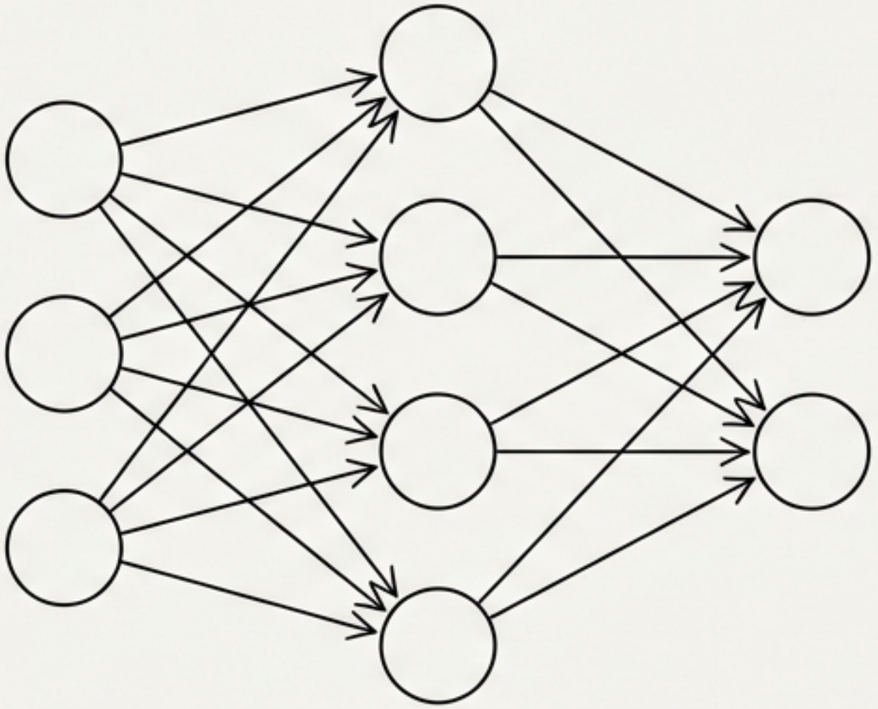
Le Calcul de la Rétropropagation



Comprendre le cœur mathématique de l'apprentissage profond.

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial w^{(1)}} \\ \frac{\partial C}{\partial b^{(1)}} \\ \vdots \\ \frac{\partial C}{\partial w^{(L)}} \end{bmatrix}$$

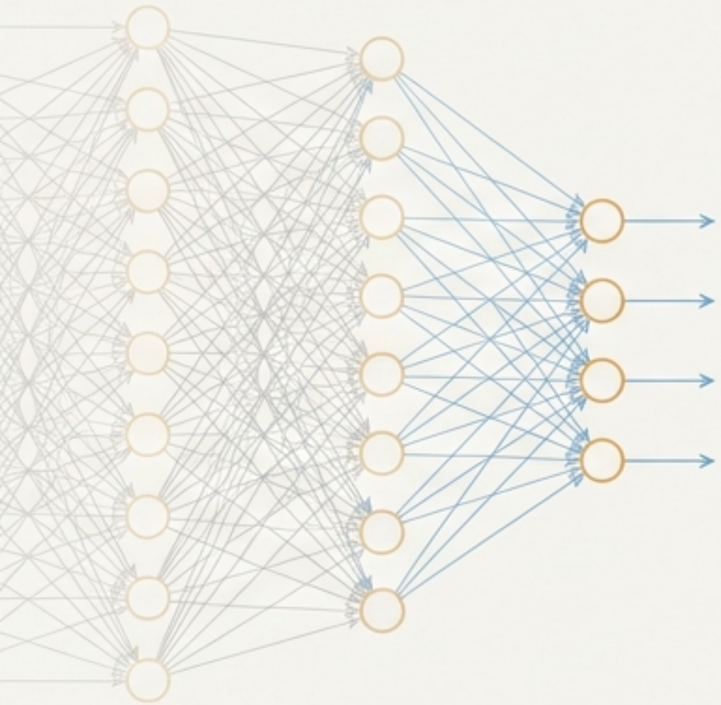
L'Objectif : Le Vecteur Gradient

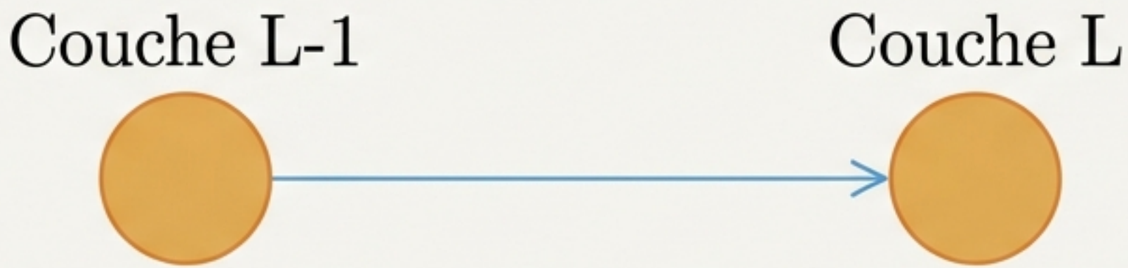
La rétropropagation est un algorithme permettant de calculer le gradient de la fonction de coût.

Ce vecteur contient toutes les modifications nécessaires (poids et biais) pour réduire l'erreur du réseau.

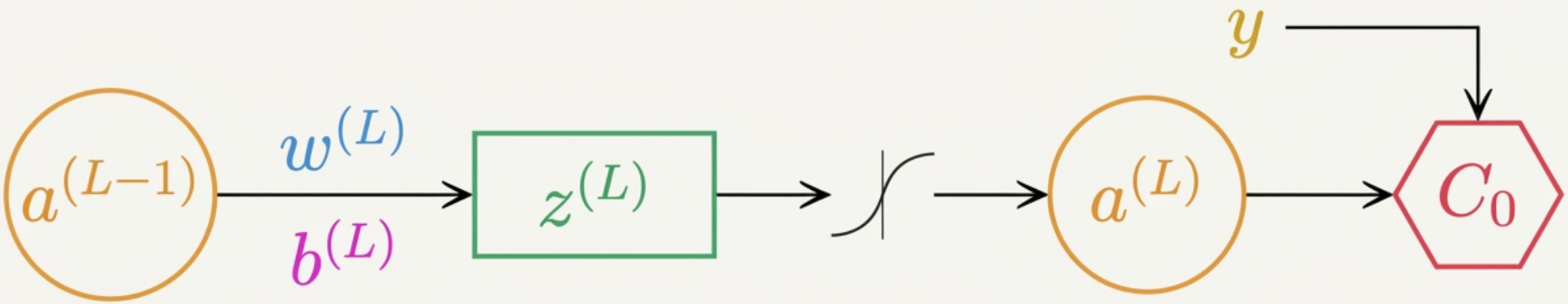
Une Simplification Stratégique

Pour comprendre la mécanique, isolons une seule ligne de neurones. Nous nous concentrons sur les deux dernières couches.





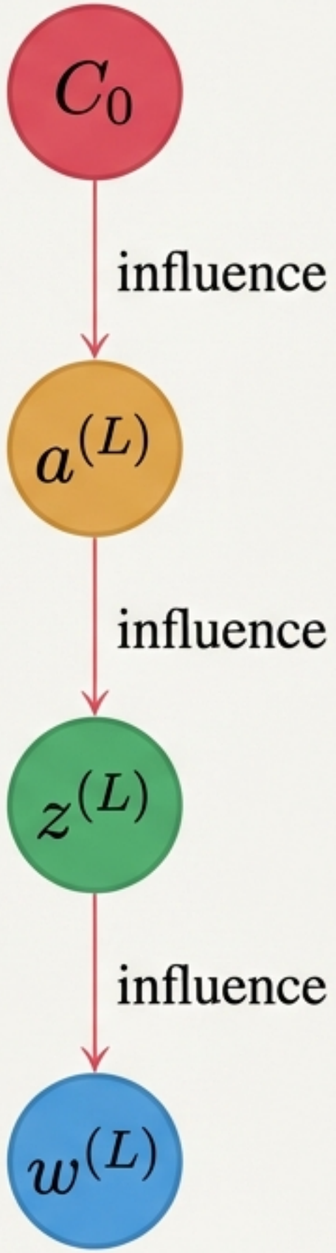
La Carte des Variables



$$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$$

$$a^{(L)} = \sigma(z^{(L)})$$

$$C_0 = (a^{(L)} - y)^2$$



L'Effet Domino (La Règle de la Chaîne)

Une petite variation de w affecte z, qui affecte a, qui affecte finalement le coût C_0.

$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdot \frac{\partial C_0}{\partial a^{(L)}}$$

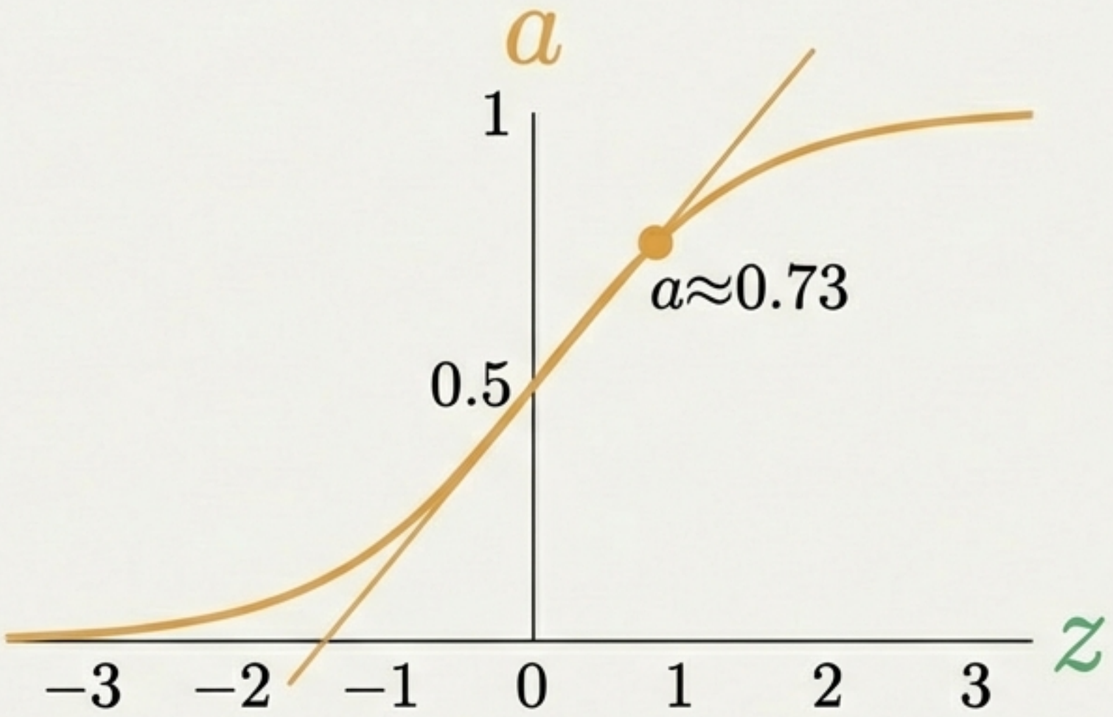
Maillon 1 : La Sensibilité du Poids

$$\frac{\partial z^{(L)}}{\partial w^{(L)}} = a^{(L-1)}$$

La dérivée de z par rapport à w est simplement l'activation précédente.

Intuition : Plus le neurone précédent est actif, plus le changement du poids impacte la somme.

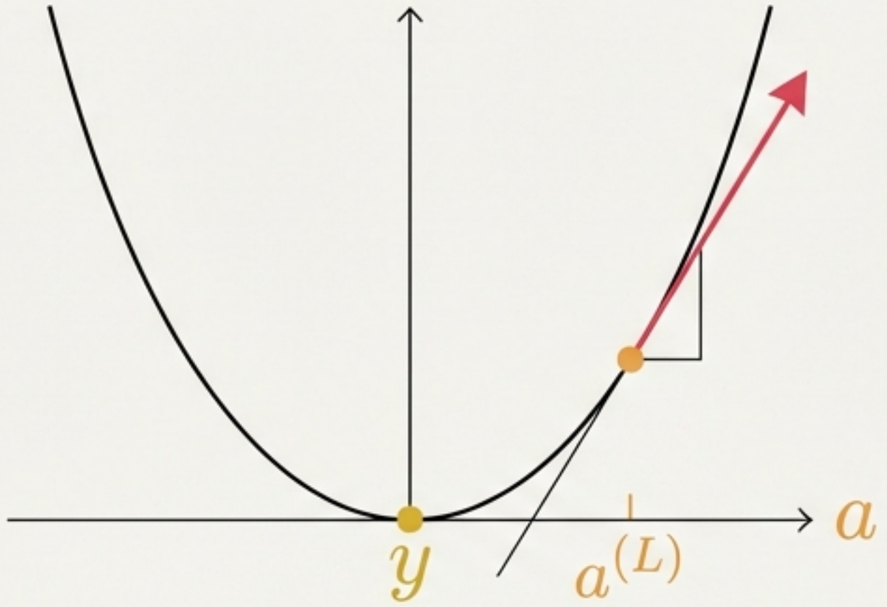
Maillon 2 : La Fonction d'Activation



$$\frac{\partial a^{(L)}}{\partial z^{(L)}} = \sigma'(z^{(L)})$$

La dérivée dépend de la pente de la fonction d'activation à la valeur z.

Maillon 3 : Le Coût Final



$$\frac{\partial C_0}{\partial a^{(L)}} = 2\left(a^{(L)} - y\right)$$

La dérivée est proportionnelle à l'erreur commise.

Si l'écart entre la sortie (a) et la cible (y) est grand, la dérivée est grande.

La Synthèse pour le Poids (w)

$$\frac{\partial C_0}{\partial w^{(L)}} = a^{(L-1)} \cdot \sigma'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

Nous avons calculé comment une variation précise de ce poids affecte le coût total pour un exemple donné.

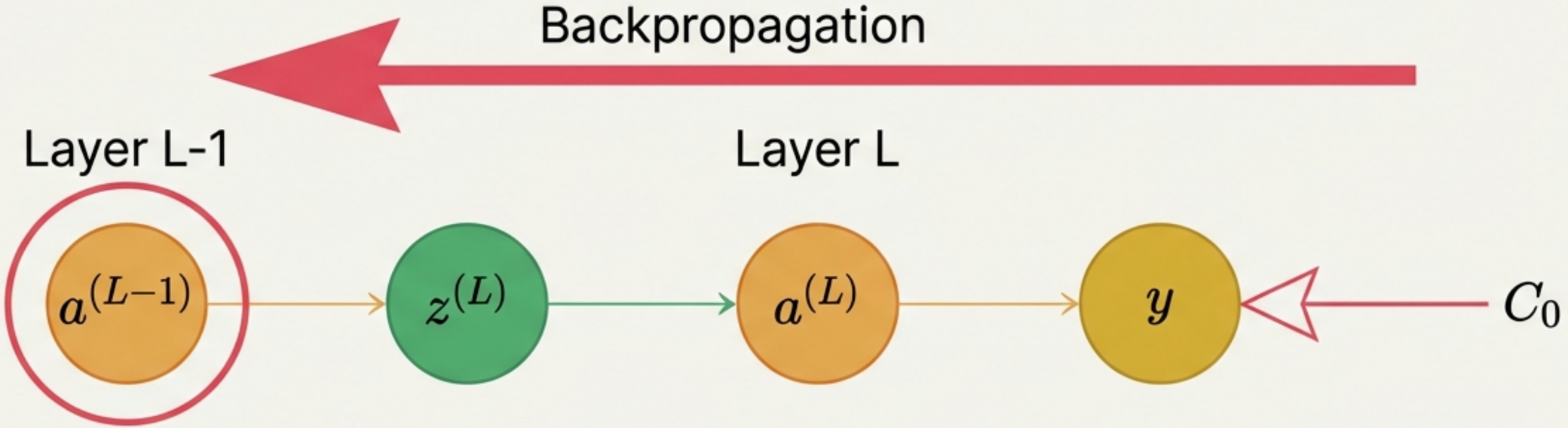
Et pour le Biais (b) ?

$$\frac{\partial C_0}{\partial w^{(L)}} = a^{(L-1)} \cdot \sigma'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

$$\frac{\partial C_0}{\partial b^{(L)}} = 1 \cdot \sigma'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

La logique est identique, sauf que la dérivée de z par rapport à b est toujours 1.

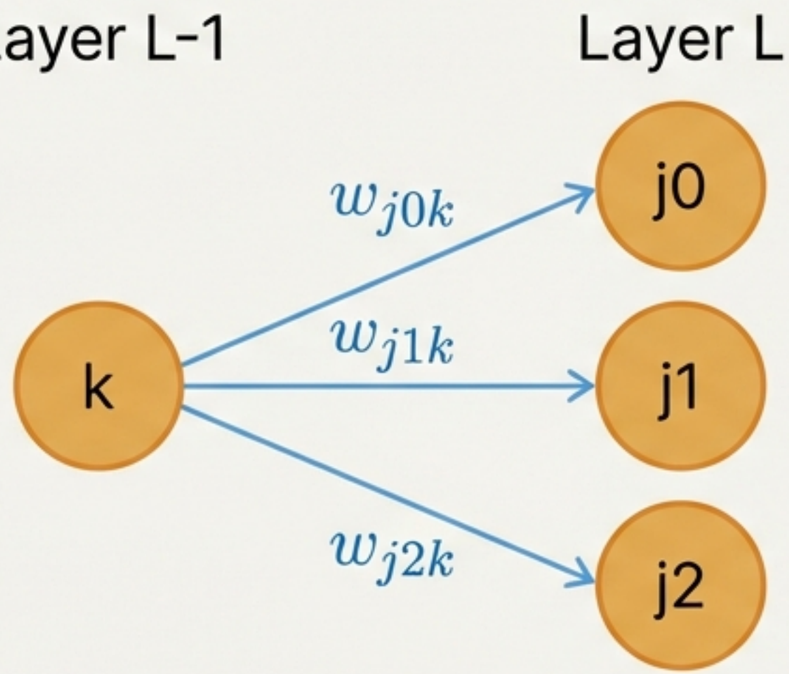
Remonter le Courant (Backpropagation)



Pour calculer les poids des couches précédentes, nous devons d'abord connaître la sensibilité du coût par rapport à l'activation précédente $a^{(L-1)}$.

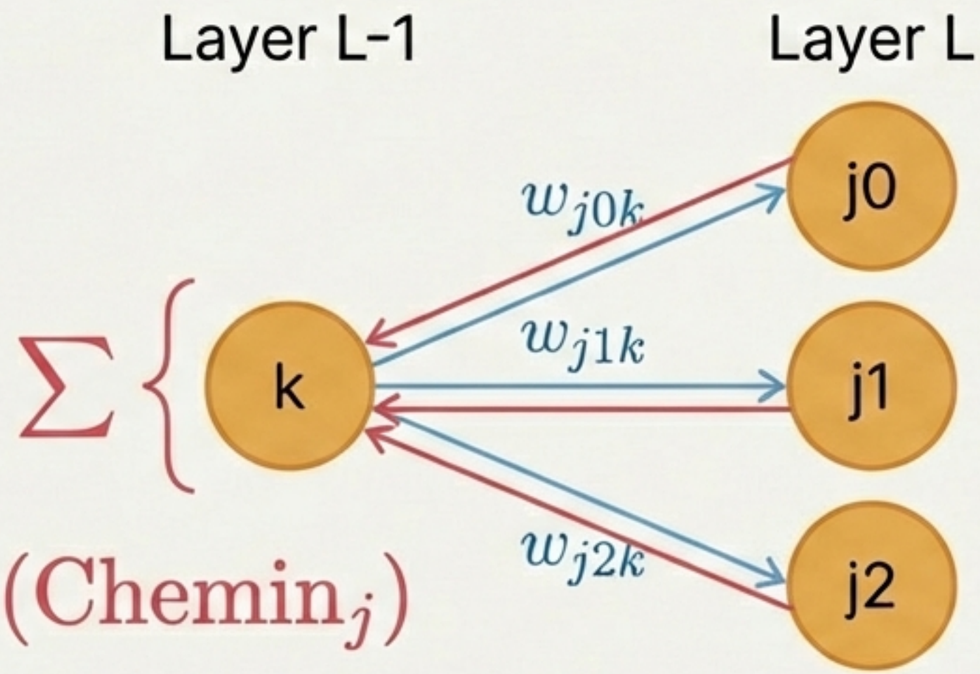
On réapplique la règle de la chaîne vers l'arrière.

La Complexité Réelle (Indices j et k)



Dans un vrai réseau, un neurone k influence plusieurs neurones de la couche suivante.

La Somme des Influences

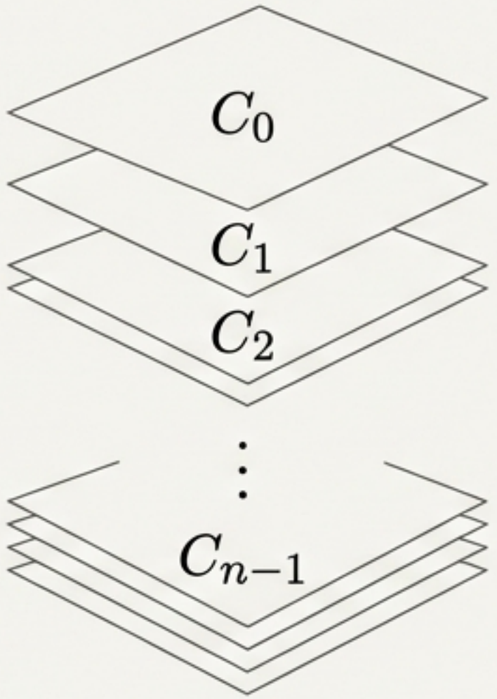


$$\sum \left\{ \right.$$

$$\frac{\partial C_0}{\partial a_k^{L-1}} = \sum_j (\text{Chemin}_j)$$

L'erreur du neurone k est la somme de toutes les erreurs qu'il a provoquées dans la couche suivante.

Du Cas Unique à l'Ensemble (Batch)



$$\frac{\partial C}{\partial w} = \frac{1}{n} \sum_{k=0}^{n-1} \left(\frac{\partial C_k}{\partial w} \right)$$

Le Gradient final est la moyenne des gradients calculés pour chaque exemple d'entraînement.

L’Intelligence n’est que du Calcul

$$\begin{bmatrix} \frac{\partial C}{\partial w} \\ \frac{\partial C}{\partial b} \end{bmatrix}$$ ✓ ✓

Ce qui semble être de ‘l’apprentissage’ est en réalité une descente de gradient, calculée étape par étape grâce à la règle de la chaîne.

Nous avons trouvé notre boussole.