**Idées d'Implémentation**

**Optimisation des descentes de gradient pour nouvelles connexions**

**Gradient différentiel par connexions (Adaptive Gradient Scaling)** : Modifier la magnitude des gradients en fonction de l'importance ou de l'utilité estimée de chaque connexion inhabituelle. Cela permettrait de prioriser l'apprentissage des connexions les plus prometteuses sans perturber le reste du réseau.

**Gradient séparé multi-échelle (Multi-scale Gradient Descent)** : Implémenter plusieurs canaux de rétropropagation agissant à différentes profondeurs du réseau, permettant aux connexions longues d'apprendre sur des fréquences différentes de mise à jour, selon leur position et leur portée.

**Gradient combiné avec attention (Attention-based Gradient Computation)** : Utiliser des poids d'attention calculés dynamiquement pour ajuster le gradient appliqué à chaque connexion inhabituelle. Cela permettrait au réseau de "se concentrer" sur les connexions les plus pertinentes à chaque itération.

**Gradient d'apprentissage mixte (Mixed Learning Rate Gradient)** : Définir un taux d'apprentissage distinct pour les connexions standards et les connexions inhabituelles, ajusté dynamiquement avec des heuristiques basées sur l'évolution de leur impact sur la perte.

**Filtrage de gradient (Gradient Gating)** : Introduire des portes (gating functions) contrôlant la rétropropagation vers certaines connexions longues seulement si elles contribuent de manière significative à la sortie. Cela permet de filtrer le bruit dans l’apprentissage des connexions peu influentes.

**Normalisation des gradients par topologie (Topology-aware Gradient Normalization)** : Ajuster les gradients en fonction de la distance topologique entre les neurones connectés (ex. : plus la distance est grande, plus le gradient est régularisé).

**Techniques adaptatives pour placement dynamique des connexions**

**Algorithmes évolutionnaires (Evolutionary Strategies)** : Implémenter des populations de topologies différentes, avec mutation et croisement aléatoires de connexions inhabituelles, sélectionnant les topologies les plus performantes à chaque génération.

**Apprentissage par renforcement (Reinforcement Learning Placement)** : Définir un agent RL (ex. : PPO ou DQN) dont l'action est d'activer/désactiver ou créer/supprimer une connexion, avec comme récompense la réduction de l’erreur ou l'amélioration de la généralisation du réseau.

**Placement basé sur l'importance du gradient (Gradient Importance Based Placement)** : Observer les gradients inter-neurones potentiels (même non connectés) et créer dynamiquement des connexions là où le gradient implicite est le plus élevé, révélant un fort potentiel d’apprentissage.

**Optimisation par sparsité adaptative (Adaptive Sparsity Optimization)** : Appliquer une pénalisation L1 ou entropique sur les poids des connexions inhabituelles pour forcer une sélection naturelle vers un graphe de connexions minimal mais efficace.

**Algorithmes de clustering dynamique (Dynamic Clustering Algorithms)** : Identifier en ligne les groupes de neurones fortement corrélés dans leur activation, et connecter automatiquement ces clusters via des connexions inter-groupes.

**Apprentissage basé sur l'incertitude (Uncertainty-based Connection Growth)** : Ajouter des connexions là où le réseau manifeste une forte incertitude (par ex. : par l’entropie des sorties ou la variance de prédiction entre batchs).

**Recherche par gradient structurel (Structural Gradient Search)** : Étendre la rétropropagation pour inclure un terme de gradient structurel qui suggère les connexions absentes dont l’ajout réduirait le plus la fonction de perte.

**Exploration aléatoire guidée (Guided Random Walk on Graph)** : Simuler un parcours aléatoire guidé dans le graphe de neurones pour explorer les connexions inhabituelles possibles en se basant sur la densité d’activation.

**Techniques d'évaluation de performance**

**Validation croisée adaptative (Adaptive Cross-Validation)** : Adapter dynamiquement la validation croisée pour évaluer en temps réel l'efficacité des connexions ajoutées ou retirées.

**Évaluation par métriques hybrides (Hybrid Metric Evaluation)** : Combiner des métriques classiques (précision, rappel, RMSE) avec des métriques spécifiques à la topologie du réseau pour juger les connexions proposées.

**Autres idées expérimentales**

**Auto-encoder pour pré-apprentissage (Autoencoder Pretraining)** : Utiliser un auto-encoder pour préentraîner les sous-réseaux afin de mieux déterminer les connexions inter-réseaux initiales.

**Perturbation stochastique (Stochastic Perturbation)** : Introduire des perturbations aléatoires des connexions durant l'entraînement pour tester la robustesse et l'efficacité dynamique des nouvelles connexions.