



Le génie pour l'industrie

Laboratoire 3 ELE-767

Apprentissage machine en intelligence artificielle

**Conception et entraînement d'un réseau
du type LVQ pour la classification de formes.**

**Saddat Mohammad, Caverio-Linares Gabriella et
Frija-Altarc Liam**

**MOHS27119206
CAVG08599602
FRIL04049603**

**Rapport Présenté à
Chakib Tadj**

31/03/2019

Introduction	3
Étapes de développement	3
Contraintes complémentaires ajoutées	4
Contrainte complémentaire 1: Interface graphique	4
Contrainte complémentaire 2: nombre de prototypes par classe	4
Contrainte complémentaire 3: taux d'apprentissage adaptatif	4
Amélioration	5
Différentes simulations effectuées	5
Évolution de la performance	7
a) L'apprentissage	7
b) La validation croisée	7
c) La généralisation	8
Comparaison avec MLP	8
Conclusions	8
Annexe	9

Introduction

Lors de ce laboratoire, nous avons fait une application, à l'aide du langage de programmation Python, permettant d'identifier 10 chiffres à travers des vecteurs de signal de paroles. Afin d'atteindre notre but, nous avons construit un réseau de neurones compétitif du type LVQ.

Étapes de développement

1. Prérequis :

La première étape de développement a été de lister tous les prérequis, les concepts à utiliser et les contraintes du laboratoire.

2. Choix de Structure:

Ensuite, nous avons séparé chaque étape de l'algorithme en fonctions qui seront implémentées dans le code .

3. Apprentissage, Validation croisée et test de généralisation:

Nous avons intégré une limite de temps pour l'apprentissage. Lors de l'implémentation de la validation croisée, nous avons intégré une fonction de calcul de performance afin de confirmer le bon fonctionnement de notre réseau. Le calcul de performance nous permettra de faire le choix de la meilleure configuration de réseau de neurone. Le code relié à la généralisation reste le même que celui du laboratoire précédent.

4. Interface:

La dernière étape a été de créer une interface GUI permettant à l'utilisateur de choisir les caractéristiques principales de son réseau de neurone sans s'aventurer dans le code.

Contraintes complémentaires ajoutées

Contrainte complémentaire 1: Interface graphique

Nous avons implémenté une interface graphique sur web. Elle nous permet de sélectionner les fichiers d'entraînement, de tests généralisés, de validations et de configurations. Cette interface nous permet de choisir le nombre de prototypes par classe, choisir le set de bases de données voulue, choisir le taux d'apprentissage, choisir le taux d'apprentissage adaptatif, choisir le temps d'apprentissage et ajouter du bruit à nos bases de données. Nous pouvons aussi visualiser la performance de notre LVQ pendant l'apprentissage et la validation croisée dans un graphique où l'on voit la performance selon le nombre d'époque.

Lab 3: LVQ par G. Cavero, L. Frijia, S. Mohammed



Figure 1. Interface web

Contrainte complémentaire 2: nombre de prototypes par classe

Nous permettons à l'utilisateur de changer le nombre de prototypes par classe. Dans notre implémentation, on fait référence à ce nombre avec la constante K.

Contrainte complémentaire 3: taux d'apprentissage adaptatif

Lors de l'apprentissage, nous avons implémenté le "adaptive learning rate". Cette méthode consiste à faire diminuer le facteur de performance après chaque époque. Les prototypes vont donc avoir de grandes variations lors des premières époques, mais les variations diminuent linéairement. Il y aura donc une diminution de la vitesse d'apprentissage. Un réseau a besoin d'apprendre rapidement au début, mais, plus il se rapproche de sa version optimal moins il a besoin d'apprendre. Cette méthode permet d'avoir une meilleure performance.

Amélioration

Nous avons l'option de rajouter du bruit à nos données. Cela nous permet, donc, de doubler le nombre de données dédiées à l'apprentissage, avoir un meilleur apprentissage et ainsi avoir une meilleure performance.

Différentes simulations effectuées

Nombre d'entrées:

Nous avons simulé avec 40, 50 et 60 sets de base de données. Nous avons comparé la performance lors de la validation croisée et constaté que la performance est meilleure avec un set de 40 données. Voici le tableau résumant les performances obtenues.

Tableau 1. Comparaison du nombre d'entrée

Éta	Set de donnée	# d'époques	# de K	Éta adaptatif	Ajout de bruit	Performance VC
0.1	40	5	5	Non	Non	77%
0.1	50	5	5	Non	Non	68%
0.1	60	5	5	Non	Non	73%

Nombre de représentant par classe:

Nous avons simulé de 5 à 100 représentants(K) par classe. Nous avons comparé la performance lors de la validation croisée et constaté que la performance est meilleure avec 20 représentants par classe. Voici le tableau résumant les performances obtenues.

Tableau 2. Comparaison du nombre de représentant par classe

Éta	Set de donnée	# d'époques	# de K	Éta adaptatif	Ajout de bruit	Performance VC
0.1	40	5	5	Non	Non	77%
0.1	40	5	10	Non	Non	73%
0.1	40	5	20	Non	Non	80%
0.1	40	5	50	Non	Non	77%
0.1	40	5	75	Non	Non	75%
0.1	40	5	100	Non	Non	73%

Nombre d'époques et temps d'apprentissage:

Nous avons simulé de 1 à 20 époques. Nous avons comparé la performance lors de la validation croisée et constaté que la performance est meilleure avec 10 époques. Voici le tableau résumant les performances obtenues.

Tableau 3. Comparaison de performance en fonction du nombre d'époques

Éta	Set de donnée	# d'époques	# de K	Éta adaptatif	Ajout de bruit	Performance VC
0.1	40	1	20	Non	Non	71%
0.1	40	2	20	Non	Non	78%
0.1	40	5	20	Non	Non	73%
0.1	40	10	20	Non	Non	83%
0.1	40	15	20	Non	Non	78%
0.1	40	20	20	Non	Non	76%

Taux d'apprentissage adaptatif:

Nous avons simulé avec et sans taux d'apprentissage adaptatif. Nous avons comparé la performance lors de la validation croisée et constaté que la performance est meilleur avec taux de performance. Voici le tableau résumant les performances obtenues.

Tableau 4. Comparaison de performance en fonction du taux d'apprentissage adaptatif

Éta	Set de donnée	# d'époques	# de K	Éta adaptatif	Ajout de bruit	Performance VC
0.1	40	10	20	Oui	Non	85%
0.1	40	10	20	Non	Non	83%

Taux d'apprentissage:

Nous avons simulé le taux d'apprentissage de 0.05 à 0.2. Nous avons comparé la performance lors de la validation croisée et constaté que la performance est meilleur avec le taux d'apprentissage à 0.1. Voici le tableau résumant les performances obtenues.

Tableau 5. Comparaison de performance en fonction du taux d'apprentissage

Éta	Set de donnée	# d'époques	# de K	Éta adaptatif	Ajout de bruit	Performance VC
0.1	40	10	20	Non	Non	83%
0.2	40	10	20	Non	Non	74%
0.05	40	10	20	Non	Non	80%

Ajout du bruit:

Nous avons simulé avec et sans l'ajout du bruit. Nous avons comparé la performance lors de la validation croisée et constaté que la performance reste la même avec ou sans ajout de bruit. Nous avons décidé de garder la configuration avec ajout de bruit puisqu'elle double le nombre de données d'apprentissage. Cela permet un meilleur apprentissage et on ne risque pas de tomber dans le surapprentissage, puisque la performance est à 85%.

Tableau 6. Comparaison de performance en fonction de l'ajout du bruit

Éta	Set de donnée	# d'époques	# de K	Éta adaptatif	Ajout de bruit	Performance VC
0.1	40	10	20	Oui	Oui	85%
0.1	40	10	20	Oui	Non	85%

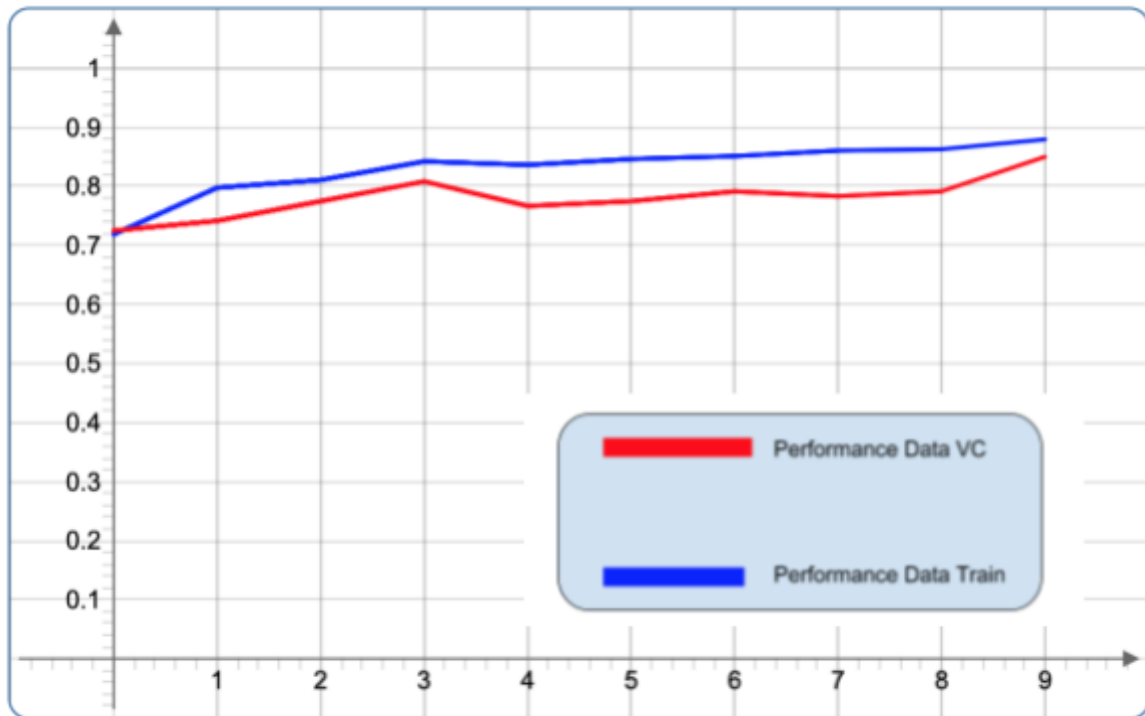


Figure 2. Graphique de la meilleure configuration

Si l'on récapitule le réseau de neurones le plus performant à un taux d'apprentissage de 0,1 adaptatif, 10 époques, 20 représentants pour chaque classe, 40 entrées et est entraîné par la base de donnée d'entraînement à laquelle nous avons ajouté les données d'entraînement bruitées.

Évolution de la performance

a) L'apprentissage

Analyse de la courbe de performance d'apprentissage:

Nous constatons que la courbe de performance d'apprentissage commence à 70% ,ce qui est déjà assez haut ,pour ensuite finir à 98%. L'erreur est donc de 2%, ce qui est normal lors de l'apprentissage. Le fait que la performance d'apprentissage est aussi haut depuis le début s'explique par le fait que les signaux de voix de chaque classe ce ressemble. Nous pouvons constater une certaine tendance logarithmique, mais pas autant qu'avec le MLP. Dans la **figure 2** , on constate qu'il y a de grandes variation de la performance d'apprentissage au début de l'entraînement , puis qu'elles diminuent plus le nombre d'époques augmente. Cela est dû au taux d'apprentissage qui est adaptatif.

b) La validation croisée

Analyse de la courbe de performance VC:

Dans notre implémentation, nous faisons la validation croisée après chaque époque, ce qui explique l'augmentation de la performance de VC. Plus le réseau apprend, plus il a une performance élevée lors de la VC et plus l'erreur diminue. On constate dans la **figure 2** qu'elle passe de 70% à 85%, ce qui laisse une erreur de 15%. La performance VC de début est haute tout comme celle de l'apprentissage. La performance VC varie drastiquement entre chaque époque et semble moins stable qu'avec un réseau MLP.

c) La généralisation

Analyse de performance du test de généralisation:

Lors du test de généralisation on obtient une performance de 78%, ce qui est une performance acceptable.

Comparaison avec MLP

Nous constatons que le LVQ performe mieux que le MLP lors de la validation croisée. La meilleure performance de notre MLP est de 80% tandis qu'avec notre LVQ, on est capable d'aller chercher jusqu'à 85%. Le nombre de calculs arithmétiques, la complexité des calculs et d'organisation des données est moindre pour un réseau LVQ. Ce fait permet de diminuer le temps nécessaire au développement de ce type de réseau qui en plus d'être plus simple est plus performant. Par contre pour le LVQ, le temps de calcul pour chaque époque augmente. Avec le MLP, chaque époque se déroule en environ 5 secondes, cependant avec le LVQ elle prend environ 10 secondes par époque avec 20 représentants par classe. La durée d'exécution augmente encore plus lorsqu'on augmente le nombre de représentants par classe. On constate aussi que la performance d'apprentissage et de validation croisée sont très hautes au départ de l'entraînement. Selon nous, cela permet de dire que le réseau LVQ se rapproche plus au comportement de l'être humain qu'un réseau MLP.

Conclusions

En conclusion, à travers ce laboratoire nous avons réalisé qu'un réseau de neurones compétitifs LVQ est très simple en terme de nombre de calculs et est très performant. Il imite mieux le comportement humain qu'un réseau MLP.

Annexe

Fichier contenant les représentants : lvqbest.txt