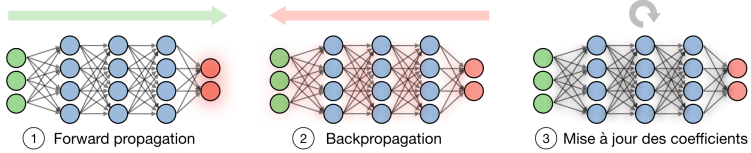


- Étape 1 : Prendre un lot de training data et effectuer une forward propagation pour calculer le loss.
- Étape 2 : Backpropaguer le loss pour obtenir le gradient du loss par rapport à chaque coefficient.
- Étape 3 : Utiliser les gradients pour mettre à jour les coefficients du réseau.



3.3 Réglage des paramètres

3.3.1 Initialisation des coefficients

□ **Initialisation de Xavier** – Au lieu de laisser les coefficients s’initialiser de manière purement aléatoire, l’initialisation de Xavier permet d’avoir des coefficients initiaux qui prennent en compte les caractéristiques uniques de l’architecture.

□ **Apprentissage par transfert** – Entraîner un modèle d’apprentissage profond requière beaucoup de données et beaucoup de temps. Il est souvent utile de profiter de coefficients pré-entraînés sur des données énormes qui ont pris des jours/semaines pour être entraînés, et profiter de cela pour notre cas. Selon la quantité de données que l’on a sous la main, voici différentes manières d’utiliser cette méthode :

Taille du training	Illustration	Explication
Petite		Gèle toutes les couches, entraîne les coefficients du softmax
Moyenne		Gèle la plupart des couches, entraîne les coefficients des dernières couches et du softmax
Grande		Entraîne les coefficients des couches et du softmax en initialisant les coefficients sur ceux qui ont été pré-entraînés

3.3.2 Optimisation de la convergence

□ **Taux d’apprentissage** – Le taux d’apprentissage (en anglais *learning rate*), souvent noté α ou η , indique la vitesse à laquelle les coefficients sont mis à jour. Il peut être fixe ou variable. La