

TRAIN A SMARTCAB TO DRIVE

Módulo de Aprendizagem por Reforço
Nanodegree Engenheiro de Machine Learning Udacity

Tarefas

TAREFA 1:

Implementar um Agente Condutor Básico

Para começar, sua única tarefa é fazer o táxi inteligente movimentar-se ao redor do ambiente. Nesse momento, você não deverá se preocupar com nenhuma política de otimização de condução. Note que o agente condutor está dando as seguintes informações a cada intersecção:

- O próximo ponto de navegação é relativo a sua própria localização e direção.
- O estado do semáforo na intersecção e a presença de veículos em direção contrária vindo de outras direções.
- O atual tempo que sobrou do prazo estipulado. Para completar essa tarefa, simplesmente faça com que seu agente condutor escolha uma ação qualquer do conjunto de ações possíveis (None , 'forward' , 'left' , 'direita') a cada intersecção, desconsiderando a informação de entrada acima. Defina a simulação de execução do prazo final, enforce_deadline para False e observa como ele executa

TAREFA 2:

Informar o Agente condutor

Agora que seu agente condutor é capaz de se movimentar pelo ambiente, sua próxima tarefa é identificar um conjunto de estados que são apropriados para a modelagem do táxi inteligente e o ambiente. A fonte principal de variáveis de estado são as atuais entradas na intersecção, mas nem todas podem precisar de representação. Você pode optar por definir explicitamente os estados ou usar algumas combinações de entrada como um estado implícito. A cada passo de tempo, processe as entradas e atualize o estado atual do agente utilizando a variável `self.state`. Continue com a obrigação do prazo da simulação `enforce_deadline` sendo ajustado para `False` e observe como seu agente condutor agora reporta a mudança de estado ao progredir da simulação.

TAREFA 3:

Implementar um Agente Condutor Q-Learning

Com o seu agente condutor bem capacitado para interpretar as informações de entrada e ter um mapeamento dos estados do ambiente, sua próxima tarefa é implementar um algoritmo QLearning para o seu agente condutor escolher a melhor ação a cada passo, baseado nos Qvalues do estado e ação atuais. Cada ação tomada pelo táxi inteligente vai produzir uma recompensa que dependerá do estado do ambiente. O agente condutor QLearning vai precisar considerar essas recompensas ao atualizar os Qvalues. Uma vez implementado, ajuste a obrigação do prazo da simulação `enforce_deadline` para `True` . Execute a simulação e observe como os táxis inteligentes se movimentam no ambiente em cada tentativa.

TAREFA 4:

Melhorar o Agente Condutor Q-Learning

Sua última tarefa para este projeto é melhorar seu agente condutor até que, depois de treinado o suficiente, o táxi inteligente seja capaz de alcançar o destino dentro do tempo alocado de maneira segura e eficiente. Parâmetros do algoritmo do QLearning, como a taxa de aprendizagem (α), o fator de desconto (γ) e a taxa de exploração (ϵ), todos contribuem para a habilidade do agente condutor de aprender a melhor ação para cada estado. Para melhorar o sucesso do seu táxi inteligente:

- Ajuste o número de tentativas, `n_trials` , na simulação para 100.
- Execute a simulação com obrigação do prazo `enforce_deadline` , ajuste para `True` (você vai precisar reduzir o atraso da atualização `update_delay` e ajuste o `display` para `Falso`).
- Observe a aprendizagem do agente condutor e a taxa de sucesso do táxi inteligente particularmente durante os últimos testes.
- Ajuste um ou vários dos parâmetros acima e itere esse processo. Esta tarefa estará completa uma vez que você tenha chegado naquilo que você determinou como melhor combinação de parâmetros requisitados para que o agente condutor aprenda com sucesso.

PERGUNTAS

PERGUNTA 1:

Observe o que você vê do comportamento do agente ao realizar ações aleatórias. O táxi inteligente eventualmente chega ao seu destino? Há outras observações interessantes a serem feitas?

PERGUNTA 2:

Quais estados definidos por você são apropriados para modelar o táxi inteligente e o ambiente? Por que você acredita que cada um desses estados são apropriados para esse problema?

PERGUNTA OPCIONAL: Quantos estados no total existem para o táxi inteligente nesse ambiente? Esse número parece razoável dado que o objetivo do QLearning é aprender e tomar decisões informadas sobre cada estado? Por que e por que não?

PERGUNTA 3:

Reporte os valores diferente para os parâmetros sintonizados em sua implementação básica de QLearning. Para quais conjuntos de parâmetros o agente melhor se desempenha? Quão bom é o último desempenho do agente condutor?

PERGUNTA 4:

Seu agente se aproxima de encontrar um política ótima, por exemplo, chegar ao destino no tempo mínimo possível e sem incorrer nenhuma penalidade? Como você descreveria uma política ótima para este problema?