

Apport d'un modèle Bayésien lors de l'utilisation de réseaux de neurones pour la classification d'images

Comparaison de 2 versions du modèle AlexNet pour la classification d'images, exemple sur un jeu d'images d'insectes



BNN | Avantages V Inconvénients X

Ecole d'Automne
DataSciences 2025

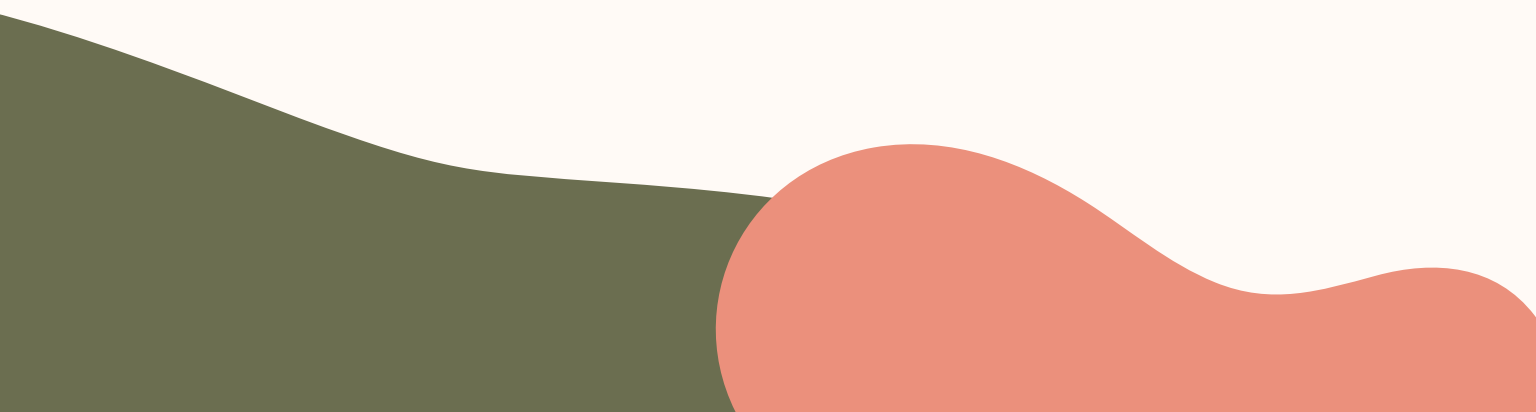
V Incertitude = erreur d'un modèle, recul sur des prédictions

V Hypothèses a priori (possibilité d'utiliser des connaissances a priori)

X Temps de calcul important

X Hypothèses a priori (problème si pas de connaissances a priori)



1. Quelles différence de qualité de prédiction entre CNN et BNN ?
 2. Quelles différences de compromis quantité de données et temps d'entraînement entre les CNN et les BNN ?
- 

Sommaire.

- 1 Démarche et données
- 2 AlexNet : Architecture et fine-tuning d'un CNN
- 3 Méthode : BNN vs CNN
- 4 Résultats

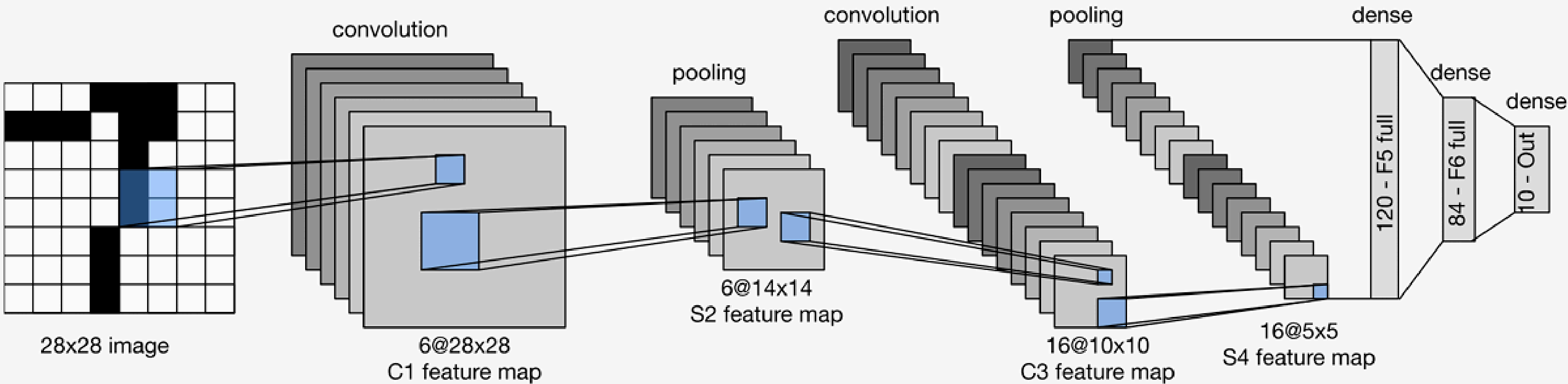
Démarche :

- Choix du modèle CNN
- CNN : mise en place et fine tuning
- Comment passer d'un CNN à un BNN
- BNN : mise en œuvre et fine tuning
- Résultats : comparaison et conclusion

Données :

- "Insect_Detect_classification_v2"
- 21.000 images - 70x70
- 27 classes





Pooling :

Entrée

8	9	10	11	12
6	7	8	9	10
4	5	6	7	8
2	3	4	5	6
0	1	2	3	4

kernel

$$\begin{bmatrix} 3 & 2 \\ 15 & 10 \end{bmatrix}$$

output

$$= \begin{bmatrix} & & 33 & 24 \\ & & 135 & 100 \\ 6 & 6 & & \\ 0 & 10 & & \end{bmatrix}$$

output

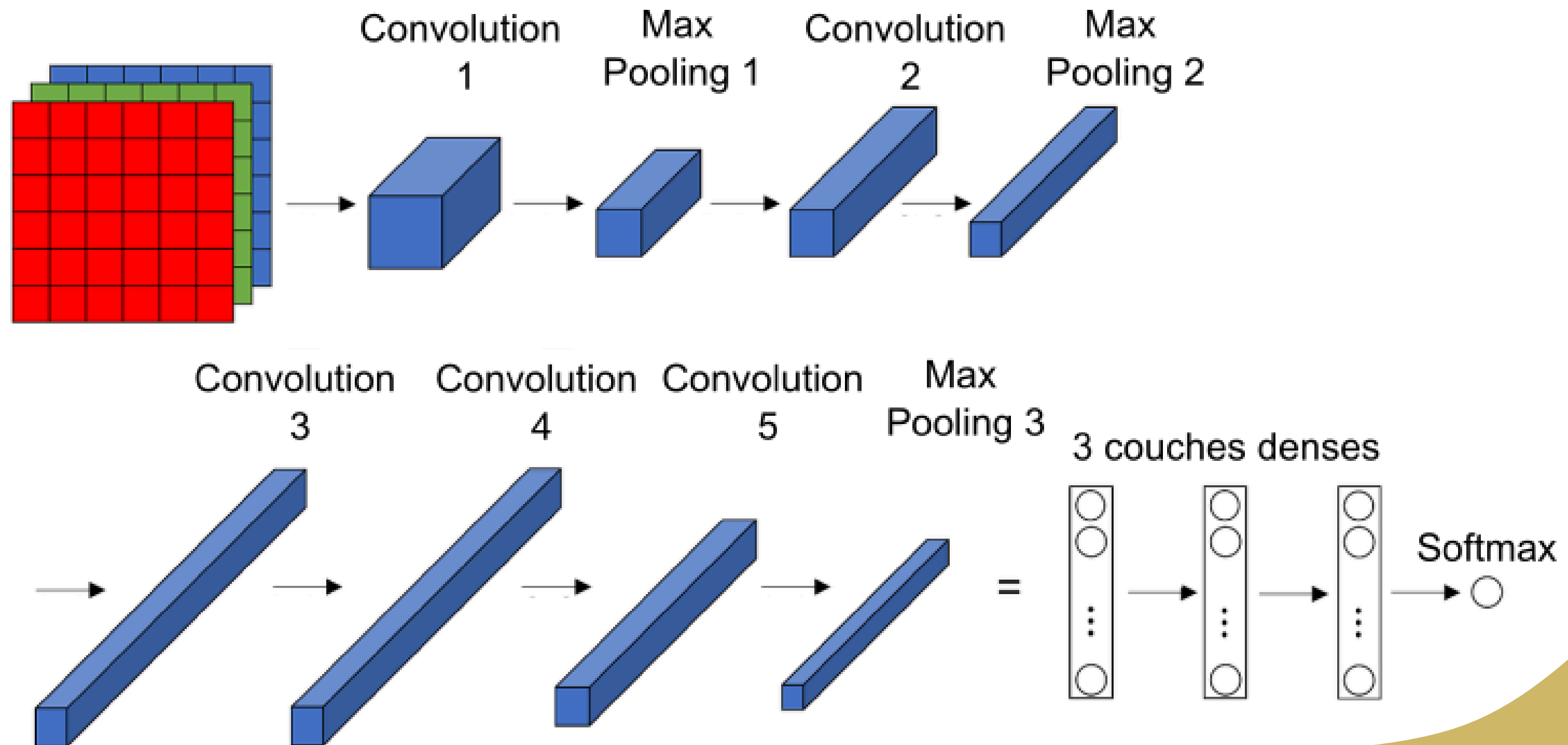
$$\begin{bmatrix} & 135 \\ 10 & \end{bmatrix}$$

2×2
max-pool

AlexNet

Architecture

Ecole d'Automne
DataSciences 2025



1) Traitement des données : ●

- Création de sous jeux de données (5 img / 10 img / 50 img / 100 img / 500 img)
- Fonction dé-zip
- Transformation des images pour afficher du 227x227 + normalisation

2) Fine-tunning : ●

- Importation du pré-trained AlexNet
- Modification du nombre de classes de la dernière couche
- Définition : optimizer (SGD) + fonction loss (cross-entropy)
- Boucle train sur 50 epoches

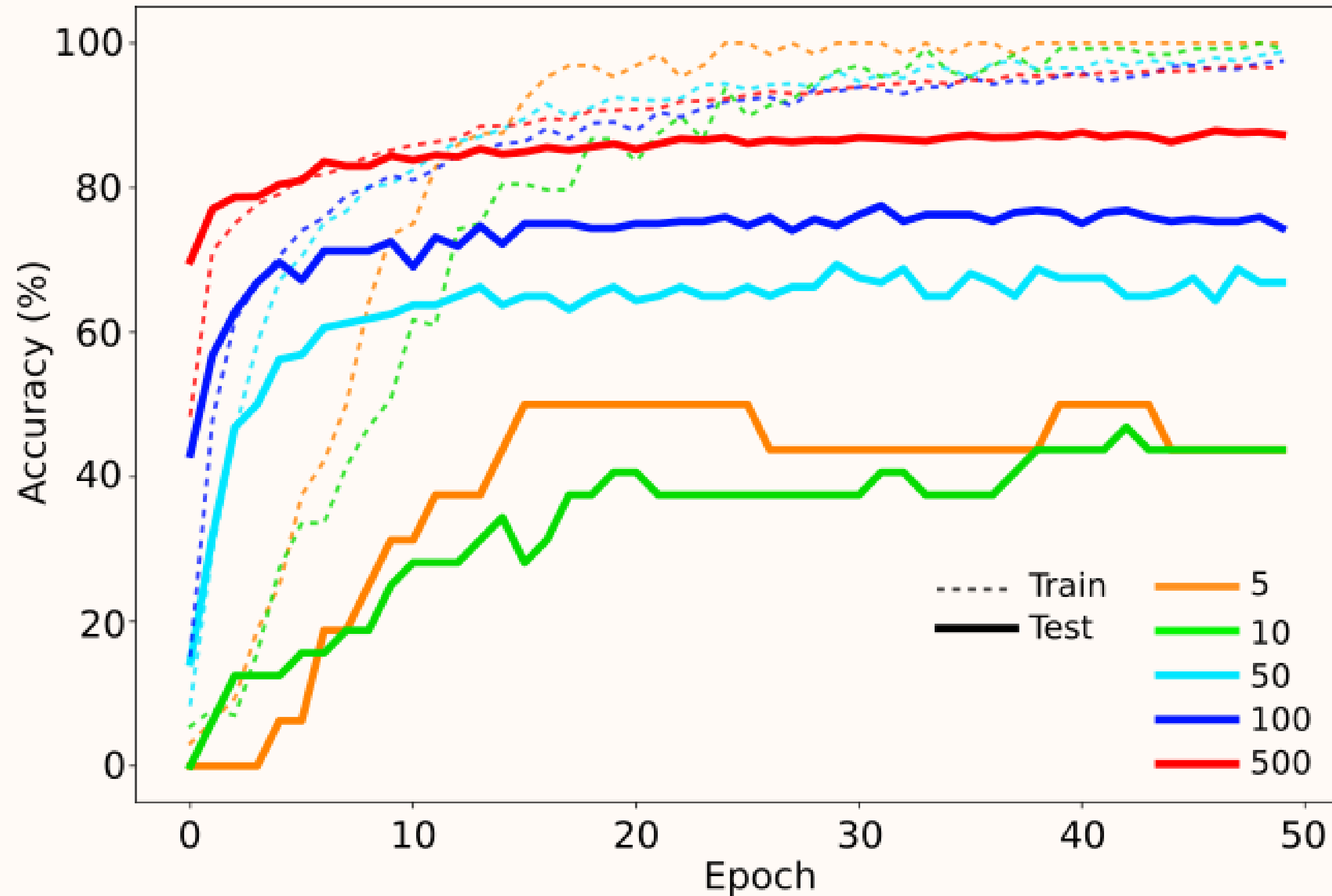
3) Plot ●

AlexNet

CNN : PLOT

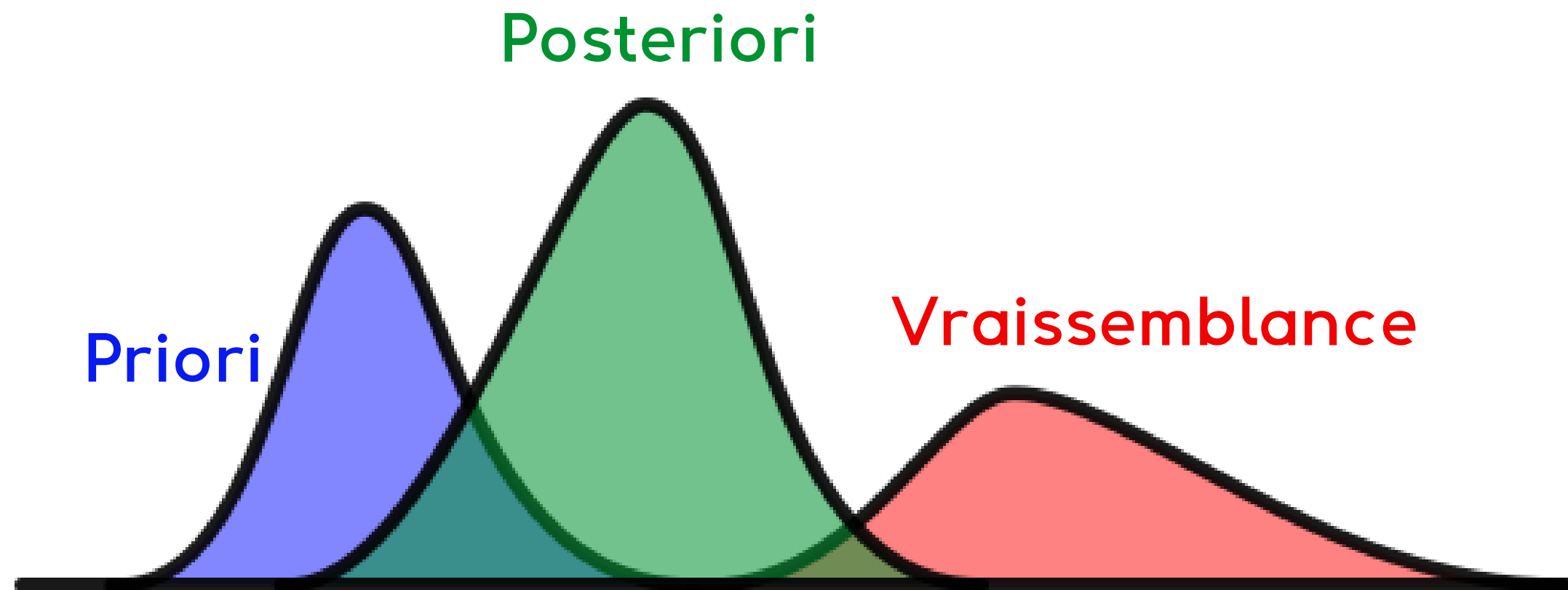
Ecole d'Automne
DataSciences 2025

Variation de l'accuracy en fonction des époques pour
5, 10, 50, 100 et 500 images avec AlexNet



Concept du bayésien :

- Apport de connaissances = **probabilité a priori**
- **Vraisemblance**
- Mise à jour des connaissances = **probabilité à postérieure**



Méthode : CNN vs BNN

Ecole d'Automne
DataSciences 2025

Principe

Qu'est-ce qu'un CNN en bayésien ?

→ Différences CNN et BNN :

Les paramètres ne sont pas des nombres mais des distributions !

- En réseau de neurones, les paramètres sont : **les poids W** !

D'où :

En CNN : Poids = nombres -----> $W = (1, 3, 0.5)$

En BNN : Poids = distribution de probabilité -----> $W = (\text{distribution}_1, \text{distribution}_3, \text{distribution}_{0.5})$

Méthode : CNN vs BNN

Ecole d'Automne
DataSciences 2025

Loi de Bayes

Probabilité des paramètres sachant nos données = compromis entre la probabilité d'observer nos données sachant les paramètres et la probabilité des paramètres sans connaître nos données

x = Entrées de la couche

y = Sortie de la couche

w = Poids des neurones

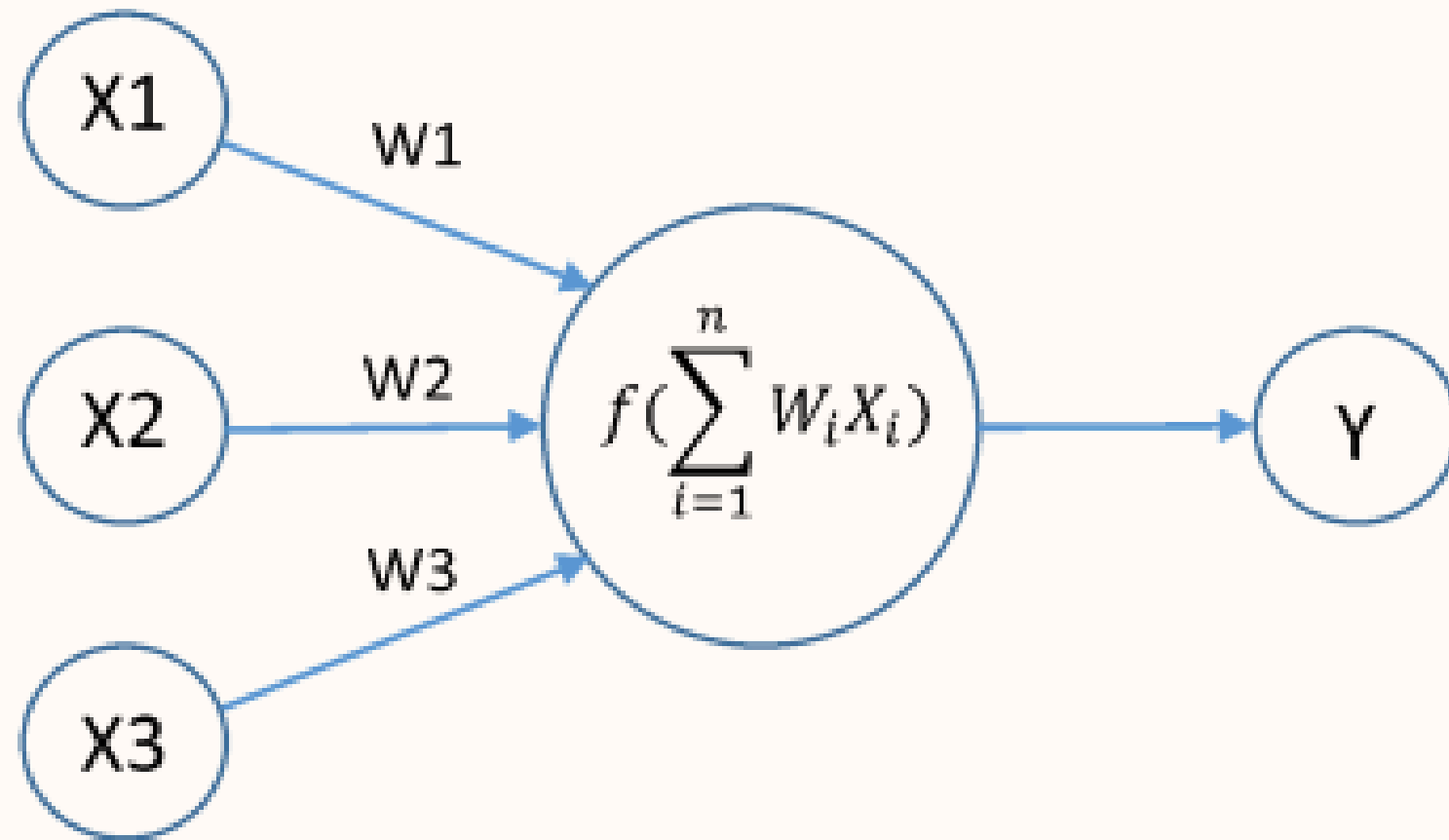
$$\text{Proba a posteriori} \quad \Pr(x, w|y) = \frac{\text{Vraisemblance} \quad \text{Proba a priori} \quad \Pr(y|x, w) \quad \Pr(x, w)}{\Pr(y)}$$

Méthode : CNN vs BNN

Gestion des poids

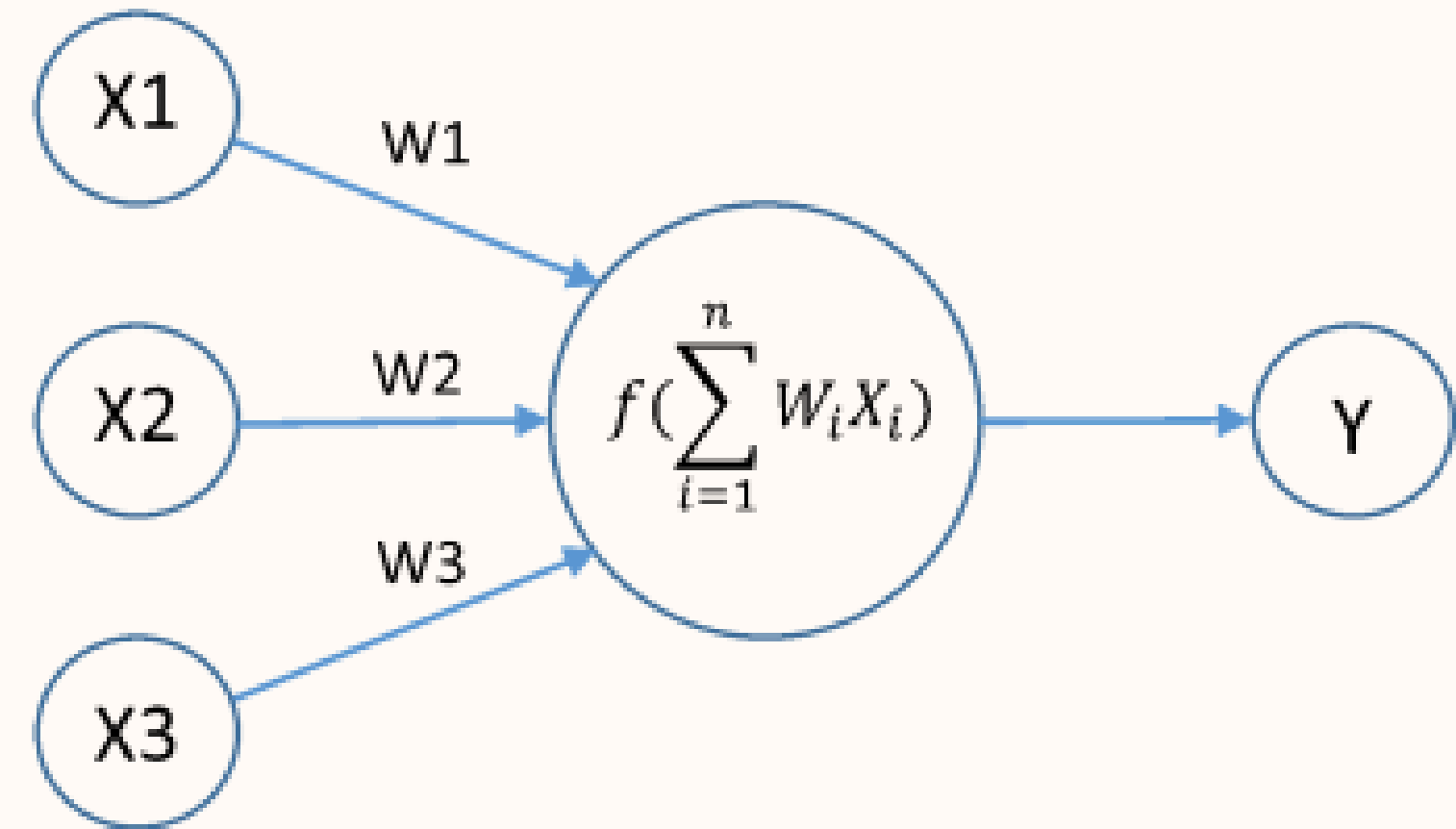
Ecole d'Automne
DataSciences 2025

NN



$$y = w_1 x_1 + w_2 x_2 + w_3 x_3$$

BNN



$$y = w_1 x_1 + w_2 x_2 + w_3 x_3$$

Avec:

$$w_1 \sim N(\mu_1, \sigma_1^2)$$

$$w_2 \sim N(\mu_2, \sigma_2^2)$$

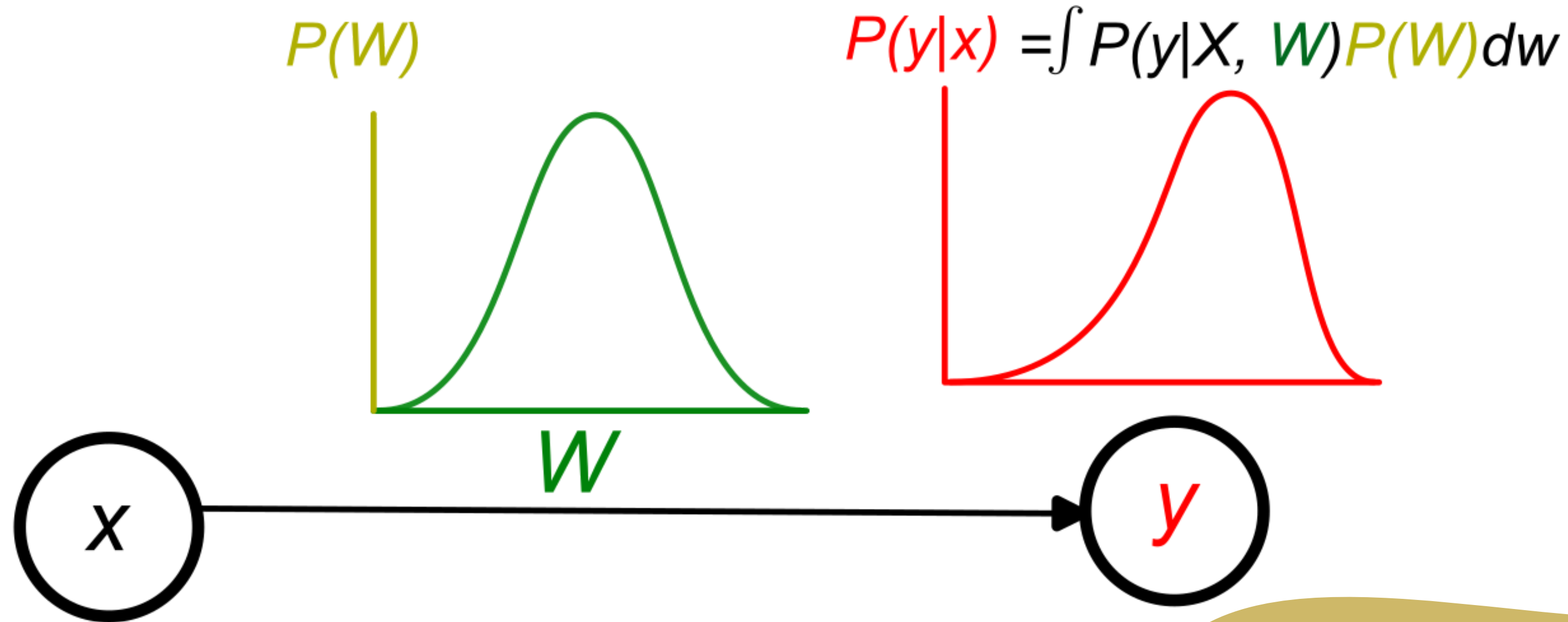
$$w_3 \sim N(\mu_3, \sigma_3^2)$$

Méthode : CNN vs BNN

Gestion des poids : BNN

Ecole d'Automne
DataSciences 2025

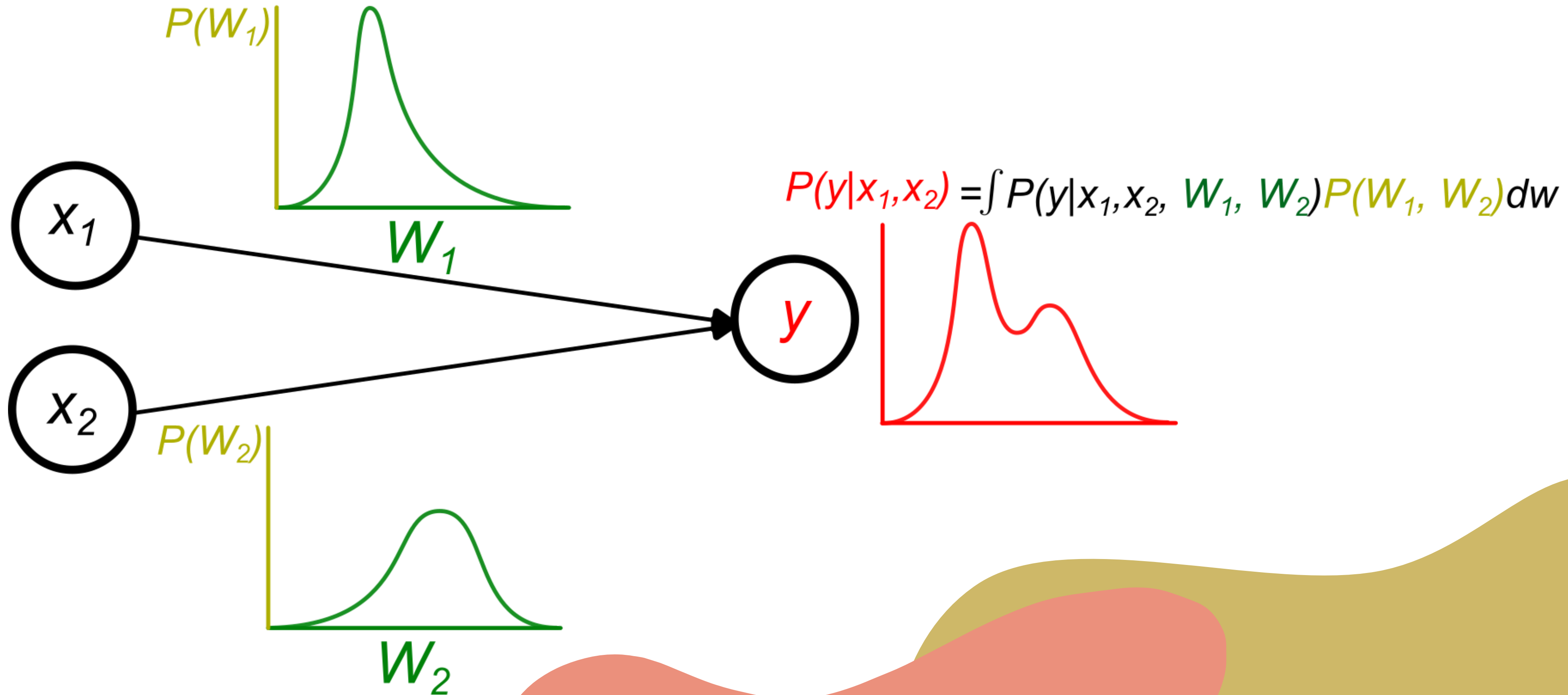
Pour un modèle entraîné:



Méthode : CNN vs BNN

Ecole d'Automne
DataSciences 2025

Gestion des poids : BNN



Méthode : CNN vs BNN

Ecole d'Automne
DataSciences 2025

Entraînement

CNN

$$D_{train} = x_1, \dots, x_n, y_1, \dots, y_n$$

Entraînement:

$$w^* = \arg \max_w \sum_{i=1}^n \log(\textcolor{red}{\Pr(y_i|x_i, w)})$$

Vraisemblance

Prédiction:

$$\textcolor{violet}{\Pr(\hat{y}|x_{new}, w^*)} \Leftrightarrow \hat{y} = f(x_{new}, w^*)$$

BNN

Entraînement:

$$\mu^*, \sigma^* = \arg \max_{\mu, \sigma} \sum_{i=1}^n \log(\textcolor{red}{\Pr(y_i|x_i, w)}) + \log(\textcolor{blue}{\Pr(w, x_i)})$$

Vraisemblance *Proba a priori*

\propto *Proba a posteriori*

$$w^* \sim \mathcal{N}(\mu^*, \sigma^*)$$

Prédiction:

$$\Pr(\hat{y}|x_{new}, D_{train}) = \int \textcolor{violet}{\Pr(\hat{y}|x_{new}, w^*)} \textcolor{green}{\Pr(w^*|D_{train})} dw^*$$

Proba a posteriori

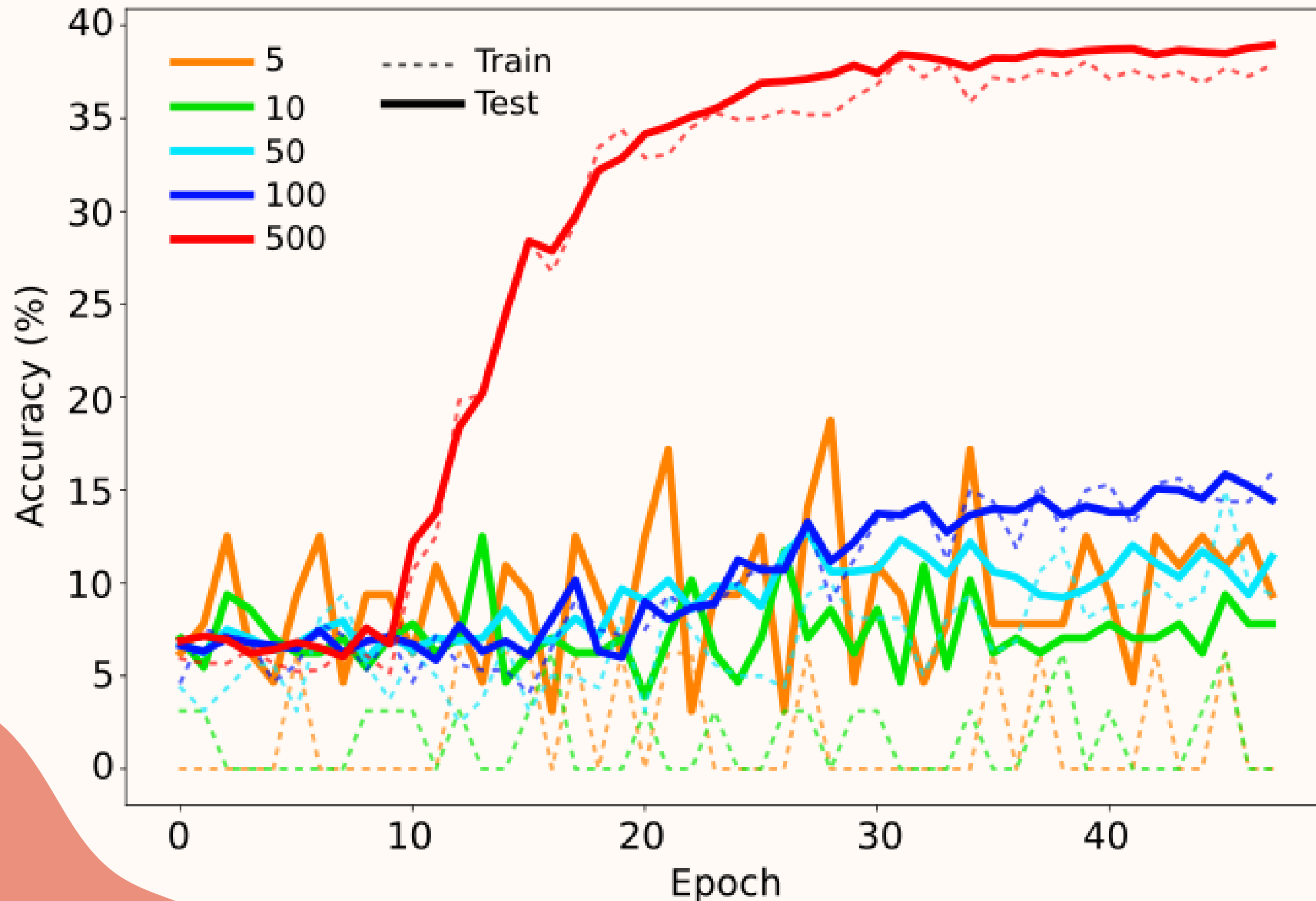
→ Hypothèse du modèle de BNN : Distribution normale des poids

Résultats

AlexNet bayésien

Ecole d'Automne
DataSciences 2025

Variation de l'accuracy en fonction des époques pour
5, 10, 50, 100 et 500 images avec AlexNet bayésien



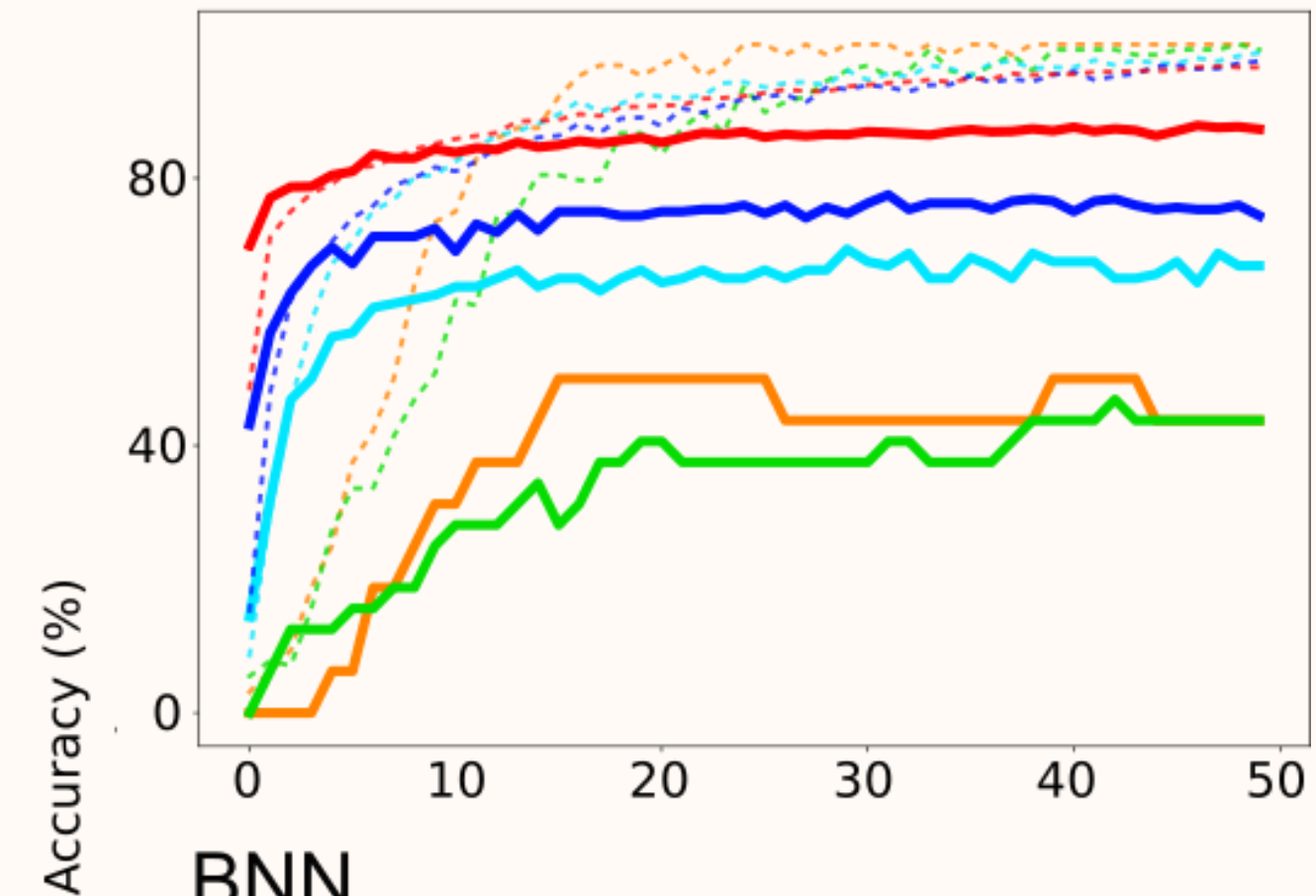
Conclusion

Ecole d'Automne
DataSciences 2025

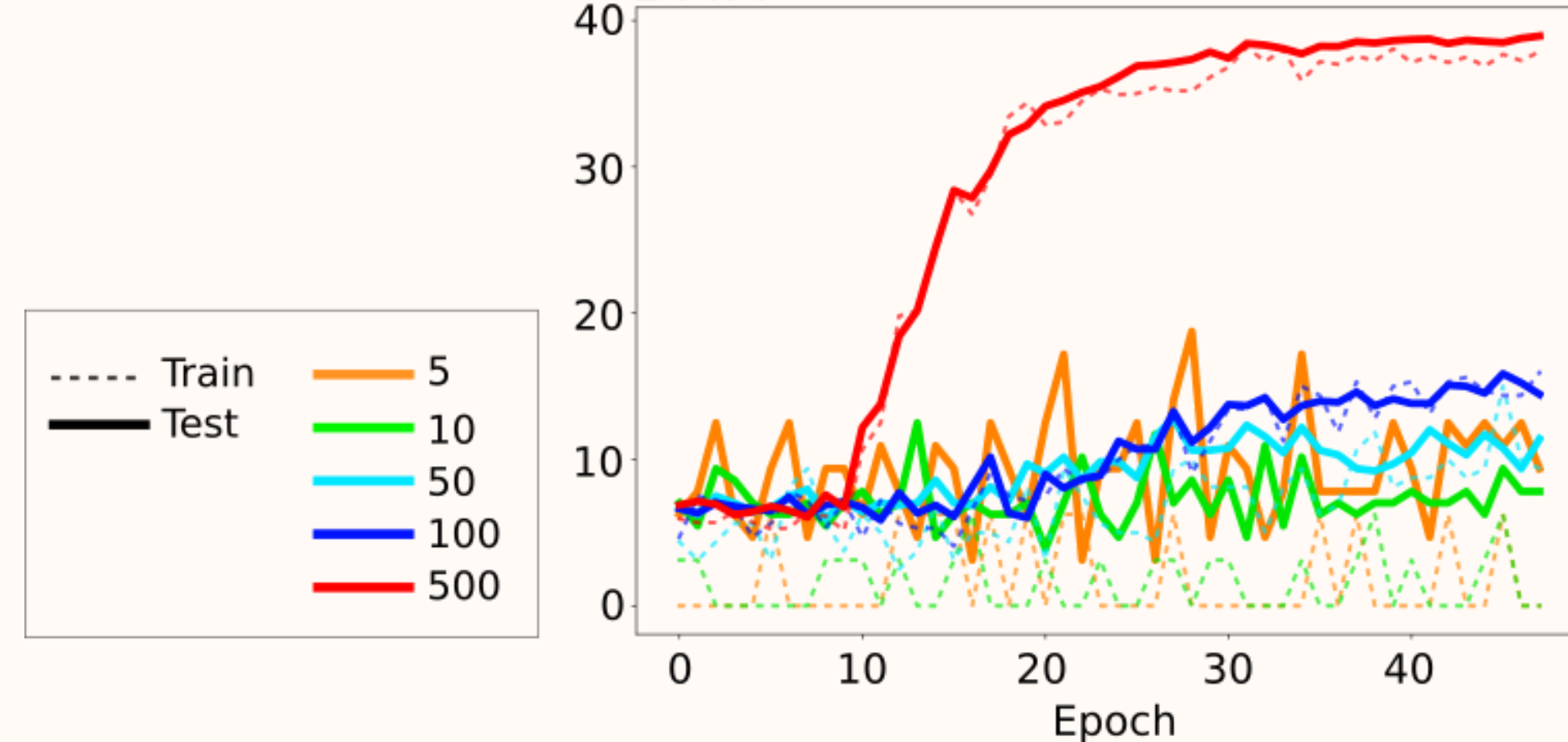
BNN:

- Temps de calcul prohibitif
- Quantité de données d'apprentissage nécessaire supérieur
- Accuracy inférieur pour la même quantité de données

CNN



BNN

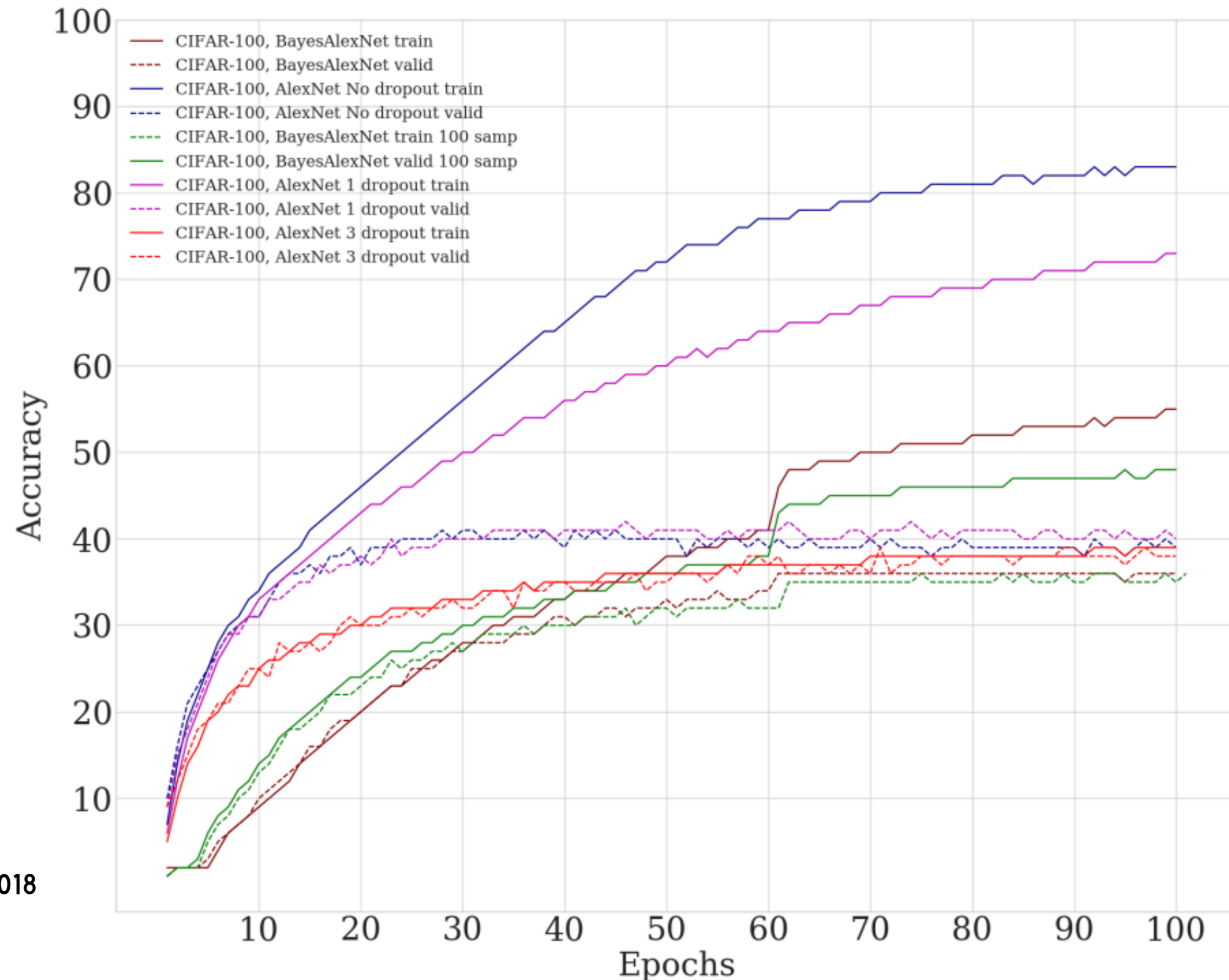


Conclusion

Résultat de la littérature:

- Meilleure performance du BNN
- Écarts d'accuracy entre CNN et BNN plus faibles

Shridhar. K. 2018



- Shridhar. K. 2018, A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference. PhD Thesis, 38 pp.
- Lien GitHub de la thèse Shridhar 2028
<https://github.com/kumar-shridhar/PyTorch-BayesianCNN>
- Lien vers le code de AlexNet
<https://medium.com/@whyamit404/writing-alexnet-from-scratch-in-pytorch-15dfbf06fefe>

A FAIRE :

Ce soir

- Nettoyer Git
- Clean Code BNN parfaitement + Expliquations (txt) + annotations (#..)
- Finir diapo + le corriger + dernier check
- Résultats BNN ? Conclusion BNN ? revoir ce qu'on a a en dire
- DOC TENTATIVE BNN
- MAITRISER PARFAITEMENT CNN oral
- MAITRISER PARFAITEMENT BNN oral
- Questionnaire auto évaluation
- Questionnaire évaluation autre groupe
- dernière couche dense modifiée uniquement ! passage de 4096 à 20 ????