

Rattrapage IA applications ING2

CY Tech

23 Mai 2023

1 Apprentissage par renforcement 1

Soit le graphe décrivant la mise en production d'un modèle d'IA avec quelques étapes importantes et la probabilité de transition entre celles-ci.



Les étapes sont : { Récupération des données (R), Labélisation des données (L), Prétraitement des données (P), Construction et entraînement d'un modèle (C), Test du modèle (T), Mise en production (M) } .

- Les probabilités de transition sont les suivantes :

- a) Récupération des données \rightarrow Labélisation avec une probabilité de 0.9 ou Récupération des données (0.1)
- b) Labélisation des données \rightarrow Prétraitement des données (0.75) ou Construction d'un modèle (0.2) ou Récupération (0.05),
- c) Construction et entraînement d'un modèle \rightarrow Test du modèle (0.8) ou Mise en production (0.2),
- d) Test du modèle \rightarrow Mise en production (0.5) ou Récupération des données (0.25) ou Labélisation (0.25),
- e) Mise en production \rightarrow Mise en production (1).

Les récompenses associées sont de 10 pour la mise en production, 8 pour une arrivée sur le Test du modèle, 1 pour la Construction du modèle et 4 pour la Récupération des données. Les autres seront à 0.

- 1. Faire un schéma récapitulatif et donner la matrice de transition.
- 2. Donner trois exemples de suites d'états possibles.
- 3. En utilisant le système de récompense, expliquer comment calculer la valeur de chaque état.

2 Apprentissage par renforcement 2

On considère un agent qui se déplace sur le terrain de la figure 1 ci-après : où la case verte (ligne 2, colonne 3) est l'objectif à atteindre. Les cases noires sont des obstacles.

- 4. Proposer une modélisation (détaillée) de ce problème.
- 5. Donner la définition d'une stratégie et donner un exemple de stratégie non optimale et un autre d'une stratégie optimale.
- 6. Donner la différence entre les méthodes model-based et model free. Pour chacune d'elles, donner un algorithme.

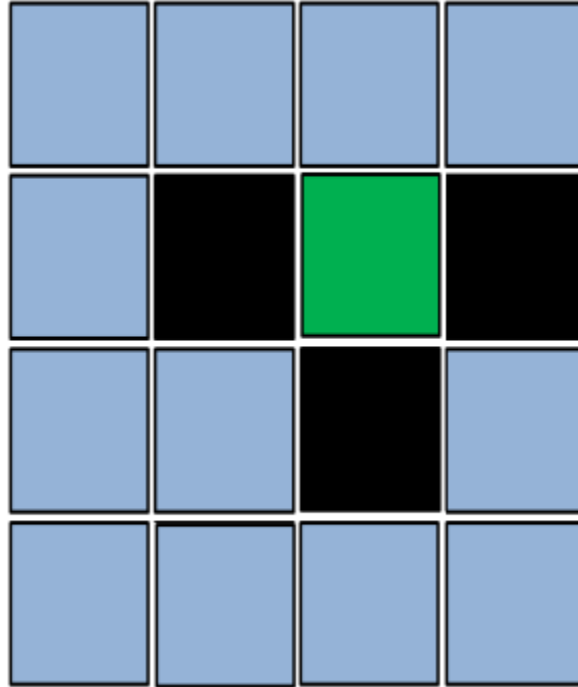


Figure 1: Gridworld

3 Deep Learning

On considère le jeu de données CIFAR10 constitué de 60000 images représentant les classes d'animaux et objets avec leurs labels :

Description	Label
Avion	0
Automobile	1
Oiseau	2
Chat	3
Cerf	4
Chien	5
Grenouille	6
Cheval	7
Bateau	8
Camion	9

NB : les classes sont équilibrées : il y a autant d'images pour chaque classe.

7. Sur la figure 2, commenter les étapes 2 et 3.
8. Sur la figure 3, décrire (sans répétition) les différentes couches.
9. À partir de la figure 3, retrouver les hyperparamètres manquants.
10. Sur la figure 5, commenter/décrire/expliciter les hyperparamètres *loss*, *optimizer* et *metric*.



```
# Load the data and split it between train and test sets
(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()

# Model / data parameters
num_classes = 10
input_shape = (32, 32, 3)

# étape 1
x_train = x_train.astype("float32") / 255
x_test = x_test.astype("float32") / 255

# étape 2
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
```

Figure 2

```
model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

model.summary()
```

Figure 3

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dropout (Dropout)	(None, 2304)	0
dense (Dense)	(None, 10)	23050
Total params: 42,442		
Trainable params: 42,442		
Non-trainable params: 0		

Figure 4

```
batch_size = 128
epochs = 15

model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])

model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, validation_split=0.1)
```

Figure 5

4 Optimisation

4.1 Compréhension

11. Définir la notion de problème à variables discrètes et problème à variables continues. Citer une méthode pour résoudre chacun d'entre eux.
12. On souhaite utiliser une métaheuristique pour résoudre un problème d'optimisation dont on ne connaît absolument pas l'optimum. Comment évaluer la pertinence de sa réponse et sa stabilité ?
13. Définir la notion de convergence pour l'algorithme du recuit simulé.

4.2 Application

Nous souhaitons générer de manière automatique l'emploi du temps des ING2 GSI et MI, considérant le fait qu'il n'y a qu'un seul groupe par parcours.

Générer un emploi du temps consiste à positionner un ensemble de n créneaux de 1h30 dans la semaine (on supposera qu'il existe au moins une configuration admissible).

Un créneau consiste à affecter une heure de début, un cours, un groupe d'étudiants et une salle (avec une capacité).

Un cours représente un couple prof / séance. Une séance est un CM ou un TD.

Entre deux créneaux, il est nécessaire d'avoir 15 min de pause réglementaire et on ne fait pas cours avant 8h, de 12h30 à 13h30 et après 20h.

14. Proposer une représentation pour décrire une solution candidate.
15. Étudier les contraintes d'admissibilité/faisabilité (explicites et implicites).
16. Quelle métaheuristique pourriez-vous utiliser pour résoudre ce problème ? Justifier.