

Méthode de sac de mots visuel (BoW) est une **méthode de description & classification d'images**

Outils utiliser pour le développement de la méthode BoW : matrice de corrélation, K-Means, (*chatgpt : histogramme et SIFT, SVM, KMeans*)

L'opérateur SIFT est un descripteur local car il extrait des caractéristiques à partir de régions spécifiques de l'image indépendamment de la transformation d'échelle ou rotation

Image indexée = image RVB

Régression linéaire → Si a est très petite alors la **descente du gradient** est lente ; Si a est très élevée alors la **descente du gradient saute le minimum** ⇔ ne converge pas ou diverge

Under-fitting → prédiction trop loin des vraies réponses

Over-fitting → prédiction trop proche des vraies réponses ⇔ **mauvais et doit toujours être évité** car il **prédit très bien les données d'apprentissage** mais **mal les données de test**

Cause de l'Over-fitting : modèle très complexe ; nombre de paramètres du modèle très élevé ; base d'apprentissage trop petite

Machine learning VS deep learning :

Machine learning : Entrée → **extraction de caractéristiques** → **classification** → sortie

Deep learning : Entrée → **extraction de caractéristiques + classification** → sortie

CNN (réseau de Neurones Feedforward) **possède plusieurs couches** qui apprennent progressivement des caractéristiques complexes :

- 1) Niveau bas → formes simple ; peu de couches (~ 3) → suffit de détecter des bords, lignes et formes basiques
- 2) Niveau intermédiaire → formes plus complexes ; - 10 couches → besoin de détecter formes, couleurs & motifs plus détaillés
- 3) Niveau avancé → formes très complexes ; + 16 couches → apprentissage de détails fins comme les traits du visage

Couches spécifiques du CNN :

- Couche convolutive (CL) →
 - Opérateur de convolution, concept clé en traitement du signal (*1D pour les signaux et 2D les images*)
 - Permet d'avoir une image en sortie à partir d'une image en entrée
 - Représentation linéaire (somme des produits)
 - Le noyau de convolution est utilisé pour extraire des caractéristiques visuelles (détection de contours, floutage, lissage).
- Couche de Pooling (PL) → Sous échantillonnage (opérateur moyen et max)
- Couche ReLU (RL) → facilite l'optimisation de la descente du gradient ; introduit la représentation non linéaire
- Couche entièrement connectée (FC) → imite les classifieurs MLP ; tâche de classification

MLP VS CNN :

MLP → Organisation des neurones en une séquence de couches

CNN → Organisation des neurones en une séquence de couches 3D (appelées profondeur)

Pooling Spatial → réduit la dimension de chaque caractéristiques tout en conservant les informations importantes → type : moyen ; max ; somme

Pour un simple objet sur une image → classification + localisation

Pour plusieurs objets sur une image → détection d'objet + segmentation d'instance

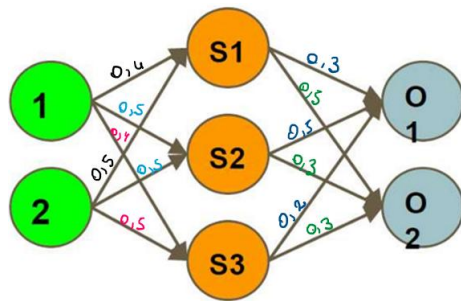
Evaluation de la performance d'une méthode : calcul de son erreur sur la base d'apprentissage et sur la base de test (séparée) → possibilité que l'erreur sur la base de test soit très faible et que sur la base d'apprentissage elle soit très grande ; intéressant d'utiliser plusieurs bases d'apprentissages et de test

Opérateur Kmeans → classifieur non supervisé

Pour prédire la catégorie :

La fonction d'activation (F), les valeurs seront données

Exemple : Pour cet exemple, la fonction d'activation est : Sigmoid : $\sigma(x) = \frac{1}{1+e^{-x}}$



$$S1 = 1*0,4 + 2*0,5 = 1,4 \Leftrightarrow S1 = \frac{1}{1+e^{-1,4}} = 0.8$$

$$S2 = 1*0,5 + 2*0,5 = 1,5 \Leftrightarrow S2 = \frac{1}{1+e^{-1,5}} = 0.81$$

$$S3 = 1*0,4 + 2*0,5 = 1,4 \Leftrightarrow S3 = \frac{1}{1+e^{-1,4}} = 0.8$$

$$O1 = S1*0,3 + S2*0,5 + S3*0,2 = 0,80 \Leftrightarrow O1 = \frac{1}{1+e^{-0,8}} = 0.69$$

$$O2 = S1*0,5 + S2*0,3 + S3*0,3 = 0,88 \Leftrightarrow O2 = \frac{1}{1+e^{-0,88}} = 0.70$$

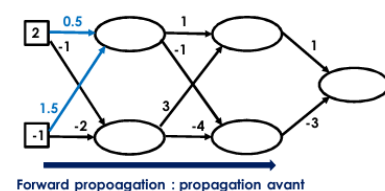
Ici O1 et O2 sont trop proche pour pouvoir prédire la catégorie donc X est dans C1 ou C2

Pour entrainer le modèle :

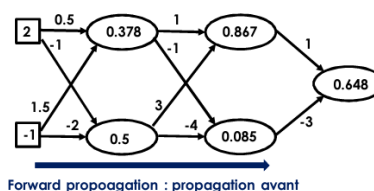
On nous donne : le vecteur d'entrée, le vecteur de sortie et les poids → le but : vérifier qu'avec les poids données, on retrouve bien le vecteur de sortie qui est donné

Etape 1 : propagation avant → on va calculer toutes les sorties jusqu'à la sortie finale

Exemple : Pour cet exemple, la fonction d'activation est : Sigmoid : $\sigma(x) = \frac{1}{1+e^{-x}}$; Vecteur d'entrée : X = [-2 ; 1] ; Vecteur de sortie : y = 1



Forward propagation : propagation avant



Forward propagation : propagation avant

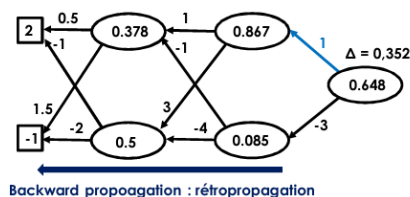
On applique la même formule que pour prédire la catégorie

On remarque que le vecteur de sortie est différent.

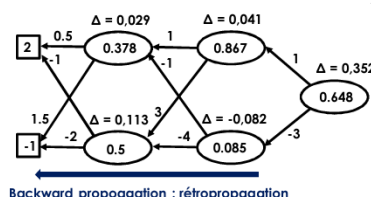
Etape 2 : calcul de la différence entre la sortie attendue et la sortie qu'on a : $\Delta = y - S$

Dans notre exemple, $\Delta = 1 - 0,648 = 0,352$

Etape 3 : rétropropagation → on va calculer le delta pour chaque sortie ($\Delta_j = g_j(1 - g_j) \sum_k w_{j,k} \Delta^k$)



Backward propagation : rétropropagation

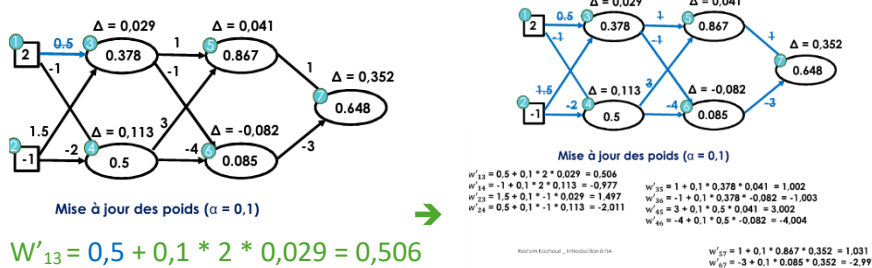


Backward propagation : rétropropagation

$$\Delta = 0,867 (1 - 0,867) * 1 * 0,352 = 0,041$$

Etape 4 : mise à jour des poids \rightarrow On utilise la formule : $w'_{ij} = w_{ij} + \alpha * x^i * \Delta^j$
On va avoir besoin d'une valeur de α , s'il n'est pas donné il faut prendre $\alpha = 0,1$

Dans notre exemple :



$$W'_{13} = 0,5 + 0,1 * 2 * 0,029 = 0,506$$

Schéma de l'architecture CNN correspondant à un code :

S'il n'y a pas d'encodeur (réduction de dimension) et un décodeur (reconstruction de l'entrée) c'est un classifieur

Exemple :

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, MaxPooling2D
model = Sequential()
model.add(layers.Conv2D(256, (3, 3), activation='relu', input_shape=(227, 227, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='sigmoid'))
model.add(layers.Dense(1, activation='sigmoid'))
```

Ici c'est un classifieur. L'entrée est de dimension 3 (227, 227, 3). La sortie sera de dimension 1

Schéma :

