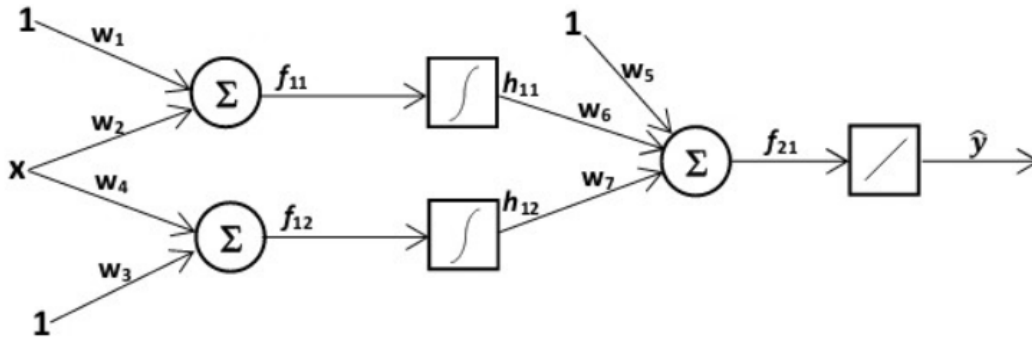


Exercice 1

Soit le réseau de neurones multicouches décrit par le graphe suivant :



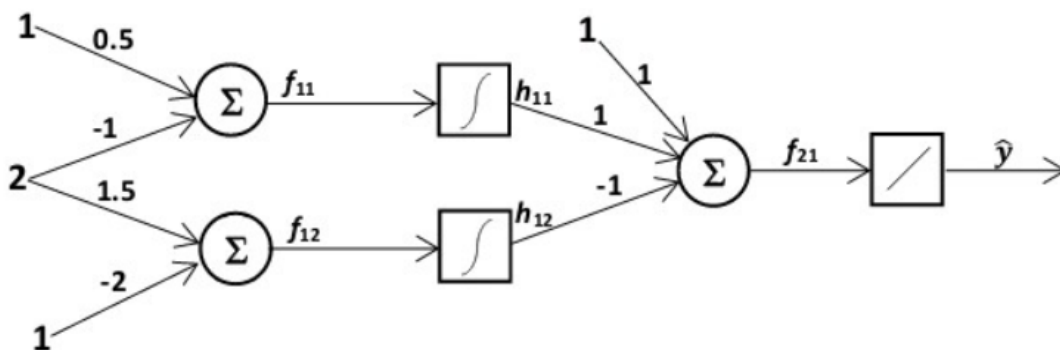
1. Donner les formules mathématiques qui déterminent les sorties intermédiaires f_{11} , f_{12} , h_{11} , h_{12} , f_{21} ainsi que la sortie finale \hat{y} .
2. Soit la fonction d'erreur :

$$E(\mathbf{w}) = (y - \hat{y})^2$$

En appliquant l'algorithme de propagation en arrière (backpropagation), trouver les expressions des mises à jour des paramètres Δw_j pour $j = 1, \dots, 7$.

Exercice 2

Soit le réseau de neurones multicouches décrit par le graphe suivant :



Soit la donnée $(\mathbf{x}, y) = (2, 1)$:

1. Déterminer les sorties intermédiaires f_{11} , f_{12} , h_{11} , h_{12} , f_{21} ainsi que la sortie finale \hat{y} , en utilisant successivement les fonctions d'activation ReLU, Sigmoid et Tanh.
2. Calculer les paramètres Δw_j et w_j pour $j = 1, \dots, 7$ après une itération de mise à jour (en considérant le paramètre d'apprentissage $\alpha = 0.1$).

Objectif

L'objectif de ce TP est de comprendre l'impact des fonctions d'activation sur les sorties d'un perceptron. Pour cela, importez le fichier *perceptron.xlsx*.

Données du problème

- Matrice des poids :

$$W = \begin{bmatrix} 1 & -1 \\ 1 & -2 \\ 1 & -3 \end{bmatrix}$$

- Vecteur d'entrée :

$$X = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

- Biais :

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Travail à faire

1. Sortie linéaire brute :

Calculez les activations linéaires $Z_i = W_i \cdot X + b_i$ pour chaque neurone $i \in \{1, 2, 3\}$.

2. Application des fonctions d'activation :

Créer des fonction qui permettent de calculez les sorties activées $h_i = f(Z_i)$ pour chaque fonction d'activation suivante :

- ReLU : $f(z) = \max(0, z)$
- Sigmoïde : $f(z) = \frac{1}{1 + e^{-z}}$
- Tanh : $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

3. Représentation graphique : Réalisez les courbes des trois fonctions d'activation suivantes : **ReLU**, **Sigmoïde** et **Tanh**. Indiquez clairement sur chaque courbe les valeurs de Z_i obtenues précédemment. Précisez également la fonction utilisée dans chaque graphique. Enfin, réalisez un seul graphique combiné affichant les trois courbes sur un même plan pour faciliter la comparaison visuelle.

Objectifs

- De manipuler un réseau de neurones multicouches (MLP) avec `scikit-learn`,
- D'étudier l'influence des fonctions d'activation sur la frontière de décision,
- D'explorer l'effet des hyperparamètres (architecture, taux d'apprentissage, régularisation...),
- De visualiser les performances via une réduction de dimension (PCA).

Étapes à suivre

1. Chargement et exploration des données

- Quelle base de données allez-vous utiliser ? (Nom et description)
- Combien y a-t-il d'observations et de variables ?
- Quelles sont les classes à prédire ?

2. Préparation des données

- Séparer les données en deux ensembles (apprentissage et test).
- Appliquer une normalisation des variables explicatives.
- Réduire la dimension des données à 2 composantes principales à l'aide de l'ACP.
- Visualiser les individus projetés dans le plan des deux premières composantes (coloriés par classe).

3. Entraînement d'un réseau de neurones de base

- Entraîner un MLP avec une seule couche cachée de 10 neurones.
- Quelle est la fonction d'activation utilisée par défaut ?
- Afficher les scores d'apprentissage et de test.
- Comment interpréter l'écart entre les deux scores ?

4. Étude des fonctions d'activation

- Comparer les fonctions d'activation suivantes : `identity`, `logistic`, `tanh`, `relu`.
- Pour chaque fonction :
 - (a) Entraîner un modèle avec les mêmes paramètres.
 - (b) Afficher le score sur les données de test.
 - (c) Tracer la frontière de décision dans l'espace PCA.
- Quelle fonction semble donner la meilleure séparation ?

5. Effet de l'architecture du réseau

- Tester plusieurs architectures en variant :
 - (a) le nombre de neurones dans une seule couche (5, 10, 50),
 - (b) le nombre de couches cachées : ((10,), (10, 10), (30, 30), etc.).
- Comment évolue la performance ?
- Que constatez-vous en termes de surapprentissage ?

6. Hyperparamètres avancés

Explorer les effets combinés des hyperparamètres suivants :

- Fonction d'activation : identity, logistic, tanh, relu
- Taux d'apprentissage (`learning_rate_init`) : 0.001, 0.01
- Paramètre de régularisation (`alpha`) : 0.0001, 0.001
- Architecture : (5,), (10,)
- Nombre d'itérations : 300
- Batch size : 32
- Early stopping : activé
- Pour chaque combinaison, entraîner un modèle et afficher :
 - la précision sur le test,
 - une interprétation de la stabilité du modèle.
- Quelle combinaison donne les meilleurs résultats ?
- Comment expliquer l'influence du batch size et de l'early stopping ?

7. Synthèse et interprétation

- Résumez l'influence des fonctions d'activation.
- Résumez l'influence des différents hyperparamètres.
- Quelle combinaison offre un bon compromis entre biais et variance ?

Fin