Fondamentaux des réseaux de neurones

Compétences travaillées : calcul de propagation avant, interprétation des prédictions, normalisation des données, comptage de paramètres, descente de gradient, rétropropagation, fonctions d'activation, softmax et bases des convolutions.

Version web

Ex 1 — Forward pass et interprétation d'un neurone sigmoïde

g5Ka

★ ★

On considère un réseau réduit à un seul neurone sigmoïde, de paramètres $w = 2$ et $b = -1$, recevant une entrée $x = 1$. On rappelle que $\sigma(z) = \frac{1}{1+e^{-z}}$ et que $\sigma(1) \approx 0{,}73$.

1. Calculer $z = wx + b$, puis $\hat{p} = \sigma(z)$.
2. La vraie classe est $y = 0$. Le modèle se trompe-t-il ? Avec quel degré de confiance prédit-il la classe 1 ?
3. Sans calculer la perte d'entropie croisée binaire (BCE) exacte, anticiper si elle sera plutôt faible ou forte. Justifier.

Ex 2 — Effet de la normalisation sur le gradient

I54c

★ ★

On dispose de deux exemples $x^{(1)} = 1000$ et $x^{(2)} = 2000$. La normalisation standard est effectuée avec une moyenne $\mu = 1500$ et un écart-type $\sigma = 500$.

1. Calculer les valeurs normalisées $\tilde{x}^{(1)}$ et $\tilde{x}^{(2)}$.
2. Si on oublie de normaliser et qu'on initialise $w = 0{,}01$, comment le gradient $\frac{\partial J}{\partial w}$ se compare-t-il proportionnellement à celui obtenu avec les données normalisées ? Qu'est-ce que ça change en pratique ?

Ex 3 — Comptage de paramètres d'un MLP

GFrG

★

On considère un réseau de neurones entièrement connecté avec une entrée de dimension 3, une couche cachée de 4 neurones ReLU, et une sortie scalaire.

1. Combien de paramètres ce réseau contient-il au total ? Détailler par couche.
2. Si on double la taille de la couche cachée (8 neurones), combien de paramètres supplémentaires ajoute-t-on ?
3. Même question pour une image 8×8 aplatie en entrée avec une couche cachée de 32 neurones.

Ex 4 **Nombre de paramètres d'un réseau de neurones** nEkp

⋆

On considère deux réseaux de neurones ayant une seule couche cachée contenant h neurones et une unique sortie.
Le premier réseau reçoit une seule variable d'entrée : son architecture est $1 \to h \to 1$.
Le second réseau reçoit trois variables d'entrée : son architecture est $3 \to h \to 1$.
L'objectif est de comprendre comment le nombre total de paramètres dépend du nombre de variables d'entrée, lorsque le nombre de neurones cachés reste fixé.

1. Exprimer en fonction de h le nombre de paramètres du réseau $3 \to h \to 1$. Détailler par couche.
2. Exprimer en fonction de h le nombre de paramètres du réseau $1 \to h \to 1$.
3. Combien de paramètres supplémentaires le réseau $3 \to h \to 1$ possède-t-il par rapport au réseau $1 \to h \to 1$? Interpréter ce résultat.
4. Application numérique : retrouver le résultat lorsque $h = 8$.

Ex 5 **Une étape de descente de gradient sur un modèle linéaire** L9Tu

⋆ ⋆

On cherche à ajuster un modèle linéaire de la forme

$$\hat{y} = wx + b,$$

où x désigne l'entrée, $\hat{y}$ la valeur prédite par le modèle, et y la valeur cible.
On considère un seul exemple d'entraînement :

$$x = 2, \qquad y = 1.$$

Les paramètres initiaux du modèle sont

$$w = 3, \qquad b = 0,$$

et on choisit un taux d'apprentissage

$$\alpha = 0{,}1.$$

On utilise la fonction de coût quadratique moyenne

$$J = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)^2 .$$

Dans cet exercice, comme on ne considère qu'un seul exemple, on a $m = 1$.
L'objectif est de calculer à la main une étape de descente de gradient, puis d'interpréter le sens de la mise à jour obtenue.

1. Calculer la prédiction initiale $\hat{y}$.
2. Calculer le gradient du coût par rapport à w, noté $\frac{\partial J}{\partial w}$.
3. Effectuer une itération de descente de gradient pour calculer la nouvelle valeur de w.
4. Calculer la nouvelle prédiction obtenue après cette mise à jour. La prédiction se rapproche-t-elle de la cible ?
5. Sans refaire tous les calculs, expliquer pourquoi il était cohérent que la mise à jour fasse diminuer w.

6. Si l'on oubliait le facteur $\frac{1}{m}$ dans la formule du gradient, quelle serait ici la nouvelle valeur de w ? Cet oubli est-il problématique dans ce cas ?

Ex 6 **Effet du taux d'apprentissage sur la descente de gradient** MLC3

★ ★

On se place dans la situation suivante : $w = 3$, gradient $= 4$.

1. Avec $\alpha = 0{,}1$: quelle est la nouvelle valeur de w ?
2. Avec $\alpha = 1$: quelle est la nouvelle valeur de w ?
3. Avec $\alpha = 10$: quelle est la nouvelle valeur de w ? Que se passe-t-il si le gradient reste ≈ 4 à l'itération suivante ?

Ex 7 **Rétropropagation et problème du neurone mort** gkVL

★ ★ ★

On étudie, sur un exemple simple, le calcul de la propagation avant et de la rétropropagation du gradient dans un réseau de neurones de régression.
Le réseau possède :

— une seule variable d'entrée ;
— une couche cachée de 2 neurones avec fonction d'activation ReLU ;
— une seule sortie réelle.

On adopte la convention suivante : les exemples sont rangés par lignes et les poids sont multipliés à droite. Ainsi, pour un lot de données X, on a

$$Z^{[1]} = XW^{[1]} + b^{[1]}, \qquad A^{[1]} = \mathrm{ReLU}(Z^{[1]}), \qquad \hat{y} = A^{[1]}W^{[2]} + b^{[2]}.$$

Dans cet exercice, on considère un seul exemple, donc

$$X = \begin{bmatrix} 1 \end{bmatrix}, \qquad y = \begin{bmatrix} 0 \end{bmatrix}.$$

Les paramètres du réseau sont

$$W^{[1]} = \begin{bmatrix} 1 & -2 \end{bmatrix}, \quad b^{[1]} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \quad W^{[2]} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad b^{[2]} = \begin{bmatrix} 0 \end{bmatrix}.$$

On utilise la fonction de coût

$$J = \frac{1}{2}(\hat{y} - y)^2.$$

Dans la suite, le symbole $\odot$ désigne le produit terme à terme, et

$$\mathrm{ReLU}'(z) = \begin{cases} 1 & \text{si } z > 0, \\ 0 & \text{si } z < 0. \end{cases}$$

1. Calculer $Z^{[1]}$, puis $A^{[1]}$. En déduire la prédiction $\hat{y}$.
2. Calculer le coût J.

3. Calculer

$$\delta^{[2]} = \hat{y} - y, \qquad dW^{[2]} = (A^{[1]})^\top \delta^{[2]}, \qquad db^{[2]} = \delta^{[2]}.$$

4. Calculer

$$dA^{[1]} = \delta^{[2]}(W^{[2]})^\top, \qquad \delta^{[1]} = dA^{[1]} \odot \mathrm{ReLU}'(Z^{[1]}).$$

Que remarque-t-on pour le deuxième neurone caché ?

5. En déduire

$$dW^{[1]} = X^\top \delta^{[1]}, \qquad db^{[1]} = \delta^{[1]}.$$

6. Quels paramètres seront modifiés lors d'une étape de descente de gradient ?

Ex 8 — Calcul de la fonction softmax et perte entropie croisée — 01Bu

★ ★

Un réseau multiclasse produit les scores bruts $z = [3,\ 1,\ 0]$ pour 3 classes. On rappelle que $\mathrm{softmax}(z)_k = \frac{e^{z_k}}{\sum_j e^{z_j}}$, et que $e^0 = 1$, $e^1 \approx 2{,}7$, $e^3 \approx 20$.

1. Calculer approximativement $\hat{p}_1$, $\hat{p}_2$, $\hat{p}_3$.
2. Vérifier que leur somme vaut bien 1.
3. La vraie classe est 1. La perte CCE vaut $-\log(\hat{p}_1)$. Est-elle faible ou forte ?

Ex 9 — Réseau entièrement linéaire : équivalence avec un neurone simple — 4G7c

★ ★ ★

On considère un réseau $1 \to 3 \to 1$ dont toutes les activations sont linéaires (identité). Ce réseau est-il différent d'un neurone simple $1 \to 1$? Justifier. La réponse change-t-elle par rapport au cas $1 \to 1 \to 1$ avec $h = 1$?

Ex 10 — Convolution 2D à la main — mode valid — i7Xp

★

On considère l'image I et le noyau K suivants :

$$I = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \qquad K = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

On utilise le mode **valid** (pas de padding) et un stride de 1.

1. Quelle est la taille de la carte de sortie ? Justifier à l'aide de la formule $\mathrm{out} = \lfloor (\mathrm{in} + 2p -$

$k)/s\rfloor + 1$.

2. Calculer à la main tous les coefficients de la carte de sortie $S = I \star K$.
3. Que détecte intuitivement ce noyau K ?

Ex 11 Calcul de taille de sortie ⋆ pL3m

Pour chacune des configurations suivantes, on rappelle la formule :

$$\text{out} = \left\lfloor \frac{\text{in} + 2p - k}{s} \right\rfloor + 1,$$

où p désigne le padding, k la taille du noyau et s le stride.

1. Compléter le tableau suivant en calculant la taille de sortie pour chaque configuration.

#	Entrée	Noyau	Padding	Stride
a	10×10	3×3	0	1
b	10×10	3×3	1	1
c	10×10	5×5	2	1
d	16×16	3×3	0	2
e	28×28	5×5	0	1
f	32×32	3×3	1	2

2. Pour les cas b et c : que remarque-t-on sur la taille de sortie par rapport à l'entrée ? Comment appelle-t-on ce type de padding ?

Ex 12 Interprétation de filtres classiques ⋆ ⋆ kR9v

On donne les quatre noyaux suivants :

$$K_A = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad K_B = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix},$$

$$K_C = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix}, \quad K_D = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

1. Associer chaque noyau à l'un des rôles suivants : **lissage**, **détection de contours tous azimuts**, **détection de contours verticaux (Sobel X)**, **Laplacien discret**.
2. Pour K_A : calculer la somme de ses coefficients. Qu'implique ce résultat sur les zones uniformes de l'image ?
3. Pour K_B : calculer la somme de ses coefficients. Qu'implique ce résultat sur les zones uniformes ?
4. Pour K_C : si on l'applique à une image dont toutes les colonnes sont identiques, que vaut la sortie ? Pourquoi ?
5. Calculer $K_B + 9K_A$ (addition élément par élément). Interpréter le résultat.

Ex 13 — Effet du stride sur la taille et la résolution spatiale — nT2w

★ ★

On dispose d'une image 6×6 et d'un noyau moyenneur 3×3 (tous les coefficients valent $1/9$), avec padding $p = 1$.

1. Calculer la taille de sortie pour les strides $s = 1$, $s = 2$ et $s = 3$.
2. Pour stride $= 2$: combien de fois le noyau se déplace-t-il en ligne ? En colonne ?
3. Deux pixels voisins de l'image originale (distance 1) correspondent-ils toujours à deux pixels voisins dans la sortie avec stride $= 2$? Qu'est-ce que cela implique sur la résolution spatiale ?
4. On souhaite obtenir une sortie de taille 3×3 à partir d'une image 7×7, avec un noyau 3×3 et **sans padding**. Quel stride faut-il utiliser ?

Ex 14 — Noyaux séparables et gain de calcul — sB4q

★ ★ ★

Un noyau K de taille $k \times k$ est dit **séparable** s'il peut s'écrire comme le produit extérieur de deux vecteurs :

$$K = \mathbf{v} \cdot \mathbf{h}^\top,$$

où $\mathbf{v} \in \mathbb{R}^k$ est un vecteur colonne et $\mathbf{h} \in \mathbb{R}^k$ un vecteur ligne. Dans ce cas, appliquer K revient à effectuer deux convolutions 1D successives : d'abord $\mathbf{h}$ horizontalement, puis $\mathbf{v}$ verticalement. On considère le noyau gaussien 3×3 :

$$K_g = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}.$$

1. Vérifier que K_g est séparable en trouvant $\mathbf{v}$ et $\mathbf{h}$ tels que $K_g = \mathbf{v} \cdot \mathbf{h}^\top$.
2. Soit une image de taille $N \times N$ et un noyau de taille $k \times k$. Comparer le nombre de multiplications pour appliquer K_g en 2D directement versus en deux passes 1D successives. Quel est le gain ?
3. Le noyau Sobel $K_C = \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix}$ est-il séparable ? Si oui, trouver sa décomposition.

Ex 15 — Composition de convolutions et champ réceptif — cF8j

★ ★ ★

On empile deux couches de convolution successives, chacune avec un noyau 3×3, padding $= 0$ et stride $= 1$.

1. Après la première couche, quelle est la taille de la sortie si l'entrée est 7×7 ?
2. Après la deuxième couche, quelle est la taille finale ?
3. Quel est le champ réceptif d'un pixel de la sortie finale (c'est-à-dire la zone de l'image d'entrée qui a influencé sa valeur) ? Montrer le raisonnement.

4. Comparer avec une unique convolution 5×5 appliquée directement sur l'image 7×7 (padding $= 0$). Même champ réceptif ? Même taille de sortie ?
5. Combien de paramètres comporte chaque approche (sans biais) ? Quelle configuration est plus économique ?

Ex 16 **Convolution 2D complète à la main avec padding same** mH6z

★ ★ ★

On considère l'image I et le noyau K suivants :

$$I = \begin{pmatrix} 1 & 0 & 1 & 2 \\ 3 & 1 & 0 & 1 \\ 2 & 2 & 1 & 0 \\ 0 & 1 & 3 & 1 \end{pmatrix}, \qquad K = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}.$$

On applique ce filtre en mode **same** (padding $p = 1$) avec stride $= 1$.

1. Écrire l'image paddée I_p (6×6 avec des zéros sur les bords).
2. Rappeler le noyau renversé $K' = \text{flip}(K)$. Que remarque-t-on ?
3. Calculer les 4 coefficients de la ligne du milieu de la sortie, aux positions $(1, 0)$, $(1, 1)$, $(1, 2)$, $(1, 3)$ (indexation ligne/colonne à partir de 0).
4. Interpréter le résultat : ce filtre est-il symétrique ? Que détecte-t-il ?

Ex 17 **Équivariance par translation et max pooling** eT5r

★ ★ ★ ★

Soit I une image discrète et $T_d(I)$ l'image I translatée de d pixels vers la droite, définie par $[T_d(I)](i, j) = I(i, j - d)$. On note $\star$ la convolution 2D discrète :

$$(I \star K)(i, j) = \sum_{m,n} I(i - m,\, j - n)\, K(m, n).$$

1. Démontrer formellement que $T_d(I) \star K = T_d(I \star K)$.
2. Pourquoi cette propriété est-elle précieuse pour la reconnaissance d'objets dans une image ?
3. Le max pooling brise-t-il cette propriété d'équivariance ? Illustrer sur un exemple 1D avec un signal $[0, 1, 0, 3]$, un signal translaté d'un pixel $[0, 0, 1, 0]$ (bord droit ignoré), et pool size $= 2$.
4. Malgré cette rupture, pourquoi dit-on que le max pooling confère une certaine **invariance** par translation ? Distinguer équivariance et invariance.