UNIVERSITÉ CÔTE D'AZUR

Bases de l'IA

Traitement d'image (cont.)

Elena CABRIO

elena.cabrio@univ-cotedazur.fr

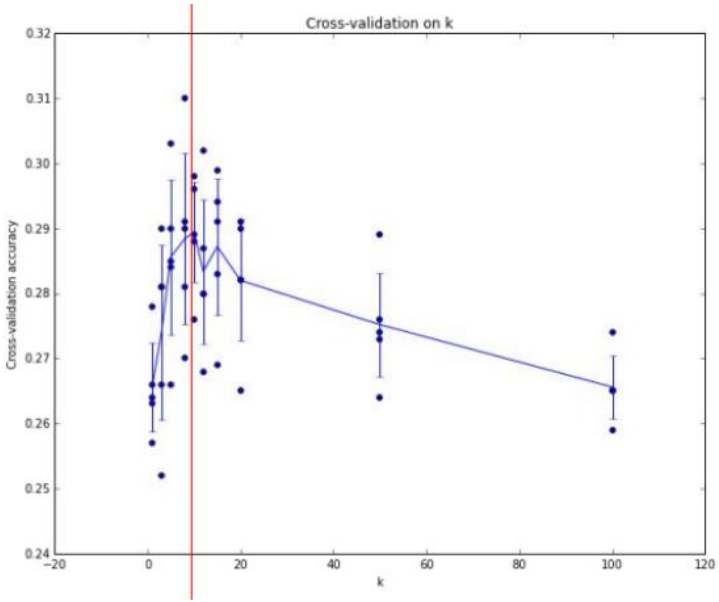
UNIVERSITÉ CÔTE D'AZUR

Plan pour cette séance

- Classification des images
  - K-plus proches voisins (cont.)
  - Classificateurs linéaires

Slides from Fei-Fei Li, Justin Johnson, Serena Yeung http://vision.stanford.edu/teaching/cs231n/

k plus proches voisins (cont.)



- Exemple de validation croisée 5-folder pour la valeur de K.
- Chaque point : résultat unique.
- La ligne passe par la moyenne, les barres indiquent l'écart-type.
- Il semble que k~=7 fonctionne le mieux pour ces données.

Récap : Comment choisir les hyperparamètres ?

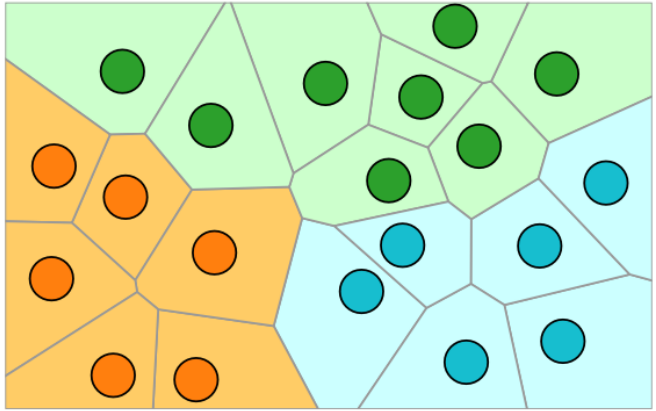
UNIVERSITÉ CÔTE D'AZUR

- Méthodologie
  - Entraînement et test
  - Entraînement, validation, test

- Entraînement pour apprendre le modèle
- Valider pour définir les hyperparamètres
- Tester pour comprendre la « généralisabilité »

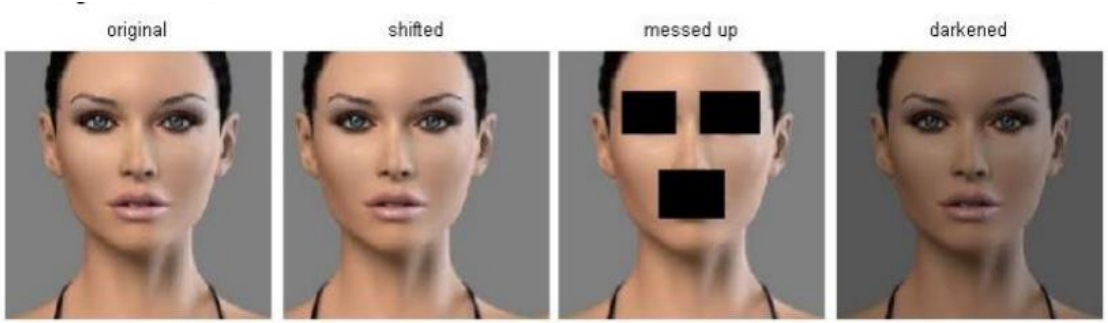
kNN -- Complexité et stockage

- N images d'apprentissage, M images de test
- Entraînement : O(1)
- Test : O(MN)
- Hmm…
  - Normalement, c'est l'inverse qu'il faut faire
  - Entraînement lent (ok), test rapide (nécessaire)



kNN

- kNN sur des images **jamais vues**
- Très mauvaises performances dans la phase de test
- Les mesures de distance au niveau des images entières peuvent être très peu intuitives

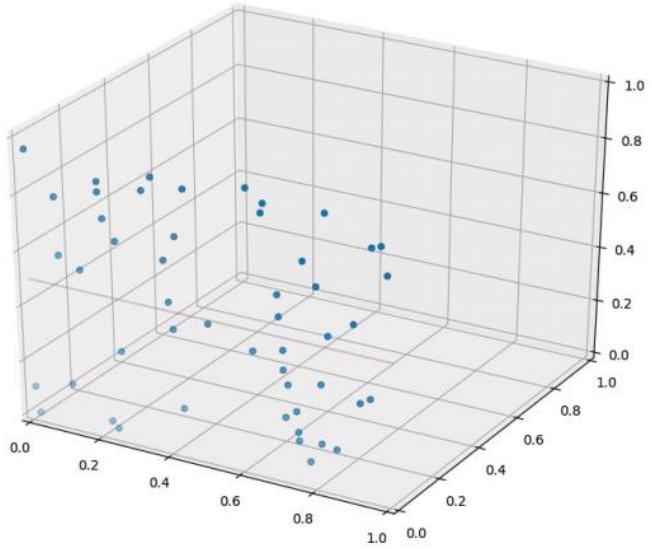


- Les trois images ont la même distance L2 par rapport à l'image de gauche.

Problèmes avec KNN : La « malédiction » de la dimensionnalité

UNIVERSITÉ CÔTE D'AZUR

- Au fur et à mesure que le nombre de dimensions augmente, la même quantité de données devient plus éparse.
- La quantité de données dont nous avons besoin finit par être exponentielle dans le nombre de dimensions.

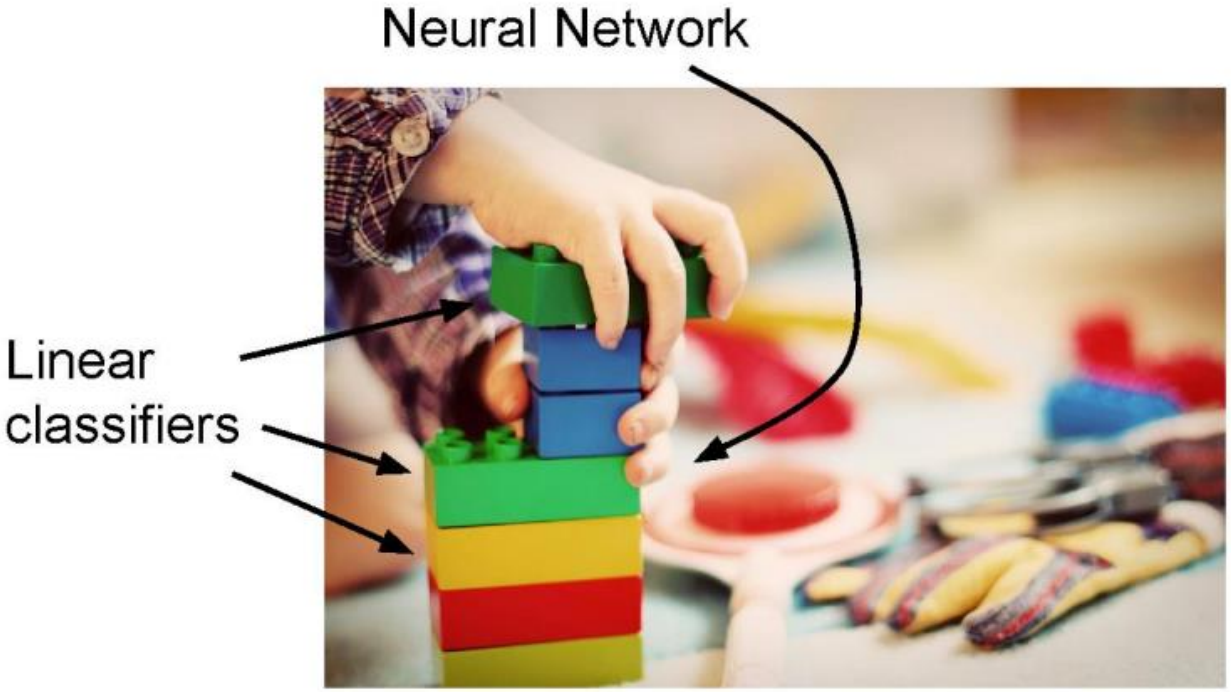


Animation from https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote02_kNN.html

K les plus proches voisins: Résumé

- En classification d'images, nous partons d'un ensemble d'images et d'étiquettes, et nous devons prédire les étiquettes sur l'ensemble de test
- Le classificateur des K-voisins les plus proches prédit les étiquettes sur la base des exemples d'apprentissage les plus proches
- La métrique de distance et K sont des hyperparamètres.
- Choisir les hyperparamètres à l'aide de l'ensemble de validation et ne l'exécuter sur l'ensemble de test qu'une seule fois, à la fin !

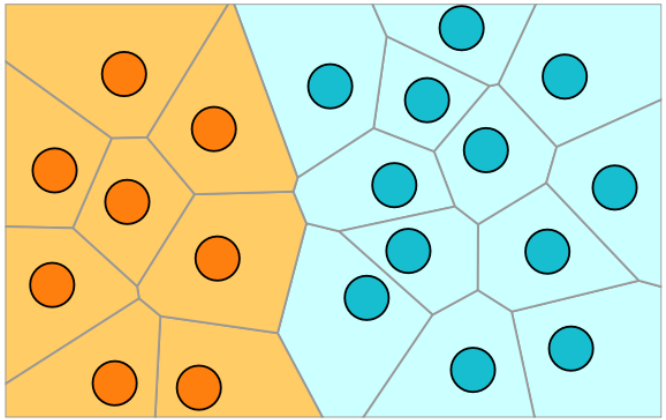
Classificateurs linéaires



Classification linéaire vs. K le plus proches voisins

UNIVERSITÉ CÔTE D'AZUR

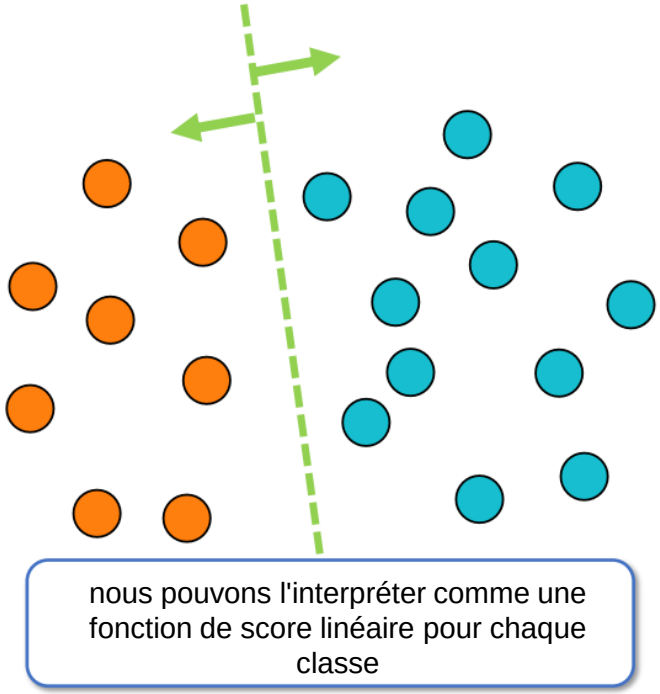
- Voisins les plus proches
  - Stocker chaque image
  - Recherche les plus proches voisins au moment du test et attribution de la même classe



Classification linéaire vs. K le plus proches voisins

UNIVERSITÉ CÔTE D'AZUR

- K-Voisins les plus proches
  - Stocker chaque image
  - Recherche les plus proches voisins au moment du test et attribution de la même classe
- Classificateur linéaire
  - Stocker les hyperplans qui séparent le mieux les différentes classes
  - Nous pouvons calculer le score d'une classe continue en calculant la distance par rapport à l'hyperplan.



Fonction de score

class scores

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Fonction de score: f

Parametric approach

image parameters

f(**x**,**W**)

10 numbers, indicating class scores

[32x32x3]
array of numbers 0...1
(3072 numbers total)

Classification paramétrique: classificateur linéaire

$$f(x, W) = Wx$$

10x1 10x3072 3072x1

10 numbers, indicating class scores

[32x32x3]
array of numbers 0...1

parameters, or “weights”

Classification paramétrique: classificateur linéaire

$f(x, W) = Wx$

10x1 10x3072 3072x1

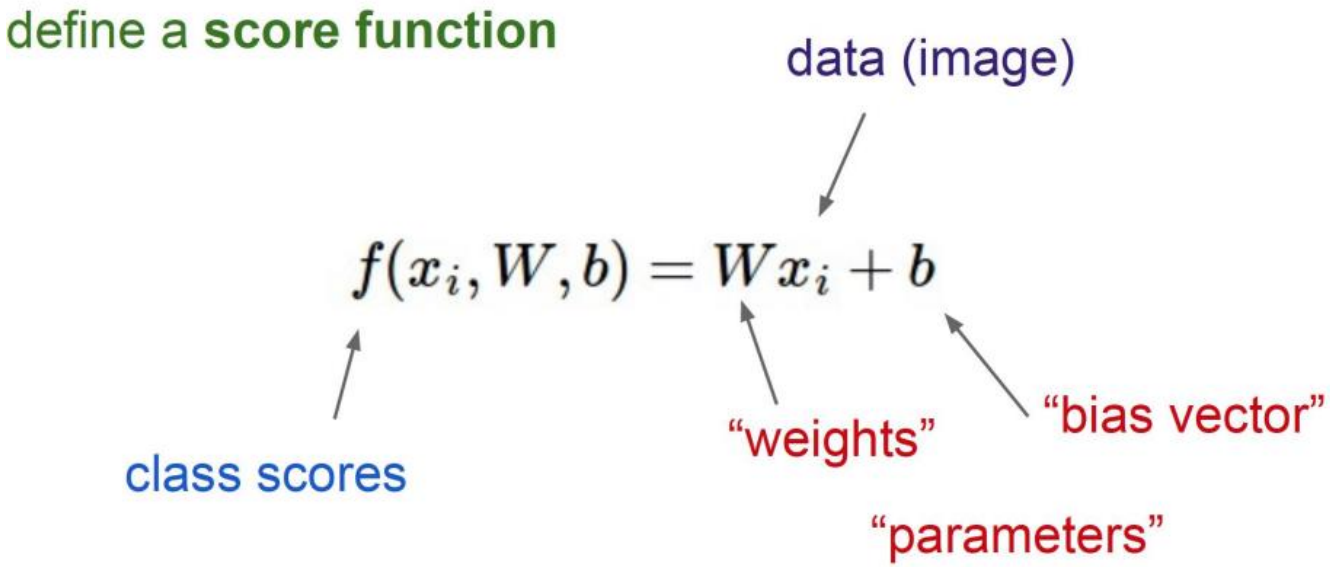
(+b) 10x1

10 numbers, indicating class scores

[32x32x3]
array of numbers 0...1

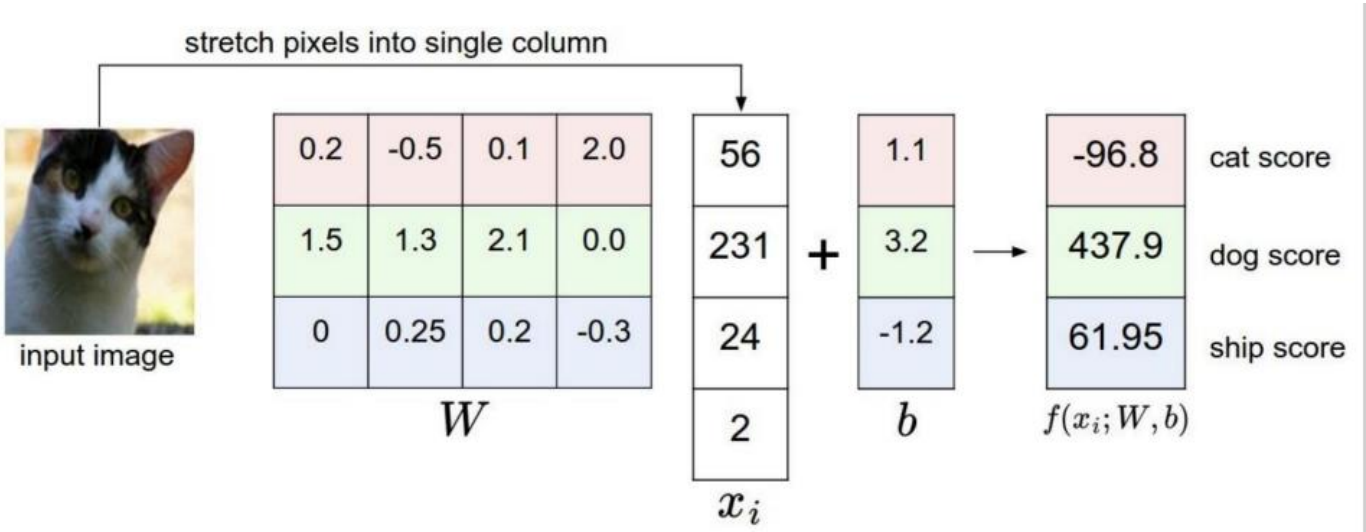
parameters, or “weights”

Classificateur linéaire

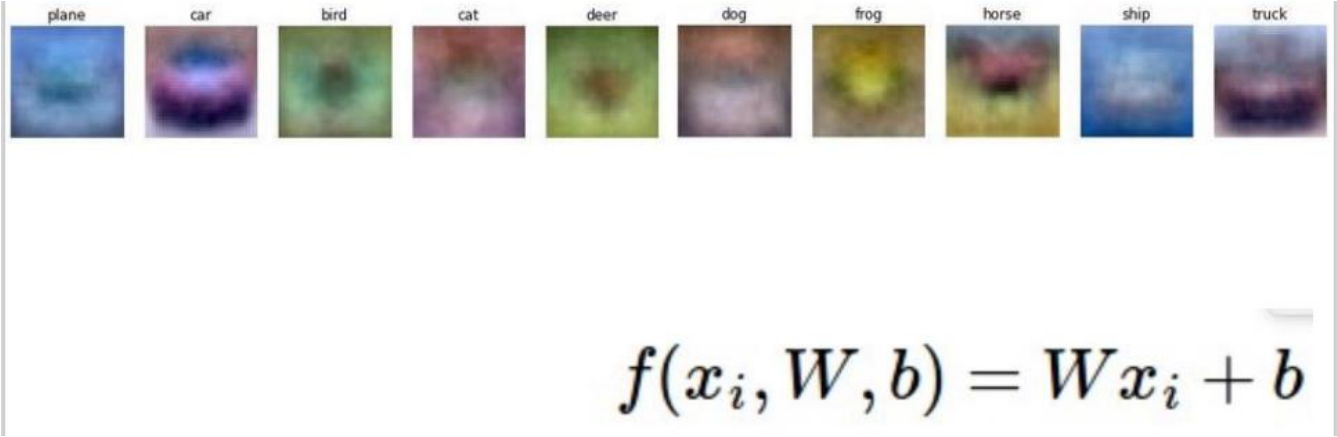


Classificateur linéaire

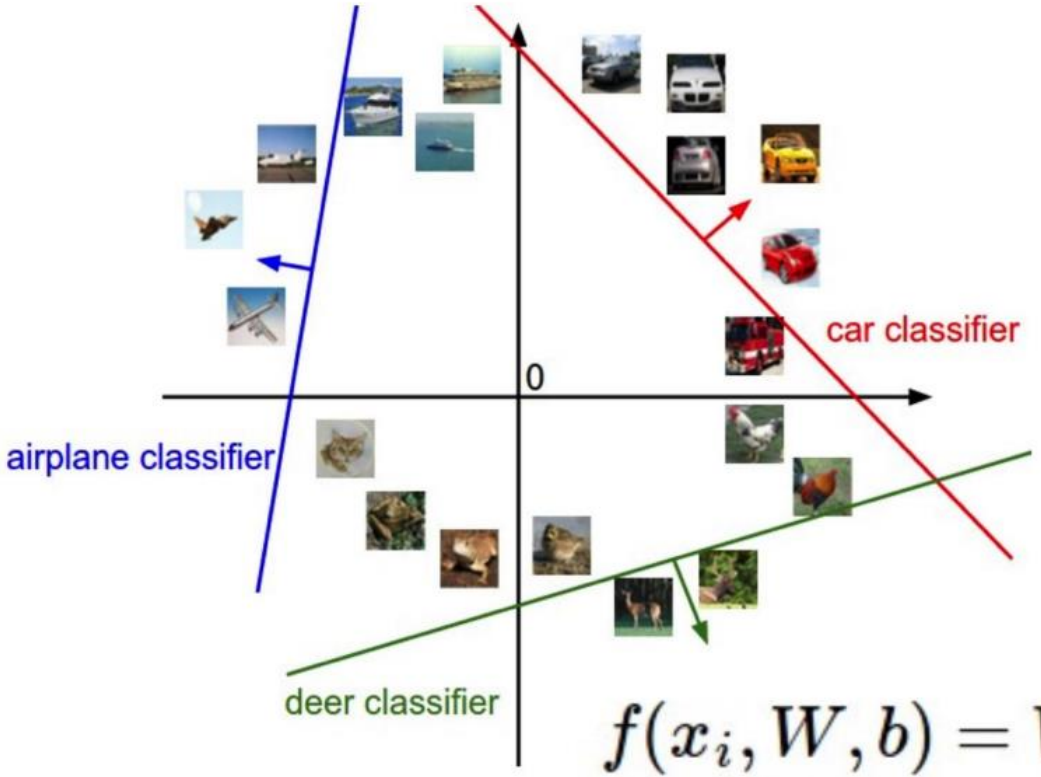
Exemple avec une image de 4 pixels et trois classes : (chat/chien/navire)



Interprétation : Correspondance des modèles



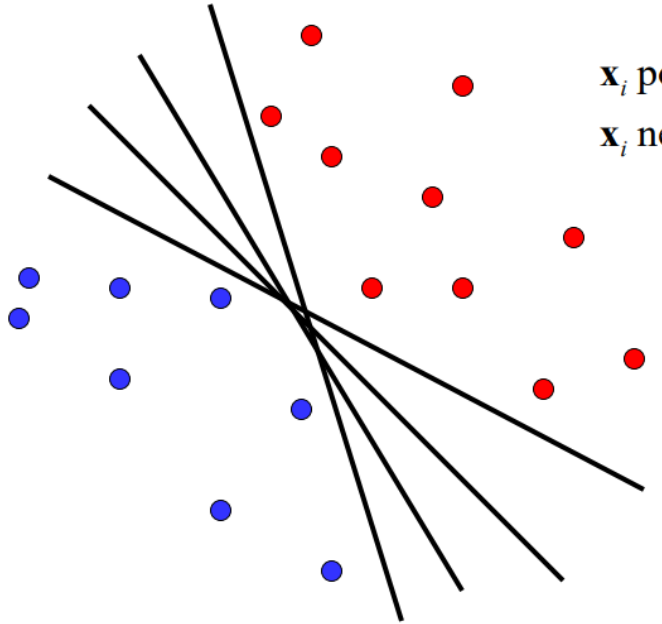
Interprétation géométrique



$$f(x_i, W, b) = Wx_i + b$$

Classificateurs linéaires

Trouver une fonction linéaire (hyperplan) pour séparer les exemples positifs et négatifs



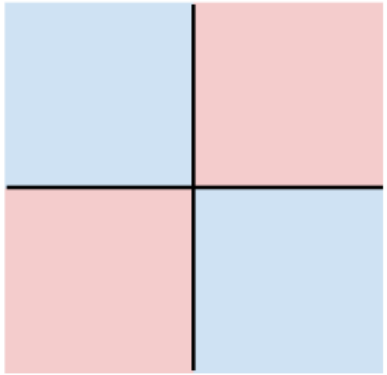
$$\mathbf{x}_i \text{ positive:} \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$
$$\mathbf{x}_i \text{ negative:} \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Quel est le meilleur hyperplan ?

Cas difficiles pour un classificateur linéaire

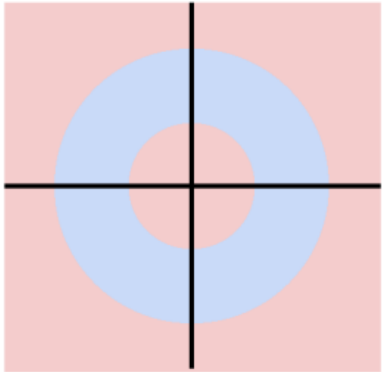
Class 1:
First and third quadrants

Class 2:
Second and fourth quadrants



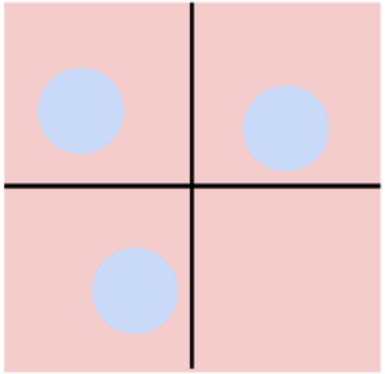
Class 1:
1 <= L2 norm <= 2

Class 2:
Everything else



Class 1:
Three modes

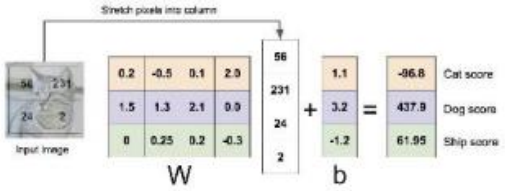
Class 2:
Everything else



Classification linéaire : trois points de vue

Point de vue algébrique

$$f(x,W) = Wx$$

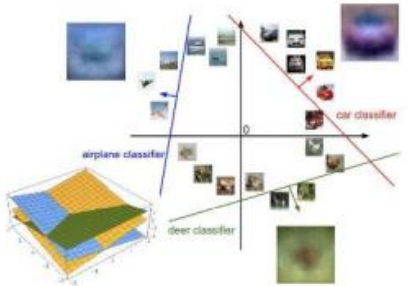


Point de vue visuel

un modèle par classe

Point de vue géométrique

hyperplans découpant l'espace



Jusqu'à présent : définition d'une fonction de score (linéaire) $f(x,W) = Wx + b$

- Exemple de scores de classe pour 3 images pour certaines valeurs de W.
- Comment savoir si ce W est bon ou mauvais ?



airplane	-3.45	-0.51	3.42
automobile	-8.87	**6.04**	4.64
bird	0.09	5.31	2.65
cat	**2.9**	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	**-4.34**
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

Résumé

- Méthodes d'apprentissage
  - K-voisins les plus proches (k-Nearest Neighbors)
  - Classification linéaire
- Le classificateur produit une fonction de score donnant un score à chaque classe
- Comment définir la qualité d'un classificateur sur la base des données d'apprentissage ? (Spoiler : définir une fonction de « perte », loss function)

Classification linéaire

airplane	-3.45	-0.51	3.42
automobile	-8.87	**6.04**	4.64
bird	0.09	5.31	2.65
cat	**2.9**	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	**-4.34**
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

Output scores

À FAIRE :

- Définir une fonction de perte qui quantifie notre mécontentement à l'égard des scores sur l'ensemble des données d'apprentissage.
- Trouver un moyen efficace de calculer les paramètres qui minimisent la fonction de perte (optimisation)

Classification linéaire

Supposons :

3 exemples de training, 3 classes.

Avec certains W, les scores $f(x, W) = Wx$ sont:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

Une fonction de perte (**loss function**) indique la qualité de notre classificateur.

Étant donné un ensemble de données d'exemples, où

x_i est une image et

y_i une étiquette (entier)

la perte (loss) sur l'ensemble de données est la somme des pertes sur les exemples :

$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

Fonction de perte, fonction de coût/objectif

UNIVERSITÉ CÔTE D'AZUR

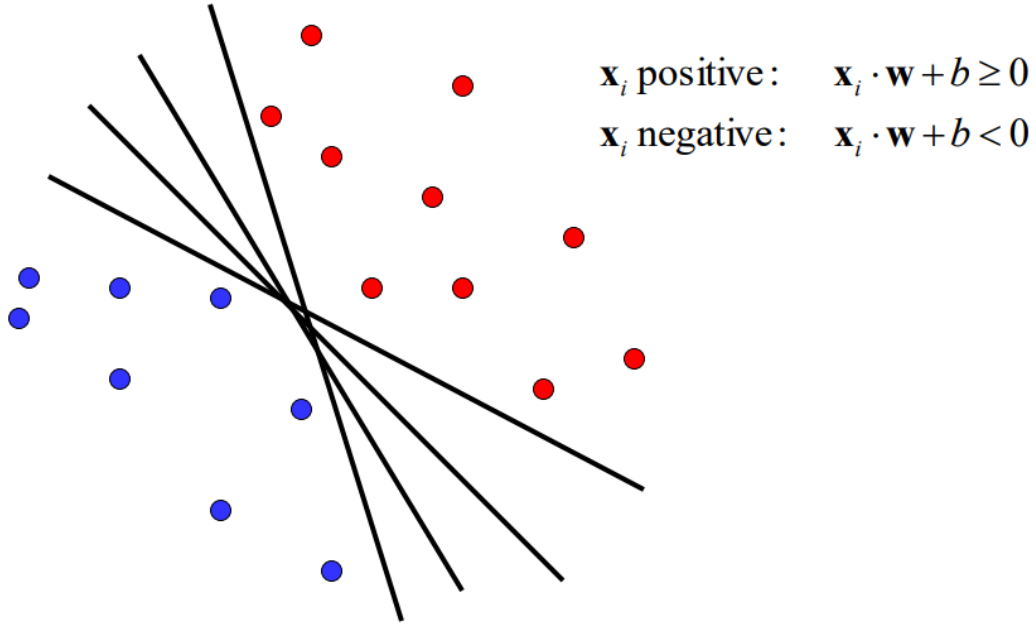
- Étant donné les étiquettes du golstandard (y_i), les scores $f(x_i , W)$
  - dans quelle mesure sommes-nous mécontents des scores ?
- La fonction de perte ou la fonction objectif/coût mesure le mécontentement.
- Au cours de l'apprentissage, nous voulons trouver les paramètres W qui minimisent la fonction de perte.

Exemple plus simple : classification binaire

- Deux classes (par exemple, "chat" et "pas chat")
  - Classes "positives" et "négatives".

Classificateurs linéaires

Trouver une fonction linéaire (hyperplan) pour séparer les exemples positifs et négatifs

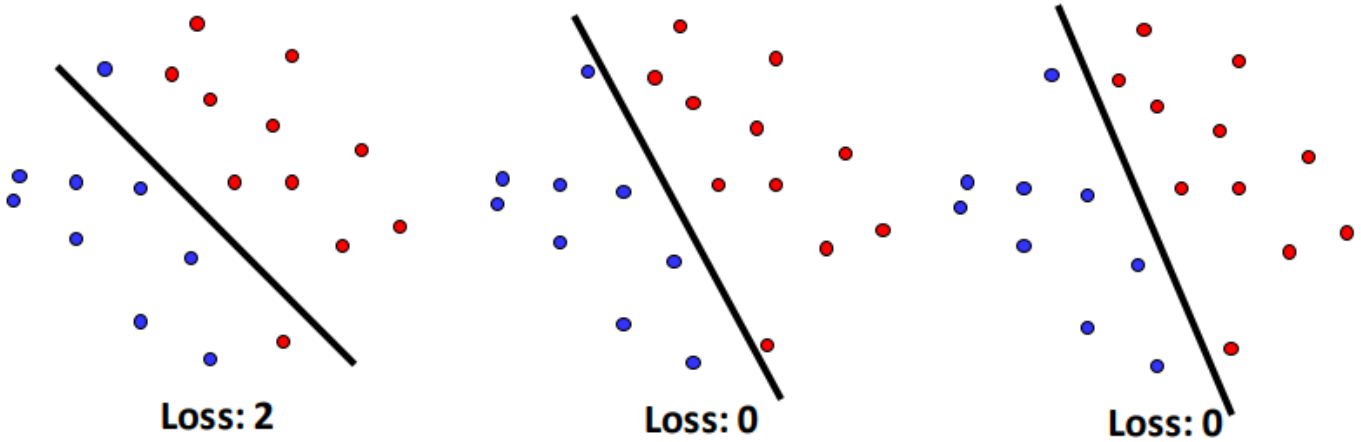


Quel est le meilleur hyperplan ?

Quelle est une bonne fonction de perte ?

Une possibilité:

- Nombre d'exemples mal classés



- Problèmes : discret, ne peut pas briser les égalités
- Nous voulons que la perte conduise à une *bonne généralisation*
- Nous voulons que la perte fonctionne pour plus de 2 classes

Classificateur Softmax

UNIVERSITÉ CÔTE D'AZUR

$$f(x_i, W) = Wx_i \quad \text{(score function)}$$

$$\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}$$ softmax function

- Exemple avec trois classes

$$[1, -2, 0] \rightarrow [e^1, e^{-2}, e^0] = [2.71, 0.14, 1] \rightarrow [0.7, 0.04, 0.26]$$

- Interprétation : écrasement des valeurs en probabilités comprises entre 0 et 1 $P(y_i \mid x_i; W)$

Perte d'entropie croisée (cross entropy loss)

UNIVERSITÉ CÔTE D'AZUR

$$f(x_i, W) = Wx_i$$ (score function)

Perte d'entropie croisée (cross entropy loss)

$$f(x_i, W) = Wx_i \quad \text{(score function)}$$

f_{y_i} : score of correct class

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \qquad L_i = -f_{y_i} + \log\sum_j e^{f_j}$$

We call L_i *cross-entropy loss*

Perte d'entropie croisée (cross entropy loss)

UNIVERSITÉ CÔTE D'AZUR

$$f(x_i, W) = Wx_i \quad \text{(score function)}$$

f_{y_i} : score of correct class

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \qquad L_i = -f_{y_i} + \log\sum_j e^{f_j}$$

We call L_i *cross-entropy loss*

$$P(y_i \mid x_i; W)$$

nous minimisons l'opposé de la log-vraisemblance (negative log likelihood)

Perte

- La perte d'entropie croisée n'est qu'une des pertes possibles
  - Une propriété intéressante est qu'elle réinterprète les scores comme des probabilités, qui ont un sens naturel

- Les fonctions de perte SVM (marge maximale) étaient également populaires
  - Mais actuellement, l'entropie croisée est la fonction de perte de classification la plus courante.

Résumé

- Disposer d'une fonction de score et d'une fonction de perte
- Actuellement, la fonction de score est basée sur un classificateur linéaire
- Ensuite, elle sera généralisée aux réseaux neuronaux convolutifs
- Trouver W et b pour minimiser la perte

$$L = \frac{1}{N} \sum_i -\log \left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}} \right) + \lambda \sum_k \sum_l W_{k,l}^2$$