

Révision ANN

Total questions: 12

Worksheet time: 9mins

Instructor name: Mr. oualid miloudi

Name

Class

Date

1. **Quelle est la fonction d'activation la plus couramment utilisée dans les couches cachées ?**

- a) C. ReLU
- b) D. Softmax
- c) A. Sigmoid
- d) B. Tanh

2. Quelle fonction est utilisée pour la sortie d'un réseau de classification binaire ?

- a) A. ReLU
- b) B. Softmax
- c) C. Sigmoid
- d) D. Tanh

3.

```
model = Sequential()  
model.add(Dense(64, activation='relu', input_shape=(100,)))  
model.add(Dense(1, activation='sigmoid'))
```

Que fait ce code ?

- a) A. Crée un réseau avec 64 couches cachées
- b) B. Crée un réseau avec 64 neurones d'entrée
- c) C. Crée un réseau avec 1 couche cachée de 64 neurones et une couche de sortie
- d) D. Crée une fonction d'activation sigmoïde

4.

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Que fait la ligne suivante ?

- a) A. Entraîne le modèle
- b) B. Évalue les performances du modèle
- c) C. Prépare le modèle pour l'entraînement
- d) D. Sauvegarde le modèle

5. **Que signifie input_shape=(100,) dans la première couche ?**

- a) B. Chaque entrée a 100 features
- b) D. L'entrée sera redimensionnée à 100x1
- c) A. 100 échantillons d'entrée
- d) C. Il y a 100 neurones dans la couche

6.

```
model = Sequential()  
model.add(Dense(32, activation='relu'))  
model.add(Dense(3, activation='softmax'))  
model.compile(optimizer='adam', loss='binary_crossentropy')
```

Quel est le problème avec ce code ?

- a) D. Il manque la méthode fit()
- b) A. binary_crossentropy ne convient pas pour une sortie à 3 classes
- c) B. Dense(3, activation='softmax') est invalide
- d) C. Rien, le code est correct

7.

```
from keras.layers import Dropout  
model.add(Dropout(0.5))
```

Que fait ce code ?

- a) B. Ignore aléatoirement 50% des neurones pendant l'entraînement
- b) D. Gèle 50% des poids du réseau
- c) C. Supprime 50% des neurones de manière permanente
- d) A. Double le taux d'apprentissage

8. Que contient history.history après l'entraînement ?

- a) A. Les poids du réseau
- b) C. Les couches du modèle
- c) D. L'historique des métriques par époque (loss, accuracy, etc.)
- d) B. Les hyperparamètres

9. Comment met-on en place une régularisation dans un réseau de neurones avec Keras ?

- a) B. En ajoutant Dropout() ou kernel_regularizer dans les couches
- b) C. En augmentant le nombre de couches cachées
- c) A. En ajoutant un argument loss='regularization' dans compile()
- d) D. En activant un mode de régularisation dans fit()

10. Quel code ajoute une régularisation l2 correcte ?

- a) Dense(64, activation='relu', regularizer='l2')
- b) Dense(64, activation='relu', loss='l2')
- c) Dense(64, activation='relu', kernel_initializer='l2')
- d) Dense(64, activation='relu', kernel_regularizer=l2(0.01))

11. Quel code complet est correct pour classifier 3 catégories ?

a)

```
model = Sequential()
model.add(Dense(100, activation='relu', input_shape=(100,)))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

b)

```
model = Sequential()
model.add(Dense(10, activation='relu', input_shape=(100,)))
model.add(Dense(1, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

c)

```
model = Sequential()
model.add(Dense(10, activation='relu', input_shape=(100,)))
model.add(Dense(1, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

d)

```
model = Sequential()
model.add(Dense(128, activation='relu', input_shape=(100,)))
model.add(Dense(2, activation='softmax'))
model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
```

12. Quel code est correct pour une classification de scans médicaux en “sain” ou “malade”?

a)

```
model = Sequential()
model.add(Dense(10, activation='relu', input_shape=(100,)))
model.add(Dense(2, activation='sigmoid'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

b)

```
model = Sequential()
model.add(Dense(10, activation='relu', input_shape=(100,)))
model.add(Dense(2, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

c)

```
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(100,)))
model.add(Dense(2, activation='softmax'))
model.compile(optimizer='adam', loss='mse', metrics=['accuracy'])
```

d)

```
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(100,)))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Answer Keys

1. a) C. ReLU
2. c) C. Sigmoid
3. c) C. Crée un réseau avec 1 couche cachée de 64 neurones et une couche de sortie
4. c) C. Prépare le modèle pour l'entraînement
5. a) B. Chaque entrée a 100 features
6. b) A. binary_crossentropy ne convient pas pour une sortie à 3 classes
7. a) B. Ignore aléatoirement 50% des neurones pendant l'entraînement
8. c) D. L'historique des métriques par époque (loss, accuracy, etc.)
9. a) B. En ajoutant Dropout() ou kernel_regularizer dans les couches
10. d) Dense(64, activation='relu', kernel_regularizer=l2(0.01))
11. b)
12. d)

