

Antoine Yang 16241010 et Jeremy Lu 16241058

1. on peut regarder que la surface de ddp connue est le plus joli (comme un *magatama*). La surface de RN est un peu comme ddp mais il est plus carré. La surface de quadratique est seulement un parabole.

Les trois discriminateurs ont différentes performances. Ddp connu est le meilleur (le taux est environ 0.93 et ne change pas). quadratique est le pire (le taux est environ 0.51 et ne change pas). mais pour RN. le taux est de plus en plus grand mais ne peut pas arriver le taux d'apprentissage). Et N_c plus grand, plus de ligne en pointillé. Le nombre de neurons est le nombre de weights. Alors γ doit correspondre le nombre de classes. Le pas du gradient plus grand, le gradient descend plus vite par pas. Le nombre de passes est plus grand, le nombre d'itérations est plus grand, le JT est plus petit, l'erreur est plus petite, le taux est plus stable.

par les expériences, on sait que P_{app} est plus grand, donc le taux d'apprentissage est plus grand, enfin environ 1, la performance est la meilleure. P_{gen} est plus grand, le taux de généralisation est plus près de le taux d'apprentissage et JT plus petit, l'erreur est plus petite, le taux de généralisation est plus stable.

Question:

Est-ce que P_{gen} doit être supérieur à P_{app} ?

Est-ce qu'il fait mieux si on ajoute une autre couche cachée.

2. la performance ne change pas.

3. le nombre est plus grand, la performance est pire qu'avant. (parce qu'on a plus de N_c)

4. il n'y a pas beaucoup de différence quand on utilise le modèle de l'équation (49). mais plusieurs paramètres sont plus nets quand on utilise le modèle de l'équation (51). et le taux de FA gaussien (environ 0.5) est beaucoup plus petit que le taux de FA gaussien amplitude inconnue (environ 0.9), et l'écart type est plus grand. parce que A change le paramètre et il est peut être plus net ou moins net avec un bruit blanc gaussien, donc il faut le discriminateur de FA gaussien (amplitude inconnue) qui a la meilleure performance.

5. on peut découvrir que quand on utilise la même base pour l'apprentissage et la généralisation. Le taux de généralisation est plus grand (environ 0.9), plus proche de le taux d'apprentissage (environ 1). la performance est pas mal. le nombre de chiffres est plus petit, les erreurs sont plus petites, le taux d'apprentissage plus tôt arrive un régime stable. Le taux de généralisation est plus proche de le taux d'apprentissage

que quand on utilise différentes bases pour l'apprentissage et la généralisation. Le taux de généralisation est plus petit (environ 0.5), plus loin de le taux d'apprentissage (environ 1).

La performance est pire..

En fait, pour améliorer les performances du réseau de neurones, on peut:

1. assurer qu'on utilise la même base pour l'apprentissage et la généralisation.

2. ajouter P_{app}

3. assurer les classes est assez petit, N_c est pratique

4. ajouter P_{gen}