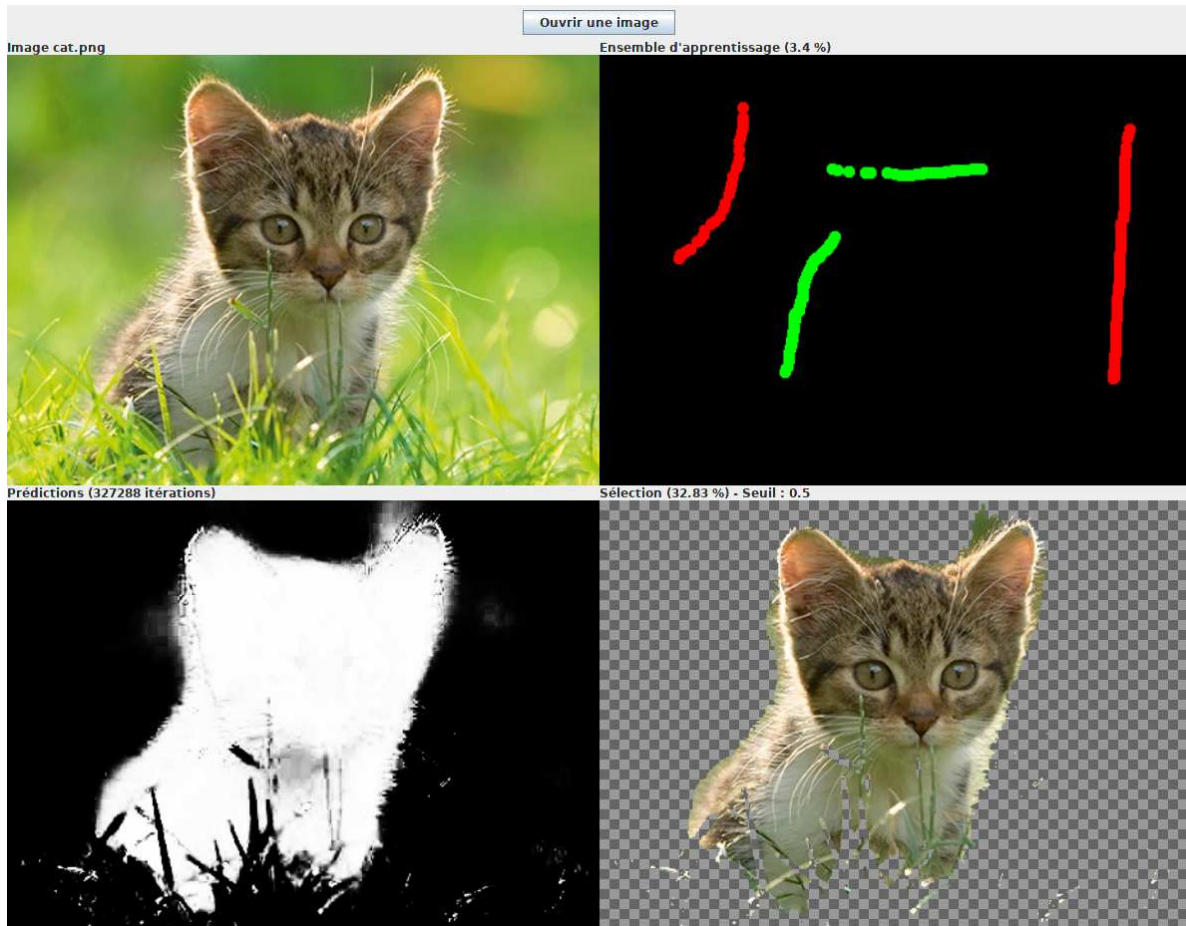


Intelligence artificielle

Master 1 Informatique

TP 2

Classification dans une image



On souhaite finaliser un programme permettant de détourer un objet dans une image. L'interface graphique proposée se compose de quatre cadrans.

Dans le cadran en-haut à gauche se trouve l'image traitée. Dans celui en-haut à droite est affiché l'ensemble d'apprentissage : vert pour les exemples positifs, rouge pour les exemples négatifs et noir pour les pixels qui ne font pas partie de l'ensemble d'apprentissage. Dans le cadran en-bas à gauche est affichée une image en niveaux de gris qui représente les degrés d'appartenance des pixels à la classe positive. Enfin, en-bas à droite se trouve l'image de l'objet détouré selon un seuil sur le degré d'appartenance (fixé initialement à 0,5).

L'interaction avec le programme se fait via la souris. Le clic gauche ajoute des pixels positifs à l'ensemble d'apprentissage alors que le clic droit ajoute des pixels négatifs. Le clic du milieu retire des pixels de l'ensemble d'apprentissage. La molette permet de modifier le seuil sur le degré d'appartenance.

Chaque fois que l'ensemble d'apprentissage est modifié, l'algorithme d'apprentissage est exécuté pendant 500 ms, puis l'affichage est mis à jour. Les données traitées sont les pixels de l'image, représentés par trois attributs numériques (rouge, vert et bleu) entre 0 et 1.

Perceptron multicouche

On souhaite utiliser le Perceptron multicouche comme méthode de classification supervisée, dans le *package* `ImageClassification.class`. La classe abstraite `Neuron` est capable de définir son niveau d'activation et de le retourner. Les classes qui étendent la classe `Neuron` sont :

- les neurones de la couche d'entrée, qui définisse leur activation selon la valeur d'un attribut ;
- les neurones des couches cachées et de la couche de sortie, qui calculent leur activation selon la somme pondérée des entrées ;
- le neurone de biais (singleton), dont l'activation est toujours 1.

La classe `MultiLayerPerceptron` étend la classe abstraite `Classification`. La classe est actuellement incomplète. Le constructeur `public MultiLayerPerceptron (int nbInputs, int nbHidden, int nbNeurons)` doit créer le réseau de neurone. La méthode `protected void initialization ()` peut rester vide. La méthode `public double predict (double [] object)` retourne le degré d'appartenance estimé d'une observation en activant le réseau. La méthode `public double learn ()` réalise l'apprentissage du réseau en utilisant la rétropropagation.

Activation

Écrire la méthode `private void activate (double [] object)` qui calcul le niveau d'activation de tous les neurones dans le réseau en fonction d'un object en entrée.

Rétropropagation

Écrire la méthode `private void backpropagation (double [] object, int label)` qui réalise **une** itération de l'algorithme de rétropropagation du gradient.

Chat

Appliquer le programme sur l'image `cat.png` pour détourer le chat. Que se passe-t-il si on ne rentre que des exemples positifs ?

Maison bleue

Appliquer le programme sur l'image `blue_house.png` pour détourer la maison. Comment peut-on améliorer le résultat ?