(x,y) = \frac{1}{K}\sum_{k=1}^{K} \sigma(w_k^T x + b_k \sigma(w_k^T y + b_k)) \text{ avec } w_k \sim \mathcal{N}(0, l^{-2} I_Q) \text{ et } b_k \sim p(b)$$

Les K termes peuvent être assimilés à K neurones cachés dans le réseau de neurones. On peut paramétrer le processu gaussien en posant :

$$\begin{cases} w_l & \sim \mathcal{N}(0, l^{-2} I_Q) \\ w_d & \sim \mathcal{N}(0, l-2 I_K) \\ b_k & \sim p(b) \\ W_1 & = [w_k]_{k=1} K \\ W_2 & = [w_d]_{d=1} D \\ b & = [b_k]_{k=1} K \\ \omega & = \{W_1, W_2, b\} \\ p(y*|x*, \omega) & = \mathcal{N}(y^*, \sqrt{\frac{1}{K}} \sigma(x*W_1 + b) W_2, \tau - 1 I_N) \end{cases}$$

On utilise alors l'estimation a posteriori obtenue par la méthode variationnelle. On a :

$$\begin{cases} q_\theta(\omega) & = q_\theta(W_1) q_\theta(W_2) q_\theta(b) \\ q_\theta(W_1) & = \prod_{q=1}^{Q} q_\theta(w_q) \\ q_\theta(W_q) & = p_1 \mathcal{N}(m_q, s^2 I_K) + (1 - p_1) \mathcal{N}(0, s^2 I_K) \end{cases}$$

$q_\theta(W_1)$ est un mélange de gaussiennes. L'ensemble de nos paramètres variationnelles sont alors : $\theta = \{M_1, M_2, m, p_1, p_2, s\}$.

Finalement, pour obtenir le modèle avec le dropout, on fixe certains paramètres, p_1, p_2 et s. On prend notamment une valeur de s petite de sorte que la divergence KL prenne une valeur finie. En normalisant par la constante $\frac{1}{N\tau}$, on obtient l'estimateur non-biaisé de $\mathcal{L}_{VI}$ suivant :

$$\mathcal{L}_{GP-MC} \propto -\frac{1}{2N}\sum_{n=1} N ||y_n - \hat{y_n}||_2^2 - \frac{p_1 l2}{2N_\tau} ||M_1||_2^2 - \frac{p_2 l2}{2N\tau} ||M_2||_2^2 - \frac{l^2}{2N\tau} ||m||_2^2$$

Pour effectuer nos prédictions, on utilise alors l'espérance empirique :

$$\mathbb{E}(y^*) \simeq \frac{1}{T}\sum_{t=1}^{T} \hat{y_t} \text{ avec } \hat{y_t} \sim \text{Réseau de neurones avec dropout}(x^*)$$

T représente le nombre de "dropout" différents effectués sur la base de test. On utilise la variance pour mesurer l'incertitude :

$$Var(y^*) \simeq \frac{1}{T}\sum_{i=1}^{T} \hat{y_t}^T \hat{y_t} - \mathbb{E}(y^*)^T \mathbb{E}(y^*) + \tau^{-1} I$$

Cette approche est qualifiée de "MC dropout" par Y. Gal et Z. Ghahramani (le terme MC réfère à "Monte Carlo") [4].

3 Applications et prolongements

Nous nous intéressons désormais aux applications de la méthode présentée. Dans un premier temps, nous illustrons celle-ci en utilisant les résultats des auteurs [4], puis nous prolongeons l'analyse en tentant de visualiser l'incertitude dans le cas d'un modèle de classification et dans le cas d'un modèle de régression implémenté sur *Python* en utilisant d'autres jeux de données.

3.1 Résultats des auteurs

Y. Gal et Z. Ghahramani testent leur méthode de prise en compte de l'incertitude (MC dropout) pour des problèmes de régression et de classification. Pour les régressions, les auteurs utilisent les données de concentration en CO_2 de l'atmosphère mesurée au Mauna Loa (Keeling et al., 2004), les données *MNIST* (LeCun & Cortes, 1998) pour la classification. Ils montrent que les réseaux de neurones bayésiens permettent de réduire l'incertitude en apprentissage profond sans altérer la précision des modèles ou leur complexité computationnelle. Nous présentons ici les résultats dans le cas de la régression.

1) Résultats qualitatifs

Les estimations sont issues de réseaux de neurones à 5 couches cachées et 1024 unités par couche. La fonction d'activation utilisée est la fonction ReLu ("Rectified Linear Unit"). Par ailleurs, le dropout est utilisé entre chaque couche avec une probabilité fixée à 0.1.

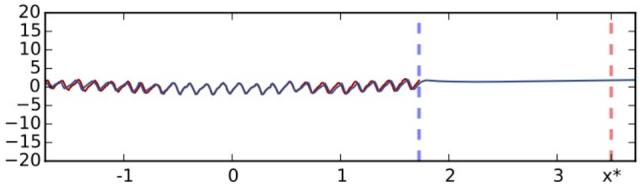


Figure 4: Dropout traditionnel.

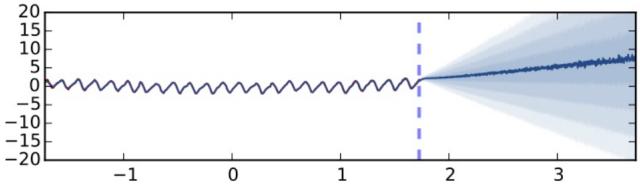


Figure 5: Approche bayésienne.

Les figures 4 et 5 représentent la prédiction de la valeur moyenne de la concentration atmosphérique en CO_2 (valeur normalisée). Les données observées sont en rouge (à droite du trait en pointillés bleu) et la moyenne des prédictions est en bleue. Plus précisément, la figure 4 correspond à un réseau de neurones avec un dropout traditionnel, i.e. qui ne permet pas d'appréhender l'incertitude, tandis qu'un MC dropout est employé dans la figure 5. Les

différentes nuances de bleu reflètent l'intensité de l'incertitude (ici chaque nuance est égale à un demi écart-type).

On constate aisément que le modèle de la figure 4 prédit, avec certitude, une valeur nulle pour le point x^* (ligne en pointillés rouge) tandis que la figure 5 prend en compte l'incertitude inhérente à l'extrapolation des données.

2) Résultats quantitatifs

Ensuite, Y. Gal et Z. Ghahramani comparent la qualité de l'estimation de l'incertitude obtenue grâce à leur méthode avec deux autres approches d'inférence, à savoir la rétropropagation probabiliste (PBP - Graves, 2011 [5]) et l'inférence variationnelle dans les réseaux de neurones bayésiens (VI - Hernandez-Lobato et Adams [6], 2015). Les comparaisons portent sur la racine carrée de l'erreur quadratique (RMSE) et sur la vraisemblance prédictive. Cette dernière mesure la qualité d'ajustement d'un modèle (plus cette mesure est élevée, plus l'ajustement est précis).

Le tableau 1 indique que la technique du dropout est la plus performante pour mesurer l'incertitude : le RMSE est minimisé tandis que la vraisemblance prédite est maximale[1].

			Avg. Test RMSE and Std. Errors			Avg. Test LL and Std. Errors		
Dataset	*N*	*Q*	**VI**	**PBP**	**Dropout**	**VI**	**PBP**	**Dropout**
Boston Housing	506	13	4.32 ±0.29	3.01 ±0.18	**2.97 ±0.19**	-2.90 ±0.07	-2.57 ±0.09	**-2.46 ±0.06**
Concrete Strength	1,030	8	7.19 ±0.12	5.67 ±0.09	**5.23 ±0.12**	-3.39 ±0.02	-3.16 ±0.02	**-3.04 ±0.02**
Energy Efficiency	768	8	2.65 ±0.08	1.80 ±0.05	**1.66 ±0.04**	-2.39 ±0.03	-2.04 ±0.02	**-1.99 ±0.02**
Kin8nm	8,192	8	**0.10 ±0.00**	**0.10 ±0.00**	**0.10 ±0.00**	0.90 ±0.01	0.90 ±0.01	**0.95 ±0.01**
Naval Propulsion	11,934	16	**0.01 ±0.00**	**0.01 ±0.00**	**0.01 ±0.00**	3.73 ±0.12	3.73 ±0.01	**3.80 ±0.01**
Power Plant	9,568	4	4.33 ±0.04	4.12 ±0.03	**4.02 ±0.04**	-2.89 ±0.01	-2.84 ±0.01	**-2.80 ±0.01**
Protein Structure	45,730	9	4.84 ±0.03	4.73 ±0.01	**4.36 ±0.01**	-2.99 ±0.01	-2.97 ±0.00	**-2.89 ±0.00**
Wine Quality Red	1,599	11	0.65 ±0.01	0.64 ±0.01	**0.62 ±0.01**	-0.98 ±0.01	-0.97 ±0.01	**-0.93 ±0.01**
Yacht Hydrodynamics	308	6	6.89 ±0.67	**1.02 ±0.05**	1.11 ±0.09	-3.43 ±0.16	-1.63 ±0.02	**-1.55 ±0.03**
Year Prediction MSD	515,345	90	9.034 ±NA	8.879 ±NA	**8.849 ±NA**	-3.622 ±NA	-3.603 ±NA	**-3.588 ±NA**

Table 1: Performance de l'approche bayésienne pour différentes bases de données (N représente le nombre d'observations et Q le nombre de variables).

Le principal avantage de l'approche de Y. Gal et Z. Ghahramani est que celle-ci peut s'appliquer à n'importe quel réseau (différents degrés de complexité, différentes structures, etc.). Ainsi, les modèles d'apprentissage profond qui sont des "boîtes noires" peuvent désormais être enrichis d'une mesure représentant l'incertitude.

[1]Notons que la base de donnée "Yacht" est une exception – la méthode PBP offre de meilleurs résultats.

3.2 Illustration de la méthode du MC dropout dans le cas d'une classification et d'une régression

3.2.1 MC dropout dans un modèle de classification

Dans cette section, nous présentons deux réseaux de neurones que nous avons construits en intégrant la méthode MC dropout. Dans le premier nous employons une partie des données *MNIST* utilisées auparavant par les auteurs. En revanche, notre plus faible puissance de calcul nous contraint à réduire la base de données initiale en une base de données ne comportant que 3000 images; 2000 utilisées pour l'apprentissage et 1000 utilisées pour la base de test. De plus nous conservons uniquement des images correspondant aux chiffres 0 et 1 dans le but de réduire la complexité de calcul. Nous implémentons un réseau de neurones avec deux couches cachées de 20 neurones, et un dropout avant la couche d'activation. La probabilité associée au dropout est ici de 0.5.

Structure du réseau de neurones:

1. Première couche cachée:

$$\sigma(XW_1 + b_1)$$

$$\sigma(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

2. Deuxième couche caché:

$$\sigma(\sigma(XW_1 + b_1)W_2 + b_2)$$

3. Dropout avant le neurone d'activation:

$$\sigma_1(\sigma(\sigma(XW_1 + b_1)W_2 + b_2)P_1W3 + b_3)$$

$$\sigma_1((z_1, z_2)) = (\frac{e^{z_1}}{e^{z_1} + e^{z_2}}, \frac{e^{z_2}}{e^{z_1} + e^{z_2}})$$

Où W_1, W_2, W_3 sont les matrices des poids, b_1, b_2, b_3 sont les biais de chaque couche et P_1 est une matrice de Bernouilli de probabilité p=0.5, correspondant à la loi a priori utilisée pour le dropout. Les deux couches cachées possèdent 20 neurones.

Nous estimons des paramètres à l'aide de la méthode de descente de gradient, puis effectuons nos prédictions à partir du réseau de neurones entraîné avec la méthode MC dropout. Rappelons que notre objectif n'est pas de trouver l'algorithme optimal pour effectuer la prédiction mais d'observer l'incertitude à l'aide de la méthode du MC dropout tout en contrôlant la qualité de la prédiction. Dans le tableau suivant sont présentés les résultats de la prédiction. Nous pouvons voir que malgré l'ajout de la couche dropout les résultats de la classification sont toujours relativement bons.

Table 2: Matrice de confusion du réseau de neurones MC dropout classifieur.

	Chiffre réel 0	Chiffre réel 1
Chiffre prédit 0	420	3
Chiffre prédit 1	4	573

Par la suite, nous analysons l'incertitude du modèle en nous inspirant de la méthode utilisée par les auteurs [4]. Après entraînement de notre modèle, nous le testons sur 1000 nouvelles images. Chaque image est testée 100 fois dans le modèle MC dropout, c'est-à-dire qu'on exécute la propagation *forward* 100 fois. Cette méthode nous donne ainsi 100 probabilités d'être égal au chiffre 1 pour chaque image. La moyenne de ces 100 probabilités et l'application de la fonction *softmax* au seuil de 0.5 nous donne la prédiction de notre modèle. Or, comme l'expliquent les auteurs, l'intérêt est de connaître avec quelle certitude la prédiction est faîte. Dans notre cas nous voulons savoir si la prédiction du chiffre 1 ou 0 est faite avec certitude. Pour ce faire, nous représentons la prédiction (courbe bleue) et les 100 probabilités (points gris) pour chaque image dans le graphique ci-dessous:

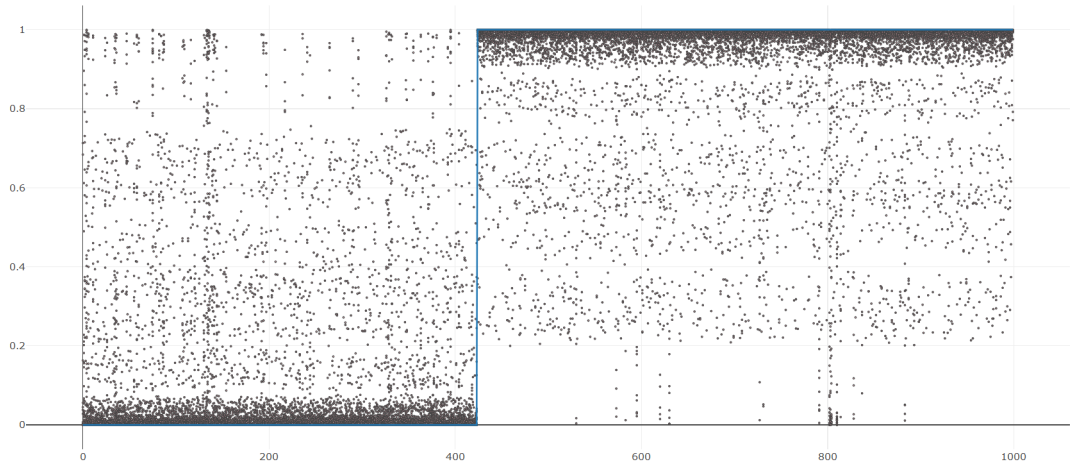


Figure 6: Incertitude du modèle MC dropout classifieur.

Si notre modèle prédisait le chiffre avec une certitude très élevée, alors tous les marqueurs gris seraient très près de la courbe bleue, ce qui n'est pas vraiment le cas sur notre graphique. En revanche, il est intéressant de noter que la distribution des marqueurs se rapproche nettement de la courbe ce qui signifie que l'incertitude n'est pas trop élevée. Ce degré d'incertitude, mesuré en fait par la variance ajustée de la variance a priori, est un critère non-négligeable pour le choix d'un modèle ou d'un paramètre. Il aurait été bien sûr intéressant de comparer plusieurs modèles selon ce paramètre mais les difficultés de calcul nous ont contraint à un seul modèle de classification.

3.2.2 MC dropout dans un modèle de régression

Après avoir évalué et tenté de visualiser l'incertitude dans le cas d'une classification, nous nous intéressons à un réseau de neurones MC dropout dans le cas de la régression. Nous

nous appuyons sur des données de consommation électrique d'un foyer au cours du temps. Il s'agit de données temporelles couvrant la période de décembre 2006 à novembre 2007. Cette période correspond à plus de 2.000.000 données. Nous nous restreignons aux 75.000 premières observations pour des raisons pratiques de capacité de calcul.

Nous nous intéressons alors à prédire la consommation globale d'électricité du foyer dans le temps en fonction de variables explicatives telles que la tension moyenne, l'intensité, etc... Nous avons au total 5 variables explicatives et une variable à prédire, toutes étant quantitatives. Nous rappelons que notre objectif est de visualiser l'incertitude et non d'évaluer la qualité de notre modèle au niveau de la prédiction. En revanche, nous devons également illustrer le fait que l'ajout de la couche dropout ne rend pas le modèle trop inefficace dans la prédiction.

Pour ce faire, nous implémentons un réseau de neurones à une couche cachée de 20 neurones et un dropout avant la couche d'activation. Nous séparons notre base en une base d'apprentissage comportant 50.000 observations et une base de test comportant 25.000 observations.

Structure du réseau de neurones:

1. Première couche cachée:

$$\sigma(XW_1 + b_1)$$

$$\sigma(z) = Relu(z) = z * max(z, 0)$$

2. Dropout avant le neurone d'activation et fonction d'activation linéaire:

$$\sigma(XW_1 + b_1)P_1W2 + b_2)$$

Où W_1, W_2, sont les matrices des poids, b_1, b_2, sont les biais de chaque couche et P_1 est une matrice de Bernouilli de probabilité p=0.05, correspondant à la loi a priori utilisée pour le dropout.

Nous calculons les estimations des paramètres à l'aide de la méthode de descente de gradient, puis effectuons nos prédictions à partir du réseau de neurones entraîné avec la méthode dropout. Le graphique suivant illustre l'écart et la similitude entre les 1000 premières prédictions et les valeurs réelles:

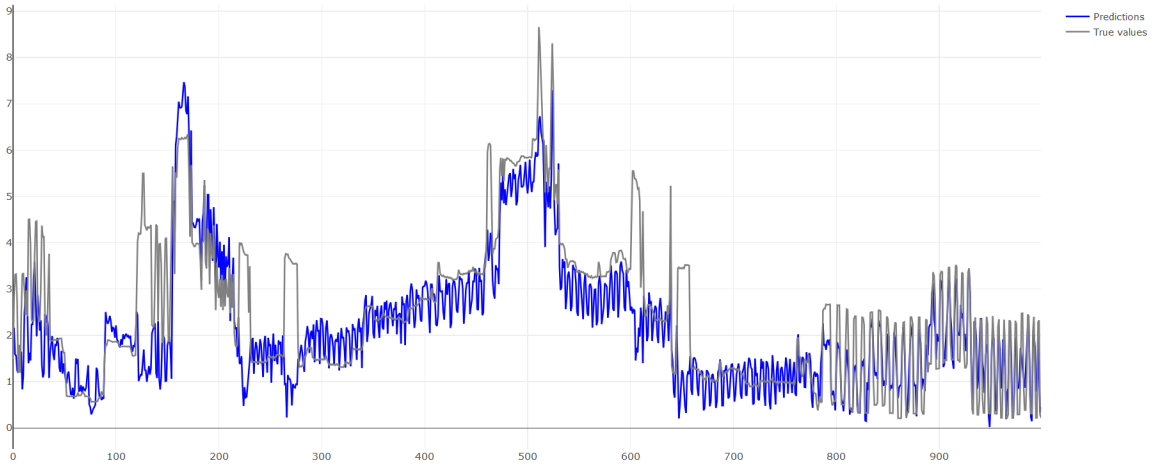


Figure 7: Prédictions du modèle MC dropout régresseur.

Si les deux courbes sont assez proches, on ne peut repérer l'incertitude à l'aide de ce graphique. Néanmoins on peut raisonnablement dire que le dropout n'affecte pas excessivement la qualité des prédictions.

Comme pour le problème de classification, nous représentons les prédictions ainsi que 50 des 100 sorties (50 seulement pour une visualisation plus facile) prédictions dropout. Nous obtenons ainsi une illustration de l'incertitude du modèle:

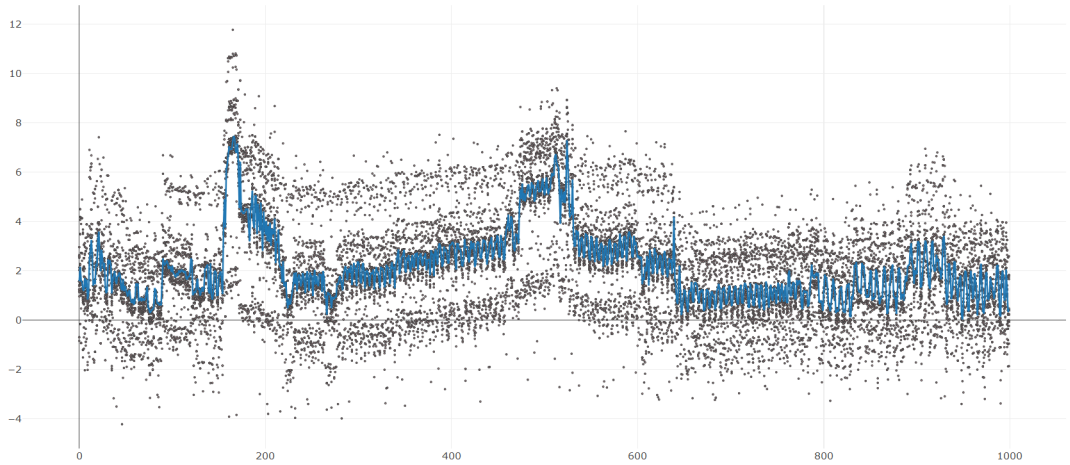


Figure 8: Incertitude du modèle MC dropout régresseur.

Contrairement à certains modèles de séries temporelles où la variance augmente au cours du temps, on peut voir ici que l'incertitude est assez élevée car les valeurs sont basses et certaines peuvent être négatives (en raison d'une fonction d'activation linéaire). Mais la variance autour de la prédiction moyenne n'augmente pas et donc l'incertitude n'augmente pas à mesure que l'on s'éloigne temporellement des observations de la base d'entraînement.

4 Discussions

La méthode présentée par Y. Gal et Z. Ghahramani en 2015 présente certaines limites. Pour pallier celles-ci, les auteurs proposent des prolongements. Des approches alternatives ont été développées.

Une des limites du MC dropout est que cette méthode ne peut pas être utilisée en temps réel (apprentissage en ligne) du fait de l'approximation de Monte Carlo. De plus, la technique de l'inférence variationnelle est beaucoup critiquée car elle sous-estime la variance et donc l'incertitude [10],[2]. Récemment (2017), Y. Gal a proposé une nouvelle méthode qui repose sur la minimisation de "l'alpha divergence" pour éviter cette sous-estimation [7].

Par ailleurs, selon I. Osband [8], Y. Gal et Z. Ghahramani confondent les notions de risque et d'incertitude alors qu'elles sont fondamentalement différentes. En effet, le risque peut être vu comme l'erreur stochastique inhérente à un modèle tandis que l'incertitude permet d'appréhender la confusion liée aux paramètres du modèle. Dans ces circonstances, la méthode du dropout mesure le risque et non l'incertitude.

Outre cette différence conceptuelle, I. Osband et al. [9] ont mis en exergue les mauvaises performances de la technique du MC dropout (ne converge pas asymptotiquement). Ils proposent une approche reposant sur la technique du bootstrap pour mesurer l'incertitude et obtiennent de bons résultats empiriques. Sans entrer dans les détails de la démonstration (Annexe A du papier [9]), présentons l'intuition des auteurs. Imaginons un problème de régression avec un jeu de données très bruitées. Si on ajuste un réseau de neurones dont le critère est l'erreur quadratique moyenne (MSE), celui-ci converge sûrement vers la valeur moyenne des données (i.e. le réseau tient compte des valeurs aberrantes). La technique du dropout étant locale, tous les échantillons du dropout restent concentrés autour de cette moyenne (cf. Figure 9 - les données aberrantes sont à droite). A l'inverse, avec la technique du bootstrap, les réseaux construits peuvent comprendre différents sous-échantillons de données bruitées et donc produire des estimations de l'incertitude plus adéquates.

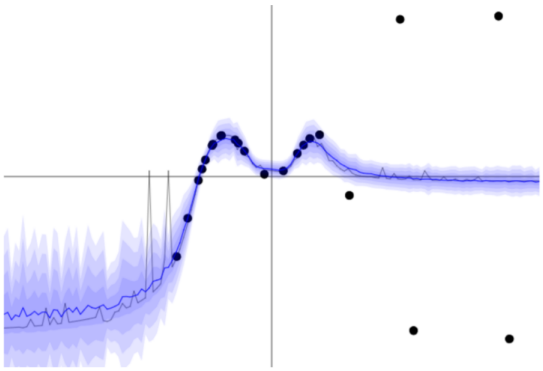


Figure 9: Dropout et données aberrantes: convergence vers la moyenne et non-prise en compte de l'incertitude [9].

Dans ce contexte, Osband et al. ont démontré l'avantage principal de la technique du MC dropout, à savoir le fait qu'elle peut s'appliquer à n'importe quel réseau, n'est pas vérifié empiriquement.

Notons que Y. Gal [3] a partiellement répondu aux critiques de I. Osband. D'une part, il a reconnu que deux notions distinctes d'incertitude existaient, à savoir l'incertitude "aléatoire" (capture le bruit inhérent à l'environnement) et "épistémique" (capture notre ignorance concernant le modèle le plus approprié pour modéliser nos données) ; la technique du MC dropout mesurant la seconde forme d'incertitude. En outre, Y. Gal et al. [3] ont introduit la méthode du "concrete dropout" qui est un prolongement du MC dropout dans lequel la valeur de la probabilité de dropout pour chaque couche peut être optimisée (cela permet d'influencer la magnitude de l'incertitude épistémique). Par ailleurs, dans son blog, Y. Gal a justifié les mauvaises performances de la méthode du dropout pointées par Y. Osband (cf. Figure 9) par le fait que ce dernier utilisait une régression homoscédastique (le bruit est identiquement distribué). Y. Gal a démontré qu'en modélisant les données bruitées avec une régression hétéroscédastique, l'incertitude liée aux points aberrants est mieux représentée (cf. Figure 10).

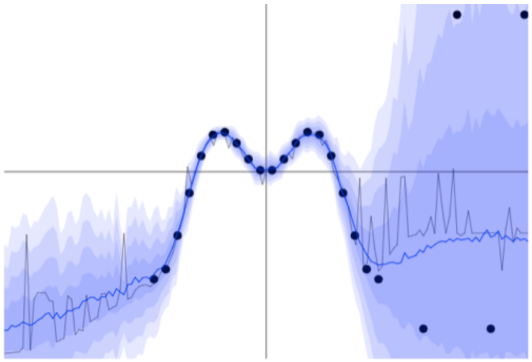


Figure 10: Dropout et données aberrantes: régression hétéroscédastique et prise en compte de l'incertitude.

Pour conclure, il n'y a pas de consensus dans la littérature : la mesure de l'incertitude en apprentissage profond reste une question ouverte.

5 Conclusion

Cet article propose une méthode d'analyse de l'incertitude dans les réseaux de neurones en utilisant le lien entre processus gaussiens et dropout. Les auteurs montrent qu'un réseau de neurones avec une certaine profondeur, des fonctions non-linéaires et auquel le dropout est appliqué avant chaque couche est mathématiquement équivalent à l'estimation d'un processus gaussien profond. L'idée est d'utiliser la technique du dropout (connue pour pallier le sur-apprentissage) afin de mesurer l'incertitude ; le dropout étant relativement simple à introduire au sein d'un réseau de neurones.

Y. Gal et Z. Ghahramani présentent leurs méthodes dans le cadre de la régression et de la classification. Nous avons reproduit leur approche avec différentes données et avons obtenu des résultats satisfaisants. Les auteurs proposent également une application à un problème d'apprentissage par renforcement. Le but ici est d'apprendre, à partir d'expériences, ce qu'il convient de faire en différentes situations, de façon à optimiser une récompense quantitative au cours du temps.

Nous aurions pu mener diverses extensions comme par exemple utiliser une loi gaussienne et non une loi de Bernouilli pour les couches dropout. Ainsi, on aurait pu comparer les résultats et s'approcher un peu plus de l'approximation des processus gaussiens. Nous aurions pu également faire varier la probabilité du dropout pour tenter de choisir le paramètre de manière optimale (approche utilisée dans [3]). Néanmoins notre implémentation simple dans les cas de la classification et de la régression nous a permis de visualiser l'incertitude tout en conservant une certaine qualité de prédiction.

Bibliographie

[1] Dahl et al. "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition". In: *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING* 20.1 (2012), pp. 30–42. DOI: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/dbn4lvcsr-transaslp.pdf.

[2] Yarin Gal. "Uncertainty in Deep Learning. PhD thesis". In: *University of Cambridge* (2016).

[3] Yarin et al. Gal. "Concrete Dropout". In: *Advances in Neural Information Processing Systems (NIPS 2017)* (2017).

[4] Zoubin Gal Yarin; Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of the 33th International Conference on Machine Learning (ICML-16)* (2016).

[5] A. Graves. "Practical variational inference for neural networks". In: *Advances in Neural Information Processing Systems (NIPS 2011)* (2011).

[6] J M Hernandez-Lobato and R P. Adams. "Probabilistic backpropagation for scalable learning of bayesian neural networks". In: *Proceedings of the 32th International Conference on Machine Learning (ICML-15)* (2015).

[7] Yingzhen Li and Yarin Gal. "Dropout Inference in Bayesian Neural Networks with Alpha-divergences". In: *Proceedings of the 34th International Conference on Machine Learning (ICML-17)* (2017).

[8] Ian Osband. "Risk versus Uncertainty in Deep Learning: Bayes, Bootstrap and the Dangers of Dropout". In: *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (2016).

[9] Ian et al. Osband. "Deep Exploration via Bootstrapped DQN". In: *Advances in Neural Information Processing Systems 29 (NIPS 2016)* (2016).

[10] Re Turner and M. Sahani. "Two problems with variational expectation maximisation for time-series models. " In: *Inference and Estimation in Probabilistic Time-Series Models* (2011).