Bachelor en Science Informatique
UNIGE - Faculté des Sciences - Département d'Informatique

Sensibilité structurelle des réseaux convolutionnels aux directions des bases de Fourier

Auteur :
David Tournier Marrucho

Responsable :
Stephane Marchand Maillet

29 mai 2026

Table des matières

Table des notations

Notation	Description
Indices, images et fréquences	
N	Taille spatiale des images considérées. Dans les expériences sur CIFAR, $N = 32$.
$\mathbb{Z}_N$	Ensemble des indices $\{0, \dots, N-1\}$, interprétés modulo N.
(m, n)	Coordonnées spatiales d'un pixel : m pour la direction verticale, n pour la direction horizontale.
(u, v)	Fréquence de Fourier 2D : u pour la fréquence verticale, v pour la fréquence horizontale.
$w(u, v)$	Indice aplati associé à la fréquence (u, v), utilisé après vectorisation.
x, X, Y	Image ou signal d'entrée, image d'entrée matricielle et image de sortie.
Fourier et convolution	
h	Filtre de convolution.
$h_{i,j}$	Filtre reliant le canal d'entrée j au canal de sortie i.
$*$	Convolution circulaire.
ϕ_k	Vecteur de Fourier discret 1D associé à la fréquence k.
$\Phi_{u,v}$	Onde de Fourier 2D associée à la fréquence (u, v).
F_N	Matrice de Fourier 1D dont les colonnes sont les vecteurs ϕ_k.
Q_N	Matrice de Fourier 2D vectorisée, définie par $Q_N = F_N \otimes F_N$.
$\mathrm{vec}(X)$	Vectorisation de l'image X en empilant ses colonnes.
λ_f, $\lambda_{u,v}^{i,j}$	Valeurs propres associées à l'action d'un filtre de convolution sur une fréquence de Fourier.
Représentation matricielle du réseau	
I_r	Matrice identité de taille $r \times r$. En particulier, I_N désigne l'identité de taille N, et I_{m_k} l'identité agissant sur les canaux de la couche k.
M	Matrice représentant une couche de convolution, ou l'opérateur linéaire global selon le contexte.
$M^{(k)}$	Matrice représentant la k-ième couche de convolution de $\mathbb{C}^{m_k N^2}$ vers $\mathbb{C}^{m_{k+1} N^2}$.
m_{in}, m_{out}	Nombre de canaux d'entrée et de sortie d'une couche de convolution considérée isolément.
m_k	Nombre de canaux en entrée de la couche k.
L	Matrice décrivant les mélanges de canaux dans le domaine de Fourier.
$G^{(w)}$	Matrice de mélange associée à une fréquence donnée w.

Notation	Description
σ	Valeur singulière, utilisée pour mesurer l'amplification d'une direction d'entrée.
a, b	Vecteurs singuliers droit et gauche associés à une fréquence donnée.
Perturbations et protocole expérimental	
ϵ	Amplitude maximale autorisée pour la perturbation, mesurée en norme ℓ_∞.
$p_{u,v}$	Motif spatial réel construit à partir de la fréquence (u, v).
$\delta_{u,v}$	Perturbation de Fourier normalisée associée à la fréquence (u, v).
$\tilde{x}$	Image perturbée obtenue après ajout de $\delta_{u,v}$ puis clipping dans $[0, 1]$.
f	Modèle de classification étudié.
$\mathcal{D}$	Ensemble d'images utilisé pour l'évaluation.
$\mathrm{FR}(u, v)$	*Fool ratio* obtenu pour la fréquence (u, v), c'est-à-dire la proportion de prédictions modifiées.
H	Heatmap de sensibilité fréquentielle contenant les valeurs de $\mathrm{FR}(u, v)$.
$(u^\star, v^\star)$	Fréquence qui maximise le *fool ratio*.

1 Introduction

Les réseaux de neurones convolutionnels, souvent appelés CNN pour *Convolutional Neural Networks*, inspirés par le cortex visuel des vertébrés, occupent une place centrale en vision par ordinateur. Ils sont utilisés pour reconnaître des objets, classer des images, détecter des formes ou encore extraire automatiquement des informations visuelles.

Leur efficacité vient en grande partie de leur capacité à apprendre des motifs locaux dans les images, puis à les combiner progressivement pour construire des représentations plus abstraites.

Fragilités et attaques adversariales

Cependant, malgré leurs très bonnes performances, ces modèles peuvent présenter des fragilités structurelles. Une image peut être faiblement modifiée, de telle manière que cela devienne imperceptible à l'oeil humain, tout en entraînant un changement important dans la prédiction du réseau. Ce phénomène est connu sous le nom de **perturbation adversariale** [13].

Un aspect particulièrement intéressant de ces perturbations est leur caractère parfois universel. Dans certains cas, il est possible de trouver une perturbation qui est directement basée sur l'architecture du réseau (sa manière de fonctionner), exploitant une faiblesse structurelle du fonctionnement interne de ce dernier. Ainsi ce genre de perturbation, une fois trouvée, peut être appliquée à n'importe quelle image, **indépendamment** de son contenu, et tromper le modèle, pour une proportion importante d'images, dans ses prédictions. On parle alors de perturbation adversariale **universelle** [10].

La question centrale est alors de comprendre d'où vient cette sensibilité. Une première explication possible consiste à l'attribuer aux données d'entraînement ou à l'optimisation du modèle. Dans ce travail, nous nous intéressons à une autre piste : **la structure même des réseaux convolutionnels**. En effet, les CNN reposent sur des couches de convolution, et ces couches possèdent des propriétés mathématiques particulières.

L'approche par la base de Fourier

Pour étudier cette question, nous choisissons d'utiliser la base de Fourier. Une image peut être décrite pixel par pixel, mais elle peut aussi être décomposée en motifs fréquentiels : des variations lentes, des oscillations rapides, des motifs horizontaux, verticaux ou diagonaux. Ces motifs fréquentiels donnent une autre manière de décrire les structures présentes dans une image.

Ainsi, analyser une image non plus seulement dans le domaine spatial, c'est-à-dire à partir de ses pixels, mais aussi dans le domaine fréquentiel, permet de mieux comprendre quelles composantes de l'image influencent le comportement du réseau et quelles directions de perturbation peuvent devenir particulièrement sensibles.

Ce lien constitue le point de départ du travail de Tsuzuku et Sato [14], qui montrent que certains réseaux convolutionnels sont particulièrement sensibles à des directions issues de la base de Fourier. Leur idée est que certaines perturbations adversariales universelles peuvent être

mieux comprises en étudiant la manière dont les couches convolutionnelles traitent les fréquences.

Comment l'architecture convolutionnelle des CNN induit-elle des directions de vulnérabilité privilégiées dans le domaine de Fourier, et dans quelle mesure cette sensibilité structurelle permet-elle de mieux comprendre l'existence de perturbations adversariales universelles ?

L'objectif de ce rapport est de reprendre cette idée et de la reconstruire petit à petit de manière plus détaillée en introduisant les éléments nécessaires pour comprendre les résultats, de telle manière que n'importe quel lecteur avec une connaissance de base en algèbre linéaire et en machine learning puisse comprendre les implications de ce travail.

Organisation du rapport

Pour répondre à cette problématique, le rapport est organisé de la manière suivante :

- **Chapitre 2 :** Présentation du contexte général : réseaux convolutionnels, robustesse et perturbations adversariales.
- **Chapitre 3 :** Présentation des fondements théoriques reliant convolution et base de Fourier. On y montre notamment l'émergence naturelle des fonctions de Fourier dans l'analyse linéaire.
- **Chapitre 4 :** Présentation de la reproduction expérimentale, du protocole utilisé, des *heatmaps* de sensibilité fréquentielle, puis discussion des résultats et des limites.
- **Conclusion :** Synthèse des enseignements et perspectives de recherche.

2 Contexte

Ce chapitre fixe les notions utiles pour la suite : les CNN comme modèles fondés sur la convolution, la robustesse comme stabilité de la prédiction, et les perturbations adversariales comme directions dans l'espace des images.

2.1 Les réseaux convolutionnels en vision par ordinateur

Une image numérique peut être vue comme un tenseur $X \in \mathbb{R}^{C\times H\times W}$, où C désigne les canaux couleur, et H, W la taille spatiale. Un classifieur apprend une fonction

$$f : \mathbb{R}^{C\times H\times W} \longrightarrow \{1, \ldots, K\},$$

qui associe à chaque image une classe parmi K catégories.

Les réseaux convolutionnels exploitent la structure spatiale de l'image. Contrairement à une couche entièrement connectée, une convolution applique le même filtre local à toutes les positions. Ce partage des poids permet de détecter un motif, par exemple un bord ou une texture, indépendamment de sa position.

En empilant plusieurs couches, le réseau combine des motifs locaux en représentations plus abstraites. Les premières couches capturent des structures simples, tandis que les couches profondes construisent des informations plus liées aux objets. La figure 2.1 illustre ce principe.

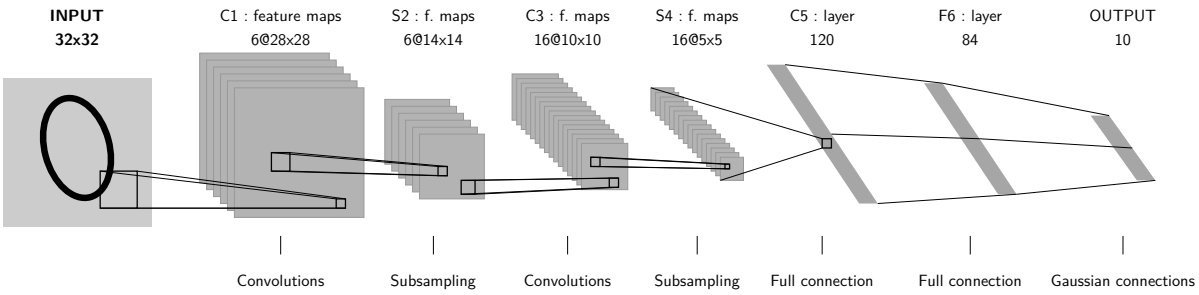


FIGURE 2.1 – Architecture du réseau de neurones convolutionnel LeNet-5. Adapté de [8].

L'output de taille 10 correspond ici à une prédiction parmi les dix chiffres possibles.

2.2 Robustesse des modèles de vision

La performance est souvent mesurée sur des images propres, proches de celles utilisées pendant l'entraînement. Cette mesure ne suffit pas : un modèle peut être précis, mais instable face à de petites variations.

Soit une image x et une perturbation δ. Un modèle est robuste localement si sa prédiction reste stable lorsque x est remplacée par $x + \delta$:

$$f(x + \delta) = f(x) \qquad \text{pour des perturbations } \delta \text{ suffisamment petites.}$$

2.3 Perturbations adversariales

Une perturbation adversariale est une modification choisie pour changer la prédiction d'un modèle, tout en restant faible ou peu visible pour l'oeil humain [13]. Ce n'est pas un bruit aléatoire : elle suit une direction qui influence fortement la sortie du réseau.

À partir d'une image x, on construit une image perturbée

$$\widetilde{x} = x + \delta,$$

avec une perturbation δ suffisamment petite, mais telle que

$$f(\widetilde{x}) \neq f(x).$$

Dans ce rapport, ces perturbations servent surtout d'outil d'analyse : elles indiquent quelles directions de l'espace des images sont fortement amplifiées par le modèle.

2.4 Perturbations universelles et directions de vulnérabilité

Dans le cas le plus simple, une perturbation adversariale est construite pour une image donnée. Elle dépend donc fortement de cette image.

Une perturbation adversariale universelle repose sur une idée plus forte : trouver une seule perturbation δ, que l'on ajoute à de nombreuses images différentes [10].

Une telle perturbation n'exploite donc pas seulement une particularité d'une image précise. Elle exploite une direction sensible du modèle, liée aux poids appris, aux données, aux non-linéarités, mais aussi à la structure de l'architecture. C'est ce dernier point qui motive ce rapport.

Si l'on considère une image comme un point dans un espace de grande dimension, une perturbation est un déplacement. Certaines directions ont peu d'effet, tandis que d'autres traversent rapidement une frontière de décision, comme illustré dans la figure 2.2.

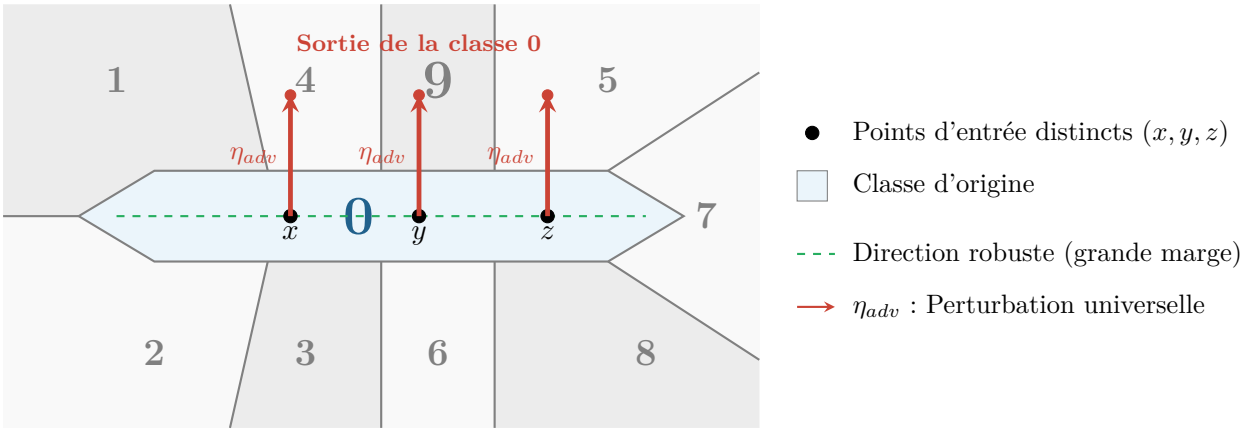


FIGURE 2.2 – Illustration d'une perturbation universelle dans une région de décision anisotrope. La même direction de perturbation η_{adv} est appliquée à plusieurs points d'entrée distincts x, y et z et les fait sortir de la région associée à la classe 0.

3 Fondements théoriques : convolution et base de Fourier

3.1 Objectif de cette section

Cette section a pour objectif de reconstruire le raisonnement théorique proposé par Tsuzuku et Sato [14]. Il s'agit de comprendre pourquoi les réseaux convolutionnels présentent une sensibilité particulière à certaines directions de la base de Fourier, en caractérisant mathématiquement ces vulnérabilités.

L'idée générale repose sur la structure mathématique des couches convolutionnelles. Pour l'étudier, nous nous placerons dans un cadre linéaire simplifié, en faisant fi des éléments classiquement introduits le long du processus de décision d'un CNN pour en briser la linéarité (tels que les fonctions d'activation non linéaires et le *max-pooling*).

Les développements qui suivent utilisent deux rappels classiques d'algèbre linéaire : la représentation matricielle d'une application linéaire, et le lien entre vecteurs propres, valeurs propres et diagonalisation. Ces rappels sont donnés en annexes A et B.

3.2 Conventions de Fourier et de vectorisation

Avant d'étudier les convolutions, nous fixons les conventions utilisées dans tout le chapitre.

Dans la suite, toutes les fréquences et tous les indices spatiaux seront interprétés modulo N.

3.2.1 Base de Fourier 1D et transformée discrète

Pour un entier $N \geq 1$, on note

$$\mathbb{Z}_N = \{0, \dots, N-1\}$$

l'ensemble des indices pris modulo N. Concrètement, cela signifie que l'indice N est identifié à l'indice 0, l'indice $N+1$ à l'indice 1, et ainsi de suite. Cette convention sera utilisée dans tout le chapitre.

Pour chaque fréquence $k \in \mathbb{Z}_N$, on définit le vecteur de Fourier discret normalisé

$$\phi_k[n] = \frac{1}{\sqrt{N}} e^{2\pi i k n/N}, \qquad n \in \mathbb{Z}_N. \tag{3.1}$$

Le vecteur ϕ_k représente une onde discrète de fréquence k. Les petites valeurs de k correspondent à des variations lentes, tandis que les fréquences plus élevées correspondent à des oscillations plus rapides sur les N points du signal. Le facteur $1/\sqrt{N}$ sert à normaliser le vecteur, ce qui donne

$$\|\phi_k\|_2 = 1.$$

On regroupe tous ces vecteurs dans une même matrice :

$$F_N = \begin{pmatrix} | & | & & | \\ \phi_0 & \phi_1 & \cdots & \phi_{N-1} \\ | & | & & | \end{pmatrix}. \tag{3.2}$$

La matrice F_N contient donc, colonne par colonne, toutes les directions de Fourier possibles pour un signal de taille N. Avec la normalisation choisie, ses colonnes forment une base orthonormée de $\mathbb{C}^N$, c'est-à-dire

$$F_N^H F_N = I_N, \qquad F_N^{-1} = F_N^H.$$

La transformée de Fourier discrète consiste alors à exprimer un signal $x \in \mathbb{C}^N$ dans cette base. Avec notre convention, les coefficients de Fourier de x sont donnés par

$$\widehat{x} = F_N^H x, \tag{3.3}$$

c'est-à-dire, composante par composante,

$$\widehat{x}[k] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] e^{-2\pi i k n/N}. \tag{3.4}$$

Le coefficient $\widehat{x}[k]$ mesure la contribution de la fréquence k dans le signal.

Comme F_N est unitaire, on peut reconstruire le signal à partir de ses coefficients de Fourier par

$$x = F_N \widehat{x}, \tag{3.5}$$

ou encore, composante par composante,

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \widehat{x}[k] e^{2\pi i k n/N}. \tag{3.6}$$

3.2.2 Base de Fourier 2D

On passe maintenant des signaux 1D aux images 2D. Pour une image $X \in \mathbb{C}^{N \times N}$, on utilise les indices spatiaux

$$(m, n) \in \mathbb{Z}_N^2.$$

L'indice m désigne la position verticale, et l'indice n la position horizontale.

De la même manière, une fréquence 2D est décrite par un couple

$$(u, v) \in \mathbb{Z}_N^2.$$

L'indice u correspond à la fréquence verticale, tandis que l'indice v correspond à la fréquence horizontale.

Pour chaque couple de fréquences (u, v), on définit l'image de Fourier 2D

$$\Phi_{u,v}[m, n] = \phi_u[m]\phi_v[n] = \frac{1}{N} e^{2\pi i(um+vn)/N}. \tag{3.7}$$

Cette formule signifie qu'une onde de Fourier 2D est obtenue comme le produit d'une onde verticale et d'une onde horizontale sous forme matricielle : $\Phi_{u,v} = \phi_u \phi_v^T$.

En faisant varier les fréquences u et v, on obtient différents motifs périodiques dans l'espace des images. Visuellement, ces motifs peuvent prendre la forme de bandes, de grilles ou d'oscillations orientées, dont la densité dépend directement du couple (u, v). Ainsi, ajouter une direction de Fourier à une image revient à lui superposer un motif géométrique périodique.

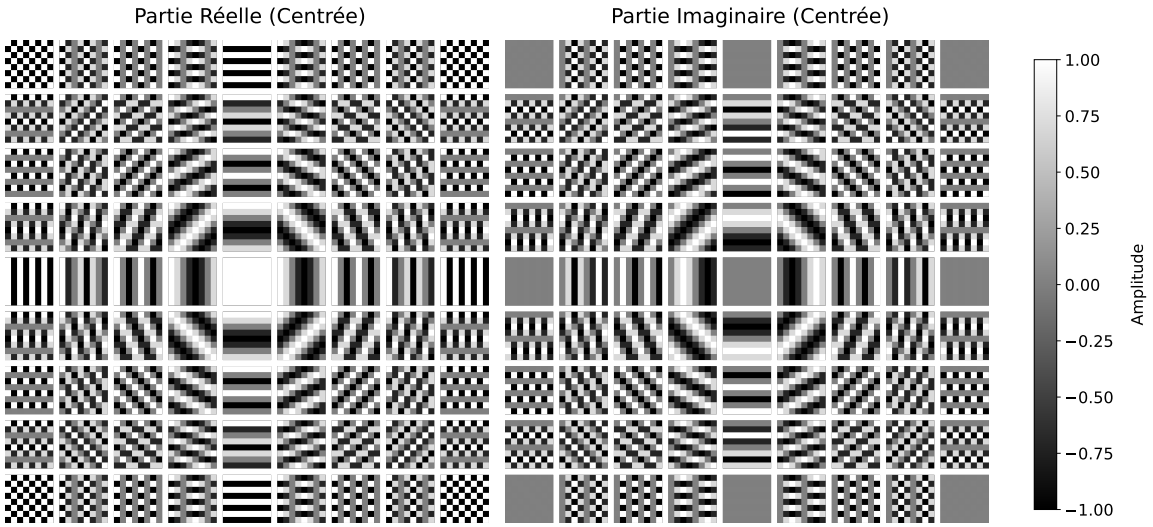


FIGURE 3.1 – Visualisation des bases de Fourier 2D pour tous les couples de fréquences (u, v) d'une image de taille 9×9. La partie réelle et la partie imaginaire sont affichées séparément. Les basses fréquences sont situées au centre des grilles, tandis que les hautes fréquences se trouvent en périphérie.

Dans les figures, on représentera parfois les fréquences de manière centrée, avec les basses fréquences au centre de l'image. Cette convention est pratique pour la visualisation. En revanche, dans les formules, les indices restent toujours dans $\{0, \ldots, N-1\}$ et sont interprétés modulo N.

3.2.3 Vectorisation et produit de Kronecker

Dans tout le chapitre, nous vectorisons les images en empilant les colonnes. Pour $X \in \mathbb{C}^{N\times N}$, on pose donc

$$\mathrm{vec}(X) = \begin{pmatrix} X[0,0] \\ X[1,0] \\ \vdots \\ X[N-1,0] \\ X[0,1] \\ \vdots \\ X[N-1,N-1] \end{pmatrix}. \tag{3.8}$$

On lit d'abord la première colonne, puis la deuxième, et ainsi de suite. Avec cette convention, l'entrée $X[m,n]$ correspond à l'indice vectorisé

$$m + Nn.$$

Nous utiliserons aussi le produit de Kronecker, noté $\otimes$. Défini comme suit, soient $A \in \mathbb{C}^{m\times n}$ et $B \in \mathbb{C}^{p\times q}$, alors la matrice par blocs de taille $mp \times nq$ est définie par

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}.$$

Il permet notamment de construire une base de Fourier 2D à partir de deux bases de Fourier 1D. Les propriétés algébriques de la vectorisation et du produit de Kronecker sont déjà connues en analyse matricielle [5].

Avec notre convention de vectorisation par colonnes, on a en particulier

$$\mathrm{vec}(ab^T) = b \otimes a. \tag{3.9}$$

En appliquant cette identité à l'image de Fourier

$$\Phi_{u,v} = \phi_u \phi_v^T,$$

on obtient

$$\mathrm{vec}(\Phi_{u,v}) = \phi_v \otimes \phi_u. \tag{3.10}$$

Ainsi, dans toute la suite du chapitre, la colonne de Q_N associée à la fréquence 2D (u,v) sera notée

$$q_{u,v} = \phi_v \otimes \phi_u.$$

On note alors

$$Q_N = F_N \otimes F_N \tag{3.11}$$

la base de Fourier 2D vectorisée. Par unitarité de F_N :

$$Q_N^H Q_N = I_{N^2}.$$

Les colonnes de Q_N forment donc une base orthonormée de $\mathbb{C}^{N^2}$.

Avec la notation précédente, la colonne associée à la fréquence (u, v) est

$$(Q_N)_{w(u,v)} = q_{u,v} = \phi_v \otimes \phi_u.$$

L'indice $w(u, v)$ est simplement l'indice aplati qui permet de passer du couple de fréquences (u, v) à un seul indice de colonne dans Q_N. Il est défini par

$$w(u, v) = u + Nv. \tag{3.12}$$

Réciproquement, si l'on connaît un indice aplati $w \in \{0, \dots, N^2 - 1\}$, on retrouve le couple de fréquences correspondant par

$$u = w \bmod N, \qquad v = \left\lfloor \frac{w}{N} \right\rfloor.$$

3.3 Convolution discrète en dimension 1

Avant d'étudier les images, nous commençons par le cas plus simple d'un signal discret périodique représenté par sa période $x = (x[0], x[1], \dots, x[N-1])$ avec $x[i] \in \mathbb{R}$. Soit $x \in \mathbb{R}^N$ notre signal et $h = (h[0], h[1], \dots, h[N-1]) \in \mathbb{R}^N$ notre filtre (qui sera supposé de même taille que le signal, quitte à être complété par des zéros). La convolution circulaire entre h et x est définie par :

$$y[n] = (h * x)[n] = \sum_{k=0}^{N-1} h[k]\, x[(n-k) \bmod N] \quad [15]$$

L'opération modulo signifie que les indices du signal sont pris dans $\mathbb{Z}_N$. Ainsi, lorsqu'un indice sort de l'intervalle $\{0, \dots, N-1\}$, il est ramené à sa classe correspondante dans $\mathbb{Z}_N$. Le signal est donc interprété comme périodique, comme l'illustre la figure 3.2.

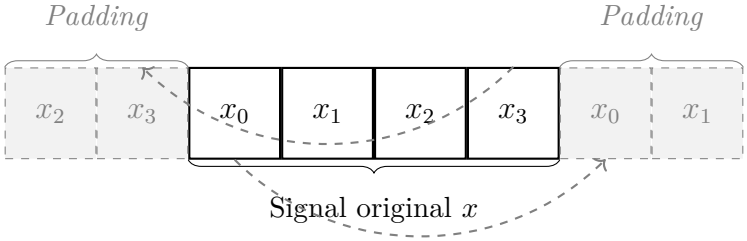


FIGURE 3.2 – Représentation du *padding* circulaire pour un signal 1D. Les bords du signal se répètent de manière périodique.

Remarque sur l'hypothèse de périodicité. Dans cette section, nous travaillons avec des convolutions circulaires. Cela revient à considérer les signaux comme périodiquement prolongés : lorsque l'on dépasse une extrémité du signal, on revient de l'autre côté. Cette hypothèse est très utile mathématiquement, car elle permet de diagonaliser les convolutions par la base de Fourier discrète.

Elle ne correspond cependant pas toujours au comportement exact des images réelles aux bords : si les extrémités ne se raccordent pas naturellement, la périodisation peut créer une discontinuité artificielle. Ce point est discuté plus en détail en annexe C.1.

3.4 Les fonctions de Fourier comme fonctions propres de la convolution circulaire

Cas unidimensionnel On se place dans l'espace des signaux discrets périodiques de période N :

$$V_N = \{x : \mathbb{Z}_N \to \mathbb{C}\}.$$

Un élément de V_N peut être vu soit comme une fonction discrète périodique, soit comme un vecteur de $\mathbb{C}^N$.

Pour un filtre $h \in \mathbb{C}^N$ fixé, la convolution circulaire définit un opérateur linéaire

$$\mathcal{C}_h : V_N \to V_N, \qquad \mathcal{C}_h(x) = h * x.$$

Pour chaque fréquence $f \in \{0, \dots, N-1\}$, on utilise le vecteur de Fourier normalisé ϕ_f défini en (3.1).

Le résultat fondamental liant la convolution circulaire et la base de Fourier s'écrit alors :

$$\mathcal{C}_h(\phi_f) = \lambda_f \phi_f, \tag{3.13}$$

c'est-à-dire, composante par composante :

$$(h * \phi_f)[n] = \lambda_f \phi_f[n],$$

où

$$\lambda_f = \sum_{k=0}^{N-1} h[k] e^{-2i\pi fk/N}.$$

Il est important de remarquer que λ_f dépend du filtre h et de la fréquence f, mais pas de la position n. Ainsi, lorsque l'entrée de l'opérateur de convolution circulaire est une onde de Fourier, la convolution ne modifie pas sa forme spatiale : elle se contente de la multiplier par un scalaire complexe. Autrement dit, λ_f est une valeur propre et ϕ_f une fonction propre de l'opérateur de convolution circulaire $\mathcal{C}_h$.

Démonstration.

$$\begin{aligned}
(\mathcal{C}_h(\phi_f))[n] &= (h * \phi_f)[n] \\
&= \sum_{k=0}^{N-1} h[k]\, \phi_f[(n-k) \bmod N] \\
\text{(périodicité de } \phi_f) \quad &= \sum_{k=0}^{N-1} h[k]\, \phi_f[n-k] \\
\text{(propriété de l'exponentielle)} \quad &= \sum_{k=0}^{N-1} h[k] e^{-2\mathrm{i}\pi fk/N} \phi_f[n] \\
&= \left(\sum_{k=0}^{N-1} h[k] e^{-2\mathrm{i}\pi fk/N} \right) \phi_f[n] \\
&= \lambda_f \phi_f[n].
\end{aligned}$$

□

Ce résultat correspond au théorème de convolution discret : dans le domaine de Fourier, une convolution circulaire devient une multiplication fréquence par fréquence. Ainsi, lorsque l'entrée est une onde de Fourier pure ϕ_f, la convolution ne change pas sa fréquence ; elle la multiplie seulement par un scalaire. On obtient donc 3.13

3.4.1 Conséquence matricielle : diagonalisation des matrices circulantes

Pour un filtre h fixe, la convolution circulaire 1D définit une application linéaire de $\mathbb{C}^N \to \mathbb{C}^N$ car pour tous $x, y \in \mathbb{C}^N$ et tous $\alpha, \beta \in \mathbb{C}$ nous avons que

$$h * (\alpha x + \beta y) = \alpha(h * x) + \beta(h * y)$$

Démonstration.

$$\begin{aligned}
(h * (\alpha x + \beta y))[n] &= \sum_{k=0}^{N-1} h[k]\, (\alpha x + \beta y)[(n-k) \bmod N] \\
&= \sum_{k=0}^{N-1} h[k]\, (\alpha x[(n-k) \bmod N] + \beta y[(n-k) \bmod N]) \\
&= \alpha \sum_{k=0}^{N-1} h[k]\, x[(n-k) \bmod N] + \beta \sum_{k=0}^{N-1} h[k]\, y[(n-k) \bmod N] \\
&= \alpha(h * x)[n] + \beta(h * y)[n].
\end{aligned}$$

□

Comme cette opération est linéaire, le rappel de l'annexe A garantit qu'elle peut être représentée, une fois une base choisie, par une matrice $C_h \in \mathbb{C}^{N \times N}$ telle que pour tout signal d'entrée $x \in \mathbb{C}^N$:

$$y = C_h x.$$

où C_h est une matrice circulante construite à partir du filtre h telle que la première ligne est organisée de manière à reproduire $y[0]$, puis chaque ligne suivante est obtenue par décalage circulaire :

$$C_h = \begin{pmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{pmatrix}.$$

En effet, $y[0] = h[0]x[0] + h[1]x[N-1] + \cdots + h[N-1]x[1]$ correspond à la première ligne de C_h multipliée par le vecteur x, en appliquant itérativement la même logique on obtient les autres éléments de y.

En choisissant la base canonique de $\mathbb{C}^N$, l'opérateur linéaire $\mathcal{C}_h$ est représenté par la matrice circulante C_h. La relation $\mathcal{C}_h(\phi_f) = \lambda_f \phi_f$ peut alors être réécrite sous forme matricielle :

$$C_h \phi_f = \lambda_f \phi_f \tag{3.14}$$

Ainsi, chaque fonction de Fourier ϕ_f est un vecteur propre de la matrice circulante C_h, avec valeur propre associée λ_f. Ce qui signifie que dans le cadre circulaire, la convolution agit diagonalement dans la base de Fourier.

En utilisant la matrice de Fourier F_N définie en (3.2), on regroupe les valeurs propres dans la matrice diagonale :

$$\Lambda_h = \mathrm{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{N-1}).$$

L'égalité vectorielle (3.14) pour toutes les fréquences devient :

$$\begin{aligned} & & C_h F_N &= F_N \Lambda_h \\ \text{(par unitarité de } F_N\text{)} \qquad & \iff & C_h &= F_N \Lambda_h F_N^H. \end{aligned} \tag{3.15}$$

3.5 Convolution circulaire en dimension 2

Nous étendons maintenant le résultat précédent au cas des images carrées. Soit $X \in \mathbb{C}^{N \times N}$ une image et soit $h \in \mathbb{C}^{N \times N}$ un filtre. La convolution circulaire 2D est définie par

$$Y[m,n] = (h * X)[m,n] = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} h[a,b]\, X[(m-a) \bmod N, (n-b) \bmod N]. \tag{3.16}$$

Nous utilisons les images de Fourier 2D définies en (3.7) :

$$\Phi_{u,v}[m,n] = \phi_u[m]\phi_v[n].$$

Comme en dimension 1, un décalage circulaire d'une base de Fourier ne change pas sa forme, mais la multiplie par un facteur complexe :

$$\phi_u[(m-a) \bmod N] = e^{-2\pi iua/N}\phi_u[m], \qquad \phi_v[(n-b) \bmod N] = e^{-2\pi ivb/N}\phi_v[n].$$

Ainsi,

$$\Phi_{u,v}[(m-a) \bmod N, (n-b) \bmod N] = e^{-2\pi i(ua+vb)/N}\Phi_{u,v}[m,n],$$

ou $(ua+vb) \in \mathbb{Z}_N$.

En appliquant la convolution circulaire 2D à $\Phi_{u,v}$, on obtient

$$\begin{aligned}(h * \Phi_{u,v})[m,n] &= \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} h[a,b]\,\Phi_{u,v}[(m-a) \bmod N, (n-b) \bmod N] \\ &= \left(\sum_{a=0}^{N-1}\sum_{b=0}^{N-1} h[a,b]e^{-2\pi i(ua+vb)/N}\right)\Phi_{u,v}[m,n]. \end{aligned} \tag{3.17}$$

On définit donc

$$\lambda_{u,v} = \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} h[a,b]e^{-2\pi i(ua+vb)/N}. \tag{3.18}$$

On obtient finalement

$$h * \Phi_{u,v} = \lambda_{u,v}\Phi_{u,v}. \tag{3.19}$$

3.6 Représentation matricielle d'une convolution 2D

Nous utilisons maintenant la vectorisation par colonnes définie en (3.8). Comme la convolution circulaire est linéaire, il existe une matrice

$$M \in \mathbb{C}^{N^2 \times N^2}$$

telle que

$$\operatorname{vec}(Y) = M \operatorname{vec}(X). \tag{3.20}$$

Autrement dit, le coefficient de M situé à la ligne $m+Nn$ et à la colonne $r+Ns$ indique comment le pixel $X[r,s]$ contribue au pixel $Y[m,n]$.

La matrice M en question applique le filtre h sur chaque pixel de notre image X en une seule opération. À cette fin, la matrice M possède une structure bloc-circulante à blocs circulants (BCCB) comme illustré ci-dessous :

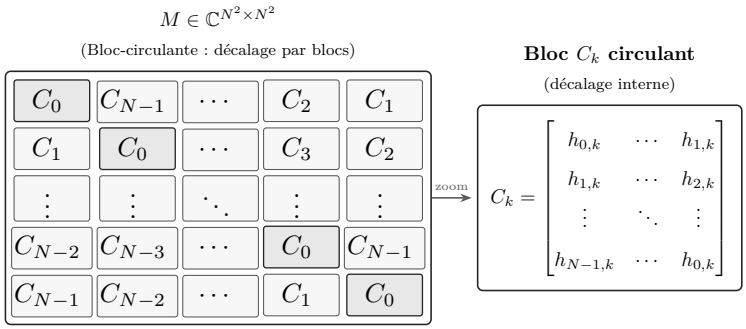


FIGURE 3.3 – Structure compacte d'une matrice BCCB (doublement circulante).

On dit alors que M est une matrice **BCCB** (*Block-Circulant with Circulant Blocks* [7]). Le détail est donné en annexe C.2.

Le résultat spectral obtenu pour la convolution circulaire 2D s'écrit alors, sous forme vectorisée,

$$M \operatorname{vec}(\Phi_{u,v}) = \lambda_{u,v} \operatorname{vec}(\Phi_{u,v}).$$

3.6.1 Diagonalisation par le produit de Kronecker des bases de Fourier

D'après le résultat précédent 3.19 comme $q_{u,v} = \operatorname{vec}(\Phi_{u,v})$, on obtient

$$M q_{u,v} = \lambda_{u,v} q_{u,v}.$$

l'égalité matricielle pour toutes les fréquences devient

$$\begin{aligned} & MQ_N = Q_N \Lambda_h \\ \iff & \text{(unitarité de } Q_N\text{)} \quad M = Q_N \Lambda_h Q_N^H. \end{aligned} \tag{3.21}$$

où Λ_h est la matrice diagonale contenant les valeurs propres $\lambda_{u,v}$ dans le même ordre que les colonnes $q_{u,v}$ de Q_N.

3.7 Extension au cas multi-canaux

Jusqu'ici, nous avons considéré le cas d'une convolution mono-canal, c'est-à-dire que les images d'entrée et de sortie sont constituées d'un seul canal (par exemple, une image en niveaux de gris). Maintenant, nous allons considérer le cas plus général d'une convolution multi-canaux :

$$X \in \mathbb{C}^{m_{in} \times N \times N}, \qquad Y \in \mathbb{C}^{m_{out} \times N \times N},$$

où m_{in} et m_{out} désignent le nombre de canaux d'entrée et de sortie respectivement.

Ainsi, pour chaque canal de sortie Y_i, nous avons que :

$$Y_i = \sum_{j=1}^{m_{in}} h_{i,j} * X_j \qquad \text{pour } i = 1, \ldots, m_{out}, \tag{3.22}$$

où $h_{i,j}$ est le filtre de convolution appliqué au canal d'entrée j pour produire le canal de sortie i.

3.7.1 Mélange des canaux à fréquence fixée

Soit une fréquence spatiale (u, v) et l'image de Fourier correspondante $\Phi_{u,v}$. On suppose que chaque canal d'entrée contient cette même fréquence spatiale, mais avec une amplitude différente :

$$X_j = \alpha_j \Phi_{u,v}, \qquad j = 1, \dots, m_{\text{in}}.$$

Le vecteur

$$\alpha = (\alpha_1, \dots, \alpha_{m_{\text{in}}})^T$$

contient donc les amplitudes de cette fréquence dans les différents canaux d'entrée.

En injectant ces entrées dans (3.22) et par linéarité de la convolution, on obtient, pour le canal de sortie i,

$$\begin{aligned} Y_i &= \sum_{j=1}^{m_{\text{in}}} h_{i,j} * X_j \\ &= \sum_{j=1}^{m_{\text{in}}} \alpha_j (h_{i,j} * \Phi_{u,v}) \\ &= \sum_{j=1}^{m_{\text{in}}} \alpha_j \lambda_{u,v}^{i,j} \Phi_{u,v} \\ &= \left(\sum_{j=1}^{m_{\text{in}}} \lambda_{u,v}^{i,j} \alpha_j \right) \Phi_{u,v}. \end{aligned}$$

On introduit alors la matrice de mélange associée à la fréquence (u, v) :

$$G^{(u,v)} \in \mathbb{C}^{m_{\text{out}} \times m_{\text{in}}}, \qquad G_{i,j}^{(u,v)} = \lambda_{u,v}^{i,j}.$$

Si $\beta \in \mathbb{C}^{m_{\text{out}}}$ désigne les amplitudes de sortie de cette même fréquence, on peut écrire simplement

$$\beta = G^{(u,v)} \alpha.$$

Cette écriture résume le rôle d'une convolution multi-canaux : pour une fréquence spatiale fixée, elle effectue un mélange linéaire entre les canaux.

3.7.2 Forme matricielle globale

Pour écrire l'opération multi-canaux sous forme matricielle globale, on définit :

$$\mathbf{x} = \begin{pmatrix} \text{vec}(X_1) \\ \vdots \\ \text{vec}(X_{m_{\text{in}}}) \end{pmatrix} \in \mathbb{C}^{m_{\text{in}} N^2}, \qquad \mathbf{y} = \begin{pmatrix} \text{vec}(Y_1) \\ \vdots \\ \text{vec}(Y_{m_{\text{out}}}) \end{pmatrix} \in \mathbb{C}^{m_{\text{out}} N^2}.$$

Pour chaque paire de canaux (i, j), on note $B^{i,j}$ la matrice de convolution spatiale associée au filtre $h_{i,j}$, c'est-à-dire la matrice qui applique la convolution entre le canal d'entrée j et le

canal de sortie i. On utilise la base de Fourier 2D vectorisée Q_N définie en (3.11). Pour chaque paire de canaux (i, j), la matrice $B^{i,j}$ se diagonalise donc sous la forme

$$B^{i,j} = Q_N D^{i,j} Q_N^H,$$

où

$$D^{i,j} = \mathrm{diag}(\lambda_{u,v}^{i,j})_{(u,v)\in\{0,\dots,N-1\}^2}.$$

Les coefficients diagonaux de $D^{i,j}$ sont ordonnés selon l'indice aplati $w(u, v)$ défini en (3.12).

À présent nous pouvons introduire la matrice globale $M \in \mathbb{C}^{m_{out}N^2 \times m_{in}N^2}$ qui représente la couche de convolution complète sous la forme d'une matrice par blocs de taille $m_{out} \times m_{in}$, où chaque bloc (i, j) correspond à l'application du filtre $h_{i,j}$:

$$M = \begin{pmatrix} B^{1,1} & \dots & B^{1,m_{in}} \\ \vdots & \ddots & \vdots \\ B^{m_{out},1} & \dots & B^{m_{out},m_{in}} \end{pmatrix}.$$

Comme la même base Q_N diagonalise tous les blocs spatiaux $B^{i,j}$, on peut factoriser le changement de base spatial de part et d'autre de la matrice par blocs. On obtient alors

$$M = (I_{m_{out}} \otimes Q_N) L (I_{m_{in}} \otimes Q_N)^H \tag{3.23}$$

où $L \in \mathbb{C}^{m_{out}N^2 \times m_{in}N^2}$ est une matrice composée de blocs diagonaux :

$$L = \begin{pmatrix} D^{1,1} & \dots & D^{1,m_{in}} \\ \vdots & \ddots & \vdots \\ D^{m_{out},1} & \dots & D^{m_{out},m_{in}} \end{pmatrix}.$$

Démonstration. Par définition du produit de Kronecker, les opérations $I_{m_{out}} \otimes Q_N$ et $(I_{m_{in}} \otimes Q_N)^H = I_{m_{in}} \otimes Q_N^H$ génèrent des matrices diagonales par blocs :

$$I_{m_{out}} \otimes Q_N = \begin{pmatrix} Q_N & 0 & \cdots & 0 \\ 0 & Q_N & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_N \end{pmatrix}, \quad I_{m_{in}} \otimes Q_N^H = \begin{pmatrix} Q_N^H & 0 & \cdots & 0 \\ 0 & Q_N^H & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_N^H \end{pmatrix}.$$

En calculant le produit matriciel par blocs à gauche, Q_N se distribue sur chaque bloc des lignes correspondantes de L :

$$(I_{m_{out}} \otimes Q_N)L = \begin{pmatrix} Q_N D^{1,1} & \cdots & Q_N D^{1,m_{in}} \\ \vdots & \ddots & \vdots \\ Q_N D^{m_{out},1} & \cdots & Q_N D^{m_{out},m_{in}} \end{pmatrix}.$$

En multipliant ensuite ce résultat à droite par la seconde matrice diagonale par blocs, Q_N^H se distribue sur chaque bloc des colonnes correspondantes :

$$(I_{m_{out}} \otimes Q_N)L(I_{m_{in}} \otimes Q_N)^H = \begin{pmatrix} Q_N D^{1,1} Q_N^H & \cdots & Q_N D^{1,m_{in}} Q_N^H \\ \vdots & \ddots & \vdots \\ Q_N D^{m_{out},1} Q_N^H & \cdots & Q_N D^{m_{out},m_{in}} Q_N^H \end{pmatrix}.$$

Puisque la décomposition de la convolution mono-canal donne $B^{i,j} = Q_N D^{i,j} Q_N^H$ pour tout couple (i, j), on obtient :

$$(I_{m_{out}} \otimes Q_N)L(I_{m_{in}} \otimes Q_N)^H = \begin{pmatrix} B^{1,1} & \cdots & B^{1,m_{in}} \\ \vdots & \ddots & \vdots \\ B^{m_{out},1} & \cdots & B^{m_{out},m_{in}} \end{pmatrix} = M.$$

□

On retrouve ainsi la forme de la **Proposition 1** de l'article de Tsuzuku et Sato [14]. Elle sépare deux phénomènes :

— les matrices $(I \otimes Q_N)$ réalisent le changement de base spatial vers le domaine de Fourier ;

— la matrice centrale L contient les mélanges entre canaux, organisés fréquence par fréquence.

3.8 Empilement linéaire de convolutions

Nous allons maintenant introduire la notion d'empilement de couches de convolution. L'idée est de modéliser mathématiquement le fonctionnement d'un réseau de neurones convolutifs profond, qui est composé de plusieurs couches appliquées successivement.

Comme nous l'avons vu dans les sections précédentes, notre cadre linéaire nous permet de représenter chaque couche de convolution de stride 1 (stride = pas voir C.2 pour une illustration) par une grande matrice globale $M^{(k)}$. On note m_k le nombre de canaux en entrée de la couche k, et m_{k+1} le nombre de canaux en sortie :

$$M^{(k)} : \mathbb{C}^{m_k N^2} \longrightarrow \mathbb{C}^{m_{k+1} N^2}.$$

Ainsi, pour une entrée vectorisée $X \in \mathbb{C}^{m_1 N^2}$, la sortie de la première couche s'écrit :

$$X_1 = M^{(1)} X \in \mathbb{C}^{m_2 N^2}.$$

En appliquant une seconde couche de convolution, on obtient :

$$X_2 = M^{(2)}X_1 = M^{(2)}(M^{(1)}X) = (M^{(2)}M^{(1)})X.$$

En continuant ainsi pour d couches de convolution successives :

$$M(X) = \left(M^{(d)}M^{(d-1)}\cdots M^{(1)}\right)X. \tag{3.24}$$

3.8.1 Simplification par télescopage des transformations

L'intérêt majeur de la décomposition matricielle de l'équation (3.23) réside dans son comportement lors du produit de plusieurs couches. Considérons deux couches successives $M^{(1)}$ et $M^{(2)}$. Leur produit s'écrit :

$$\begin{aligned} M^{(2)}M^{(1)} &= \left[(I_{m_3}\otimes Q_N)L^{(2)}(I_{m_2}\otimes Q_N)^H\right]\left[(I_{m_2}\otimes Q_N)L^{(1)}(I_{m_1}\otimes Q_N)^H\right] \\ &= (I_{m_3}\otimes Q_N)L^{(2)}\underbrace{\left((I_{m_2}\otimes Q_N)^H(I_{m_2}\otimes Q_N)\right)}_{I_{m_2N^2}}L^{(1)}(I_{m_1}\otimes Q_N)^H \end{aligned}$$

Par récurrence, pour un réseau de d couches, la matrice globale M se simplifie en :

$$M = (I_{m_{d+1}}\otimes Q_N)L(I_{m_1}\otimes Q_N)^H$$

où $L = L^{(d)}L^{(d-1)}\cdots L^{(1)}$ représente l'accumulation des mélanges de canaux.

3.8.2 Indépendance fréquentielle et matrice de mélange $G^{(w)}$

Comme chaque bloc interne des matrices $L^{(k)}$ est diagonal, le produit de ces matrices conserve cette structure. En effet, le bloc (i,j) du produit de deux matrices par blocs $L^{(2)}L^{(1)}$ s'écrit

$$(L^{(2)}L^{(1)})_{i,j} = \sum_k D^{(2)}_{i,k}D^{(1)}_{k,j}.$$

Or le produit de deux matrices diagonales est diagonal, et une somme de matrices diagonales reste diagonale. Par conséquent, chaque bloc de la matrice globale $L = L^{(d)}L^{(d-1)}\cdots L^{(1)}$ est encore une matrice diagonale de taille $N^2\times N^2$.

Ainsi, dans un empilement linéaire de convolutions circulaires de stride 1, la structure par fréquence est conservée. Pour une fréquence donnée, seule la répartition entre les canaux évolue.

L'idée est alors la suivante : pour étudier ce que fait le réseau à une fréquence donnée, il n'est pas nécessaire de regarder toute la grande matrice L. Il suffit de regarder, dans chaque bloc $D^{(i,j)}$.

On regroupe ces coefficients dans une petite matrice

$$G^{(w)} \in \mathbb{C}^{m_{\text{out}}\times m_{\text{in}}},$$

définie par

$$\left(G^{(w)}\right)_{i,j} = \left(D^{(i,j)}\right)_{w,w}. \tag{3.25}$$

La matrice $G^{(w)}$ décrit donc le comportement du réseau à la fréquence w. Elle ne décrit plus toute l'image, mais seulement le mélange entre les canaux pour cette fréquence précise. C'est ce qui permet de réduire l'étude d'une grande couche convolutionnelle à l'étude d'une matrice beaucoup plus petite, fréquence par fréquence.

3.8.3 Analyse de sensibilité par décomposition en valeurs singulières (SVD)

L'opérateur de convolution multi-canaux peut être représenté par une matrice

$$M \in \mathbb{C}^{m_{\text{out}}N^2 \times m_{\text{in}}N^2}.$$

Comme nous cherchons à identifier les directions d'entrée qui sont les plus amplifiées par la couche de convolution, il vient naturellement à l'esprit d'étudier les valeurs singulières de cette matrice.

La décomposition en valeurs singulières de M s'écrit

$$M = U\Sigma V^H,$$

où U et V sont des matrices unitaires, et où Σ contient les valeurs singulières

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq 0.$$

Les colonnes de V sont appelées vecteurs singuliers droits : elles correspondent aux directions dans l'espace d'entrée. Les colonnes de U sont les vecteurs singuliers gauches : elles correspondent aux directions dans l'espace de sortie.

Une valeur singulière mesure l'amplification d'une direction d'entrée. Plus précisément, si v_i est un vecteur singulier droit associé à σ_i et si $\|v_i\|_2 = 1$, alors

$$\|Mv_i\|_2 = \sigma_i.$$

Ainsi, les grandes valeurs singulières correspondent aux directions d'entrée fortement amplifiées par l'opérateur M. En particulier,

$$\sigma_1(M) = \max_{x \neq 0} \frac{\|Mx\|_2}{\|x\|_2} = \max_{\|x\|_2=1} \|Mx\|_2 = \|M\|_2.$$

La direction associée à σ_1 est donc une direction de sensibilité maximale, lorsque cette valeur singulière est simple. Si elle est multiple, toute direction unitaire dans le sous-espace singulier associé donne la même amplification.

Dans le cadre de notre analyse, les directions de forte sensibilité du réseau correspondent donc aux vecteurs singuliers droits associés aux grandes valeurs singulières. Le lien détaillé entre SVD, vecteurs propres de $M^H M$ et l'amplification est rappelé en annexe C.6.

3.8.4 Preuve de la structure du vecteur singulier de sensibilité

Démonstration. Soit σ une valeur singulière de la matrice de mélange $G^{(w)} \in \mathbb{C}^{m_{\text{out}} \times m_{\text{in}}}$. On note

$$a \in \mathbb{C}^{m_{\text{in}}}, \qquad b \in \mathbb{C}^{m_{\text{out}}}$$

les vecteurs singuliers droit et gauche associés. On a donc

$$G^{(w)}a = \sigma b, \qquad G^{(w)H}b = \sigma a.$$

Dans l'espace complet du réseau, le vecteur singulier droit associé à cette fréquence w s'écrit

$$v = (I_{m_1} \otimes Q_N)(a \otimes e_w),$$

où e_w est le vecteur de la base canonique, égal à 1 à l'indice w et à 0 ailleurs.

On vérifie d'abord cette affirmation en appliquant l'opérateur M à ce candidat v :

$$\begin{aligned} Mv &= \left[(I_{m_{d+1}} \otimes Q_N)L(I_{m_1} \otimes Q_N)^H\right]\left[(I_{m_1} \otimes Q_N)(a \otimes e_w)\right] \\ &= (I_{m_{d+1}} \otimes Q_N)L \underbrace{\left[(I_{m_1} \otimes Q_N)^H(I_{m_1} \otimes Q_N)\right]}_{I_{m_1 N^2}}(a \otimes e_w) \\ Mv &= (I_{m_{d+1}} \otimes Q_N)L(a \otimes e_w) \end{aligned}$$

Analysons maintenant l'action de L sur le vecteur $(a \otimes e_w)$. Rappelons que L est une matrice par blocs où chaque bloc (i, j) est la matrice diagonale $D^{(i,j)}$. Le vecteur $(a \otimes e_w)$ peut être vu comme un vecteur par blocs dont le j-ième bloc est le vecteur $a_j e_w$. Ainsi, le i-ième bloc du vecteur résultant $y' = L(a \otimes e_w)$ se calcule par :

$$y'_i = \sum_{j=1}^{m_1} D^{(i,j)}(a_j e_w)$$

Puisque $D^{(i,j)}$ est une matrice diagonale, sa multiplication par le vecteur de la base canonique e_w isole simplement son w-ième élément diagonal, multiplié par e_w. Autrement dit,

$$D^{(i,j)}e_w = (D^{(i,j)})_{w,w}\, e_w$$

Or, par définition de notre matrice de mélange fréquentiel, ce coefficient $(D^{(i,j)})_{w,w}$ est exactement l'élément (i, j) de la matrice $G^{(w)}$. Nous obtenons donc directement :

$$y'_i = \sum_{j=1}^{m_1} (G^{(w)})_{i,j}\, a_j\, e_w = \left(\sum_{j=1}^{m_1} (G^{(w)})_{i,j} a_j\right) e_w$$

Le terme entre parenthèses n'est autre que la i-ième composante du produit matriciel $G^{(w)}a$. Sachant que $G^{(w)}a = \sigma b$, il vient :

$$y'_i = (\sigma b)_i e_w = \sigma b_i e_w$$

En réassemblant les blocs, nous déduisons que

$$L(a \otimes e_w) = \sigma(b \otimes e_w). \tag{3.26}$$

Nous pouvons maintenant réinjecter ce résultat dans notre équation :

$$\begin{aligned} Mv &= (I_{m_{d+1}} \otimes Q_N)\,(\sigma(b \otimes e_w)) \\ &= \sigma\left(I_{m_{d+1}} b \otimes Q_N e_w\right) \\ Mv &= \sigma\left(b \otimes (Q_N)_w\right) \end{aligned}$$

Cette équation montre que l'application de M à v produit bien un vecteur de la forme

$$Mv = \sigma u, \qquad u = b \otimes (Q_N)_w.$$

Il reste à vérifier l'équation adjointe. Comme

$$u = (I_{m_{d+1}} \otimes Q_N)(b \otimes e_w),$$

on obtient de la même manière

$$M^H u = (I_{m_1} \otimes Q_N) L^H (b \otimes e_w).$$

Or la relation $G^{(w)H} b = \sigma a$ donne

$$L^H(b \otimes e_w) = \sigma(a \otimes e_w).$$

Ainsi,

$$M^H u = \sigma(a \otimes (Q_N)_w) = \sigma v.$$

Les deux relations de la SVD sont donc vérifiées :

$$Mv = \sigma u, \qquad M^H u = \sigma v.$$

□

3.8.5 Vérification de l'orthonormalité :

Démonstration. Pour que l'ensemble des vecteurs $v = a \otimes (Q_N)_w$ forme rigoureusement la matrice unitaire V de la SVD de M, il doit constituer une base orthonormée. Cela se démontre grâce aux propriétés du produit scalaire mixte ($(A \otimes B)^H (C \otimes D) = (A^H C) \otimes (B^H D)$) :

- **Norme unitaire :** Comme a et $(Q_N)_w$ sont unitaires, $v^H v = (a^H a) \otimes ((Q_N)_w^H (Q_N)_w) = 1 \times 1 = 1$.
- **Orthogonalité intra-fréquentielle :** Pour une même fréquence w, l'orthogonalité native de deux vecteurs droits a et $\tilde{a}$ est préservée : $v^H \tilde{v} = (a^H \tilde{a}) \otimes 1 = 0$.

— **Orthogonalité inter-fréquentielle :** Pour $w \neq \tilde{w}$, les colonnes de Fourier sont orthogonales : $v^H \tilde{v} = (a^H \tilde{a}) \otimes ((Q_N)_w^H (Q_N)_{\tilde{w}}) = 0$.

□

3.8.6 Expression finale du motif de Fourier

Enfin, comme $Q_N e_w = (Q_N)_w$, l'expression du vecteur singulier devient

$$v = a \otimes (Q_N)_w.$$

Si l'indice w correspond au couple de fréquences (u, v), alors, avec la notation $q_{u,v}$ fixée dans la section 3.2, on peut écrire

$$v = a \otimes q_{u,v} = a \otimes \big((F_N)_v \otimes (F_N)_u\big). \tag{3.27}$$

Autrement dit, le vecteur singulier est composé de deux parties : un vecteur a, qui décrit le mélange entre les canaux, et une direction de Fourier $q_{u,v}$, qui décrit le motif spatial.

3.9 Skip connections et couches de normalisation

Les résultats précédents ont été établis dans un cadre linéaire simplifié : les couches sont des convolutions de stride 1, sans activation non linéaire et avec un *padding* circulaire. Dans ce cadre, nous avons montré que les directions de sensibilité maximale peuvent s'écrire sous la forme d'un vecteur de canaux multiplié par une direction de Fourier spatiale, comme dans (3.27).

Il reste cependant à vérifier que cette conclusion ne disparaît pas dès que l'on ajoute deux mécanismes courants des architectures convolutionnelles modernes : les *skip connections* et les couches de normalisation. C'est l'objet des Propositions 3 et 4 de Tsuzuku et Sato [14].

3.9.1 Qu'est-ce qu'une *skip connection* ?

Les architectures convolutionnelles modernes, comme les réseaux résiduels (*ResNet*) [4], utilisent fréquemment des *skip connections*. Le principe consiste à ajouter directement l'entrée d'un bloc à sa sortie, de sorte que le bloc n'apprenne plus une transformation complète, mais une correction de l'identité.

Si l'on note $\mathcal{F}$ la transformation réalisée par le bloc, la sortie s'écrit

$$x_{\ell+1} = \mathcal{F}(x_\ell) + x_\ell. \tag{3.28}$$

Dans le cas simple où l'entrée et la sortie ont la même dimension, la branche de raccourci est donc l'identité. Dans le cadre de notre analyse linéaire, si l'on modélise le bloc principal par M, alors le bloc résiduel entier est représenté par

$$M_{\text{skip}} = M + I, \tag{3.29}$$

où I désigne l'identité.

Cette écriture est particulièrement utile pour notre étude spectrale car elle montre que l'ajout d'une *skip connection* ne change pas la nature de l'opérateur de base, mais ajoute simplement une contribution identité au bloc convolutionnel.

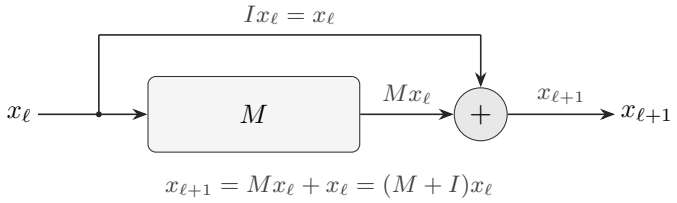


FIGURE 3.4 – Interprétation linéaire d'une *skip connection*. Le chemin principal applique l'opérateur M, tandis que la branche de raccourci applique l'identité. Le bloc résiduel est donc représenté par l'opérateur $M + I$.

3.9.2 Cas d'une skip connection

Démonstration. Analysons d'abord le cas d'une connexion résiduelle "idéale", c'est-à-dire lorsque l'entrée X et la sortie du bloc convolutionnel $M(X)$ (3.24) possèdent exactement les mêmes dimensions (même résolution spatiale et même nombre de canaux). La transformation globale s'écrit alors tout simplement :

$$M_{\text{skip}}(X) = (M^{(d)} M^{(d-1)} \dots M^{(1)})(X) + X.$$

Sous forme matricielle, cette addition équivaut à ajouter la matrice identité à notre opérateur de base, ce qui nous ramène directement à l'équation (3.29).

En pratique, lorsque la dimension des canaux change, l'addition directe devient impossible et la branche de raccourci utilise alors une projection. Si cette projection est elle-même convolutionnelle, elle entre dans le même cadre spectral que les couches étudiées précédemment. Le raisonnement reste donc identique en remplaçant l'identité par cette projection.

D'après la factorisation obtenue précédemment à l'équation (3.23), l'opérateur M peut s'écrire

$$M = (I_m \otimes Q_N) L (I_m \otimes Q_N)^H,$$

Comme Q_N est unitaire, on a aussi

$$I_{mN^2} = (I_m \otimes Q_N) I_{mN^2} (I_m \otimes Q_N)^H.$$

Ainsi,

$$\begin{aligned} M_{\text{skip}} &= M + I_{mN^2} \\ &= (I_m \otimes Q_N) L (I_m \otimes Q_N)^H + (I_m \otimes Q_N) I_{mN^2} (I_m \otimes Q_N)^H \\ &= (I_m \otimes Q_N)(L + I_{mN^2})(I_m \otimes Q_N)^H. \end{aligned}$$

La skip connection ne change donc pas la base spatiale de diagonalisation. Elle modifie uniquement la matrice centrale L en lui ajoutant l'identité.

Pour rendre cette observation plus explicite, fixons une fréquence spatiale w, correspondant à un couple de fréquences (u, v). Comme expliqué dans la section 3.8, l'action du réseau sur cette fréquence est décrite par une matrice de mélange définie en (3.25)

$$G^{(w)} \in \mathbb{C}^{m \times m}.$$

L'ajout de l'identité ne modifie pas la fréquence considérée : il ajoute simplement une contribution directe sur le même canal et la même fréquence. La matrice de mélange devient donc

$$G^{(w)}_{\text{skip}} = G^{(w)} + I_m.$$

À fréquence fixée, la skip connection ajoute donc simplement une contribution directe dans la matrice de mélange.

On peut alors appliquer exactement le même raisonnement que dans la section 3.8. En particulier, si a est un vecteur singulier droit de $G^{(w)}_{\text{skip}}$, alors le vecteur singulier droit correspondant dans l'espace complet reste de la forme

$$v = a \otimes (Q_N)_w,$$

c'est-à-dire,

$$v = a \otimes q_{u,v} = a \otimes \big((F_N)_v \otimes (F_N)_u\big),$$

Une vérification algébrique détaillée de cette affirmation est donnée en annexe C.3.

La skip connection conserve donc la forme du raisonnement précédent : pour chaque fréquence w, elle remplace simplement $G^{(w)}$ par $G^{(w)} + I_m$. Les directions singulières restent de la forme

$$a \otimes q_{u,v}.$$

Cela correspond à la Proposition 3 de Tsuzuku et Sato [14].

□

3.9.3 Cas des couches de normalisation

Démonstration. Les architectures convolutionnelles modernes utilisent aussi des couches de normalisation, notamment la *batch normalization* introduite par Ioffe et Szegedy [6]. À première vue, ces couches pourraient sembler sortir du cadre précédent, car elles ne sont pas des convolutions. Cependant, lors de la phase de test (aussi appelée *inference*), les statistiques utilisées par la batch normalization sont fixées : la moyenne et la variance ne sont plus calculées à partir du mini-batch courant, mais proviennent des statistiques apprises pendant l'entraînement. Cette stabilisation transforme une opération complexe en une transformation affine statique.

Pour un canal de sortie i, elle applique une normalisation suivie d'un changement d'échelle et d'un décalage appris. À l'inférence, les statistiques utilisées ne sont plus celles du mini-batch

courant, mais des estimations fixées pendant l'entraînement. On obtient alors

$$\mathrm{BN}_i(Y_i) = \gamma_i \frac{Y_i - \mu_i}{\sqrt{\sigma_i^2 + \varepsilon}} + \beta_i,$$

où μ_i et σ_i^2 sont les statistiques fixées du canal i, tandis que γ_i et β_i sont des paramètres appris.

Comme μ_i, σ_i^2, γ_i et β_i sont constants à l'inférence, cette transformation est affine. En posant

$$s_i = \frac{\gamma_i}{\sqrt{\sigma_i^2 + \varepsilon}}, \qquad t_i = \beta_i - s_i \mu_i,$$

on obtient

$$\mathrm{BN}_i(Y_i) = s_i Y_i + t_i.$$

Supposons d'abord que la convolution soit écrite sans biais, conformément au cadre linéaire utilisé jusqu'ici en (3.22). À l'inférence, la batch normalization agit alors canal par canal comme une transformation affine, et l'on obtient

$$\mathrm{BN}_i(Y_i) = s_i \sum_{j=1}^{m_{\text{in}}} h_{i,j} * X_j + t_i = \sum_{j=1}^{m_{\text{in}}} (s_i h_{i,j}) * X_j + t_i.$$

La partie linéaire est donc encore une convolution, avec des filtres modifiés

$$h'_{i,j} = s_i h_{i,j}.$$

Le terme constant t_i est un biais affine ajouté au canal de sortie. Comme l'analyse spectrale porte sur la partie linéaire de l'opérateur, ce terme constant ne modifie pas les directions singulières étudiées.

Ainsi, une convolution suivie d'une batch normalization à l'inférence peut être représentée comme une nouvelle convolution, avec des poids et des biais modifiés.

Il faut toutefois noter que cette conclusion concerne le mode inférence. Pendant l'entraînement, la batch normalization dépend des statistiques du mini-batch courant. L'opérateur n'est alors plus strictement le même pour chaque entrée individuelle. On retrouve bien le résultat de la Proposition 4 de Tsuzuku et Sato [14].

□

3.9.4 Conclusion

Les Propositions 3 et 4 montrent que les skip connections et les normalisations à l'inférence s'intègrent encore dans le cadre linéaire étudié ici. Elles modifient les matrices de mélange ou les paramètres effectifs des convolutions, mais pas la forme générale des directions singulières obtenues précédemment.

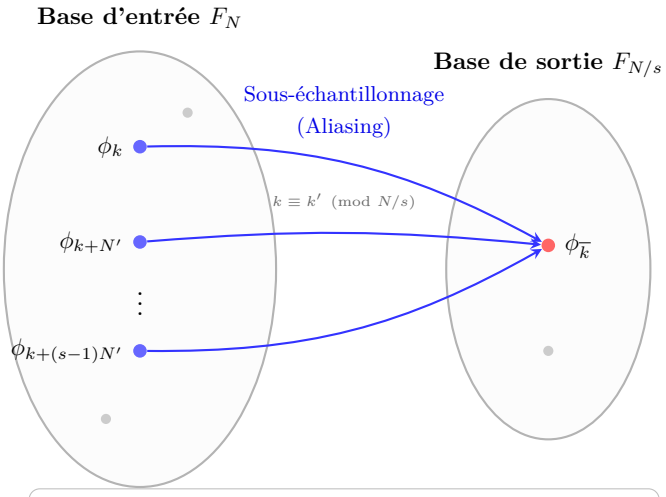
3.10 Couches de réduction : stride et pooling moyen

Cette section reprend la Proposition 5 de Tsuzuku et Sato [14]. Le but est de mieux comprendre ce qui se passe lorsqu'une couche réduit la taille spatiale de l'image. En particulier, nous allons voir pourquoi, dans ce cas, les directions singulières ne correspondent plus forcément à une seule fréquence de Fourier.

Qu'est-ce qu'un *stride* ? Avant d'aller plus loin, il convient de définir la notion de *stride*. Dans une couche de convolution standard, le filtre glisse sur l'image en se décalant d'une position à la fois : on parle alors d'un *stride* de 1.

Cependant, on peut choisir d'imposer au filtre de faire des sauts plus grands, en se décalant de s positions à chaque étape, où s est un entier strictement positif. Ce paramètre s est le *stride*. L'effet immédiat d'un $s > 1$ est de réduire la résolution spatiale de la sortie, puisque l'on ne calcule plus une valeur en chaque position, mais seulement une valeur toutes les s positions comme illustre la figure C.2.

Du point de vue fréquentiel, cette réduction de résolution a une conséquence essentielle : elle rend certaines fréquences indiscernables. En effet, lorsque l'on ne conserve qu'une position sur s, plusieurs composantes de Fourier distinctes dans l'espace d'entrée peuvent produire la même composante dans l'espace de sortie. Autrement dit, le *stride* introduit un phénomène de repliement spectral (*aliasing* [2]) comme l'illustre la figure 3.5 : des fréquences différentes avant réduction vont renvoyer vers la même fréquence après réduction. C'est précisément ce mécanisme qui jouera un rôle central dans l'analyse fréquentielle des couches de réduction.



Phénomène d'aliasing : Plusieurs ondes de fréquences différentes dans la base large F_N (espacées de $N' = N/s$) sont projetées sur une seule et même onde dans la base réduite $F_{N/s}$. Elles deviennent alors indiscernables après l'opération de réduction.

FIGURE 3.5 – Représentation ensembliste du repliement de spectre généralisé. Les flèches illustrent la perte d'injectivité : un ensemble de s fréquences sources contribue à une unique fréquence de sortie.

Pourquoi les résultats précédents ne s'appliquent plus directement Les résultats précédents reposaient sur le fait qu'une convolution circulaire de stride 1 préserve la taille spatiale et agit séparément sur chaque fréquence de Fourier. Cette séparation permet d'obtenir des vecteurs singuliers associés à une seule fréquence. Dès qu'une couche réduit la résolution spatiale, cette propriété disparaît : le sous-échantillonnage regroupe plusieurs fréquences d'entrée dans une même fréquence de sortie. C'est ce regroupement, ou aliasing, que nous allons maintenant formaliser.

3.10.1 Sous-échantillonnage spatial et aliasing fréquentiel

Pour simplifier l'analyse nous allons commencer par isoler l'effet du sous-échantillonnage, indépendamment de toute convolution. On suppose que s divise N, et on note

$$N' = \frac{N}{s}.$$

On définit l'opérateur de sous-échantillonnage

$$P_s : \mathbb{C}^N \longrightarrow \mathbb{C}^{N'}$$

par

$$(P_s x)_r = x_{sr}, \qquad r = 0, \dots, N' - 1.$$

Autrement dit, P_s conserve une valeur toutes les s positions.

Soit maintenant $(F_N)_k$ une direction de Fourier 1D de taille N. En appliquant P_s à cette direction, on obtient, pour tout $r = 0, \dots, N' - 1$,

$$\begin{aligned}
\big(P_s(F_N)_k\big)_r &= ((F_N)_k)_{sr} \\
&= \frac{1}{\sqrt{N}} \exp\left(2\pi i \frac{ksr}{N}\right) \\
&= \frac{1}{\sqrt{N}} \exp\left(2\pi i \frac{kr}{N'}\right).
\end{aligned}$$

Or l'exponentielle précédente dépend de k modulo N'. En notant

$$\overline{k} = k \bmod N',$$

et comme $N = sN'$, donc

$$\frac{1}{\sqrt{N}} = \frac{1}{\sqrt{s}} \frac{1}{\sqrt{N'}},$$

on obtient

$$P_s(F_N)_k = \frac{1}{\sqrt{s}} (F_{N'})_{\overline{k}}. \tag{3.30}$$

Ce qui signifie qu'après sous-échantillonnage, la fréquence k de taille N devient la fréquence $\overline{k}$ de taille N'. En effet, cette équation exprime précisément le phénomène d'*aliasing*. Deux fréquences k et k' telles que

$$k \equiv k' \pmod{N'}$$

deviennent identiques après sous-échantillonnage. Ainsi, pour chaque fréquence $\overline{k} \in \{0, \dots, N' - 1\}$, il existe un ensemble de s fréquences sources dans le signal original,

$$\mathcal{A}_k = \left\{ \overline{k} + qN' \mid q \in \{0, \dots, s-1\} \right\},$$

qui deviennent indistinguables après sous-échantillonnage !

Autrement dit, l'opérateur de sous-échantillonnage n'est pas **injectif** dans le domaine fréquentiel : s fréquences distinctes de l'espace de départ, espacées de N', se projettent sur une unique fréquence dans l'espace d'arrivée.

Extension au cas bidimensionnel. On étend maintenant ce raisonnement aux images de taille $N \times N$. On définit l'opérateur de sous-échantillonnage spatial 2D

$$P_s^{(2D)} : \mathbb{C}^{N \times N} \longrightarrow \mathbb{C}^{N' \times N'}$$

par

$$\left(P_s^{(2D)} X\right)_{r,t} = X_{sr,st}, \qquad r, t = 0, \dots, N' - 1. \tag{3.31}$$

Il conserve donc une ligne sur s et une colonne sur s.

Après vectorisation, on note Π_s la matrice associée à l'opérateur $P_s^{(2D)}$. Pour la direction de Fourier vectorisée

$$q_{u,v} = (F_N)_v \otimes (F_N)_u,$$

on peut appliquer le résultat 1D sur chacune des deux dimensions. En utilisant le résultat 1D obtenu en (3.30) sur chacune des deux dimensions, on a

$$\begin{aligned} \Pi_s q_{u,v} &= \Pi_s\big((F_N)_v \otimes (F_N)_u\big) \\ &= \big(P_s(F_N)_v\big) \otimes \big(P_s(F_N)_u\big) \\ &= \left(\frac{1}{\sqrt{s}}(F_{N'})_{\overline{v}}\right) \otimes \left(\frac{1}{\sqrt{s}}(F_{N'})_{\overline{u}}\right) \\ &= \frac{1}{s}(F_{N'})_{\overline{v}} \otimes (F_{N'})_{\overline{u}}. \end{aligned} \tag{3.32}$$

où

$$\overline{u} = u \bmod N', \qquad \overline{v} = v \bmod N'.$$

Ainsi, deux fréquences 2D (u, v) et (u', v') deviennent indistinguables après sous-échantillonnage dès lors que

$$u \equiv u' \pmod{N'}, \qquad v \equiv v' \pmod{N'}.$$

Pour chaque fréquence de sortie $(\overline{u}, \overline{v})$, l'ensemble des fréquences d'entrée qui se projettent sur elle est donc

$$\mathcal{A}_{\overline{u},\overline{v}} = \left\{ (\overline{u} + q_1 N',\, \overline{v} + q_2 N') \mid q_1, q_2 \in \{0, \dots, s-1\} \right\}. \tag{3.33}$$

3.10.2 Convolution avec stride et repliement spectral

Revenons maintenant au cas d'une couche convolutionnelle. Une convolution avec stride s peut être vue comme une convolution classique de stride 1, suivie d'un sous-échantillonnage spatial non commutatif. Après vectorisation, si M_h désigne l'opérateur associé à une convolution circulaire de stride 1 par un noyau h, et si Π_s désigne la matrice de sous-échantillonnage associée à $P_s^{(2D)}$, alors l'opérateur de convolution avec stride s s'écrit

$$M_h^{(s)} = \Pi_s M_h. \tag{3.34}$$

Cette écriture signifie que l'on calcule d'abord toutes les valeurs de la convolution, puis que l'on ne conserve que les positions dont les deux indices spatiaux sont des multiples de s.

Comme M_h est une convolution circulaire de stride 1, cette direction est une direction propre de l'opérateur. En utilisant la diagonalisation rappelée en (3.21), on a

$$M_h q_{u,v} = \lambda_{u,v} q_{u,v},$$

où $\lambda_{u,v}$ est la valeur propre associée à la fréquence (u, v).

En appliquant ensuite le sous-échantillonnage, et en utilisant (3.32), on obtient

$$\begin{aligned} M_h^{(s)} q_{u,v} &= \Pi_s M_h q_{u,v} \\ &= \lambda_{u,v} \Pi_s q_{u,v} \\ &= \frac{\lambda_{u,v}}{s} q_{\overline{u},\overline{v}}^{(N')}, \end{aligned} \tag{3.35}$$

où

$$q_{\overline{u},\overline{v}}^{(N')} = (F_{N'})_{\overline{v}} \otimes (F_{N'})_{\overline{u}}.$$

avec

$$\overline{u} = u \bmod N', \qquad \overline{v} = v \bmod N'.$$

L'équation (3.35) montre que la convolution avec stride conserve une partie de la structure de Fourier, mais avec une différence essentielle : plusieurs fréquences d'entrée peuvent être envoyées vers la même fréquence de sortie. Plus précisément, toutes les fréquences appartenant à la même classe d'aliasing

$$\mathcal{A}_{\overline{u},\overline{v}} = \{(\overline{u} + q_1 N', \overline{v} + q_2 N') \mid q_1, q_2 \in \{0, \dots, s-1\}\}$$

se retrouvent associées à la même fréquence de sortie $(\overline{u}, \overline{v})$.

Pour exprimer ce regroupement de manière matricielle, on introduit la matrice

$$\mathcal{S}_s \in \mathbb{C}^{N'^2 \times N^2},$$

qui décrit le repliement spectral dans la base de Fourier. Ses colonnes correspondent aux fré-

quences d'entrée (u, v), et ses lignes aux fréquences de sortie (α, β). On définit

$$(\mathcal{S}_s)_{(\alpha,\beta),(u,v)} = \begin{cases} \dfrac{1}{s}, & \text{si } \alpha = u \bmod N' \text{ et } \beta = v \bmod N', \\ 0, & \text{sinon.} \end{cases} \tag{3.36}$$

Ainsi, pour une fréquence d'entrée (u, v) donnée, la matrice $\mathcal{S}_s$ place un unique coefficient non nul sur la ligne correspondant à la fréquence de sortie

$$(u \bmod N',\, v \bmod N').$$

Ce coefficient vaut $\frac{1}{s}$, comme dans (3.32).

L'action du sous-échantillonnage sur toute la base de Fourier s'écrit alors

$$\Pi_s Q_N = Q_{N'} \mathcal{S}_s. \tag{3.37}$$

Comme Q_N est unitaire, on obtient la factorisation

$$\Pi_s = Q_{N'} \mathcal{S}_s Q_N^H. \tag{3.38}$$

Cette écriture confirme que $\mathcal{S}_s$ représente bien l'aliasing dans le domaine fréquentiel. En effet, si

$$\widehat{x} = Q_N^H x$$

désigne les coefficients de Fourier de l'entrée, alors les coefficients de Fourier de la sortie sous-échantillonnée sont donnés par

$$\widehat{y} = Q_{N'}^H \Pi_s x = \mathcal{S}_s \widehat{x}.$$

Par définition de $\mathcal{S}_s$, cela donne, pour chaque fréquence de sortie (α, β),

$$\widehat{y}_{\alpha,\beta} = \frac{1}{s} \sum_{(u,v) \in \mathcal{A}_{\alpha,\beta}} \widehat{x}_{u,v}. \tag{3.39}$$

Autrement dit, le sous-échantillonnage ne conserve pas séparément les coefficients de Fourier : il additionne les coefficients appartenant à une même classe d'aliasing. C'est ce mécanisme qui explique pourquoi, dans une couche avec stride, les directions singulières ne sont plus nécessairement associées à une seule fréquence de Fourier.

3.10.3 Composition avec une convolution de stride 1

On peut maintenant combiner le sous-échantillonnage avec une convolution circulaire de stride 1. On considère une couche convolutionnelle multi-canaux de stride 1, représentée par l'opérateur matriciel M. D'après la factorisation établie en (3.23), on a

$$M = (I_{m_{\text{out}}} \otimes Q_N) L (I_{m_{\text{in}}} \otimes Q_N)^H,$$

Une convolution avec stride s s'obtient alors en appliquant d'abord la convolution de stride

1, puis en gardant seulement une position sur s dans chaque direction spatiale. Comme cette réduction est faite séparément sur chaque canal de sortie, on introduit

$$\Pi_s^{\text{out}} = I_{m_{\text{out}}} \otimes \Pi_s.$$

L'opérateur correspondant à la convolution avec stride s s'écrit alors

$$M^{(s)} = \Pi_s^{\text{out}} M.$$

En remplaçant M par sa factorisation de Fourier et Π_s par la factorisation obtenue en (3.38), on obtient

$$\begin{aligned} M^{(s)} &= (I_{m_{\text{out}}} \otimes \Pi_s)(I_{m_{\text{out}}} \otimes Q_N) L (I_{m_{\text{in}}} \otimes Q_N)^H \\ &= (I_{m_{\text{out}}} \otimes Q_{N'})(I_{m_{\text{out}}} \otimes \mathcal{S}_s) L (I_{m_{\text{in}}} \otimes Q_N)^H. \end{aligned}$$

Cette écriture montre clairement où apparaît la différence avec le cas stride 1. Le facteur L continue d'agir fréquence par fréquence. En revanche, le facteur $\mathcal{S}_s$ regroupe ensuite plusieurs fréquences d'entrée dans une même fréquence de sortie.

Pour mieux voir ce regroupement, on reprend le même raisonnement que plus haut. Soit

$$\widehat{x}_{u,v} \in \mathbb{C}^{m_{\text{in}}} \qquad \text{et} \qquad \widehat{z}_{u,v} = G^{(u,v)} \widehat{x}_{u,v}.$$

On applique ensuite le sous-échantillonnage pour une fréquence réduite (α, β),

$$\begin{aligned} \widehat{y}_{\alpha,\beta} &= \frac{1}{s} \sum_{(u,v) \in \mathcal{A}_{\alpha,\beta}} \widehat{z}_{u,v} \\ &= \frac{1}{s} \sum_{(u,v) \in \mathcal{A}_{\alpha,\beta}} G^{(u,v)} \widehat{x}_{u,v}. \end{aligned} \tag{3.40}$$

Pour écrire cette relation sous forme matricielle, on fixe un ordre arbitraire sur les fréquences de la classe d'aliasing :

$$\mathcal{A}_{\alpha,\beta} = \{(u_1, v_1), \dots, (u_{s^2}, v_{s^2})\}.$$

On regroupe ensuite les coefficients d'entrée correspondants dans un seul vecteur :

$$\widehat{x}_{\mathcal{A}_{\alpha,\beta}} = \begin{pmatrix} \widehat{x}_{u_1,v_1} \\ \vdots \\ \widehat{x}_{u_{s^2},v_{s^2}} \end{pmatrix} \in \mathbb{C}^{m_{\text{in}} s^2}.$$

On obtient alors la matrice

$$G_s^{(\alpha,\beta)} = \frac{1}{s} \begin{pmatrix} G^{(u_1,v_1)} & G^{(u_2,v_2)} & \dots & G^{(u_{s^2},v_{s^2})} \end{pmatrix} \in \mathbb{C}^{m_{\text{out}} \times m_{\text{in}} s^2}. \tag{3.41}$$

Cette matrice est simplement une façon compacte d'écrire la somme de (3.40). En effet,

multiplier $G_s^{(\alpha,\beta)}$ par le vecteur empilé $\widehat{x}_{\mathcal{A}_{\alpha,\beta}}$ donne exactement

$$\frac{1}{s} \sum_{(u,v)\in\mathcal{A}_{\alpha,\beta}} G^{(u,v)} \widehat{x}_{u,v}.$$

On a donc par construction

$$\widehat{y}_{\alpha,\beta} = G_s^{(\alpha,\beta)} \widehat{x}_{\mathcal{A}_{\alpha,\beta}}.$$

Cette formulation permet de ramener l'étude de la sensibilité structurelle d'une convolution avec stride s à une analyse locale, fréquence de sortie par fréquence de sortie. Pour chaque fréquence réduite (α, β), il convient alors d'étudier les valeurs et vecteurs singuliers de la matrice $G_s^{(\alpha,\beta)}$.

3.10.4 Construction des vecteurs singuliers droits

Démonstration. Pour alléger les notations, on pose

$$\varphi_{u,v}^{(N)} = (F_N)_v \otimes (F_N)_u, \qquad \varphi_{\alpha,\beta}^{(N')} = (F_{N'})_\beta \otimes (F_{N'})_\alpha.$$

On fixe une fréquence de sortie (α, β) et sa classe d'aliasing $\mathcal{A}_{\alpha,\beta}$. En choisissant un ordre arbitraire sur cette classe, on écrit

$$\mathcal{A}_{\alpha,\beta} = \{(u_1, v_1), \ldots, (u_{s^2}, v_{s^2})\}.$$

Soit

$$p = \begin{pmatrix} p_{u_1,v_1} \\ \vdots \\ p_{u_{s^2},v_{s^2}} \end{pmatrix} \in \mathbb{C}^{m_{\text{in}} s^2}, \qquad p_{u_i,v_i} \in \mathbb{C}^{m_{\text{in}}},$$

un vecteur singulier droit de $G_s^{(\alpha,\beta)}$, associé à une valeur singulière σ et à un vecteur singulier gauche $q \in \mathbb{C}^{m_{\text{out}}}$. On a donc

$$G_s^{(\alpha,\beta)} p = \sigma q, \qquad G_s^{(\alpha,\beta)H} q = \sigma p. \tag{3.42}$$

À partir des blocs $p_{u,v}$, on construit le vecteur d'entrée

$$r = \sum_{(u,v)\in\mathcal{A}_{\alpha,\beta}} p_{u,v} \otimes \varphi_{u,v}^{(N)}. \tag{3.43}$$

Ce vecteur n'est donc plus associé à une seule fréquence de Fourier : c'est une combinaison de toutes les fréquences appartenant à la même classe d'aliasing.

Montrons maintenant que r est un vecteur singulier droit de $M^{(s)}$. D'après (3.40), l'action de $M^{(s)}$ sur les coefficients de Fourier associés à la classe $\mathcal{A}_{\alpha,\beta}$ est donnée par

$$\widehat{y}_{\alpha,\beta} = \frac{1}{s} \sum_{(u,v)\in\mathcal{A}_{\alpha,\beta}} G^{(u,v)} p_{u,v}.$$

Par définition de $G_s^{(\alpha,\beta)}$, cette somme s'écrit simplement

$$\widehat{y}_{\alpha,\beta} = G_s^{(\alpha,\beta)} p.$$

En utilisant la relation de singularité (3.42), on obtient donc

$$\widehat{y}_{\alpha,\beta} = \sigma q.$$

Dans le domaine spatial, cela donne

$$M^{(s)} r = \sigma q \otimes \varphi_{\alpha,\beta}^{(N')}.$$

En posant

$$w = q \otimes \varphi_{\alpha,\beta}^{(N')},$$

on obtient

$$M^{(s)} r = \sigma w. \tag{3.44}$$

Il reste à vérifier l'équation adjointe. L'adjoint de $M^{(s)}$ renvoie une fréquence de sortie (α, β) vers toutes les fréquences d'entrée de la classe $\mathcal{A}_{\alpha,\beta}$. On obtient alors

$$M^{(s)H} w = \sum_{(u,v)\in\mathcal{A}_{\alpha,\beta}} \frac{1}{s} \left(G^{(u,v)H} q \right) \otimes \varphi_{u,v}^{(N)}.$$

Or la relation

$$G_s^{(\alpha,\beta)H} q = \sigma p$$

implique, bloc par bloc, que

$$\frac{1}{s} G^{(u,v)H} q = \sigma p_{u,v}, \qquad (u,v) \in \mathcal{A}_{\alpha,\beta}.$$

En remplaçant dans l'expression précédente, on obtient directement

$$M^{(s)H} w = \sigma r. \tag{3.45}$$

Les deux relations

$$M^{(s)} r = \sigma w, \qquad M^{(s)H} w = \sigma r$$

montrent donc que r et w forment bien une paire de vecteurs singuliers droit et gauche de $M^{(s)}$.

Enfin, la normalisation est conservée. En effet, les fréquences de $\mathcal{A}_{\alpha,\beta}$ sont distinctes, donc les directions $\varphi_{u,v}^{(N)}$ correspondantes sont orthonormées. Ainsi,

$$\|r\|_2^2 = \sum_{(u,v)\in\mathcal{A}_{\alpha,\beta}} \|p_{u,v}\|_2^2 = \|p\|_2^2.$$

Donc si p est normalisé, alors r l'est aussi. De même, si q est normalisé, alors $w = q \otimes \varphi_{\alpha,\beta}^{(N')}$ est normalisé.

On retrouve ainsi le résultat annoncé par la Proposition 5 de Tsuzuku et Sato [14] : dans une couche de réduction, les directions singulières ne sont plus forcément des directions de Fourier pures. Elles peuvent être des combinaisons de fréquences appartenant à une même classe d'aliasing.

□

3.10.5 Average pooling comme cas particulier

L'*average pooling* consiste à remplacer chaque fenêtre locale de taille $s \times s$ par la moyenne de ses pixels comme illustré dans la Figure C.3. Cette opération réduit donc la résolution spatiale, comme une convolution avec stride. La différence est qu'avant le sous-échantillonnage, on applique d'abord un filtre moyenneur fixe.

Dans le cadre circulaire utilisé ici, ce filtre peut être représenté par le noyau

$$h_{\text{avg}}[a,b] = \begin{cases} \dfrac{1}{s^2}, & \text{si } a,b \in \{0,\dots,s-1\}, \\ 0, & \text{sinon,} \end{cases}$$

où les indices spatiaux sont interprétés modulo N. La convolution par h_{avg} calcule donc une moyenne locale, puis le sous-échantillonnage conserve une position sur s dans chaque direction spatiale.

Dans le cas multi-canaux, cette opération est appliquée séparément sur chaque canal. Si m désigne le nombre de canaux et si $C_{h_{\text{avg}}} \in \mathbb{C}^{N^2 \times N^2}$ désigne la matrice de convolution circulaire associée au noyau moyenneur, alors l'opérateur d'*average pooling* s'écrit

$$M_{\text{avg}}^{(s)} = (I_m \otimes \Pi_s)(I_m \otimes C_{h_{\text{avg}}}). \tag{3.46}$$

On est donc exactement dans le cas étudié précédemment : une convolution circulaire de stride 1, suivie d'un sous-échantillonnage. Le noyau moyenneur agit comme un filtre passe-bas, ce qui atténue certaines fréquences avant le repliement spectral. Mais le mécanisme d'aliasing reste le même : plusieurs fréquences d'entrée peuvent être regroupées dans une même fréquence de sortie.

Par conséquent, les vecteurs singuliers droits d'une couche d'*average pooling* peuvent eux aussi être choisis comme des combinaisons de fréquences appartenant à une même classe d'aliasing.

3.11 Contrainte image réelle

La Proposition 6 de Tsuzuku et Sato [14] vise à résoudre le problème lié à l'utilisation des méthodes que nous venons de développer dans des cas concrets. En effet, les images manipulées par un réseau sont réelles, tandis qu'une fonction de Fourier prise isolément est en général complexe. L'idée est de montrer que la contrainte de valeur réelle d'une image peut être résolue en utilisant la symétrie hermitienne de ses coefficients de Fourier. Et pour cela nous allons introduire les outils nécessaires pour comprendre cette symétrie.

Commencons par la transformée de Fourier discrète 2D [15]. Pour une image $X \in \mathbb{C}^{N\times N}$, on définit

$$\widehat{X}_{u,v} = S(X)_{u,v} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} X[m,n]\,\exp\left(-2\pi i\frac{um+vn}{N}\right), \qquad u,v \in \{0,\dots,N-1\}.$$

L'image peut être reconstruite à partir de ses coefficients de Fourier par la transformée inverse :

$$X[m,n] = \frac{1}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \widehat{X}_{u,v}\,\exp\left(2\pi i\frac{um+vn}{N}\right). \tag{3.47}$$

Ces formules sont compatibles avec les bases de Fourier introduites précédemment, à un facteur de normalisation près. Ici, nous suivons la notation $S(X)$ utilisée dans l'article [14] afin de reproduire directement la Proposition 6.

3.11.1 Symétrie hermitienne des coefficients de Fourier

La contrainte de valeur réelle d'une image se traduit par une symétrie particulière de ses coefficients de Fourier. Plus précisément, pour une image $X \in \mathbb{C}^{N\times N}$, on a

$$X \in \mathbb{R}^{N\times N} \quad\Longleftrightarrow\quad \widehat{X}_{u,v} = \widehat{X}^{*}_{(-u) \bmod N,\,(-v) \bmod N}, \qquad u,v \in \{0,\dots,N-1\}. \tag{3.48}$$

Cette propriété est la symétrie hermitienne de la transformée de Fourier d'un signal réel. Elle est classique en traitement du signal discret [11], et correspond ici à la Proposition 6 de Tsuzuku et Sato [14].

Montrons d'abord le sens direct. Supposons que X soit réelle, c'est-à-dire que

$$X[m,n]^{*} = X[m,n] \qquad \text{pour tout } m,n.$$

Par définition de la DFT 2D, on a

$$\begin{aligned}
\widehat{X}^{*}_{u,v} &= \left(\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} X[m,n]\,\exp\left(-2\pi i\frac{um+vn}{N}\right)\right)^{*} \\
&= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} X[m,n]^{*}\,\exp\left(2\pi i\frac{um+vn}{N}\right) \\
&= \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} X[m,n]\,\exp\left(-2\pi i\frac{((-u)m+(-v)n)}{N}\right) \\
&= \widehat{X}_{(-u) \bmod N,\,(-v) \bmod N}.
\end{aligned}$$

On obtient donc bien

$$\widehat{X}_{u,v} = \widehat{X}^{*}_{(-u) \bmod N,\,(-v) \bmod N}.$$

Prouvons maintenant le sens inverse. Supposons que les coefficients de Fourier vérifient la symétrie (3.48). On veut montrer que l'image reconstruite est réelle. À partir de la transformée

de Fourier inverse (3.47), on calcule le conjugué de $X[m,n]$:

$$\begin{aligned} X[m,n]^* &= \left(\frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \widehat{X}_{u,v} \exp\left(2\pi i \frac{um+vn}{N}\right) \right)^* \\ &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \widehat{X}^*_{u,v} \exp\left(-2\pi i \frac{um+vn}{N}\right). \end{aligned}$$

Pour rappel d'après l'hypothèse de symétrie hermitienne, on a

$$\widehat{X}^*_{u,v} = \widehat{X}_{(-u) \bmod N, (-v) \bmod N}.$$

Donc

$$X[m,n]^* = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \widehat{X}_{(-u) \bmod N, (-v) \bmod N} \exp\left(-2\pi i \frac{um+vn}{N}\right).$$

En effectuant le changement d'indices

$$u' = (-u) \bmod N, \qquad v' = (-v) \bmod N,$$

on obtient

$$\begin{aligned} X[m,n]^* &= \frac{1}{N^2} \sum_{u'=0}^{N-1} \sum_{v'=0}^{N-1} \widehat{X}_{u',v'} \exp\left(2\pi i \frac{u'm+v'n}{N}\right) \\ &= X[m,n]. \end{aligned}$$

Ainsi $X[m,n]^* = X[m,n]$ pour tout m,n, donc X est réelle.

On a donc bien montré l'équivalence :

$$X \in \mathbb{R}^{N \times N} \quad \Longleftrightarrow \quad \widehat{X}_{u,v} = \widehat{X}^*_{(-u) \bmod N, (-v) \bmod N}.$$

3.11.2 Conséquence pour les perturbations de Fourier

L'équivalence précédente s'applique directement aux perturbations que l'on souhaite ajouter aux images. Comme les images manipulées par le réseau sont réelles,

$$x \in \mathbb{R}^{N \times N \times c},$$

une perturbation admissible doit elle aussi être réelle dans le domaine spatial. Ainsi, pour chaque canal, ses coefficients de Fourier doivent vérifier la symétrie hermitienne :

$$\widehat{\delta}_{(-u) \bmod N, (-v) \bmod N} = \widehat{\delta}^*_{u,v}. \tag{3.49}$$

Autrement dit, si l'on utilise une fréquence (u,v), il faut aussi utiliser sa fréquence opposée

$$((-u) \bmod N,\ (-v) \bmod N)$$

avec le coefficient conjugué. Cette condition garantit que la transformée inverse de Fourier donne une perturbation réelle.

Il existe quelques cas particuliers où une fréquence est égale à sa propre fréquence opposée modulo N. Ces fréquences vérifient

$$(u, v) = ((-u) \bmod N,\ (-v) \bmod N).$$

Dans ce cas, la condition précédente impose simplement que le coefficient de Fourier soit réel. Pour N pair, cela concerne notamment les fréquences

$$(0,0), \qquad (N/2,0), \qquad (0,N/2), \qquad (N/2,N/2).$$

En dehors de ces cas particuliers, les fréquences doivent être prises par paires conjuguées.

C'est cette idée qui est utilisée par Tsuzuku et Sato dans leur *Single Fourier Attack* [14]. Pour une fréquence non auto-conjuguée, une perturbation de type *Single Fourier Attack* peut alors s'écrire sous la forme

$$\delta_{u,v} = \eta \left[(1+i)\, q_{u,v} + (1-i)\, q_{(-u) \bmod N, (-v) \bmod N} \right]. \tag{3.50}$$

Les coefficients $(1+i)$ et $(1-i)$ sont conjugués l'un de l'autre, et les deux directions de Fourier correspondent à des fréquences opposées. La perturbation respecte donc la condition (3.49), ce qui garantit qu'elle est réelle dans le domaine spatial.

Le paramètre η règle l'amplitude de la perturbation dans cette écriture théorique. En pratique, dans les expériences adversariales, on impose plutôt une borne sur les pixels, par exemple

$$\|\delta\|_\infty \leq \varepsilon.$$

3.12 Limites et portée du cadre théorique

Les résultats précédents donnent une explication mathématique claire de la place particulière des directions de Fourier dans les réseaux convolutionnels. Il faut toutefois préciser la portée exacte de ce résultat.

3.12.1 Un résultat exact dans un cadre contrôlé

Le raisonnement précédent est exact dans un cadre précis : convolutions circulaires, couches linéaires, et analyse de la sensibilité par valeurs singulières. Sous ces hypothèses, la factorisation donnée en (3.23) permet d'étudier le réseau fréquence par fréquence. Les directions singulières obtenues sont alors de la forme

$$a \otimes q_{u,v},$$

où $q_{u,v}$ décrit le motif spatial de Fourier et a le mélange entre les canaux.

Ce résultat doit donc être compris comme une propriété structurelle des convolutions, et non comme une description complète d'un CNN réel.

3.12.2 Écart avec les architectures réelles

Les architectures utilisées en pratique s'écartent de ce cadre exact. Le *zero-padding* remplace souvent le padding circulaire, ce qui empêche une diagonalisation exacte par la base de Fourier. Les couches de réduction, comme les convolutions avec stride et l'*average pooling*, introduisent un phénomène d'aliasing, déjà discuté dans la section 3.10.

Enfin, les activations ReLU et le *max-pooling* rendent le réseau non linéaire. Le réseau complet ne peut donc plus être vu comme un unique opérateur linéaire diagonalisable par Fourier. Cependant, ces opérations possèdent une structure linéaire par morceaux, ce qui permet de conserver une interprétation locale de l'analyse fréquentielle.

3.12.3 Portée locale de l'analyse fréquentielle dans les CNN non linéaires

Dans cette sous-section, on considère que f désigne l'application réalisée par le réseau jusqu'aux logits, avant l'opération finale d'*argmax* qui n'est pas une application affine.

Les réseaux utilisant des activations linéaires par morceaux, comme ReLU, peuvent être vus comme des fonctions affines par morceaux : l'espace d'entrée est découpé en régions, et sur chacune de ces régions, le réseau réalise une application affine [9, 1]. Cette observation permet de préciser la portée de l'analyse fréquentielle dans les CNN réels : le réseau complet n'est pas globalement linéaire, mais son comportement peut être décrit localement par une application affine tant que les choix d'activation et de pooling restent fixés.

Ce qui m'amène à la proposition suivante :

Proposition 3.1 (Linéarisation locale d'un CNN avec ReLU et max-pooling)**.** *Soit f un réseau convolutionnel composé de couches convolutives, d'activations ReLU et de couches de max-pooling. Pour toute entrée x ne se trouvant pas sur une frontière de changement de région linéaire, c'est-à-dire lorsque les signes des pré-activations ReLU sont non nuls et que les maxima sélectionnés dans les fenêtres de pooling sont uniques, il existe un voisinage $\mathcal{V}_x$ de x tel que, pour tout $z \in \mathcal{V}_x$,*

$$f(z) = A_x z + b_x,$$

où A_x est une matrice dépendant de la région linéaire contenant x.

Démonstration. Une activation ReLU agit coordonnée par coordonnée selon

$$\mathrm{ReLU}(t) = \max(t, 0).$$

Considérons une couche donnée du réseau, et notons a_ℓ son vecteur de pré-activations. Si le signe de chaque coordonnée de a_ℓ est fixé dans un voisinage de x, alors la ReLU agit comme une application linéaire sur cette région :

$$\mathrm{ReLU}(a_\ell) = D_\ell(x) a_\ell,$$

où $D_\ell(x)$ est une matrice diagonale dont les coefficients valent 0 ou 1 selon que les coordonnées correspondantes sont désactivées ou activées.

De même, une couche de max-pooling sélectionne, dans chaque fenêtre locale, la coordonnée de valeur maximale. Lorsque ce maximum est unique dans chaque fenêtre, les indices sélectionnés restent inchangés dans un voisinage suffisamment petit de x. Dans cette région, le max-pooling devient donc une application linéaire de sélection :

$$\mathrm{MaxPool}(z) = P_\ell(x)z,$$

où $P_\ell(x)$ est une matrice de sélection dépendant de la région considérée. Cette matrice encode à la fois le choix des maxima locaux et la réduction spatiale induite par le pooling.

Ainsi, tant que l'on reste dans une région où les signes des ReLU et les indices sélectionnés par le max-pooling ne changent pas, chaque opération du réseau est affine ou linéaire. La composition de ces opérations est donc elle-même affine. Il existe donc une matrice A_x et un vecteur b_x tels que

$$f(z) = A_x z + b_x.$$

Cette région locale peut être comprise comme l'ensemble des entrées pour lesquelles les masques d'activation ReLU et les indices sélectionnés par les couches de max-pooling restent inchangés. [1]

□

Cette proposition montre qu'un CNN composé de convolutions, de ReLU et de max-pooling est affine par morceaux. Autour d'une entrée x, tant que la perturbation ne change pas les masques d'activation ni les positions sélectionnées par le pooling, le comportement du réseau est exactement linéaire à un biais près. Ainsi, pour tout $x + \delta \in \mathcal{V}_x$, on a

$$f(x + \delta) - f(x) = J_f(x)\delta.$$

Dans une telle région, le jacobien peut être vu comme un produit de matrices associées aux différentes opérations du réseau. Par exemple, dans le cas schématique d'un empilement de blocs convolution–ReLU–pooling, on peut écrire

$$J_f(x) = W_L P_{L-1}(x) D_{L-1}(x) M_{L-1} \cdots P_1(x) D_1(x) M_1,$$

où les matrices M_ℓ représentent les convolutions, les matrices $D_\ell(x)$ les masques ReLU, les matrices $P_\ell(x)$ les sélections induites par le max-pooling, et W_L la partie linéaire finale.

Les matrices $D_\ell(x)$ et $P_\ell(x)$ dépendent de l'entrée. Elles empêchent donc, en général, une diagonalisation globale du réseau dans la base de Fourier. Une direction de Fourier n'est plus un vecteur propre exact de l'application complète.

Cependant, les opérateurs convolutionnels M_ℓ restent présents dans le produit du jacobien. L'analyse fréquentielle conserve donc une portée locale : elle ne décrit plus exactement le réseau global, mais elle peut encore expliquer pourquoi certaines fréquences sont davantage amplifiées

1. Une étude plus détaillée pourrait consister à définir explicitement cette région, par exemple comme l'ensemble des points (z) vérifiant les mêmes signes de pré-activation ReLU et les mêmes indices de maximum dans les fenêtres de pooling que l'entrée (x). L'analyse de la taille de cette région donnerait une mesure plus précise de la validité locale de la linéarisation, mais dépasse le cadre de ce travail.

que d'autres autour de certaines entrées.

On peut alors interpréter la *Single Fourier Attack* comme une exploitation empirique de cette sensibilité locale. Cette lecture prolonge l'hypothèse de linéarité de Goodfellow et al. [3].

3.12.4 Portée réelle du résultat

Le cadre théorique ne permet pas de prédire exactement quelles fréquences seront les plus sensibles pour une architecture donnée. Cette sensibilité dépend aussi des poids appris, du jeu de données, du padding, des normalisations, des non-linéarités et des couches de réduction.

En revanche, l'analyse explique pourquoi les directions de Fourier constituent des candidates naturelles : elles apparaissent directement dans l'étude linéaire des convolutions. Même dans un CNN non linéaire, cette structure peut rester partiellement visible à travers le jacobien local du modèle. La partie expérimentale servira donc à vérifier dans quelle mesure cette intuition reste observable en pratique.

4 Analyse expérimentale de la sensibilité fréquentielle des CNN

4.1 Objectif et positionnement du chapitre

Dans le chapitre précédent, nous avons montré que les fonctions de Fourier jouent un rôle particulier dans l'analyse des opérateurs de convolution. Dans le cadre linéaire idéalisé, certaines directions fréquentielles apparaissent comme des directions privilégiées de l'opérateur convolutionnel. Dans un CNN réel, composé notamment de ReLU et de max-pooling, cette propriété n'est plus globale ni exacte, mais elle peut encore être interprétée localement à travers le jacobien du modèle.

L'objectif de ce chapitre est donc de vérifier expérimentalement si cette sensibilité fréquentielle reste observable dans un CNN entraîné sur des données réelles. Nous ne cherchons pas à reproduire l'ensemble des expériences de Tsuzuku et Sato [14], mais à reprendre leur mécanisme central : tester systématiquement des perturbations construites à partir de fréquences de Fourier et mesurer leur capacité à modifier les prédictions du modèle.

Notre protocole se concentre sur une architecture de type VGG avec normalisation par batch, notée `VGGSmallBN`, entraînée sur CIFAR-10 et CIFAR-100. Pour chaque fréquence (u, v), nous construisons une perturbation universelle de Fourier, puis nous mesurons la proportion de prédictions modifiées. Cette mesure, appelée *fool ratio*, permet de construire une carte de sensibilité fréquentielle du modèle.

Notre reproduction est volontairement plus restreinte que celle de Tsuzuku et Sato, mais conserve les éléments essentiels : construction d'une perturbation de Fourier, contrôle de son amplitude, calcul du *fool ratio* et génération d'une heatmap fréquentielle.

L'amplitude principale est fixée à

$$\epsilon = 10/255,$$

comme dans les expériences CIFAR de l'article original. Nous ajoutons ensuite deux contrôles expérimentaux : une comparaison avec un bruit aléatoire de même amplitude et un balayage de plusieurs valeurs de ϵ.

Enfin, contrairement à l'algorithme de heatmap de l'article, qui évalue chaque fréquence sur un minibatch, nous évaluons chaque fréquence sur l'ensemble de test complet. Ce choix rend le calcul plus coûteux, mais fournit une estimation plus stable du *fool ratio*.

4.2 Cadre expérimental retenu

4.2.1 Jeux de données

Nous utilisons CIFAR-10 et CIFAR-100. Les deux jeux de données contiennent des images RGB de taille 32×32, ce qui permet de conserver le même domaine fréquentiel dans les deux

expériences :

$$(u, v) \in \{0, \dots, 31\}^2.$$

CIFAR-10 contient 10 classes et constitue un premier cadre de reproduction relativement stable. CIFAR-100 contient 100 classes pour la même résolution d'image ; la tâche est donc plus difficile, car les catégories sont plus nombreuses et souvent visuellement proches. La comparaison entre les deux permet d'observer si la sensibilité fréquentielle persiste lorsque la classification devient plus fine.

4.2.2 Architecture étudiée

Pour cette reproduction expérimentale, nous utilisons une architecture convolutionnelle de type VGG avec normalisation par batch, que nous appelons `VGGSmallBN`. L'objectif n'est pas de reproduire exactement une architecture VGG historique [12], mais de conserver son principe général : empiler plusieurs blocs convolutionnels simples, composés de convolutions 3×3, de normalisations par batch, d'activations ReLU et de couches de réduction spatiale.

Le réseau est composé de quatre blocs convolutionnels. Chaque bloc contient deux convolutions 3×3, chacune suivie d'une normalisation par batch et d'une activation ReLU. Le bloc se termine par un *max pooling* 2×2, qui divise par deux la résolution spatiale. La tête de classification applique ensuite un *adaptive average pooling*, aplatit le tenseur obtenu, puis utilise une couche linéaire finale. Cette architecture est résumée dans la figure 4.1.

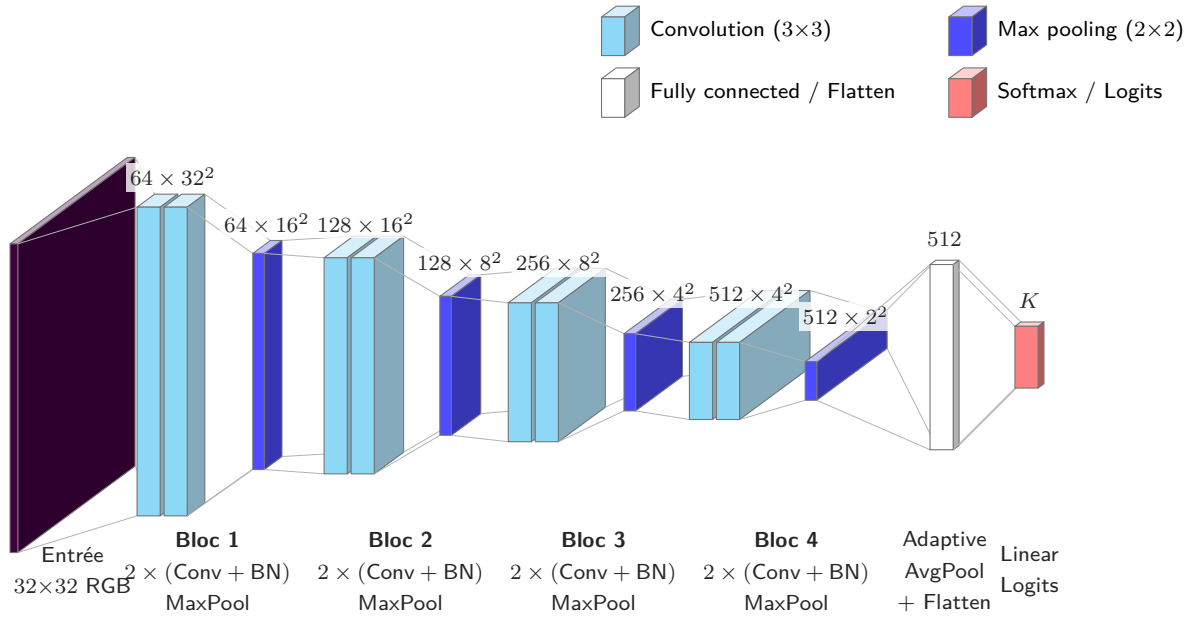


FIGURE 4.1 – Illustration de l'architecture `VGGSmallBN` utilise, où K désigne le nombre de classes du dataset : $K = 10$ pour CIFAR-10 et $K = 100$ pour CIFAR-100.

4.2.3 Paramètres d'entraînement

Paramètre	**Valeur retenue**
Nombre d'époques	100
Optimiseur	SGD
Taux d'apprentissage initial	0.1
Momentum	0.9
Weight decay	$5 \cdot 10^{-4}$
Scheduler	Cosine annealing
Taille de batch	256
Jeux de données	CIFAR-10 et CIFAR-100
Architecture	`VGGSmallBN`

TABLE 4.1 – Paramètres d'entraînement utilisés pour les modèles `VGGSmallBN`.

Le momentum permet de stabiliser les mises à jour des poids et d'accélérer l'optimisation. Le *weight decay* joue un rôle de régularisation : il pénalise les poids trop grands et limite ainsi le risque de surapprentissage.

Le taux d'apprentissage initial est fixé à 0.1, puis diminué progressivement à l'aide d'un scheduler de type *cosine annealing*. L'idée est de laisser le modèle apprendre rapidement au début de l'entraînement, puis d'effectuer des ajustements plus fins lorsque l'on se rapproche de la fin. Enfin, nous entraînons le modèle pendant 100 époques avec une taille de batch de 256, ce qui permet d'exploiter efficacement le GPU disponible tout en conservant un entraînement stable.

4.2.4 Performance de référence

Dataset	**Nombre de classes**	**Validation**	**Test**
CIFAR-10	10	93.86 %	93.07 %
CIFAR-100	100	73.04 %	72.55 %

TABLE 4.2 – Performances de référence des modèles `VGGSmallBN` avant ajout de perturbations.

La différence entre les deux datasets est attendue : CIFAR-100 contient dix fois plus de classes que CIFAR-10, avec des catégories souvent plus proches visuellement. Dans les deux cas, les performances sont suffisamment élevées pour que l'étude des perturbations soit interprétable : les modèles ne sont pas parfaits, mais ils constituent des classifieurs de référence crédibles.

4.3 Construction des perturbations de Fourier

L'idée générale est de construire une perturbation universelle. Contrairement à une attaque adversariale classique, la perturbation n'est pas optimisée séparément pour chaque image. Elle dépend uniquement d'une fréquence (u, v) et d'une amplitude ϵ. On cherche ensuite à savoir si cette direction fréquentielle, bien qu'elle ne soit pas un vecteur propre global du réseau réel, suffit à modifier les prédictions du modèle sur un grand nombre d'images.

4.3.1 Single Fourier Attack

La *Single Fourier Attack* (SFA) proposée par Tsuzuku et Sato [14] consiste à construire une perturbation à partir d'une seule fréquence de Fourier. L'idée principale est simple : au lieu d'optimiser une perturbation différente pour chaque image, on choisit une fréquence (u, v), on construit le motif spatial correspondant, puis on ajoute ce même motif à toutes les images.

Dans notre protocole, une fréquence est représentée par un couple

$$(u, v) \in \{0, \dots, N-1\}^2,$$

où N désigne la taille spatiale de l'image.

Pour une fréquence donnée (u, v), on note $\delta_{u,v}$ la perturbation construite à partir de cette fréquence. L'image perturbée est alors définie par

$$\tilde{x}_i = \mathrm{clip}_{[0,1]}\left(x_i + \delta_{u,v,i}\right) = \begin{cases} 0, & \text{si } x_i + \delta_{u,v,i} < 0, \\ x_i + \delta_{u,v,i}, & \text{si } x_i + \delta_{u,v,i} \in [0,1], \\ 1, & \text{si } x_i + \delta_{u,v,i} > 1. \end{cases} \tag{4.1}$$

où x est l'image initiale, $\tilde{x}$ l'image perturbée, et où le clipping garantit que les valeurs de pixels restent dans l'intervalle valide $[0, 1]$.

En pratique, pour chaque fréquence (u, v), nous construisons d'abord un motif spatial réel $p_{u,v}$. Ce motif est ensuite normalisé en norme ℓ_∞, afin que son amplitude maximale soit égale à l'amplitude choisie ϵ :

$$\delta_{u,v} = \epsilon \frac{p_{u,v}}{\|p_{u,v}\|_\infty}. \tag{4.2}$$

Ainsi, si une fréquence obtient un *fool ratio* plus élevé qu'une autre, ce n'est pas parce qu'elle a été testée avec une perturbation plus grande, mais parce que sa direction fréquentielle perturbe davantage le modèle.

Pour les images RGB de CIFAR-10 et CIFAR-100, le même motif spatial est appliqué aux trois canaux couleur. La perturbation finale possède donc la même structure fréquentielle dans chaque canal. Après ajout de la perturbation, l'image est ramenée dans l'intervalle $[0, 1]$, normalisée avec les statistiques du dataset considéré avant d'être donnée au modèle.

L'algorithme 1 résume la procédure utilisée dans notre implémentation.

Algorithm 1 Single Fourier Attack utilisée dans notre protocole

input : image x, fréquence (u, v), amplitude ϵ
output : image perturbée $\tilde{x}$

Construire le motif spatial réel $p_{u,v}$
Normaliser ce motif en norme ℓ_∞ :
$\delta_{u,v} \leftarrow \epsilon \dfrac{p_{u,v}}{\|p_{u,v}\|_\infty}$
Ajouter la perturbation à l'image :
$x_{\text{pert}} \leftarrow x + \delta_{u,v}$
Appliquer le clipping :
$\tilde{x} \leftarrow \text{clip}_{[0,1]}(x_{\text{pert}})$
return $\tilde{x}$

4.3.2 Contrôle de l'amplitude

Dans notre protocole principal, nous utilisons la même borne que celle utilisée dans l'article original pour CIFAR :

$$\epsilon = \frac{10}{255}.$$

Cette valeur signifie que chaque pixel peut être modifié d'au plus 10 niveaux sur une échelle de 0 à 255.

Pour chaque fréquence (u, v), la perturbation est normalisée de manière à vérifier

$$\|\delta_{u,v}\|_\infty \leq \epsilon.$$

Après ajout de la perturbation, l'image est ramenée dans l'intervalle valide $[0, 1]$ par clipping, comme dans l'équation (4.1).

La figure 4.2 montre, à titre d'exemple, le motif spatial obtenu pour la fréquence $(16, 24)$, qui correspond à la meilleure direction trouvée sur CIFAR-10.

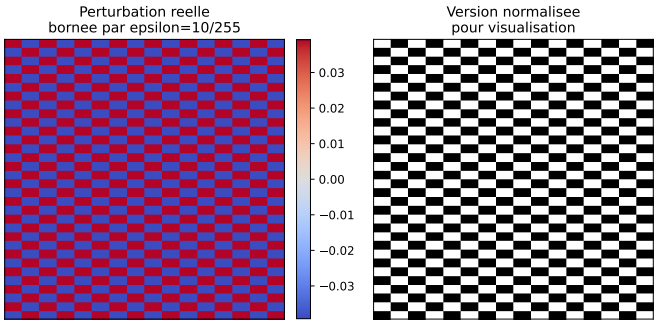


FIGURE 4.2 – Motif spatial associé à la perturbation SFA pour la fréquence $(16, 24)$. La visualisation est normalisée afin de rendre la structure du motif visible.

4.4 Mesure de la sensibilité fréquentielle

Une fois les perturbations de Fourier définies, il faut mesurer leur effet sur le modèle.

4.4.1 Définition du fool ratio

Soit f le modèle étudié, et soit $\mathcal{D} = \{x_1, \dots, x_M\}$ un ensemble d'images. Pour une fréquence (u, v) donnée, on note $\tilde{x}_k^{(u,v)}$ l'image obtenue en ajoutant à x_k la perturbation de Fourier associée à cette fréquence.

Le *fool ratio* mesure la proportion d'images pour lesquelles la prédiction du modèle change après perturbation :

$$\mathrm{FR}(u, v) = \frac{1}{M} \sum_{k=1}^{M} \mathbf{1}\left[f(x_k) \neq f\left(\tilde{x}_k^{(u,v)}\right)\right].$$

Cette mesure ne compare pas directement la prédiction au label réel. Elle compare uniquement la prédiction du modèle avant et après perturbation. Ainsi, une image est comptée comme réussie pour l'attaque lorsque la perturbation modifie la décision du modèle.

Le *fool ratio* indique donc dans quelle mesure une direction fréquentielle est capable de modifier les prédictions, indépendamment de la classe réelle des images.

4.4.2 Génération des heatmaps

Une heatmap de sensibilité fréquentielle est une matrice $H \in \mathbb{R}^{32\times32}$ dont chaque case correspond à une fréquence (u, v). La valeur $H_{u,v}$ est le *fool ratio* obtenu lorsque la perturbation de Fourier associée à cette fréquence est ajoutée aux images de test.

Le calcul suit trois étapes : d'abord, nous calculons les prédictions du modèle sur les images propres ; ensuite, pour chaque fréquence, nous appliquons la perturbation correspondante ; enfin, nous mesurons la proportion de prédictions modifiées. Cette procédure est résumée dans l'algorithme 2.

Algorithm 2 Génération d'une heatmap de sensibilité fréquentielle

input : modèle f, images de test $\mathcal{D}$, amplitude ϵ
output : heatmap $H \in \mathbb{R}^{32\times32}$

$y \leftarrow f(\mathcal{D})$ ▷ prédictions sur les images propres
for $u = 0$ **to** 31 **do**
  for $v = 0$ **to** 31 **do**
    $\delta_{u,v} \leftarrow$ construire la perturbation de Fourier de fréquence (u, v)
    $\tilde{\mathcal{D}}_{u,v} \leftarrow \mathrm{clip}_{[0,1]}(\mathcal{D} + \delta_{u,v})$
    $\tilde{y}_{u,v} \leftarrow f(\tilde{\mathcal{D}}_{u,v})$
    $H_{u,v} \leftarrow \mathrm{FoolRatio}(y, \tilde{y}_{u,v})$
  end for
end for

4.4.3 Choix de la fréquence la plus sensible

Une fois la heatmap calculée, nous sélectionnons la fréquence qui maximise le *fool ratio* :

$$(u^\star, v^\star) = \arg\max_{(u,v)} \mathrm{FR}(u, v).$$

Cette fréquence correspond à la perturbation de Fourier la plus efficace parmi toutes celles testées dans le protocole.

Cette sélection est effectuée séparément pour CIFAR-10 et CIFAR-100. Les fréquences obtenues et leur interprétation sont présentées dans la section 4.5.2.

4.5 Résultats expérimentaux

4.5.1 Heatmaps de sensibilité fréquentielle

La méthodologie décrite précédemment nous permet de construire les cartes de sensibilité fréquentielle du modèle `VGGSmallBN`. La figure 4.3 compare les résultats obtenus sur nos deux datasets pour une même amplitude de perturbation, fixée à $\epsilon = 10/255$.

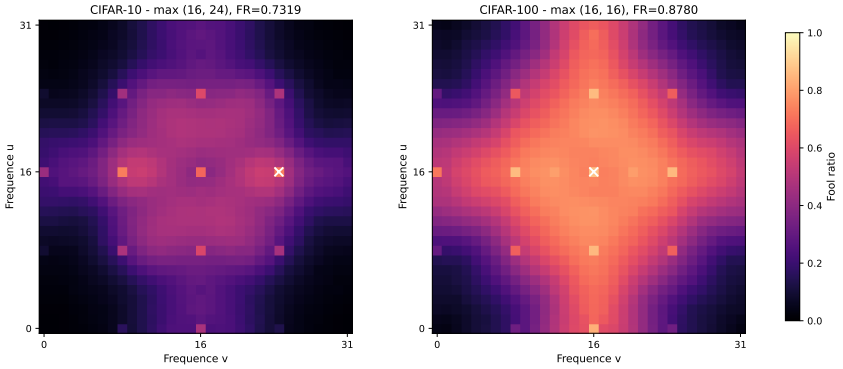


FIGURE 4.3 – Heatmaps de sensibilité fréquentielle pour `VGGSmallBN` sur CIFAR-10 et CIFAR-100.

L'observation principale est que la sensibilité n'est pas uniforme dans le domaine de Fourier. Certaines fréquences provoquent très peu de changements de prédiction, tandis que d'autres produisent un *fool ratio* beaucoup plus élevé.

Les deux heatmaps présentent une organisation structurée, avec des zones de forte sensibilité autour de certaines lignes et intersections fréquentielles, notamment près des coordonnées liées à 16. Cette structure suggère que la vulnérabilité du modèle n'est pas un effet aléatoire, mais dépend de certaines directions fréquentielles privilégiées.

Une comparaison descriptive avec le spectre moyen des images propres est donnée en annexe C.7. Cette analyse vise a donne une autre manière d'interpréter les résultats sous le prisme des statistiques fréquentielles des données brutes.

Pour compléter la lecture des heatmaps, la figure 4.4 représente la distribution des *fool ratios* sur les $32 \times 32 = 1024$ fréquences testées.

Sur CIFAR-10, la plupart des fréquences ont un effet faible ou modéré, et seules quelques directions atteignent des *fool ratios* élevés. La sensibilité du modèle est donc réelle, mais relativement localisée dans le domaine fréquentiel.

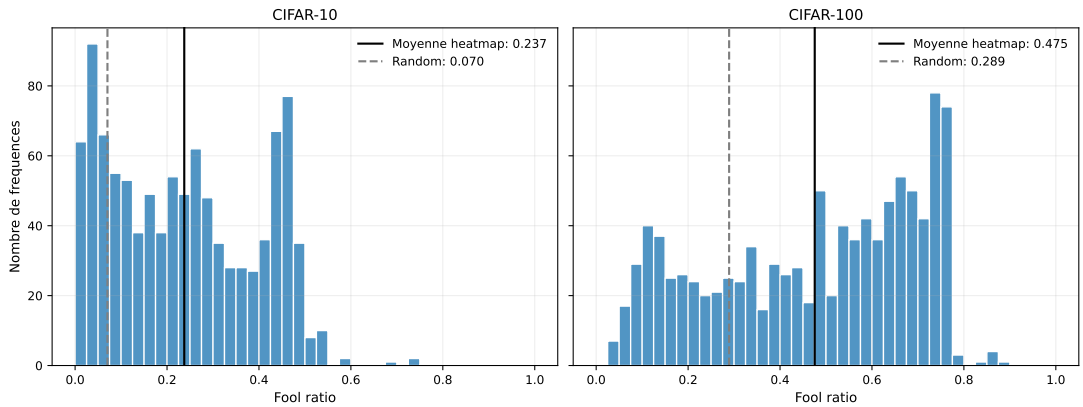


FIGURE 4.4 – Distribution des *fool ratios* obtenus sur l'ensemble des fréquences de Fourier testées.

Sur CIFAR-100, la distribution est déplacée vers la droite : la moyenne passe de 0.2375 à 0.4755. Le modèle est donc globalement plus fragile, ce qui est cohérent avec la difficulté plus élevée de la tâche discutée en annexe C.10. Cependant, cette fragilité reste structurée : toutes les fréquences ne produisent pas le même effet, et certaines directions demeurent nettement plus perturbatrices que d'autres.

L'histogramme complète donc la lecture visuelle des heatmaps. Il montre que la sensibilité n'est pas seulement non uniforme spatialement : elle est aussi très dispersée quantitativement. Une petite partie des fréquences concentre l'essentiel de l'effet perturbateur.

4.5.2 Fréquences les plus sensibles

L'analyse des heatmaps permet d'identifier les fréquences les plus perturbatrices pour chaque dataset. Le tableau 4.3 présente les trois fréquences obtenant les plus grands *fool ratios*.

	CIFAR-10		CIFAR-100	
Rang	Fréquence (u, v)	*Fool ratio*	Fréquence (u, v)	*Fool ratio*
1	$(16, 24)$	0.7319	$(16, 16)$	0.8780
2	$(16, 8)$	0.7295	$(16, 8)$	0.8585
3	$(16, 16)$	0.6765	$(24, 16)$	0.8565

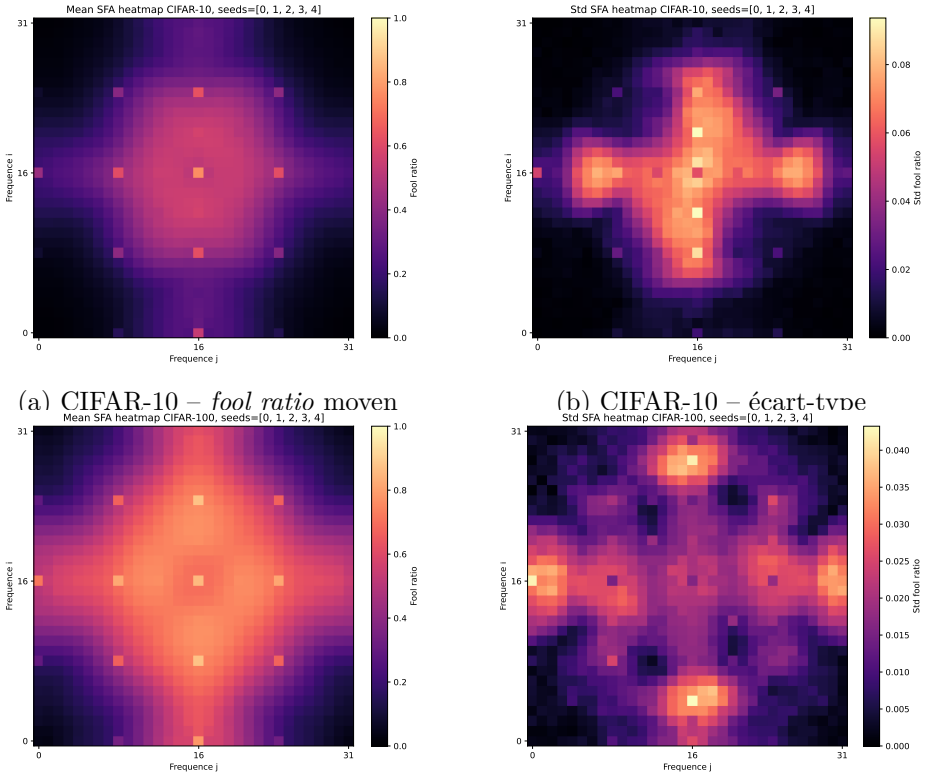
TABLE 4.3 – Fréquences les plus sensibles pour `VGGSmallBN` sur CIFAR-10 et CIFAR-100.

Les fréquences dominantes se concentrent autour d'une même famille de directions. Sur CIFAR-10, les deux premières fréquences, $(16, 24)$ et $(16, 8)$, sont conjuguées modulo 32 et obtiennent logiquement des *fool ratios* presque identiques.

Sur CIFAR-100, la fréquence la plus sensible devient $(16, 16)$, qui est sa propre conjuguée modulo 32.

4.5.3 Stabilité des zones sensibles entre plusieurs entraînements

Les heatmaps précédentes montrent que certaines fréquences sont beaucoup plus sensibles que d'autres. Cependant, une question importante reste à vérifier : cette structure est-elle propre à un entraînement particulier du modèle, ou reste-t-elle observable lorsque l'on change la seed d'initialisation ?



(a) CIFAR-10 – *fool ratio* moyen

(b) CIFAR-10 – écart-type

(c) CIFAR-100 – *fool ratio* moyen

(d) CIFAR-100 – écart-type

FIGURE 4.5 – Stabilité de la sensibilité fréquentielle de `VGGSmallBN` sur CIFAR-10 et CIFAR-100.

Pour répondre à cette question, nous avons répété l'expérience sur CIFAR-10 et CIFAR-100 avec cinq entraînements indépendants de `VGGSmallBN`. Le protocole d'entraînement est conservé identique pour chaque run. Seule la seed est modifiée.

La figure 4.5 présente, pour chaque dataset, la heatmap moyenne obtenue sur les cinq seeds ainsi que l'écart-type associé à chaque fréquence.

Les résultats montrent que la structure globale des heatmaps reste très stable entre les entraînements. Sur CIFAR-10, les zones de forte sensibilité restent localisées autour des mêmes régions fréquentielles, en particulier autour des fréquences médianes liées aux axes centraux. Sur CIFAR-100, cette structure est encore plus marquée : une large zone de forte sensibilité apparaît autour des fréquences centrales.

Le tableau 4.4 résume les principales métriques moyennées sur les cinq seeds. L'accuracy propre reste proche de celle du modèle principal, tandis que l'accuracy après attaque diminue fortement. Cela montre que le *fool ratio* élevé correspond bien à une dégradation réelle des performances du modèle, et pas seulement à des changements de prédiction sans conséquence sur la classification correcte.

Métrique	CIFAR-10	CIFAR-100
Accuracy propre moyenne	93.08 %	72.50 %
Meilleur *fool ratio* moyen	76.35 %	87.64 %
Accuracy après attaque moyenne	23.39 %	11.83 %
Taux de succès sur images initialement correctes	76.38 %	85.18 %

TABLE 4.4 – Résumé de l'expérience de stabilité multi-seeds sur CIFAR-10 et CIFAR-100.

4.5.4 Amplification interne des fréquences sensibles

Les heatmaps précédentes montrent que certaines fréquences de Fourier produisent des *fool ratios* nettement plus élevés que d'autres. Pour mieux comprendre ce phénomène, nous avons analysé la manière dont ces perturbations se propagent à l'intérieur du réseau.

Pour une perturbation δ, nous notons $G_l(\delta)$ l'amplification normalisée mesurée à la couche l. Cette quantité compare la norme moyenne de la variation d'activation provoquée par la perturbation à la norme de la perturbation d'entrée. Elle permet donc de comparer l'effet relatif de différentes fréquences dans les blocs internes du réseau et au niveau des logits.

Afin de rendre cette analyse plus robuste, nous l'avons répétée sur les cinq checkpoints utilisés dans l'expérience de stabilité multi-seeds. Les fréquences sont regroupées à partir de la heatmap moyenne : les 50 fréquences les plus sensibles, 50 fréquences de sensibilité intermédiaire, et les 50 fréquences les moins sensibles. La fréquence principale correspond à la meilleure fréquence moyenne sur les cinq seeds.

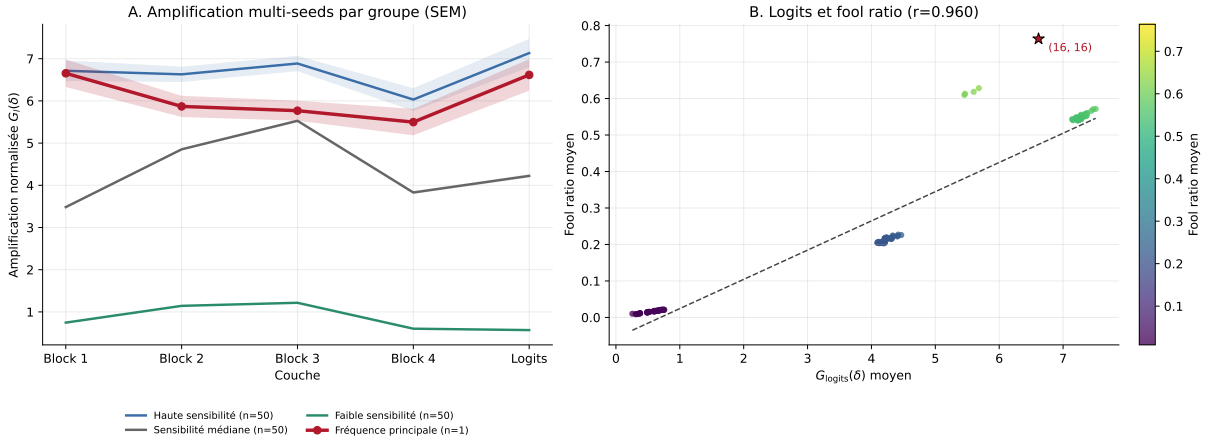


FIGURE 4.6 – Analyse multi-seeds de l'amplification interne des perturbations de Fourier dans `VGGSmallBN` sur CIFAR-10. À gauche, les fréquences sont regroupées selon leur *fool ratio* moyen. Les bandes indiquent l'erreur standard sur les cinq seeds. À droite, l'amplification moyenne au niveau des logits est comparée au *fool ratio* moyen.

La figure 4.6 montre que logiquement les fréquences les plus sensibles sont aussi celles qui produisent les amplifications internes les plus élevées.

Le panneau de droite montre une corrélation élevée entre l'amplification moyenne au niveau des logits et le *fool ratio* moyen. Cela suggère que la sensibilité finale du modèle n'est pas

seulement liée au couple (u, v) choisi, mais aussi à la manière dont cette perturbation est propagée et amplifiée par le réseau.

Cette observation fait le lien avec le cadre théorique et avec l'interprétation locale introduite au chapitre précédent : même si le réseau réel n'est pas globalement linéaire, certaines directions fréquentielles semblent être davantage amplifiées par son comportement local, ce qui peut expliquer leur effet plus important sur les prédictions.

Une visualisation plus détaillée des heatmaps internes par couche est donnée en annexe C.9.

4.5.5 Comparaison avec un bruit aléatoire

Pour vérifier que les fortes valeurs obtenues par la SFA ne proviennent pas uniquement de l'amplitude de la perturbation, nous comparons la meilleure direction de Fourier avec une baseline de bruit aléatoire. Dans cette baseline, chaque pixel est perturbé indépendamment par un bruit uniforme borné dans le même intervalle d'amplitude. La contrainte est donc la même que pour la SFA :

$$\|\delta\|_\infty \leq \epsilon, \qquad \epsilon = 10/255.$$

La différence vient de la structure de la perturbation : la SFA utilise un motif fréquentiel cohérent, tandis que la baseline aléatoire ne privilégie aucune direction particulière.

A FAIRE : Il faudrais s'assurer que $\|\delta_{SFA}\|_2 = \|\delta_{random}\|_2$

La figure 4.7 compare, pour chaque dataset, le *fool ratio* de la meilleure fréquence SFA avec le *fool ratio* moyen obtenu par bruit aléatoire.

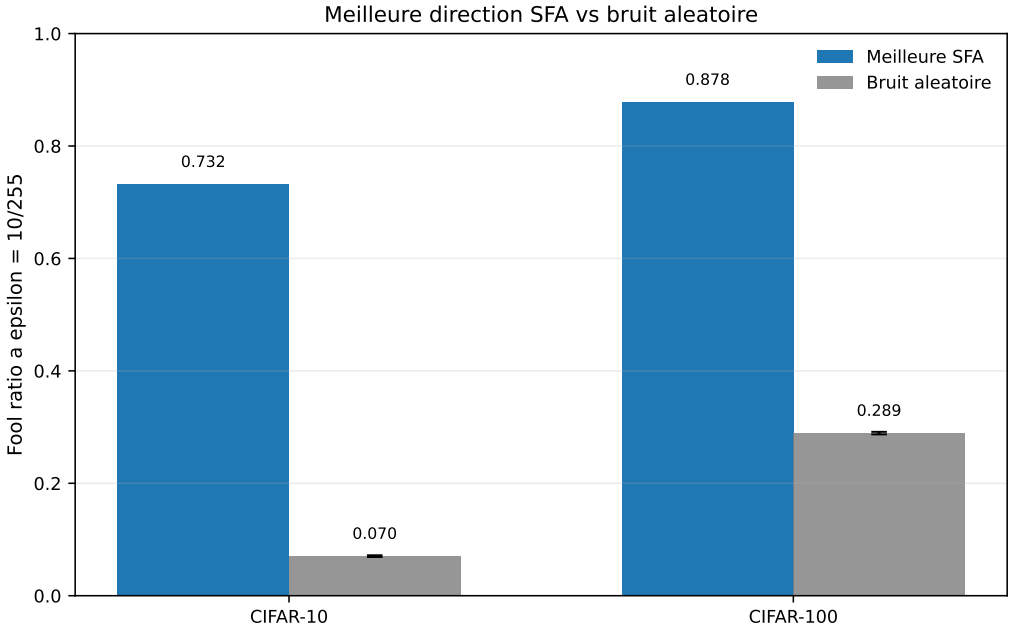


FIGURE 4.7 – Comparaison entre la meilleure direction SFA et un bruit aléatoire uniforme de même amplitude $\epsilon = 10/255$. Pour le bruit aléatoire, la barre d'erreur représente l'écart-type sur 100 essais.

Sur CIFAR-10, la différence est très nette : la meilleure direction de Fourier atteint un *fool ratio* de 0.7319, alors que le bruit aléatoire moyen atteint seulement 0.0704. La perturbation

de Fourier est donc environ dix fois plus efficace qu'un bruit aléatoire de même amplitude. Ce résultat indique que la vulnérabilité observée ne peut pas être expliquée par la seule taille de la perturbation.

Sur CIFAR-100, le bruit aléatoire est déjà plus perturbateur, avec un *fool ratio* moyen de 0.2892. Confirment que le modèle entraîné sur CIFAR-100 est globalement plus fragile. Cependant, la meilleure direction SFA atteint 0.8780, soit environ trois fois le niveau du bruit aléatoire moyen.

4.5.6 Effet de l'amplitude de la perturbation

Jusqu'ici, les résultats principaux ont été obtenus avec $\epsilon = 10/255$. Cette valeur permet de comparer nos résultats au protocole original, mais elle peut produire des motifs visibles sur des images de très faible résolution. Or une attaque adversariale a pour but de rester peu perceptible pour l'humain. Nous testons donc si les fréquences les plus sensibles conservent un effet pour des valeurs plus faibles de ϵ.

Pour cela, nous fixons la fréquence la plus sensible de chaque dataset, puis nous faisons varier ϵ dans l'ensemble suivant :

$$\epsilon \in \left\{ \frac{1}{255}, \frac{2}{255}, \frac{3}{255}, \frac{5}{255}, \frac{10}{255} \right\}.$$

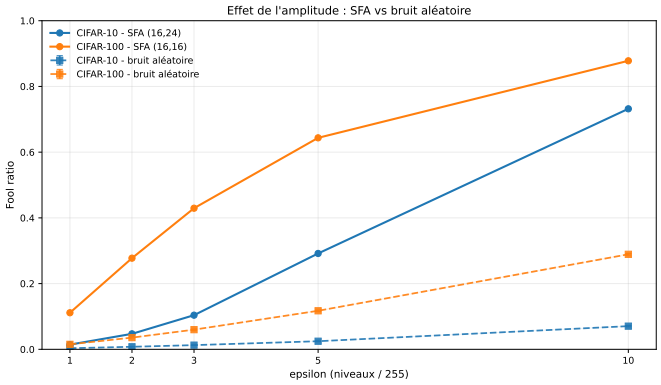


FIGURE 4.8 – Évolution du *fool ratio* en fonction de l'amplitude ϵ pour la meilleure fréquence SFA de chaque dataset.

La figure 4.8 montre une augmentation régulière du *fool ratio* lorsque ϵ augmente. Ce comportement est attendu : plus la perturbation est grande, plus elle modifie l'image d'entrée, et plus elle a de chances de déplacer la prédiction du modèle. Mais elle confirme aussi que dans ce contexte, l'amplitude de la perturbation est un facteur important de l'efficacité de l'attaque, nous obligeant rendre la perturbation plus moins discrète pour obtenir un effet plus significatif.

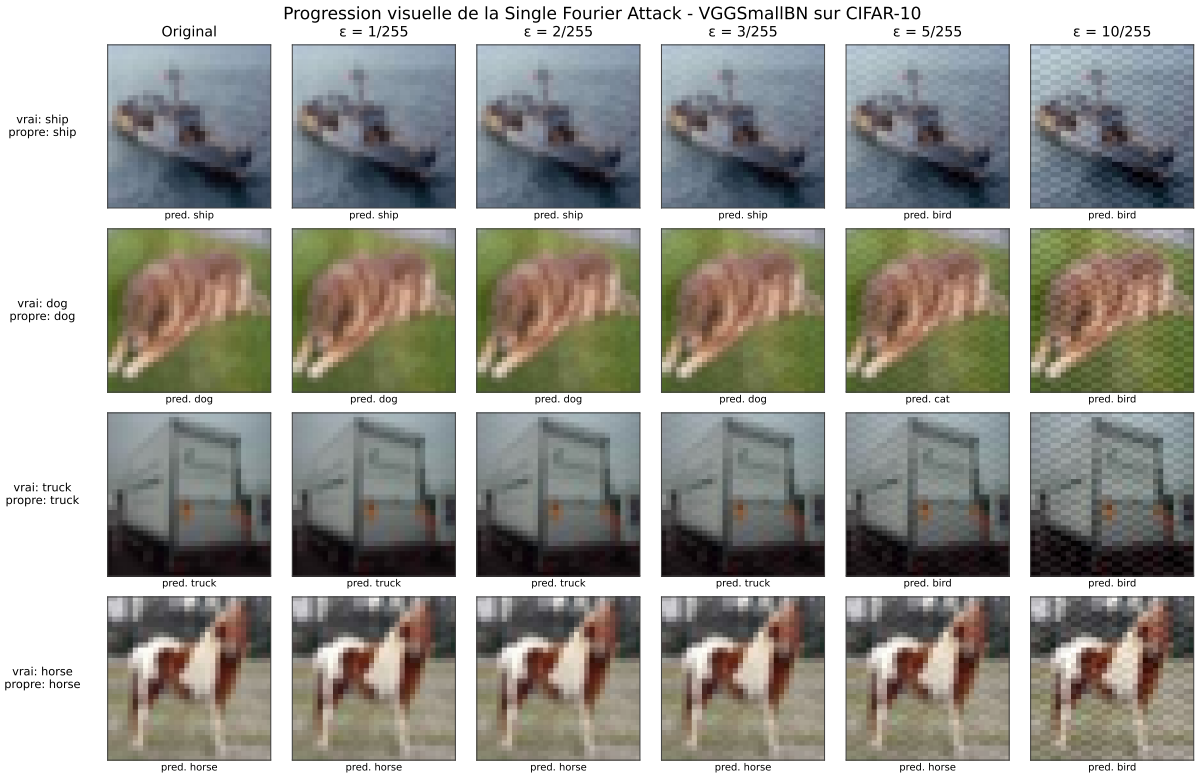


FIGURE 4.9 – Progression visuelle d'une attaque *Single Fourier Attack* sur des images de CIFAR-10.

Comme l'illustre la figure 4.9, l'augmentation de l'amplitude rend le motif de Fourier de plus en plus intrusif. On remarque notamment qu'à partir d'une valeur de $\epsilon = 3/255$, la perturbation structurée en forme de grille commence à être distinctement visible à l'oeil nu.

Il convient néanmoins de nuancer ces observations. Les expériences sont réalisées sur des images de très basse résolution (32×32 pixels) et avec une architecture volontairement restreinte (`VGGSmallBN`). Les conclusions ne doivent donc pas être généralisées directement à des images de haute résolution ou à des architectures beaucoup plus profondes. Elles montrent cependant que, dans notre cadre expérimental, l'effet des perturbations de Fourier dépend fortement de l'amplitude.

4.5.7 Lien avec le cadre théorique

Les résultats expérimentaux obtenus dans ce chapitre permettent de revenir au cadre théorique développé au chapitre précédent. Dans le modèle linéaire idéalisé, les fonctions de Fourier apparaissent comme des directions privilégiées des opérateurs de convolution. Dans un CNN réel, cette propriété ne s'applique plus directement au réseau complet, mais elle peut rester pertinente localement à travers le jacobien du modèle.

Les heatmaps obtenues expérimentalement vont dans ce sens. Si toutes les directions fréquentielles étaient équivalentes pour le modèle, les cartes de sensibilité seraient beaucoup plus uniformes. Or ce n'est pas ce que l'on observe : certaines fréquences produisent des *fool ratios* nettement plus élevés que d'autres, alors que toutes les perturbations respectent la même contrainte ℓ_∞.

La stabilité multi-seeds renforce cette interprétation. Les zones sensibles ne semblent pas dépendre uniquement d'une initialisation particulière du réseau : elles restent visibles lorsque l'entraînement est répété avec différentes seeds. La comparaison avec le bruit aléatoire montre également que l'effet ne provient pas seulement de l'amplitude de la perturbation, mais de sa structure fréquentielle.

Ces résultats montrent que l'intuition issue du cadre théorique reste observable expérimentalement pour `VGGSmallBN` : certaines directions de Fourier sont plus perturbatrices que d'autres, ce qui est compatible avec une amplification locale de ces directions dans le réseau.

Influence du mélange entre canaux. Precedement nous avions considéré qu'une direction sensible peut aussi faire intervenir un mélange entre les canaux couleur, sous la forme $a \otimes \varphi_{u,v}$. Le protocole principal utilise une version simplifiée de cette idée : le même motif de Fourier est appliqué aux trois canaux, ce qui correspond à une direction couleur de type $(1, 1, 1)$.

Une analyse complémentaire de cette simplification est proposée en annexe C.8. Elle vise à montrer que le choix du mélange entre canaux est negligeable en comparaison du choix de maximiser l'amplification de l'attaque.

4.6 Synthèse du chapitre

Les expériences réalisées confirment que la sensibilité de `VGGSmallBN` aux perturbations de Fourier est fortement non uniforme. Sur CIFAR-10 comme sur CIFAR-100, certaines fréquences provoquent beaucoup plus de changements de prédiction que d'autres, alors que toutes les perturbations respectent la même contrainte ℓ_∞.

Cette structure reste stable entre plusieurs entraînements indépendants et dépasse nettement l'effet obtenu avec un bruit aléatoire de même amplitude. Le balayage de ϵ montre toutefois que l'efficacité de l'attaque dépend fortement de l'amplitude, en particulier sur des images de basse résolution comme celles de CIFAR.

Enfin, l'analyse interne suggère que les fréquences les plus sensibles sont aussi celles qui sont le plus amplifiées dans les représentations du réseau. Ces observations ne prouvent pas que le modèle réel se comporte comme un opérateur linéaire global, mais elles montrent que la structure fréquentielle mise en évidence dans le cadre théorique reste partiellement visible dans le comportement local du modèle.

5 Conclusion

Ce travail visait à explorer le lien intrinsèque entre la structure convolutionnelle des réseaux de neurones (CNN) et l'émergence de directions de vulnérabilité privilégiées dans le domaine de Fourier. Plus précisément, notre démarche a consisté à déterminer dans quelle mesure les propriétés mathématiques inhérentes aux couches de convolution au-delà des biais liés aux données ou au processus d'entraînement permettent d'expliquer l'existence de perturbations adversariales universelles.

Pour ce faire, après avoir posé le cadre général des CNN et des attaques adversariales, nous avons reconstruit et analysé le raisonnement théorique proposé par Tsuzuku et Sato [14], en nous appuyant initialement sur un modèle linéaire simplifié.

Afin de transposer ces résultats théoriques à l'architecture réelle des CNN, où la présence d'activations ReLU et d'opérations de *max-pooling* empêche toute diagonalisation globale, nous avons introduit le concept de linéarisation locale via l'analyse du jacobien du modèle. Cette extension théorique s'est avérée cruciale pour justifier l'application de résultats fondés sur la linéarité à des architectures fondamentalement non linéaires.

L'ensemble de ces observations permet ainsi d'élucider le mécanisme par lequel une perturbation fondée sur une unique fréquence de Fourier peut impacter si sévèrement un réseau convolutionnel. L'amplification successive de cette perturbation au travers des différentes couches offre une explication structurelle probante à l'existence des perturbations universelles.

Notre démarche expérimentale vient conforter cette intuition. Les *heatmaps* générées sur l'architecture `VGGSmallBN`, pour les jeux de données CIFAR-10 et CIFAR-100, mettent en évidence une sensibilité fréquentielle fortement hétérogène. Certaines fréquences engendrent en effet des *fool ratios* significativement supérieurs, démontrant que des perturbations d'amplitude égale n'ont pas un impact équivalent : leur direction dans l'espace des entrées est un facteur déterminant. La comparaison de ces résultats avec l'injection d'un bruit aléatoire de même amplitude corrobore pleinement cette asymétrie.

Par ailleurs, les résultats soulignent l'influence logique de la complexité de la tâche de classification sur la robustesse du modèle. Le réseau s'avère globalement plus vulnérable sur CIFAR-100 que sur CIFAR-10. Cette fragilité, examinée plus en détail dans l'annexe C.10, est cohérente avec l'exigence accrue du problème : face à un nombre de classes supérieur, les frontières de décision se multiplient et se resserrent en raison de la proximité visuelle des catégories, permettant à une perturbation de modifier plus aisément la prédiction. Néanmoins, en dépit de cette vulnérabilité accrue, la sensibilité conserve une topologie structurée où certaines directions de Fourier demeurent prépondérantes.

Il faut toutefois rester prudent sur la portée de ces résultats. Notre reproduction expérimentale se limite à une architecture de type VGG et à des images de faible résolution. De plus, les fréquences les plus sensibles ont été choisies à partir de *heatmaps* calculées sur l'ensemble de test. On ne peut donc pas affirmer que tous les réseaux convolutionnels présentent exactement la même structure de vulnérabilité. En revanche, dans le cadre étudié ici, les directions de Fourier apparaissent comme un outil pertinent pour mieux comprendre certaines sensibilités des CNN.

Cette analyse peut aussi être reliée à la question de l'explicabilité des modèles. En effet, lorsqu'une fréquence particulière modifie fortement les prédictions du réseau, cela indique que cette composante de l'image joue un rôle important dans sa décision. Autrement dit, les fréquences sensibles ne servent pas seulement à construire des perturbations adversariales : elles donnent aussi des indices sur le type d'information que le réseau utilise pour classifier les images.

Une suite possible à ce travail serait donc d'utiliser ces fréquences sensibles à la fois pour renforcer les modèles et pour mieux les interpréter. Par exemple, elles pourraient servir à guider de nouvelles formes d'augmentation de données, à limiter certaines amplifications internes, ou à comparer des architectures plus robustes, notamment avec des mécanismes d'anti-aliasing.

Cette direction permettrait de faire le lien entre trois aspects importants : la robustesse adversariale, la résistance aux variations des données, et l'interprétation des décisions prises par le réseau. L'objectif ne serait donc pas seulement de montrer que certaines fréquences de Fourier sont dangereuses pour le modèle, mais aussi de comprendre comment leur identification peut aider à construire des réseaux plus stables, plus robustes et plus compréhensibles.

A Représentation matricielle d'une application linéaire

Soient V et W deux espaces vectoriels de dimensions finies sur un corps $\mathbb{K}$, où $\mathbb{K}$ désigne ici $\mathbb{R}$ ou $\mathbb{C}$. On note

$$\mathcal{B} = (e_1, \ldots, e_N)$$

une base de V, et

$$\mathcal{B}' = (e'_1, \ldots, e'_M)$$

une base de W.

Soit maintenant une application linéaire

$$f : V \to W.$$

Tout vecteur $v \in V$ se décompose de manière unique dans la base $\mathcal{B}$:

$$v = \sum_{i=1}^{N} \alpha_i e_i,$$

où les coefficients α_i sont les coordonnées de v dans la base $\mathcal{B}$.

Par linéarité de f, on a alors :

$$f(v) = f\left(\sum_{i=1}^{N} \alpha_i e_i\right) = \sum_{i=1}^{N} \alpha_i f(e_i).$$

Comme $f(e_i) \in W$, chaque vecteur $f(e_i)$ peut lui-même être décomposé dans la base $\mathcal{B}'$. Il existe donc des scalaires $a_{ji} \in \mathbb{K}$ tels que :

$$f(e_i) = \sum_{j=1}^{M} a_{ji} e'_j.$$

En remplaçant cette expression dans la formule précédente, on obtient :

$$f(v) = \sum_{i=1}^{N} \alpha_i \left(\sum_{j=1}^{M} a_{ji} e'_j\right).$$

En réorganisant les sommes selon les vecteurs de la base $\mathcal{B}'$, il vient :

$$f(v) = \sum_{j=1}^{M} \left(\sum_{i=1}^{N} a_{ji} \alpha_i\right) e'_j.$$

Si l'on note

$$[f(v)]_{\mathcal{B}'} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}, \qquad [v]_{\mathcal{B}} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix},$$

alors les coordonnées de $f(v)$ vérifient :

$$\beta_j = \sum_{i=1}^{N} a_{ji}\alpha_i.$$

On peut donc écrire sous forme matricielle :

$$[f(v)]_{\mathcal{B}'} = A[v]_{\mathcal{B}},$$

où

$$A = (a_{ji})_{\substack{1\leq j\leq M \\ 1\leq i\leq N}} \in \mathbb{K}^{M\times N}.$$

Autrement dit, la matrice A représentant l'application linéaire f est construite colonne par colonne : sa i-ème colonne est le vecteur des coordonnées de $f(e_i)$ dans la base $\mathcal{B}'$:

$$A = \begin{pmatrix} | & | & & | \\ [f(e_1)]_{\mathcal{B}'} & [f(e_2)]_{\mathcal{B}'} & \cdots & [f(e_N)]_{\mathcal{B}'} \\ | & | & & | \end{pmatrix}.$$

Ainsi, toute application linéaire entre espaces vectoriels de dimensions finies peut être représentée par une matrice, dès lors que l'on a choisi une base dans l'espace de départ et une base dans l'espace d'arrivée.

Il est important de noter que cette matrice dépend du choix des bases. Une même application linéaire peut donc avoir des représentations matricielles différentes selon les bases utilisées.

B Vecteurs propres, valeurs propres et diagonalisation

Soit V un espace vectoriel de dimension finie N sur un corps $\mathbb{K}$, et soit

$$f : V \to V$$

un endomorphisme, c'est-à-dire une application linéaire de V dans lui-même.

Un vecteur non nul $v \in V$ est appelé vecteur propre de f s'il existe un scalaire $\lambda \in \mathbb{K}$ tel que :

$$f(v) = \lambda v.$$

Le scalaire λ est alors appelé valeur propre associée au vecteur propre v.

Cette relation signifie que l'application f ne change pas la direction du vecteur v. Elle se contente de le multiplier par un facteur scalaire λ.

Considérons maintenant une base

$$\mathcal{B} = (e_1, \ldots, e_N)$$

de V. D'après le rappel de l'annexe A, l'endomorphisme f peut être représenté par une matrice dans cette base.

La question naturelle est alors la suivante : peut-on choisir une base dans laquelle la matrice de f devient particulièrement simple ?

Le cas le plus favorable est celui où la base $\mathcal{B}$ est composée uniquement de vecteurs propres de f. Supposons donc que, pour tout $i \in \{1, \ldots, N\}$, on ait :

$$f(e_i) = \lambda_i e_i.$$

Alors, dans la base $\mathcal{B}$, les coordonnées de $f(e_i)$ sont nulles partout sauf à la i-ème position :

$$[f(e_i)]_{\mathcal{B}} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \lambda_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

La matrice représentant f dans cette base a donc pour colonnes ces vecteurs de coordonnées.

Elle est diagonale :

$$A_{\mathcal{B}} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}.$$

Autrement dit, si l'on dispose d'une base de vecteurs propres, l'action de f devient très simple : chaque coordonnée est simplement multipliée par la valeur propre correspondante.

Réciproquement, si la matrice de f est diagonale dans une base donnée, alors les vecteurs de cette base sont des vecteurs propres de f.

Cette observation explique l'intérêt de chercher une base de vecteurs propres : dans une telle base, l'application linéaire ne mélange plus les directions. Elle agit indépendamment sur chaque direction de la base.

Dans le cadre de ce rapport, cette idée est appliquée à la convolution circulaire. La matrice associée à une convolution circulaire admet les vecteurs de Fourier comme vecteurs propres. Ainsi, lorsque l'on exprime le signal dans la base de Fourier, l'opérateur de convolution devient diagonal :

$$C_h = F_N \Lambda_h F_N^{-1},$$

où F_N contient les vecteurs de Fourier et où Λ_h est la matrice diagonale des valeurs propres associées.

C Éléments complémentaires

C.1 Périodicité, bords et phénomène de Gibbs

Dans ce mémoire, nous utilisons fréquemment la transformée de Fourier discrète et la convolution circulaire. Ces outils reposent implicitement sur une hypothèse de périodisation du signal. Il est donc utile de rappeler brièvement le lien entre périodicité, spectre de Fourier et effets de bord.

— Si un signal est défini sur un domaine infini, par exemple sur $\mathbb{R}$, et qu'il n'est pas périodique, alors son contenu fréquentiel est décrit par un *spectre continu*. On parle alors de transformée de Fourier.

— Si un signal est périodique, son contenu fréquentiel devient au contraire *discret* : le signal se décompose en une somme de modes harmoniques de fréquences bien déterminées. On parle dans ce cas de série de Fourier.

— Enfin, dans le cas des signaux numériques de longueur finie, on utilise souvent une hypothèse de périodisation. Le signal est alors considéré comme périodiquement prolongé : ses deux extrémités sont artificiellement raccordées, comme si l'on enroulait la feuille sur laquelle il est tracé pour former une boucle.

Cette hypothèse est mathématiquement très utile, car elle permet d'utiliser la transformée de Fourier discrète et de représenter certaines convolutions comme des convolutions circulaires. Elle peut cependant introduire un artefact lorsque les bords du signal ne se raccordent pas naturellement. Si le bord droit ne rejoint pas le bord gauche avec une valeur compatible, une discontinuité artificielle apparaît à la jonction.

Cette discontinuité est importante du point de vue de Fourier. En effet, une approximation par un nombre fini de modes de Fourier représente difficilement un saut abrupt. Elle produit alors des oscillations parasites au voisinage de la coupure. Ce comportement est connu sous le nom de phénomène de Gibbs [11].

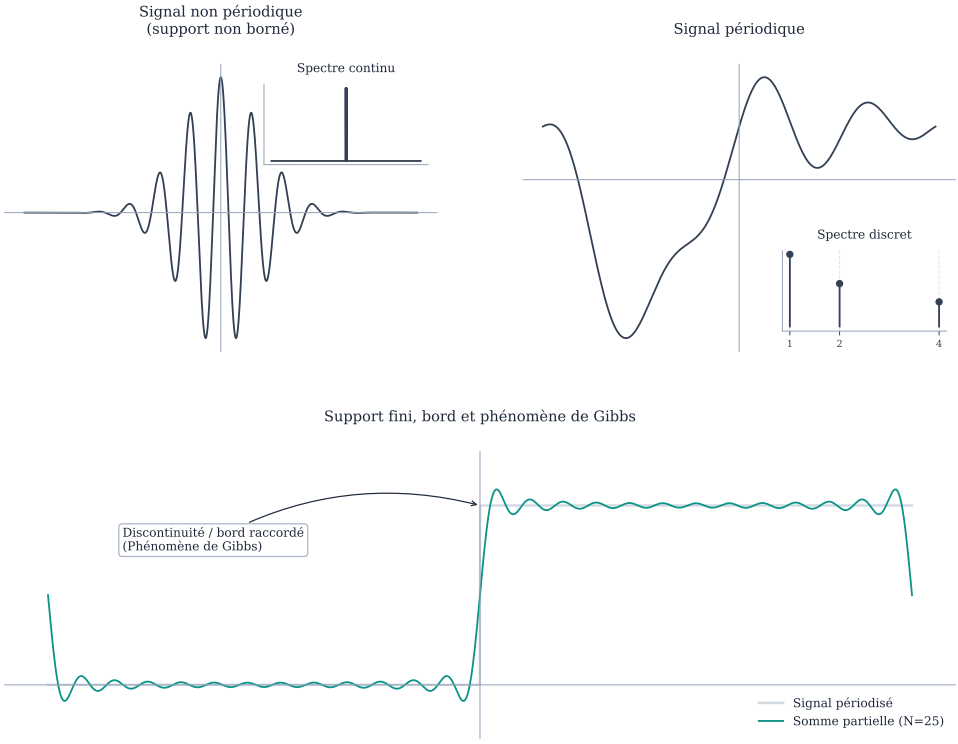


FIGURE C.1 – À gauche : un signal non périodique défini sur un domaine infini possède un spectre continu. Au centre : un signal périodique possède un spectre discret constitué de raies harmoniques. À droite : un signal de support fini peut présenter une discontinuité artificielle lorsqu'il est périodisé.

C.2 Détail de la preuve de la structure BCCB

On reprend ici le cas d'une convolution circulaire 2D mono-canal. L'objectif est de montrer, pas à pas, pourquoi la matrice M associée à cette convolution est circulante par blocs, et chacun de ses blocs est lui-même circulant.

Démonstration. On note $X \in \mathbb{C}^{N\times N}$ l'image d'entrée, $Y \in \mathbb{C}^{N\times N}$ l'image de sortie, et $h \in \mathbb{C}^{N\times N}$ le filtre. On rappelle que dans le cadre circulaire, la convolution 2D s'écrit

$$\begin{aligned} Y[m,n] &= \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} h[a,b]\, X[(m-a) \bmod N,\ (n-b) \bmod N] \\ &= \sum_{a=0}^{N-1}\sum_{b=0}^{N-1} h[(m-a) \bmod N,\ (n-b) \bmod N]\, X[a,b]. \end{aligned}$$

Avec la vectorisation par colonnes, l'entrée $X[r,s]$ correspond à l'indice vectorisé $r+Ns$, et la sortie $Y[m,n]$ correspond à l'indice vectorisé $m+Nn$. La matrice M est définie par

$$\mathrm{vec}(Y) = M\,\mathrm{vec}(X).$$

Le coefficient de M situé à la ligne $m+Nn$ et à la colonne $r+Ns$ indique donc comment le pixel $X[r,s]$ contribue au pixel $Y[m,n]$.

D'après la formule de convolution 2D, cette contribution est donnée par

$$M_{m+Nn,\, r+Ns} = h[(m-r) \bmod N,\, (n-s) \bmod N].$$

Cette formule est le point central de la preuve : le coefficient dépend seulement des décalages circulaires entre les positions d'entrée et de sortie.

On découpe maintenant la matrice M en $N \times N$ blocs, chacun de taille $N \times N$:

$$M = \begin{pmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,N-1} \\ B_{1,0} & B_{1,1} & \cdots & B_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N-1,0} & B_{N-1,1} & \cdots & B_{N-1,N-1} \end{pmatrix}.$$

Dans ce découpage, le bloc $B_{n,s}$ décrit l'effet de la colonne s de l'image d'entrée sur la colonne n de l'image de sortie.

Les coefficients de ce bloc sont donc

$$(B_{n,s})_{m,r} = M_{m+Nn,\, r+Ns} = h[(m-r) \bmod N,\, (n-s) \bmod N].$$

Fixons maintenant deux colonnes n et s. Leur décalage circulaire est

$$\delta = (n-s) \bmod N.$$

Avec cette notation, l'expression précédente devient

$$(B_{n,s})_{m,r} = h[(m-r) \bmod N, \delta].$$

Pour un δ fixé, le coefficient $(B_{n,s})_{m,r}$ dépend de m et r uniquement à travers la différence circulaire $(m-r) \bmod N$. Cela signifie que les lignes du bloc se répètent par décalage circulaire. Autrement dit, chaque bloc $B_{n,s}$ est une matrice circulante.

Pour rendre cela plus explicite, on définit le vecteur

$$h_\delta = \begin{pmatrix} h[0, \delta] \\ h[1, \delta] \\ \vdots \\ h[N-1, \delta] \end{pmatrix}.$$

Le bloc $B_{n,s}$ est alors la matrice circulante construite à partir de ce vecteur :

$$B_{n,s} = C_{h_\delta}.$$

Comme $\delta = (n - s) \bmod N$, on peut aussi écrire

$$B_{n,s} = C_{h_{(n-s) \bmod N}}.$$

Cette formule montre maintenant le deuxième niveau de circularité. Lorsque l'on change n et s, les blocs ne changent pas de manière quelconque : ils dépendent seulement du décalage circulaire $(n - s) \bmod N$. Les blocs eux-mêmes sont donc organisés de manière circulaire.

La matrice complète M peut alors s'écrire sous la forme

$$M = \begin{pmatrix} C_{h_0} & C_{h_{N-1}} & C_{h_{N-2}} & \cdots & C_{h_1} \\ C_{h_1} & C_{h_0} & C_{h_{N-1}} & \cdots & C_{h_2} \\ C_{h_2} & C_{h_1} & C_{h_0} & \cdots & C_{h_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{h_{N-1}} & C_{h_{N-2}} & C_{h_{N-3}} & \cdots & C_{h_0} \end{pmatrix}.$$

Cette écriture fait apparaître clairement les deux propriétés recherchées :

chaque bloc est circulant et les blocs sont eux-mêmes organisés circulairement.

Par conséquent, M est bien une matrice circulante par blocs avec des blocs circulants, c'est-à-dire une matrice BCCB.

□

C.3 Détail de la preuve pour les skip connections

On reprend ici le cas d'une *skip connection* identité étudié dans la section 3.9. L'objectif est de vérifier explicitement que les vecteurs singuliers droits conservent la même structure fréquentielle que dans le cas sans *skip connection*.

Démonstration. Soit $a \in \mathbb{C}^m$ un vecteur singulier droit de $G_{\text{skip}}^{(w)}$, associé à une valeur singulière σ et à un vecteur singulier gauche $b \in \mathbb{C}^m$:

$$G_{\text{skip}}^{(w)} a = \sigma b.$$

Dans l'espace complet du réseau, on considère le vecteur

$$v = a \otimes (Q_N)_w.$$

En utilisant la même notation que précédemment, cela correspond à

$$v = a \otimes (F_N)_u \otimes (F_N)_v,$$

où l'indice aplati w correspond au couple de fréquences spatiales (u, v).

Appliquons maintenant l'opérateur M_{skip} à ce vecteur :

$$M_{\text{skip}}v = (I_m \otimes Q_N)(L + I_{mN^2})(I_m \otimes Q_N)^H\big(a \otimes (Q_N)_w\big).$$

Comme $(Q_N)_w = Q_N e_w$, on a

$$a \otimes (Q_N)_w = (I_m \otimes Q_N)(a \otimes e_w).$$

Donc

$$M_{\text{skip}}v = (I_m \otimes Q_N)(L + I_{mN^2})(a \otimes e_w).$$

La présence du vecteur canonique e_w sélectionne les coefficients associés à la fréquence w. Sur cette fréquence, l'opérateur central se comporte comme la matrice de mélange augmentée

$$G_{\text{skip}}^{(w)} = G^{(w)} + I_m.$$

Ainsi, l'action sur les canaux et sur la fréquence spatiale se sépare :

$$\begin{aligned}(L + I_{mN^2})(a \otimes e_w) &= (G^{(w)}a + I_m a) \otimes e_w \\ &= (G_{\text{skip}}^{(w)}a) \otimes e_w.\end{aligned}$$

Comme a est un vecteur singulier droit de $G_{\text{skip}}^{(w)}$, on a

$$G_{\text{skip}}^{(w)}a = \sigma b.$$

On obtient donc

$$(L + I_{mN^2})(a \otimes e_w) = \sigma(b \otimes e_w).$$

En réinjectant ce résultat dans l'expression de $M_{\text{skip}}v$, il vient :

$$\begin{aligned}M_{\text{skip}}v &= (I_m \otimes Q_N)\big(\sigma(b \otimes e_w)\big) \\ &= \sigma\big((I_m b) \otimes (Q_N e_w)\big) \\ &= \sigma\big(b \otimes (Q_N)_w\big).\end{aligned}$$

Ainsi, l'application de M_{skip} au vecteur

$$v = a \otimes (Q_N)_w$$

donne bien un vecteur de sortie de même fréquence spatiale :

$$M_{\text{skip}}v = \sigma\big(b \otimes (Q_N)_w\big).$$

La fréquence spatiale $(Q_N)_w$ est donc conservée. Seuls l'amplitude globale, donnée par σ, et le profil de canaux, donné par b, sont modifiés.

Par conséquent, les vecteurs singuliers droits de l'opérateur avec *skip connection* restent de la forme

$$a \otimes (Q_N)_w,$$

c'est-à-dire

$$a \otimes (F_N)_u \otimes (F_N)_v.$$

□

C.4 Représentation graphique du stride

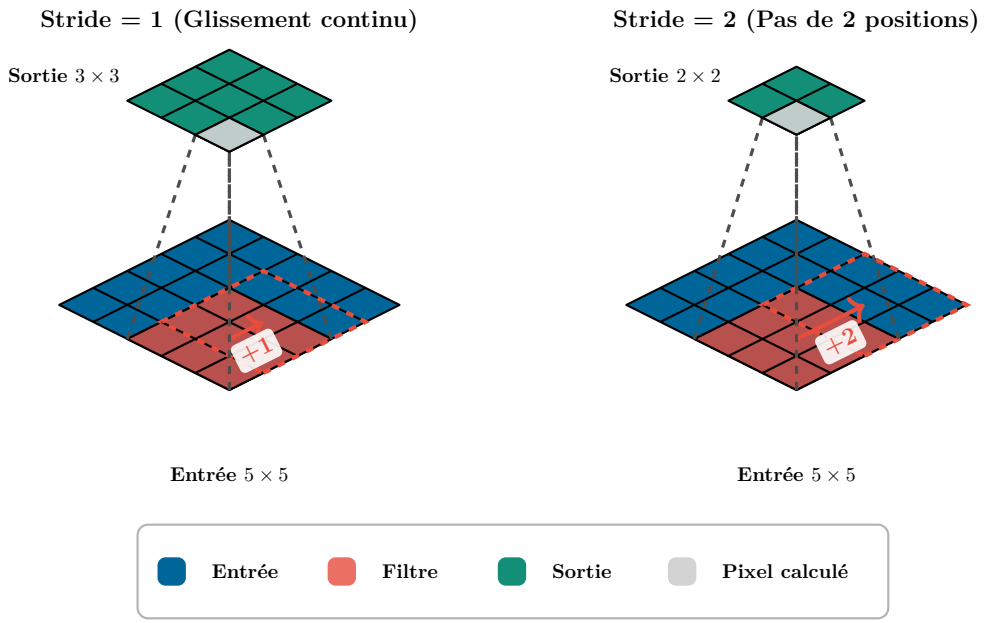


FIGURE C.2 – Effet du *stride* sur la résolution spatiale de sortie.

C.5 Illustration de l'average pooling

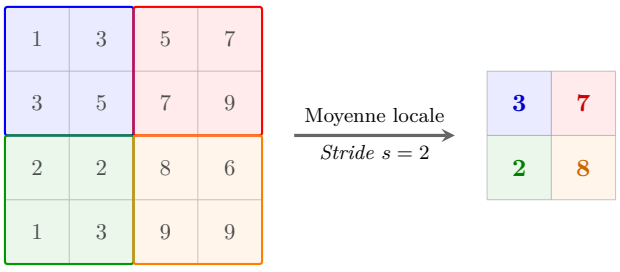


FIGURE C.3 – Illustration spatiale de l'opération d'*average pooling*. Chaque bloc de 2×2 pixels de l'image d'entrée est remplacé par sa moyenne arithmétique, réduisant ainsi la résolution spatiale par un facteur $s = 2$.

C.6 Rappel sur la SVD et les directions d'amplification

Soit

$$M \in \mathbb{C}^{m_{\text{out}}N^2 \times m_{\text{in}}N^2}$$

la matrice représentant un opérateur de convolution multi-canaux. Comme cette matrice n'est pas nécessairement carrée, on utilise la décomposition en valeurs singulières plutôt qu'une diagonalisation classique par valeurs propres.

La décomposition en valeurs singulières de M s'écrit

$$M = U\Sigma V^H,$$

où U et V sont des matrices unitaires, et où Σ est une matrice diagonale rectangulaire contenant les valeurs singulières de M :

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq 0.$$

Les colonnes de V sont appelées vecteurs singuliers droits de M : elles vivent dans l'espace d'entrée. Les colonnes de U sont appelées vecteurs singuliers gauches de M : elles vivent dans l'espace de sortie.

Pour comprendre pourquoi les vecteurs singuliers droits décrivent les directions d'amplification, on considère la matrice auto-adjointe positive $M^H M$. À partir de la SVD, on obtient

$$M^H M = (U\Sigma V^H)^H (U\Sigma V^H).$$

En utilisant $(ABC)^H = C^H B^H A^H$, il vient

$$M^H M = V\Sigma^H U^H U\Sigma V^H.$$

Comme U est unitaire, on a $U^H U = I$, donc

$$M^H M = V\Sigma^H \Sigma V^H.$$

La matrice $\Sigma^H\Sigma$ est diagonale et contient les carrés des valeurs singulières :

$$\Sigma^H\Sigma = \operatorname{diag}(\sigma_1^2, \sigma_2^2, \dots).$$

Ainsi,

$$M^H M = V \operatorname{diag}(\sigma_1^2, \sigma_2^2, \dots) V^H.$$

Cela montre que les vecteurs singuliers droits de M sont exactement les vecteurs propres de $M^H M$, et que les valeurs propres associées sont les carrés des valeurs singulières.

En particulier, si v_i désigne la i-ème colonne de V, alors

$$M^H M v_i = V D V^H v_i = V D e_i = V(\sigma_i^2 e_i) = \sigma_i^2 v_i,$$

où

$$D = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \dots).$$

Ainsi,

$$M^H M v_i = \sigma_i^2 v_i.$$

De manière équivalente, la valeur singulière σ_i mesure l'amplification induite par M dans la direction v_i. En effet,

$$\|Mv_i\|_2^2 = (Mv_i)^H (Mv_i) = v_i^H M^H M v_i = v_i^H (\sigma_i^2 v_i) = \sigma_i^2 \|v_i\|_2^2.$$

Comme V est unitaire, ses colonnes forment une base orthonormée. En particulier,

$$\|v_i\|_2 = 1.$$

On obtient donc

$$\|Mv_i\|_2 = \sigma_i.$$

La plus grande valeur singulière décrit l'amplification maximale de M parmi les entrées de norme 1 :

$$\sigma_1 = \max_{\|x\|_2=1} \|Mx\|_2.$$

Si σ_1 est simple, la direction qui réalise cette amplification maximale est le vecteur singulier droit v_1. Si σ_1 est multiple, il n'y a pas une seule direction maximale : toute direction unitaire dans le sous-espace associé à σ_1 donne la même amplification.

Dans le cadre de l'analyse fréquentielle des convolutions, les directions de forte sensibilité du réseau correspondent donc aux vecteurs singuliers droits associés aux grandes valeurs singulières.

C.7 Spectre de Fourier moyen des images propres

Cette annexe présente le spectre de Fourier moyen des images non perturbées de CIFAR-10 et CIFAR-100. L'objectif n'est pas de mesurer directement la sensibilité du modèle, mais d'observer quelles fréquences sont naturellement présentes dans les données d'entrée.

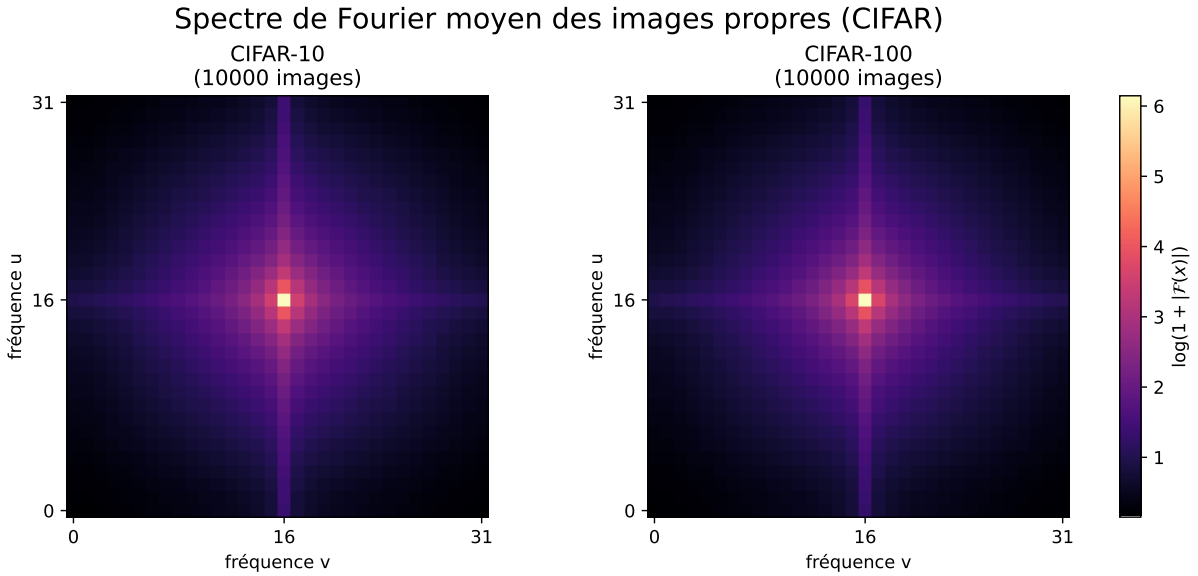


FIGURE C.4 – Spectre de Fourier moyen des images non perturbées de CIFAR-10 et CIFAR-100. Les valeurs affichées correspondent à $\log(1 + |\mathcal{F}(x)|)$, moyennées sur les images du dataset.

On observe que les deux datasets possèdent une structure fréquentielle très similaire : l'énergie est fortement concentrée autour du centre du spectre et décroît progressivement selon une forme proche d'un losange. Les lignes horizontale et verticale plus marquées indiquent également que certaines directions fréquentielles sont plus présentes que d'autres dans les images non perturbées.

Cette observation fournit une piste d'interprétation pour les heatmaps SFA. Elle montre que les images d'entrée possèdent elles-mêmes une organisation fréquentielle non uniforme. La sensibilité du modèle peut donc dépendre à la fois de l'architecture et des statistiques des données utilisées pendant l'entraînement.

Cette comparaison reste toutefois descriptive. Le spectre moyen caractérise les images d'entrée, alors que les heatmaps SFA mesurent la réponse du modèle à des perturbations imposées. Les deux objets ne doivent donc pas être confondus.

C.8 Analyse complémentaire : rôle du mélange des canaux dans la SFA

Dans le cadre théorique présenté au chapitre précédent, les directions singulières d'un modèle convolutionnel linéarisé peuvent s'écrire sous une forme du type

$$v = a \otimes q_{u,v},$$

où $q_{u,v}$ représente une composante spatiale de Fourier et où $a \in \mathbb{R}^3$ décrit le mélange entre les canaux rouge, vert et bleu.

La *Single Fourier Attack* utilisée dans le protocole principal simplifie ce cadre. Elle ne cherche pas à optimiser simultanément la fréquence spatiale et le vecteur couleur a. Le même motif de

Fourier est appliqué aux trois canaux RGB, ce qui revient à utiliser le vecteur

$$a_{\text{uni}} = (1, 1, 1).$$

L'espace de recherche est donc réduit au choix de la fréquence (u, v). Cette simplification rend l'attaque beaucoup plus simple à évaluer, puisqu'il suffit de parcourir les fréquences du domaine de Fourier. Elle est également cohérente avec la contrainte ℓ_∞ utilisée dans le protocole principal : le motif est appliqué avec l'amplitude maximale autorisée sur chacun des trois canaux.

Nous distinguons alors deux manières de contraindre le mélange entre canaux. La première conserve une contrainte comparable à la SFA standard en norme ℓ_∞, afin de tester si un autre choix de direction couleur peut améliorer l'attaque à amplitude maximale identique. La seconde utilise un budget de type ℓ_1 sur les coefficients du vecteur couleur. Dans ce cas, un vecteur comme $a = (3, 0, 0)$ est possible : le budget total est concentré sur un seul canal, ce qui ne correspond plus exactement à la contrainte de la SFA standard. Les deux variantes ne mesurent donc pas exactement la même chose ; elles servent plutôt à tester si le choix du mélange RGB peut renforcer l'effet d'une fréquence de Fourier.

Recherche de vecteurs RGB par fréquence

La figure C.5 compare la SFA standard avec des variantes où le vecteur RGB a est choisi parmi un ensemble restreint de combinaisons possibles.

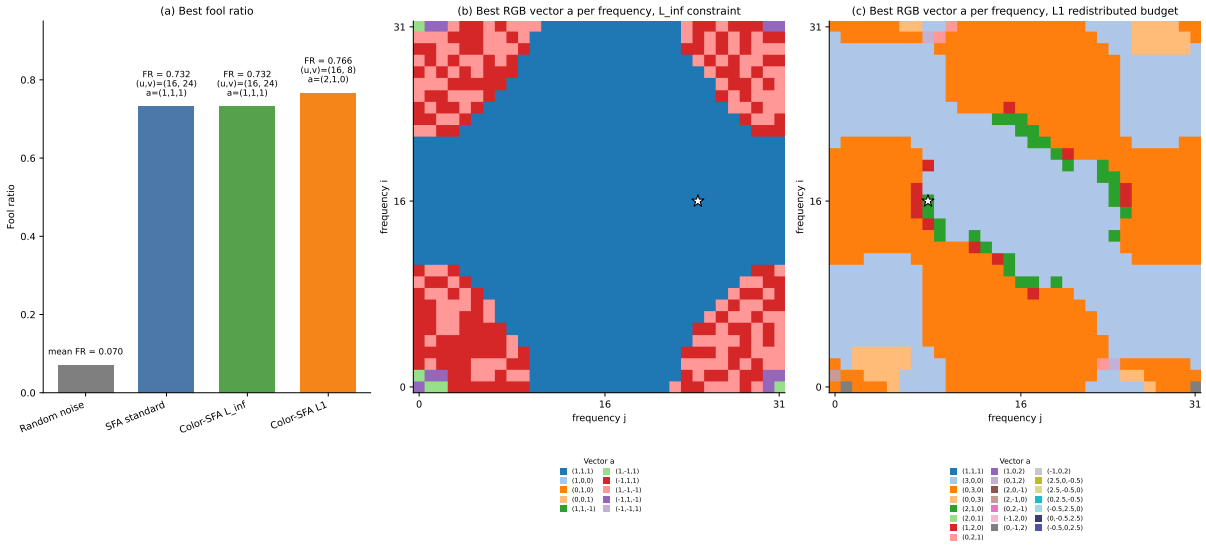


FIGURE C.5 – Comparaison entre la SFA standard et des variantes où le vecteur RGB a est choisi par fréquence. À gauche, le meilleur *fool ratio* obtenu par chaque méthode est comparé à la baseline de bruit aléatoire. Au centre et à droite, chaque case indique le meilleur vecteur RGB trouvé pour la fréquence correspondante, selon deux contraintes de budget différentes.

Dans la variante renormalisée en norme ℓ_∞, la meilleure variante colorée retrouve pratiquement le même résultat que la SFA standard. Ce qui indique que, sous cette contrainte, le choix d'un mélange RGB différent n'améliore pas significativement l'attaque par rapport à la direction uniforme $(1, 1, 1)$.

Avec la variante sous budget ℓ_1 sur les coefficients couleur, le meilleur score augmente légèrement, passant d'environ 0.732 à 0.766, avec la fréquence $(16, 8)$ et le vecteur $a = (2, 1, 0)$. Ce qui est assez modeste dans notre cas.

Comparaison de directions couleur fixes

Nous avons ensuite comparé plusieurs directions couleur simples :

$$(1, 1, 1), \quad (1, 0, 0), \quad (0, 1, 0), \quad (0, 0, 1),$$

ainsi que des directions chromatiques opposées :

$$(1, -1, 0), \quad (1, 0, -1), \quad (0, 1, -1),$$

et une direction de type luminance.

Les résultats de la figure C.6 montrent que la direction standard $(1, 1, 1)$ reste nettement la plus efficace. Les directions appliquées à un seul canal ont un effet plus faible, tandis que les directions chromatiques opposées $(1, -1, 0)$, $(1, 0, -1)$ et $(0, 1, -1)$ donnent des fool ratios presque nuls.

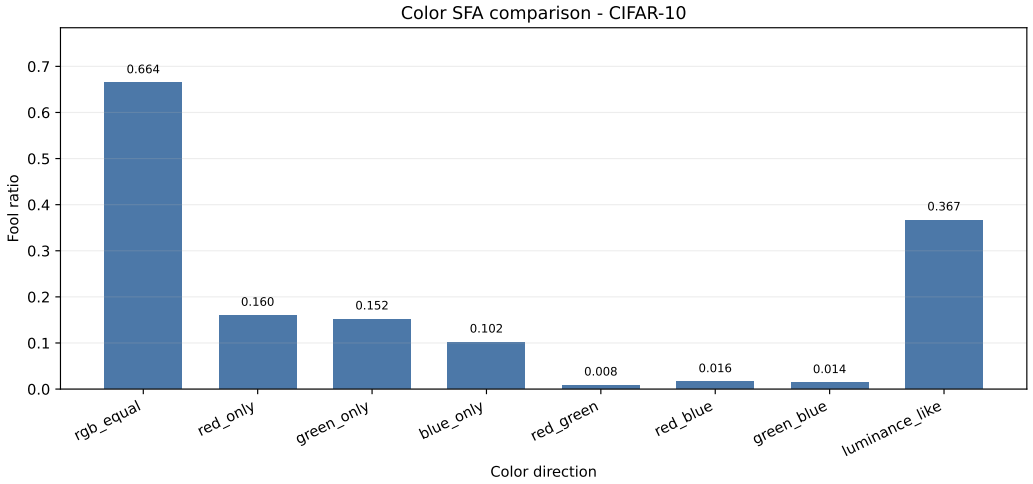


FIGURE C.6 – Comparaison de plusieurs directions couleur pour la *Single Fourier Attack* sur CIFAR-10.

Cette observation suggère que, pour la fréquence étudiée, les directions chromatiques opposées sont faiblement transmises par le réseau. Nous pouvons l'interpréter à partir du cadre multi-canaux. Pour une fréquence fixée (u, v), l'effet d'une direction couleur $a \in \mathbb{R}^3$ est décrit par la matrice de mélange $G^{(u,v)}$:

$$\beta = G^{(u,v)} a.$$

La SFA standard correspond au choix

$$a_{\text{uni}} = (1, 1, 1),$$

c'est-à-dire à une variation commune des trois canaux RGB.

Cas idéalement symétrique. Supposons que, pour la fréquence considérée, le réseau réponde de la même manière aux trois canaux d'entrée. Cela revient à supposer que les trois colonnes de $G^{(u,v)}$ sont identiques :

$$G^{(u,v)} = \begin{pmatrix} | & | & | \\ g & g & g \\ | & | & | \end{pmatrix} = g \begin{pmatrix} 1 & 1 & 1 \end{pmatrix},$$

où $g \in \mathbb{C}^{m_{\text{out}}}$ décrit la réponse fréquentielle commune du réseau.

Alors, pour toute direction couleur $a = (a_R, a_G, a_B)$, on obtient

$$G^{(u,v)} a = g(a_R + a_G + a_B).$$

L'amplification dépend donc uniquement de la somme des composantes de a. Sous une contrainte ℓ_∞ comparable à celle de la SFA standard,

$$|a_R|, |a_G|, |a_B| \leq 1,$$

cette somme est maximale en valeur absolue pour

$$a = (1, 1, 1)$$

ou pour son opposé. Ainsi, dans ce cadre symétrique simplifié, la direction uniforme a_{uni} est optimale.

Inversement, toute direction couleur vérifiant

$$a_R + a_G + a_B = 0$$

est annulée par $G^{(u,v)}$. C'est le cas des directions chromatiques opposées

$$(1, -1, 0), \qquad (1, 0, -1), \qquad (0, 1, -1).$$

Dans ce modèle simplifié, ces directions appartiennent donc au noyau de la matrice de mélange fréquentielle :

$$G^{(u,v)} a = 0.$$

Cas approximativement symétrique. Dans un réseau réel, les colonnes de $G^{(u,v)}$ ne sont pas exactement identiques. On peut cependant séparer une composante commune aux trois canaux et un terme résiduel :

$$G^{(u,v)} = g \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} + E,$$

où l'on peut prendre

$$g = \frac{1}{3} G^{(u,v)} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Le terme E mesure alors l'écart à une réponse parfaitement commune aux trois canaux.

Pour une direction chromatique opposée telle que $a_R + a_G + a_B = 0$, la composante commune disparaît :

$$G^{(u,v)} a = Ea.$$

Ainsi, si $\|E\|$ est faible, alors

$$\|G^{(u,v)} a\|_2 \leq \|E\| \, \|a\|_2.$$

Les directions chromatiques opposées ne sont donc plus annulées exactement, mais elles restent faiblement transmises lorsque le mélange fréquentiel entre les canaux est proche d'une réponse commune.

Cette proposition fournit une interprétation théorique des résultats observés dans la figure C.6. La forte efficacité de $(1, 1, 1)$ suggère que la fréquence étudiée active principalement une direction commune aux canaux, tandis que les faibles scores des directions chromatiques opposées peuvent s'expliquer par une quasi-annulation dans le mélange fréquentiel des canaux.

C.9 Heatmaps internes par couche

En complément de l'analyse d'amplification présentée dans le chapitre principal, nous visualisons ici l'effet des perturbations de Fourier à différents niveaux internes du réseau. Pour chaque fréquence, nous mesurons la variation moyenne normalisée des activations après perturbation, puis nous la comparons au *fool ratio* final.

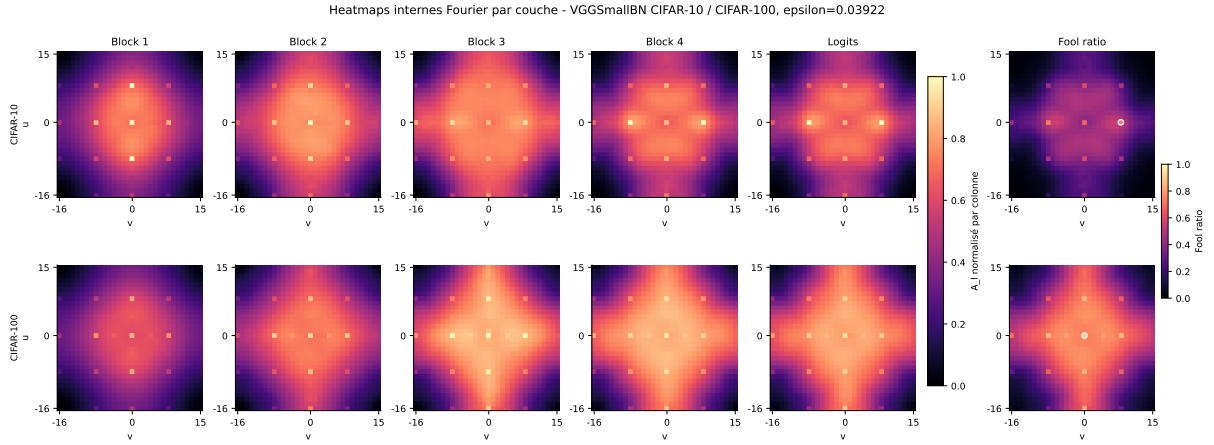


FIGURE C.7 – Heatmaps internes par couche pour `VGGSmallBN` sur CIFAR-10 et CIFAR-100. Les premières colonnes représentent la variation moyenne normalisée des activations internes après perturbation de Fourier. La dernière colonne indique le *fool ratio* final.

Ces visualisations montrent que la structure fréquentielle observée dans les heatmaps finales

apparaît déjà dans les représentations internes du réseau. Elle devient plus marquée dans les blocs profonds et dans les logits, ce qui suggère que certaines fréquences sont progressivement amplifiées au cours du traitement convolutionnel.

C.10 Analyse complémentaire des marges logits et des frontières locales

Cette annexe complète l'analyse expérimentale du chapitre 4. L'objectif est de mieux comprendre pourquoi le modèle entraîné sur CIFAR-100 apparaît plus fragile que celui entraîné sur CIFAR-10.

C.10.1 Distribution des marges logits

Pour chaque image du jeu de test, on calcule les logits produits par le modèle. On note ensuite $z_{(1)}$ le plus grand logit et $z_{(2)}$ le deuxième plus grand logit. La marge logit est définie par

$$m(x) = z_{(1)}(x) - z_{(2)}(x).$$

Cette quantité mesure basiquement l'écart entre la classe prédite et la classe concurrente la plus proche selon le modèle.

La figure C.8 compare les distributions des marges logits pour VGGSmallBN sur CIFAR-10 et CIFAR-100. On observe que la distribution de CIFAR-100 est nettement plus concentrée vers les faibles marges. La médiane est d'environ 4.15 pour CIFAR-100, contre environ 11.12 pour CIFAR-10. Cela signifie que, sur CIFAR-100, les prédictions du modèle sont en moyenne moins fortement séparées de leur seconde meilleure classe.

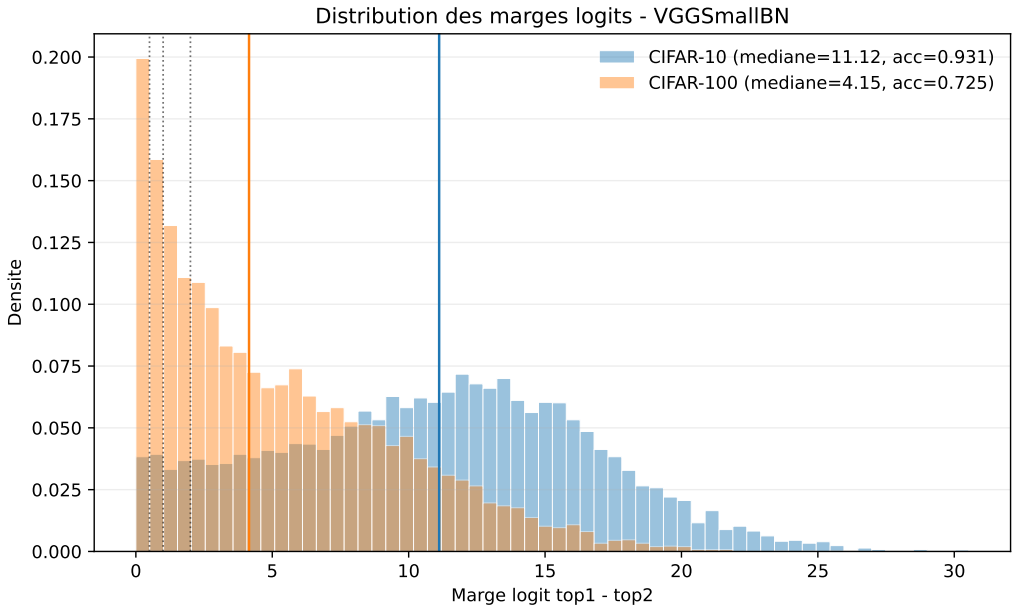


FIGURE C.8 – Distribution des marges logits pour VGGSmallBN sur CIFAR-10 et CIFAR-100. La marge est définie comme la différence entre le plus grand logit et le deuxième plus grand logit.

Ce résultat donne un appui supplémentaire à l'interprétation proposée dans le chapitre 4 : la plus grande fragilité observée sur CIFAR-100 ne vient pas seulement du choix d'une fréquence particulière, mais aussi du fait que les décisions du modèle semblent moins séparées.

Bibliographie

[1] Randall Balestriero and Richard G. Baraniuk. A spline theory of deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 374–383. PMLR, 2018.

[2] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 3 edition, 2008.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[5] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2013.

[6] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[7] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[9] Guido F. Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[10] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[11] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Pearson, 3 edition, 2009.

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[13] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[14] Yusuke Tsuzuku and Issei Sato. On the structural sensitivity of deep convolutional networks to the directions of fourier basis functions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[15] S. Voloshynovskiy. Theme 7 block 2. In *Universite de Geneve*, page 52.