

Examen – Deep Learning

Durée : 2 heures. Votre copie sera notée sur 23, et la note ramenée sur 20 (exemple : 11/23 devient 11/20).
Il peut être judicieux de terminer par l'exercice 1, plus long à traiter. Documents et calculatrice non autorisés.

1. Compréhension des notions du cours (11 points)

Expliquez les notions suivantes (2-3 lignes pour chaque item – 1pt par item) :

- Décrire le principe et l'intérêt de la validation croisée.
- Décrivez ce qu'est une matrice de confusion et quelle est son utilité.
- Décrivez en quoi consiste le *dropout* et son utilité potentielle.
- Fonctionnement une couche de convolution.
- Fonctionnement des attributs *padding* et *stride* d'une couche convolutive.

Donnez, en quelques phrases le principe général des architectures suivantes et illustrer à chaque fois à l'aide d'un schéma aussi détaillé que possible (2pts par item) :

- Le fonctionnement d'un réseau autoencodeur.
- Un réseau de neurones récurrent de type LSTM (*Long-short term memory*).
- Le fonctionnement des réseaux antagonistes génératifs

2. Calcul du nombre de paramètres (4 points)

Calculez le nombre de paramètres (poids) des réseaux de neurones ci-dessous, en détaillant les calculs :
Indication : attention aux biais.

- 10 valeurs en entrée, une couche cachée dense de 5 neurones, 10 valeurs en sortie. Code :

```
model = keras.models.Sequential()  
model.add( keras.layers.Input((10,1)) )  
model.add( keras.layers.Dense(5, activation='relu'))  
model.add( keras.layers.Dense(10, activation='softmax'))
```

- 10 valeurs en entrée, deux couches cachées denses de 10 neurones, 1 valeur en sortie.

3. CNN, paramètres, résolution d'image (3 points)

On dispose d'une image de 240*160 pixels en noir et blanc en entrée. Voici les couches successives du CNN mis en place :

- 8 couches de convolution 2D (3*3) avec padding= "same" et stride=1
- Un Maxpooling 2*2
- 16 couches de convolution 2D (3*3) avec padding="same" et stride=1
- Un Maxpooling 2*1
- 32 couches de convolution 2D (5*5) avec padding=valid et stride=1
- Un Maxpooling 4*4

- Donner, pour chaque item, la « taille » de l'image intermédiaire ainsi obtenue.
- Pour chaque couche, donner le nombre de paramètres qui seront appris.

4. Codage en python avec Keras (5 points)

Décrire le plus précisément possible ce que font les instructions ci-dessous :

1. On suppose que x est de taille 32×32 – dans chaque cas indiquer ce que fait la couche et quelle est le format de la sortie `res` :

- `res = keras.layers.dense(16, activation="relu")`
- `res = keras.layers.Conv2D(32, 3, activation="relu", strides=1, padding="valid")(x)`
- `res = keras.layers.Conv2DTranspose(8, 3, activation="relu", strides=2, padding="same")(x)`
- `res = keras.layers.Flatten()(x)`
- `res = keras.layers.MaxPooling2D((4, 4))`

2. Si RNN est un modèle que l'on a compilé, que fait l'instruction suivante (donnez les détails, et le rôle des variables données en paramètre) ?

```
RNN.fit(tr_in, tr_out, batch_size = 50, epochs = 2, validation_data = (val_in, val_out))
```

3. Si `inputs` sont les données d'entrée, et `NN1`, `NN2`, `NN3` des modèles créés préalablement, à quoi correspond le modèle `model` créé à la suite des instructions suivantes ?

```
a = NN1(inputs)
b_1 = NN2(inputs)
b_2 = NN3(inputs)
x = keras.layers.concatenate([b_1, b_2], axis=1)
y = keras.layers.Dense(10, activation='softmax', name='cnn')(x)
model = keras.Model(inputs, [a, y])
```