ÉCOLE NATIONALE SUPÉRIEURE DE TECHNIQUES AVANCÉES

ROB311 : Apprentissage pour la robotique

Parcours robotique

Année universitaire : 2020/2021

TP4: Support Vector Machines for Digit Recognition

Auteurs :

M. AHMED YASSINE HAMMAMI

MME. HANIN HAMDI

Objectifs

L'objectif de ce TP est d'utiliser l'algorithme SVM pour la reconnaissance des chiffres digitaux écrits à la main. Pour ce faire, on procède initialement à la remise en forme des données utilisées afin de réduire leurs dimensions et accélérer par la suite l'apprentissage. Ensuite, on procède à l'apprentissage automatique en utilisant deux modèles du SVM ; linéaire et non linéaire. Pour évaluer les performances de l'apprentissage, on se base sur les résultats des matrices de confusion et de la précision.

Exploration des données

Les données sont représentées sous la forme de 60000 lignes (pour le trainDataset) et 785 colonnes. Chaque ligne représente une image du dataSet. la première colonne correspond à la classe à laquelle appartient l'image (le digit qu'elle représente de 0 à 9) et le reste des cases représentent la matrice 28*28 pixels qui modélise l'image en niveau de gris et contiennent l'intensité de la couleur de chaque pixel. Enfin, on peut dire qu'il s'agit d'un problème de classification selon 10 classes.

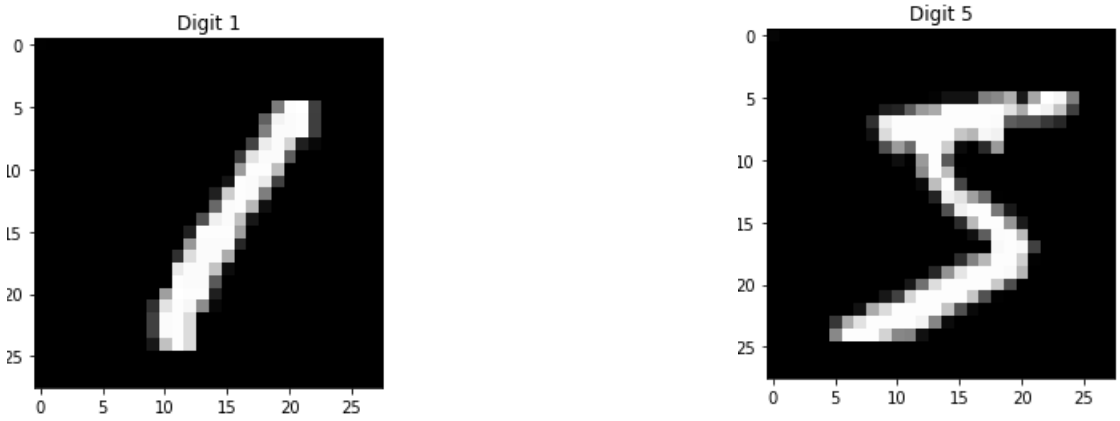

FIGURE 1 – Les données utilisées

Démarche à suivre

Pour aboutir à l'objectif du TP on suit la démarche suivante :

- Standardisation des données.
- Réduction de la dimension par PCA.
- Implémentation de l'algorithme SVM.

Standardisation des données

Étant donné que le PCA produit un sous-espace de caractéristiques (dans notre cas les pixels de chaque image) qui maximise la variance le long des axes, il est logique de normaliser les données, en particulier si elles ont été mesurées à différentes échelles. Ainsi, la standardisation (ou la normalisation) consiste à redimensionner les caractéristiques de sorte qu'elles aient les propriétés d'une distribution normale standard avec une espérance nulle et un écart-type unitaire.

Réduction de la dimension par PCA

La taille des données du test et de l'apprentissage est très grande ce qui ne rend l'apprentissage très lent voire dépassant les capacités matérielles de nos machines utilisées pour faire l'apprentissage. Ainsi, on utilise le PCA pour réduire les dimensions des données et donc pour augmenter le speed-up de l'apprentissage.

Implémentation de l'algorithme SVM

- Modèle linéaire

On a utilisé le modèle suivant :

```
model_linear = SVC(kernel='linear')
model_linear.fit(X_train_pca, Y_train)
```

FIGURE 2 – SVM, modèle linéaire

On a obtenu les valeurs suivantes pour la matrice de confusion et la précision :

```
LINEAR SVM MODEL

CONFUSION MATRIX =

[[ 964    0    3    4    1    4    2    1    1    0]
 [   0 1122    5    2    0    1    2    1    2    0]
 [   8    4  971   11    6    0    8    9   12    3]
 [   4    1   12  952    1   15    2    6   16    1]
 [   2    0    9    2  942    0    4    4    3   16]
 [   6    4    4   34    6  800   11    0   23    4]
 [  10    3   13    1    6   12  912    0    1    0]
 [   1    4   24   10    6    0    0  962    2   19]
 [   7    5    9   19    8   22    8    8  886    2]
 [   6    8    2   11   26    3    0   21    5  927]]
```

FIGURE 3 – SVM, modèle linéaire : Matrice de confusion

```
LINEAR SVM MODEL

ACCURACY =
0.9438
```

FIGURE 4 – SVM, modèle linéaire : La précision

- Modèle non-linéaire

On a utilisé le modèle suivant :

```
# model
non_linear_model = SVC(kernel='rbf')
# fit
non_linear_model.fit(X_train_pca, Y_train)
```

FIGURE 5 – SVM, modèle non-linéaire

On a obtenu les valeurs suivantes pour la matrice de confusion et la précision :

```
NON-LINEAR SVM MODEL

CONFUSION MATRIX =

[[ 968    0    2    1    0    3    3    1    2    0]
 [   0 1127    3    0    0    1    2    0    2    0]
 [   6    1  995    2    2    0    1   14   10    1]
 [   0    0    3  985    1    6    0    9    6    0]
 [   0    0    5    0  951    2    5    7    3    9]
 [   2    0    1   10    0  862    5    5    6    1]
 [   5    2    1    0    4    8  932    2    4    0]
 [   0    8   14    1    1    1    0  989    1   13]
 [   3    0    3    6    5   11    2   11  929    4]
 [   4    6    6    7   12    2    0   14    4  954]]
```

FIGURE 6 – SVM, modèle non-linéaire : Matrice de confusion

```
NON-LINEAR SVM MODEL

ACCURACY =
0.9692
```

FIGURE 7 – SVM, modèle non-linéaire : La précision