



PAC 3 – Optimització

Presentació

En aquesta PAC dissenyarem un procés basat en algorismes genètics per tal d'optimitzar el rendiment d'un sistema de vigilància i monitorització satel·litari.

Competències

En aquest enunciat es treballen en un determinat grau les següents competències general de màster:

- Capacitat per a projectar, calcular i dissenyar productes, processos i instal·lacions en tots els àmbits de l'enginyeria en informàtica.
- Capacitat per al modelat matemàtic, càlcul i simulació en centres tecnològics i d'enginyeria d'empresa, particularment en tasques d'investigació, desenvolupament i innovació en tots els àmbits relacionats amb l'enginyeria en informàtica
- Capacitat per a l'aplicació dels coneixements adquirits i de solucionar problemes en entorns nous o poc coneguts dins de contextes més amplis i multidisciplinars, essent capaços d'integrar aquests coneixements.
- Posseir habilitats per a l'aprenentatge continuat, autodirigit i autònom.
- Capacitat per a modelar, dissenyar, definir l'arquitectura, implantar, gestionar, operar, administrar y mantenir aplicacions, xarxes, sistemes, serveis i continguts informàtics.
- Capacitat per assegurar, gestionar, auditar i certificar la qualitat dels desenvolupaments, processos, sistemes, serveis, aplicacions i productes informàtics.



Les competències específiques d'aquesta assignatura que es treballen són:

- Entendre que és l'aprenentatge automàtic en el context de la Intel·ligència Artificial
- Distingir entre els diferents tipus i mètodes d'aprenentatge
- Aplicar les tècniques estudiades a un cas concret

Objectius

En aquest PAC es practicaran els conceptes del temari relacionat amb optimització, en una vessant pràctica amb dades sintètiques.

Descripció de la PAC a realitzar

Context

Un important centre d'investigació en biologia i ciències ambientals està centrant els seus esforços en identificar i controlar una plaga d'un determinat fong desconegut fins ara, que ataca de manera diferent a multitud d'espècies vegetals, posant en perill la salut dels boscos del sud d'Europa. Com la majoria de plagues, si es detecta un focus aviat, és fàcil controlar-ho i extingir-ho. A mesura que avança el temps sense actuar sobre un nou focus, més difícil és poder eradicar-ho.

Aquest centre té accés a dades obtingudes gràcies a un satèl·lit especialitzat en monitorització de vegetació i biomassa. Les dades que es necessiten per poder identificar un nou focus d'aquest tipus de fong consisteixen en imatges de 5 tipus:

- Imatges RGB
- Imatges tèrmiques
- Imatges infrarojes
- Imatges RADAR
- Imatges LiDAR



Les imatges provinents del satèl·lit es transmeten continuament en paquets de no més de 1000 imatges en total (la suma de tots els tipus d'imatges), i a on hi ha, almenys, 10 imatges de cada tipus, i com a màxim, 255 imatges d'un mateix tipus.

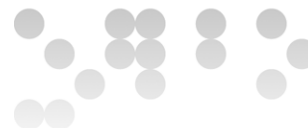
A mesura que es van rebent imatges, aquestes són processades continuament per un complex software que determina si hi ha prou indicis en alguna determinada àrea de les imatges que facin sospitar de la presència d'un focus. Aquest pas és clau per a indicar al satèl·lit que se centri en obtenir més fotos d'aquesta zona, amb millor resolució, per tal de, posteriorment, determinar si realment és necessari enviar-hi una brigada d'extinció.

Aquest software està preparat per processar seqüencialment (un darrere l'altre) els paquets que van arribant des del satèl·lit, i necessita, per cada àrea que s'explora, una determinada quantitat d'imatges de cada tipus per emetre una estimació sobre si s'ha d'analitzar amb millor resolució aquesta zona o no.

Aquest procés pot trigar hores en dur-se a terme, però aquest temps és variable, en funció de la quantitat d'imatges de cada tipus que conté el paquet que està processant, i la proporció entre els diferents tipus d'imatges.

Els enginyers de software d'aquest centre, conscients d'aquest fet, han dissenyat un simulador d'aquest programa que, donada una representació de paquet d'imatges, et retorna el temps estimat (en ms) que trigarà el software en donar resposta. La intenció d'aquest simulador és fer-ho servir per determinar quina és la millor configuració de paquet (quina és la millor quantitat d'imatges de cada tipus per paquet) per obtenir resposta en la menor quantitat de temps, ja que aquesta configuració desitjada pot ser informada al satèl·lit per a què l'adopti en la generació dels paquets que envia.

És molt important optimitzar aquest procés, donat que el satèl·lit només passa un cop per dia durant un interval limitat de temps per la nostra zona d'interès, així que hem de ser capaços d'indicar ràpidament si cal millorar la resolució de les imatges d'una determinada àrea, o si no haurem d'esperar a que ho faci al dia següent.



Activitats

La solució a aquesta pràctica es dissenyarà i executarà amb la llibreria DEAP de Python.

Cal justificar les decisions preses i analitzar els resultats obtinguts.

Activitat 1

Dissenyeu una codificació per a la representació d'un paquet de dades, pensant que aquest paquet de dades serà l'element bàsic de treball (un individu o genoma) d'un algorisme genètic que optimitzarà la seva estructura per tal de minimitzar el temps de resposta del simulador. Expliqueu la seva estructura i com la programareu a DEAP.

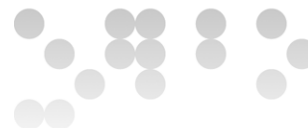
Recordeu que els paquets consten d'una determinada quantitat d'imatges, i aquestes imatges són de 5 tipus diferents.

Aquesta representació ha de ser diferent a la "trivial" que se us proporciona als fitxers adjunts. Recordeu també que una bona representació ha de permetre els mecanismes de creuament i mutació, i que quant més granular sigui aquesta representació, més efectius són aquests mecanismes.

Important: raoneu i justifiqueu la vostra decisió.

Activitat 2

El fitxer *P3.py* serà l'executable del nostre programa. Teniu diversos forats a omplir per completar les següents qüestions. També haureu de completar el fitxer *analyzer.py*. Aquest últim conté les definicions de les funcions que cridarà el nostre algorisme genètic per dur a terme l'avaluació de cada individu (és a dir, obtenir quant temps trigaria en ser processat). El funcionament resumit és que a P3 farem evolucionar les poblacions durant un cert nombre de generacions. Per a això, farem servir les funcions `analyze_performance` i la funció que haureu d'implementar per decodificar (`decode`) del fitxer *analyzer.py*.



La funció `analyze_performance` només l'heu de modificar per canviar la crida a la funció de decodificació, tal i com s'hi indica al propi fitxer. Aquesta funció farà servir la funció `simulator` per simular el temps de resposta.

IMPORTANT: el codi proporcionat no està preparat per ser executat fins que no es completi, almenys, l'apartat a). Seguiu sempre les indicacions marcades en comentaris precedits per `#TODO`. No està permès modificar la funció `simulator` ni el fitxer `proportion_factors.py`.

Prepareu l'algorisme genètic amb DEAP per resoldre el problema d'optimització plantejat. Concretament:

- a) Completeu els passos que ja teniu disponibles al fitxer `P3.py` per poder dur a terme el procés evolutiu amb valors variables de nombre de generacions, mida de les poblacions, probabilitat de creuament i probabilitat de mutació. Consulteu la documentació oficial de DEAP per escollir el mètode de mutació que millor s'adapti a les vostres necessitats.
- b) Implementeu una funció de decodificació de genoma que rebí un individu codificat segons la vostra proposta de l'activitat anterior, i retorni una tupla de 5 integers (cadascuna de les quantitats d'imatges de cada tipus). Podeu veure un exemple de decodificació trivial, on independentment de l'individu rebut, la decodificació és una tupla d'enters entre 50 i 100.
- c) Recolçant-vos en la funció de decodificació que acabeu de crear, completeu la implementació de la funció `fulfill_constraints`, que és el decorador que permet satisfer les restriccions sobre la quantitat d'imatges que conté el paquet de dades, i la quantitat d'imatges de cada tipus. Feu servir els arguments `max_imagery` i `min_imagery_type`.

Descomenteu les línies indicades per decorar els mètodes de generació de població, creuament i mutació, i comproveu el correcte funcionament del vostre codi.



Activitat 3

- Expliqueu com funcionen els operadors de creuament i mutació del programa. Quina és la finalitat de cadascun?
- Quines conseqüències tindria prescindir de l'operació de mutació?
- Expliqueu breument una codificació alternativa a la proposada a l'Activitat 1 que també permetés aplicar aquests operadors.

Activitat 4

Per a aquesta activitat, heu de realitzar un canvi al codi. A la funció `analyze_performance(individual)` del fitxer `analyzer.py`, heu de canviar el bucle `while`, de manera que passarà de ser així:

```
#Exercise 4 – Comment next line:
sim_time, remaining_samples = simulator(samples, remaining_samples)
#Exercise 4 – Uncomment next line:
#sim_time, remaining_samples = simulator(samples, remaining_samples, True)

analysis_time += sim_time
```

A ser així:

```
#Exercise 4 – Comment next line:
#sim_time, remaining_samples = simulator(samples, remaining_samples)
#Exercise 4 – Uncomment next line:
sim_time, remaining_samples = simulator(samples, remaining_samples, True)

analysis_time += sim_time
```

És a dir, comentem una línia, i descomentem altra.

- Amb aquesta configuració, és possible que en una generació l'individu guanyador ens doni un temps de simulació superior al de l'individu guanyador de l'anterior generació? Per què?
- Comproveu i discutiu el rendiment de l'algorisme genètic, tenint en compte l'efecte del nombre d'individus per població, nombre de generacions, i taxes de creuament i mutació.

Mostreu gràfiques de convergència cap a una solució en funció del nombre de generacions i del nombre d'individus per població per a les 3 combinacions de taxes de creuament i mutació que considereu més interessants. Dueu a terme varies execucions amb la mateixa configuració abans de treure conclusions.



Recursos

Bàsics

Tutorial de DEAP penjat a l'aula. Vídeo de l'assignatura sobre optimització.

Fitxers de codi adjunts.

Complementaris

Manual de teoria de l'assignatura.

Documentació de DEAP:

<http://deap.readthedocs.org/en/master/>

Criteris de valoració

Els exercicis tindran la següent valoració associada:

Activitat 1: 2 punts

Activitat 2: 4.5 punts (1.5 punts per apartat)

Activitat 3: 1.5 punts

Activitat 4: 2 punts

Raoneu la resposta en tots els exercicis. Les respostes sense justificació no rebran puntuació.

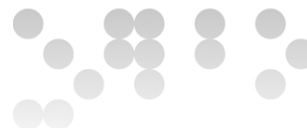
Format i data de lliurament

La PAC s'ha de lliurar abans del proper **20 de maig (inclòs)**.

La solució a entregar consisteix en un informe en format PDF fent servir aquest enunciat com a plantilla més els fitxers de codi (*.py) que heu creat per resoldre la prova. Tots aquests fitxers s'han de comprimir en un fitxer ZIP.

Adjunteu el fitxer a un missatge a l'apartat de **Lliurament i Registre d'AC (RAC)**. El nom del fitxer ha de ser *CognomsNom_IA_PAC3* amb l'extensió *.zip*.

Per a dubtes i aclariments sobre l'enunciat, adreceu-vos al consultor responsable de la vostra aula.



Nota: Propietat intel·lectual

Sovint és inevitable, en produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho en el marc d'una pràctica dels estudis del Màster en Informàtica, sempre i això es documenti clarament i no suposi plagi en la pràctica.

Per tant, en presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en què es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU, GPL ...). L'estudiant haurà d'assegurar-se que la llicència que sigui no impedeix específicament seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright.

Hauran, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.