

Direction Régionale : Casablanca – Settat

Cité des métiers et des compétences

Contrôle continu N° 1 : APPRENTISSAGE PROFOND (M108)
Année 2024/2025

Filière : Intelligence Artificielle

Durée : 2H

Niveau : TS

Variante : 1

Groupe : IA101

Barème : .../20

Nom et Prénom :

Partie théorique :

QCM (...../10 points) : Cocher la bonne réponse.

1. Une perte (loss) faible signifie que : (1 point)
 - Le modèle a surappris les données
 - Le modèle a été mal initialisé
 - Les prédictions sont proches des vraies valeurs
 - Le réseau est trop complexe
2. Pourquoi est-il important d'ajouter un biais dans un neurone lorsque certaines entrées, comme "nombre d'enfants = 0", ont un sens réel ? (1 point)
 - Parce que le biais empêche l'entrée d'être ignorée même si sa valeur est nulle
 - Le biais remplace les valeurs nulles par la moyenne des données
 - Parce que le biais transforme les entrées nulles en sorties négatives
 - Le biais supprime les entrées trop faibles pour améliorer la précision
3. À quoi servent les poids dans un réseau de neurones artificiels ? (1 point)
 - À ajuster la vitesse de propagation des données entre les couches
 - À stocker les sorties précédentes pour les utiliser plus tard
 - À déterminer l'importance de chaque entrée dans le calcul
 - À convertir les fonctions d'activation en fonctions linéaires
4. Une courbe d'apprentissage qui montre une perte d'entraînement qui diminue mais une perte de validation/test qui augmente indique : (1 point)
 - Un bon entraînement
 - Un underfitting
 - Un overfitting
 - Aucune réponse
5. Une faible précision à l'entraînement et à la validation/test, ainsi qu'une perte élevée sur les deux, indiquent généralement : (1 point)
 - Une bonne généralisation
 - Un underfitting
 - Un overfitting
 - Aucune réponse
6. Quel est le rôle principal d'un neurone dans un réseau de neurones artificiels ? (1 point)
 - Transmettre uniquement les données d'entrée vers la sortie sans transformation
 - Mémoriser tous les exemples du jeu de données dans une base interne
 - Appliquer une sommation sur les entrées pondérées, puis une fonction d'activation
 - Ajouter du bruit aléatoire pour améliorer la diversité des prédictions

7. Quel moyen aide à réduire l'overfitting ?

(1 point)

- Ajouter encore plus de couches
- Enlever le biais
- Utiliser la rétropropagation deux fois
- Appliquer une régularisation comme Dropout

8. À quoi sert 'StandardScaler' dans un pipeline de deep learning ?

(1 point)

- À encoder les labels
- À régulariser la fonction de perte
- À normaliser les données
- À réduire les dimensions

9. Quelle fonction d'activation est adaptée pour un problème à plusieurs classes (ex : "Chat", "Chien", "Oiseau") ?

(1 point)

- tanh
- sigmoid
- softmax
- relu

10. Après avoir appliqué une fonction sigmoïde à la somme pondérée d'un neurone, on obtient une sortie de 0,975. Que cela signifie-t-il dans un problème de classification binaire ?

(1 point)

- Le modèle est sûr à 97,5% que l'exemple est négatif
- Le score est trop faible pour être utilisé
- La sortie doit encore passer par une autre couche
- Le modèle prédit avec 97,5% de confiance que l'exemple appartient à la classe 1

Exercice 1 : Compléter les parties manquantes du code ci-dessous (..../10 points)

Importer les bibliothèques

(0.5 point)

```
import numpy as np
import pandas as pd
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense , Dropout
from sklearn.model_selection import train_test_split
from tensorflow.keras import regularizers
from sklearn.datasets import make_classification
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

```
X, y = make_classification(n_samples=1000, n_features=20, n_classes=2)
-----, -----, -----, ----- = ----- (X, y, test_size=0.2, random_state=42)
```

Création d'un modèle séquentiel simple, Évaluation du modèle et Faire une prédition

(4 points)

```
model = _____()
model.add(_____(64, activation='_____, (X_train.shape[1],)))
model.add(_____(32, activation='_____'))
model.add(_____(_____, activation='_____'))
```

```
model._____ (optimizer='_____, loss='_____, metrics=['accuracy'])
```

```
history=model.fit(X_train, y_train, _____=10, _____=32, verbose=1)
```

```
predictions = model._____(X_test[:5])
print("Predictions:", predictions)
```

```
scaler = _____  
X_train = scaler._____ (X_train)  
X_test = scaler._____ (X_test)
```

```
model = _____()  
model.add(_____(128, activation='_____',  
    input_shape=(30,),  
    kernel_regularizer=_____.12(_____))  
  
model.add(_____(64, activation='_____'))  
model.add(_____(0.5))  
model.add(_____(_____))
```

Afficher les courbes d'apprentissage

(0.5 point)

```
plt.plot(_____.history[‘_____’],label='loss')  
plt.plot(_____.history[‘_____’],label='val_loss')  
plt.legend()  
plt.show()
```

1. Qu'aurait-il fallu modifier si le problème avait plus de 3 classes ?

(1.5 points)

-1ére modification :

-2éme modification :

-3éme modification :