Formation Machine Learning

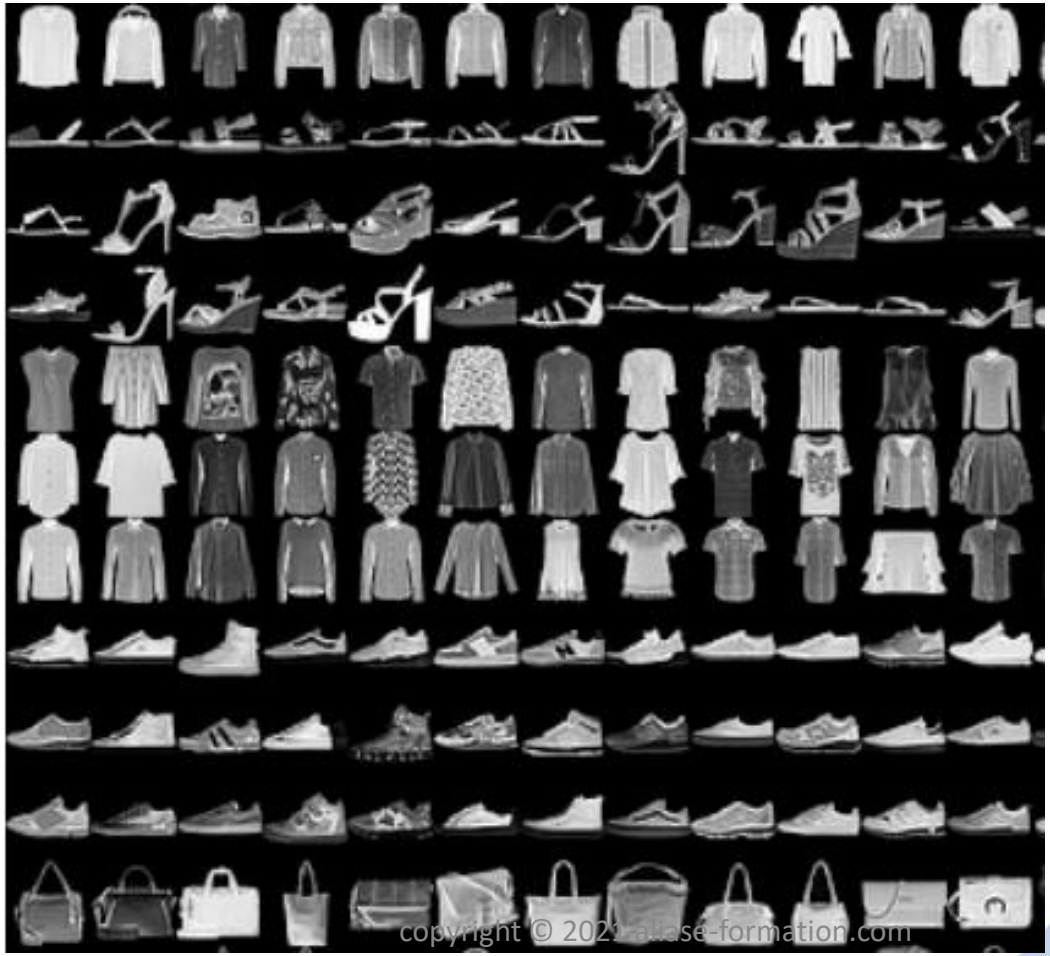
Atelier Pratique AP-ML5

ALIASE

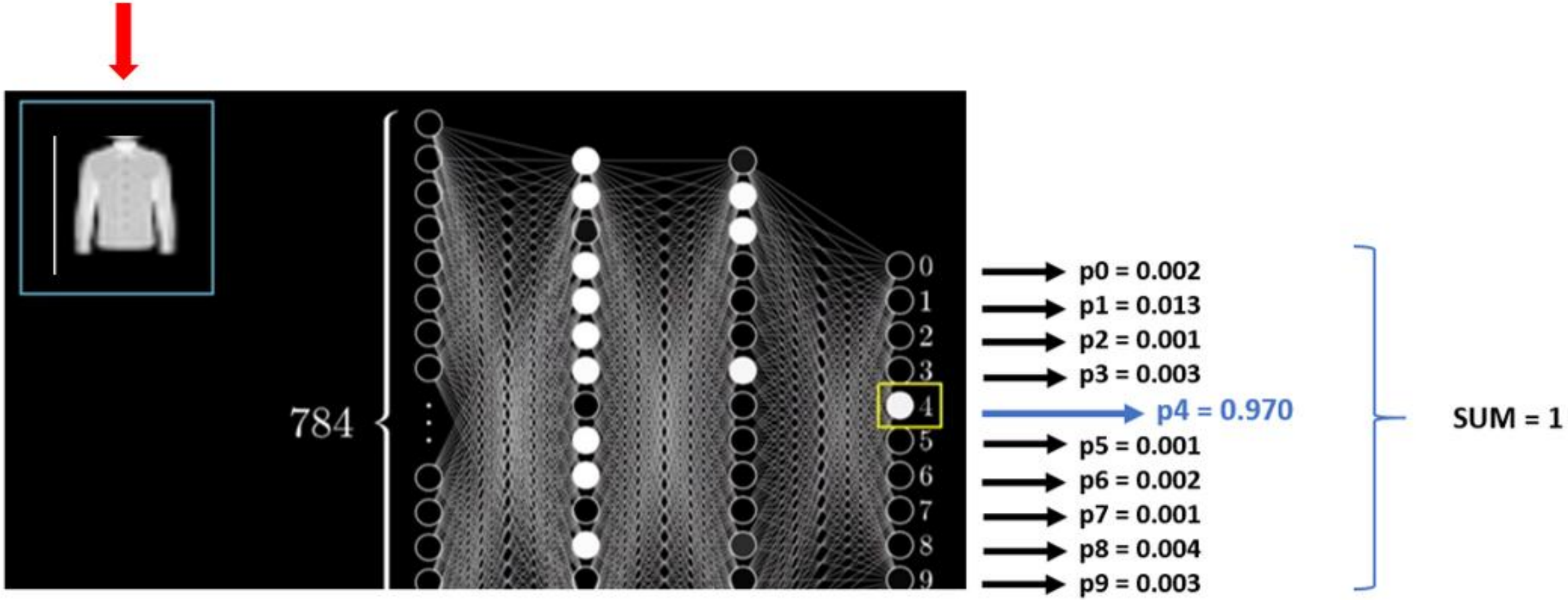
Atelier Pratique AP-ML5 : Exercice ML5.3

Définir un modèle de réseaux de neurones qui reconnaît des images d'habits.

Ecrire le même programme dans l'exercice précédent mais en utilisant le dataset d'images ***FASHION MNIST***



Atelier Pratique AP-ML5 : Exercice ML5.3



Atelier Pratique AP-ML5 : Exercice ML5.3

ALIASE
IA - SERVICE & FORMATION

Chargement du dataset d'images ***FASHION MNIST***

Dataset = 70,000 images d'habits, classés en 10 categories. Resolution : 28 x 28 pixels

```
from tensorflow.keras import layers, models, datasets

dataset = datasets.fashion_mnist

(X_train, Y_train), (X_test, Y_test) = dataset.load_data()
```

L'importation des images retourne 4 tableaux Numpy :

X_train, Y_train : Training Set

X_test, Y_test : Test Set

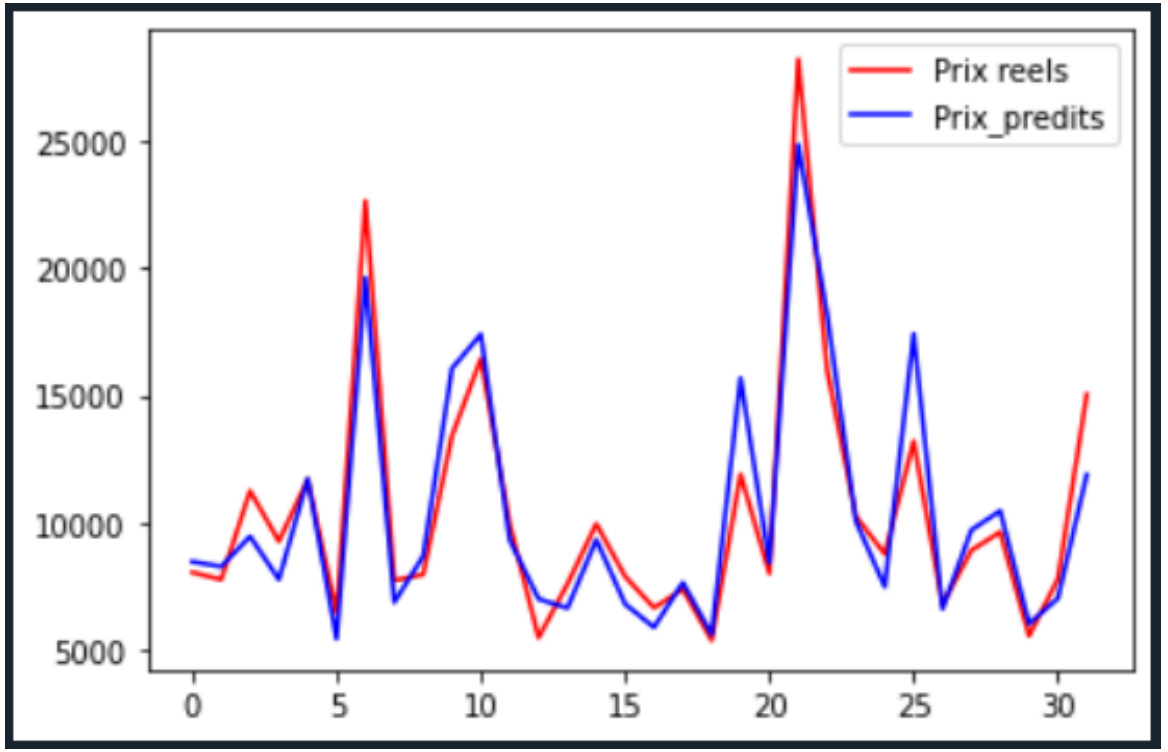
Solution → Voir MLExo5.3.py

Atelier Pratique AP-ML5 : Exercice ML5.6.1

ALIASE
IA - SERVICE & FORMATION

Regression avec un réseau de neurones MLP

Refaire l'exercice de predictions des **prix des voitures** avec un réseau MLP. Dataset = **autos.xlsx**



Solution → Voir MLExo5.6.py

Atelier Pratique AP-ML5 : Exercice ML5.6.2

ALIASE
IA - SERVICE & FORMATION

Regression avec un réseau de neurones MLP

Outil COMPLET

- **Encodage des features** : décider avec quelle fonction (*OneHot ? OrdinalEncoder ?)*
- **Valeurs abérrantes** : détection et suppressions des outliers
- **Features Selection** : Application d'une méthode parmi les 4 selon le nb de features
- **Normalisation des features** : avec quelle fonction
- **Normalisation des labels** : décider si on doit normaliser les labels (Y), et avec quelle fonction

Solution → Voir MLExo5.6.2.py

Atelier Pratique AP-ML5 : Exercice ML5.2

ALIASE
IA - SERVICE & FORMATION

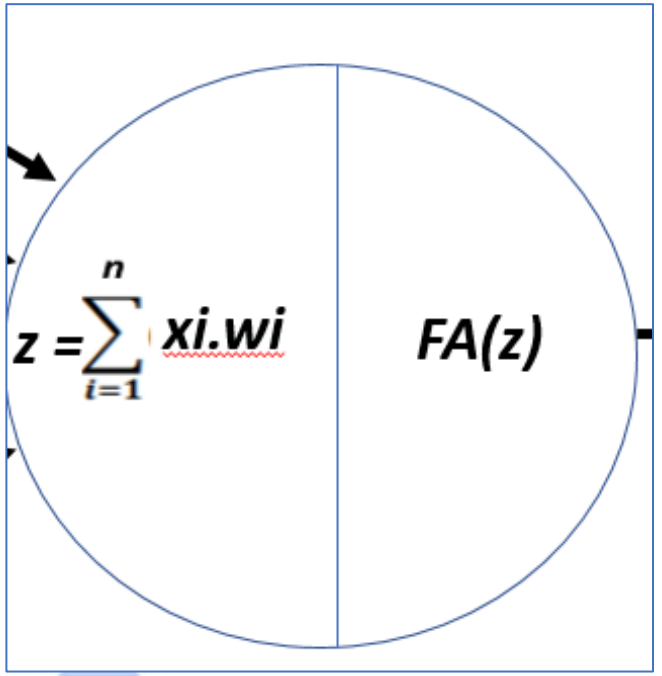
Objectif : Définir votre propre fonction softmax *:* ***def my_softmax(…)***

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

- Modifier le programme dans l'exercice précédent en utilisant votre propre fonction softmax

```
def create_model():
    Layer0  = layers.Input(shape=(64,))
    Layer1 = layers.Dense(256, activation='relu')
    Layer2 = layers.Dense(128, activation='relu')
    Layer3 = layers.Dense(10 , activation='softmax')

    model  = models.Sequential( [ Layer0,  Layer1,  Layer2,  Layer3 ] )
    return model
```



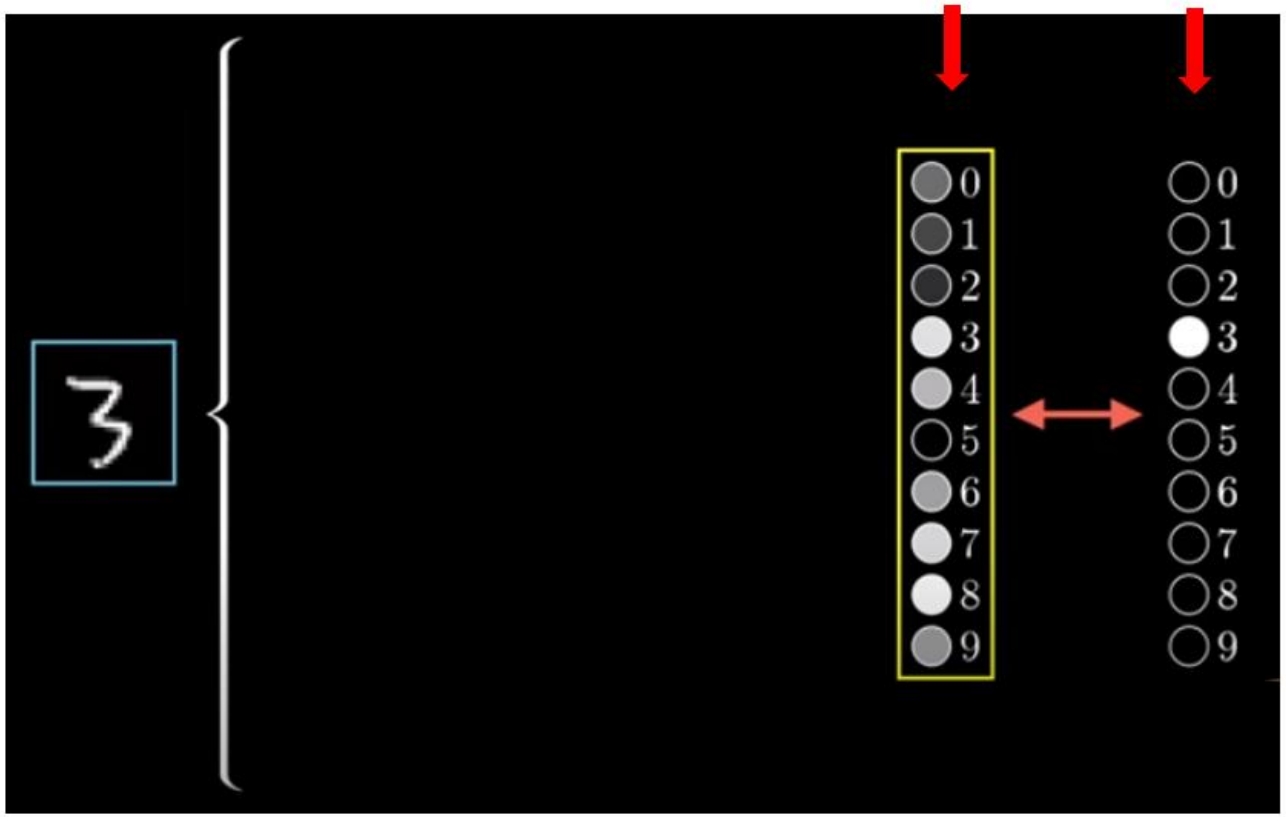
Atelier Pratique AP-ML5 : Exercice ML5.2

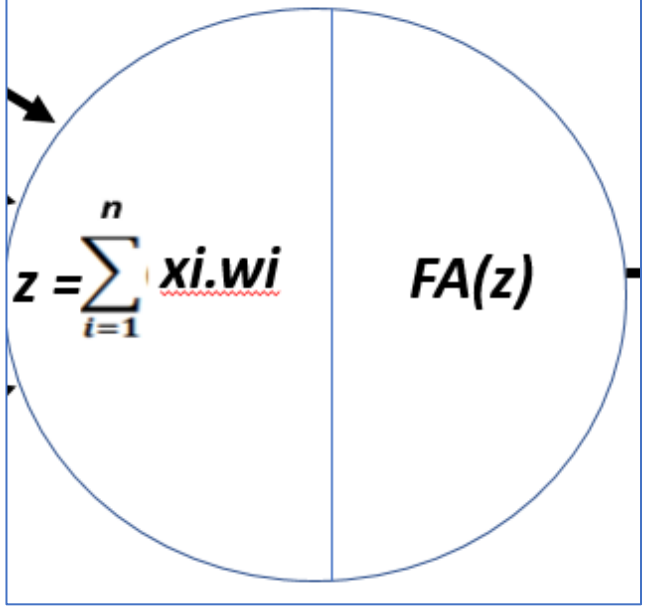
ALIASE
IA - SERVICE & FORMATION

def my_softmax(PreActiv):

- Entrée : un vecteur de 10 valeurs (**Préactivations**)
- Sortie : un vecteur de 10 valeurs (**Activations**)

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$





Solution → Voir MLExo5.2.py