

Virtual Trainer

Progetto di Fondamenti di Intelligenza Artificiale

Daniele Fabiano
Mariantonietta Maselli

Progetto combinato?

Non proprio

Virtual Trainer nasce con l'idea di integrare un modulo di intelligenza artificiale in un *software* ben più grande, ma... per questioni logistiche, non è stato possibile mettere insieme i due sistemi.

Caratteristiche salienti di SmartGym:

- applicazione di supporto agli atleti di una palestra e ai personal trainer
- dematerializzazione della scheda esercizi
- piena libertà all'utente nella scelta degli esercizi da eseguire



Come si potrebbe migliorare questo software?

Il processo di creazione di una scheda esercizi può essere lungo e noioso.

I più inesperti e gli indecisi potrebbero trovarsi in difficoltà... così tanti esercizi, quali scegliere?

I più impazienti potrebbero voler passare subito all'allenamento, lasciando a qualcun altro il compito di scegliere gli esercizi.

Ecco il perché di Virtual Trainer:

- Automatizzare il processo di creazione di una scheda degli esercizi
- Utilizzare l'intelligenza artificiale per creare schede uniche, adatte alle esigenze e alle caratteristiche fisiche di ciascun atleta



Formulazione del problema

Stati: ogni esercizio aggiunto/rimosso dalla scheda definisce un cambio di stato.

Azioni: l'aggiunta e la rimozione di un esercizio.

$\{add(es.1), add(es.2)...\},$
 $\{remove(es.1), remove(es.3)\}$

Modello di transizione: $scheda1\{(es.1),(es.2)\}$

$risultato(scheda1, add(es.3)) = scheda1\{(es.1),(es.2),(es.3)\}$

Test obiettivo: la scheda ha raggiunto il numero prestabilito di esercizi.

Specifiche PEAS

Performance: *come definiamo la misura di prestazione?*

Attraverso l'**affinità** degli esercizi e della scheda. Il significato di affinità sarà approfondito successivamente.

La scheda con affinità maggiore è quella migliore, l'agente è performante quando restituisce una scheda con affinità alta.

Specifiche PEAS

Environment: *come definiamo l'ambiente?*

Nell'ambiente sono compresi:

- gli **esercizi** candidati per comporre una scheda.
- l'**atleta** che desidera usufruire della generazione della scheda.

Specifiche PEAS

Actuators: *come fa l'agente a intraprendere azioni?*

L'agente *riceve* informazioni sugli esercizi e sull'atleta.

È necessario che sia in grado di *processarle* attraverso delle **funzioni**.

I risultati delle funzioni *influenzeranno* l'azione di aggiunta di un esercizio alla scheda.

Specifiche PEAS

Sensors: *come fa l'agente a percepire l'ambiente?*

In un archivio di dati dove saranno memorizzate:

- le informazioni relative alle **proprietà fondamentali degli esercizi** per la risoluzione del problema (nome, parte del corpo allenata, scopo dell'esercizio).
- le **caratteristiche dell'atleta** (peso, altezza, esperienza nella pratica della palestra).

Proprietà dell'ambiente	Virtual Trainer...
Completamente osservabile	conosce in ogni istante lo stato dell'ambiente
Deterministico	sceglie il prossimo esercizio da inserire sulla base degli esercizi già scelti
Sequenziale	è influenzato nella scelta dagli esercizi precedentemente selezionati
Statico	sceglie gli esercizi da un insieme fisso
Discreto	sceglie gli esercizi tra un numero di esercizi limitati
Singolo	è l'unico agente coinvolto

Una possibile soluzione...



Algoritmi genetici!

Strumenti di intelligenza artificiale ispirati alla teoria dell'evoluzione di Darwin:

- Selezione naturale
- Adattamento
- Teoria della sopravvivenza

...vengono utilizzati per risolvere problemi di ottimizzazione computazionalmente difficili.

Perché questa scelta?



1

Funzionamento

Evolvono le soluzioni candidate e producono individui di volta in volta sempre migliori rispetto ad una funzione obiettivo, fino a raggiungere l'ottimo o un'altra condizione di terminazione.



2

Tecnica di ricerca veloce

Esplorano rapidamente lo spazio di ricerca e con una buona configurazione vengono prodotti risultati vicini all'ottimo in tempi ragionevoli.



3

Valida scelta per problemi multiobiettivo

Nel nostro caso, sarà la media di due funzioni a determinare l'adeguatezza degli individui.

Setup dell'algoritmo



1

Come abbiamo codificato gli individui?

Come un array di interi, ogni intero fa riferimento all'identificativo univoco di ciascun esercizio.



2

Come viene inizializzata la popolazione?

- Esercizi scelti tra una o più parti del corpo indicate dall'utente.
- O, con l'intero set di esercizi disponibili.



3

Come vengono valutati gli individui?

$$fitness(s) = \langle Avg(affinità), Avg(sfida) \rangle$$

Funzione di affinità

- Dell'esercizio rispetto al traguardo dell'atleta.

Abbiamo individuato tre traguardi:

- Mettere massa
- Tonificare
- Dimagrire

Il traguardo sarà stabilito calcolando l'IMC dell'atleta e utilizzando una tabella che indica il traguardo per ciascun valore dell'IMC.

** utilizzabile da persone in salute.*



Funzione di affinità

- Dell'esercizio rispetto al traguardo dell'atleta.

Ogni esercizio ha una tipologia:

- **Aerobico:** camminata, corsa, saltare la corda...
- **Anaerobico:** piegamenti, trazioni, squat...

A seconda della tipologia, sarà più indicato per raggiungere un certo traguardo rispetto a un altro.



Funzione di affinità

- Dell'esercizio rispetto al traguardo dell'atleta.

Sfruttando il **valore dell'IMC dell'atleta** e la **tipologia dell'esercizio**, abbiamo costruito una tabella che restituisce il valore della funzione di affinità per ciascun caso.

Si cercherà di massimizzare la seguente funzione:

$$affinita(a, e) = y$$



Funzione di sfida

- Per l'atleta, rispetto alla difficoltà dell'esercizio.

Distinguiamo quattro stadi di esperienza per l'atleta:

- Principiante
- Intermedio
- Esperto
- Massimo esperto

Un esercizio può essere:

- Facile
- Medio
- Difficile



Funzione di sfida

- Per l'atleta, rispetto alla difficoltà dell'esercizio.

Si cercherà di massimizzare la seguente funzione:

$$sfida(a, e) = SFIDAMAX - |esperienza_{atleta} - difficoltà_{esercizio}|$$



Preference sorting

Trovandoci di fronte a un vettore di funzioni di fitness, avremo bisogno di selezionare le soluzioni ottimali tra molte opzioni.

Il **preference sorting** ci permette di classificare le soluzioni ottenute in base alle preferenze specificate. Effettueremo il preference sorting rispetto alla funzione di affinità.

Vincoli del problema

A metà della generazione della scheda: l'algoritmo si fermerà per chiedere maggiori informazioni all'utente su come proseguire.

Al termine di ogni iterazione: verrà presentata la scheda all'utente e avrà la possibilità di conservare la scheda corrente, scegliere la precedente, o generarne una nuova.

Stopping condition

L'algoritmo si ferma dopo un certo numero di generazioni determinate.

Operatori genetici

L'algoritmo migliore per ogni operatore genetico è stato stabilito mediante la sperimentazione empirica.

Selezione:

- *3-way tournament*
- *Truncation*

Crossover:

- *One point*
- *Two point*

Mutazione:

- *Random resetting*

Implementazione

Strumenti utilizzati:

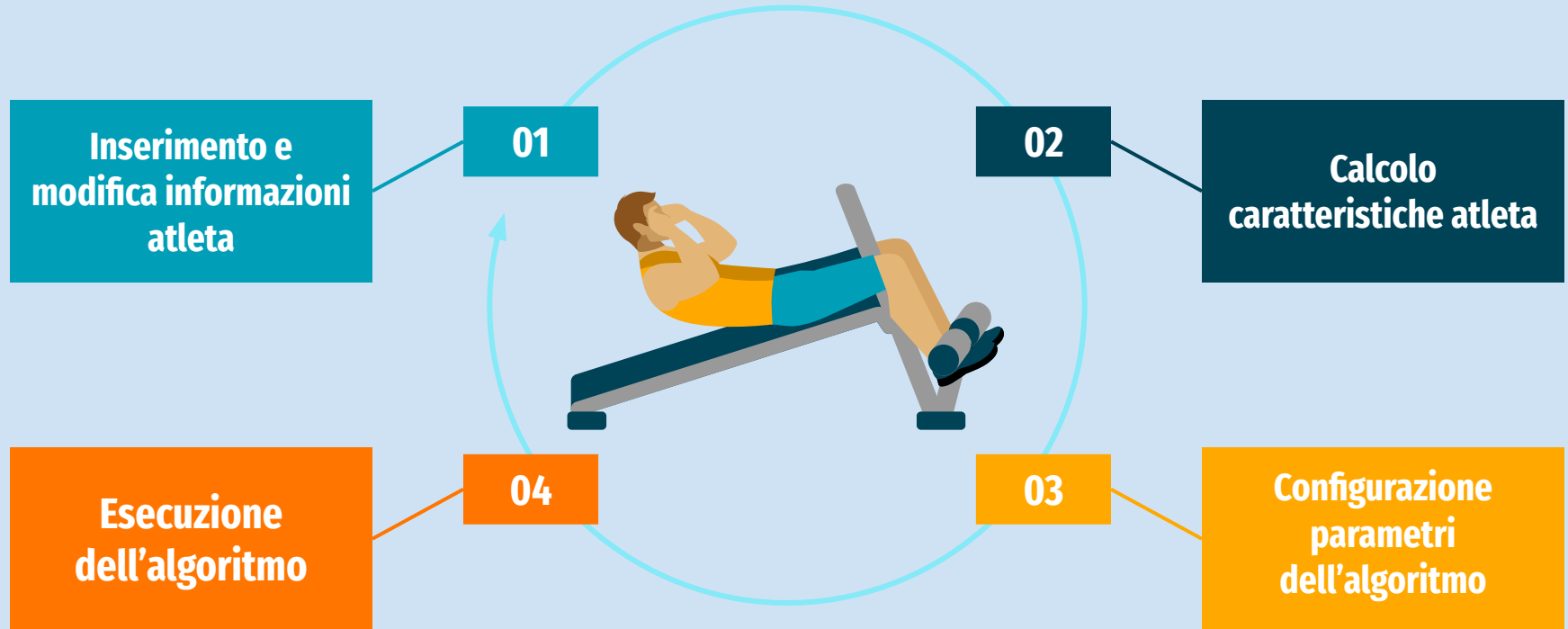
- Python
- Framework DEAP



DISTRIBUTED
EVOLUTIONARY
ALGORITHMS IN
PYTHON

** l'attuale implementazione è una versione semplificata dell'algoritmo che non tiene conto dei vincoli relativi all'interattività con l'utente.*

Il codice in breve



Fine

Slide di backup

IMC	Categoria	Traguardo
16 <= IMC < 19	Sottopeso	Mettere Massa
19 <= IMC < 20	Tendente Sottopeso	Mettere Massa
20 <= IMC < 24	Peso forma	Tonificare
24 <= IMC < 25	Tendente Sovrappeso	Dimagrire
25 <= IMC < 30	Sovrappeso	Dimagrire

IMC	Tipologia Esercizio	Affinità
16 <= IMC < 19	Aerobico	100
16 <= IMC < 19	Anaerobico	300
19 <= IMC < 20	Aerobico	200
19 <= IMC < 20	Anaerobico	300
20 <= IMC < 24	Aerobico	300
20 <= IMC < 24	Anaerobico	300
24 <= IMC < 25	Aerobico	300
24 <= IMC < 25	Anaerobico	200
25 <= IMC <= 30	Aerobico	300
25 <= IMC <= 30	Anaerobico	100



$$\text{esperienza}(a) = 120 + (m * 10)$$

Principiante: $120 \leq e \leq 360$

Intermedio: $370 \leq e \leq 600$

Esperto: $610 \leq e \leq 840$

Massimo esperto: $e > 840$

Facile: $d = (120 + 360)/2 = 240$

Medio: $d = (370 + 600)/2 = 485$

Difficile: $d = (610 + 840)/2 = 725$

SFIDAMAX = 800

$$sfida(a, e) = SFIDAMAX - | \text{esperienza}_{\text{atleta}} - \text{difficoltà}_{\text{esercizio}} |$$



Selezione:

1. *3-way tournament*
2. *Truncation*

Crossover:

3. *One point*
4. *Two point*

Mutazione:

5. *Random resetting*

- Configurazione (1, 4, 5):

AVG(affinità(s)) = 273.33, AVG(sfida(s)) = 428.66

- **Configurazione (2, 4, 5):**

AVG(affinità(s)) = 286.66, AVG(sfida(s)) = 533.83

- Configurazione (1, 3, 5):

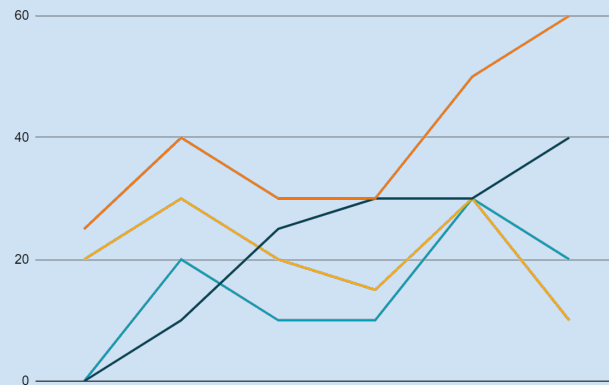
AVG(affinità(s)) = 273.32, AVG(sfida(s)) = 477.33

- Configurazione (2, 3, 5):

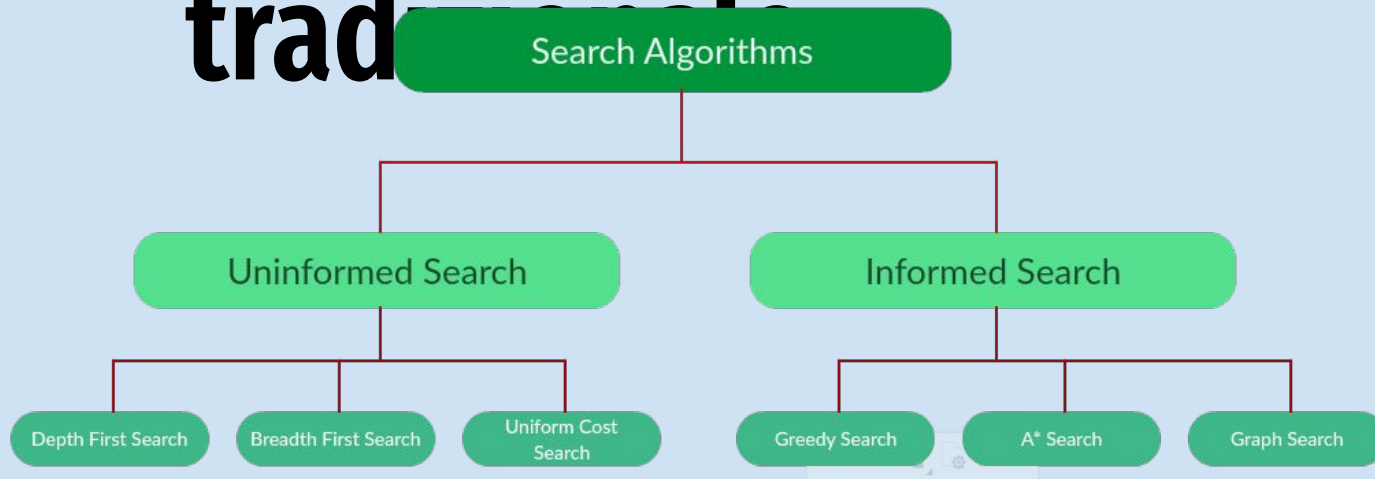
AVG(affinità(s)) = 273.32, AVG(sfida(s)) = 509.40

Proof of Concept

Abbiamo cercato più soluzioni per risolvere il problema, tutte scartate per diversi motivi.



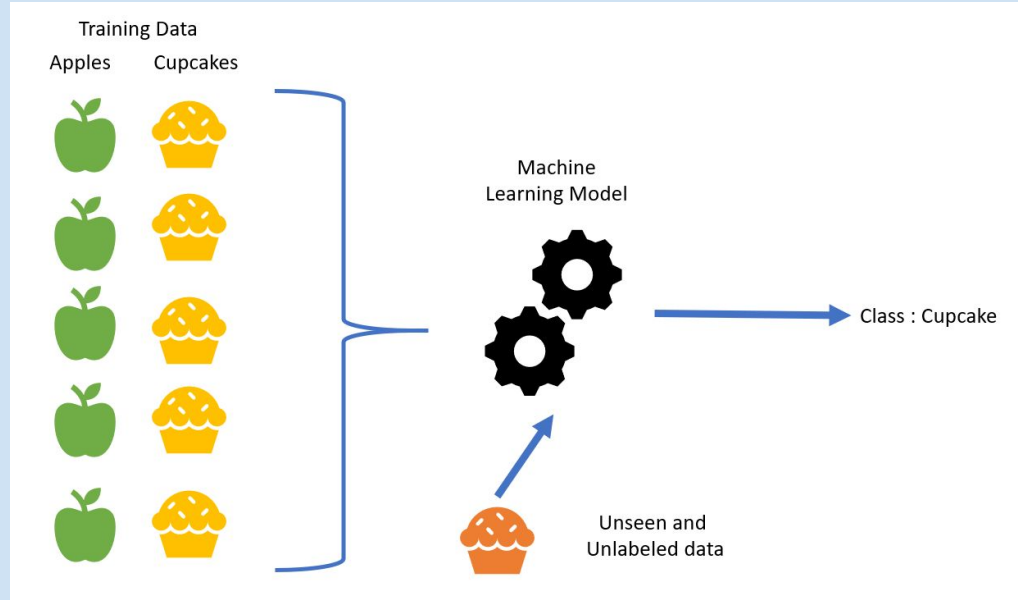
Ricerca tradizionale



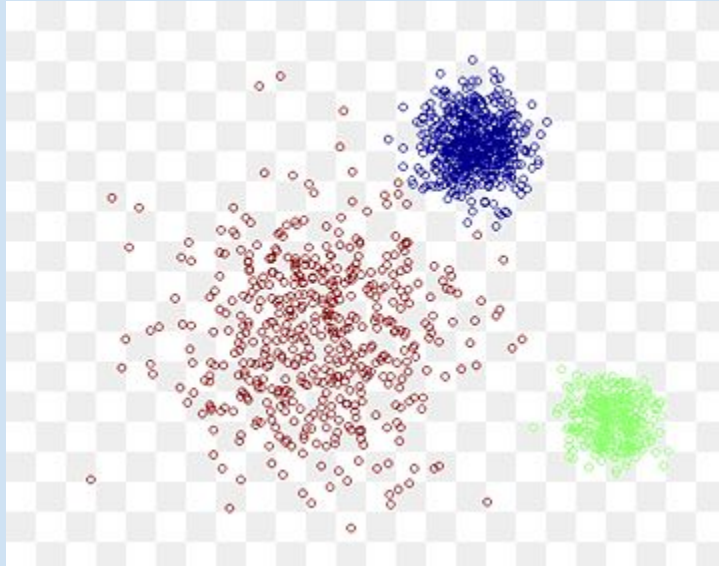
Battaglia all'ultimo muscolo



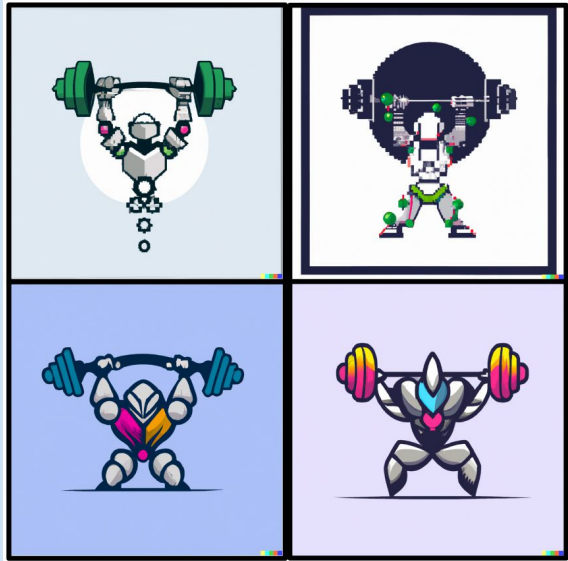
BMIClassifier



BodyClustering



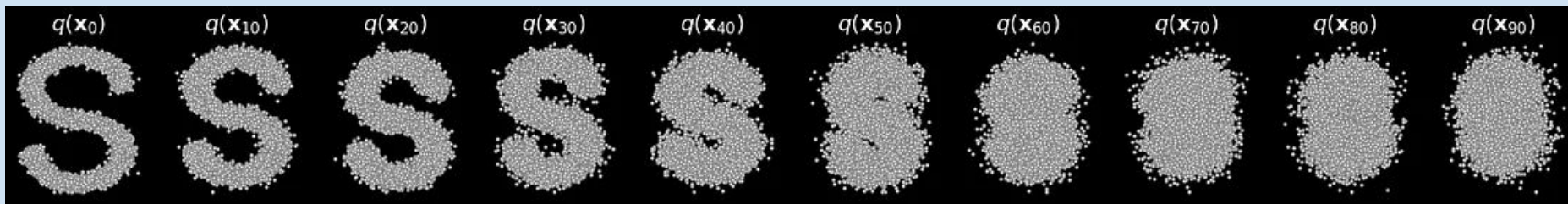
Curiosità



Il quadretto di immagini è stato generato attraverso uno strumento di Deep Learning chiamato DALL-E 2 di OpenAI.

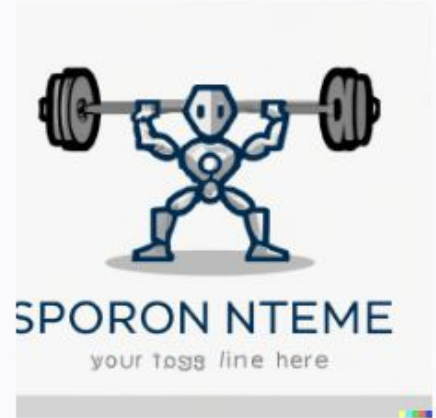
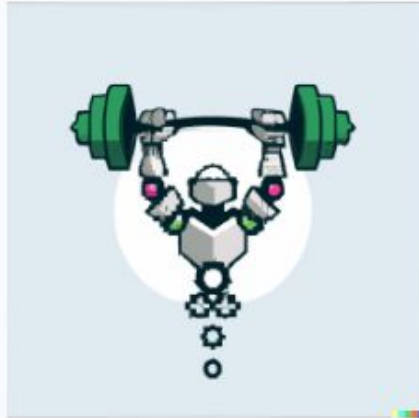
 DALL-E 2

Diffusion

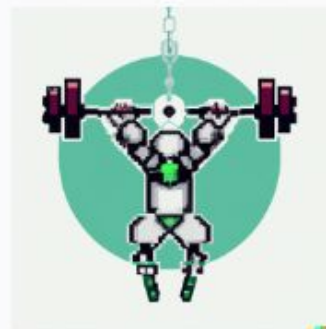
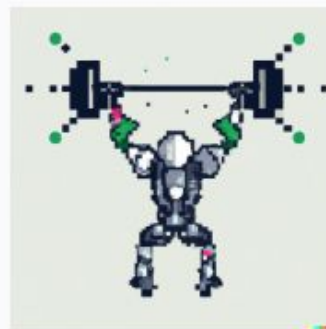
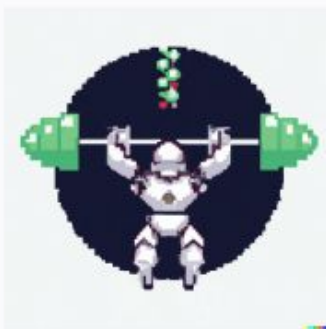


Frase 1:

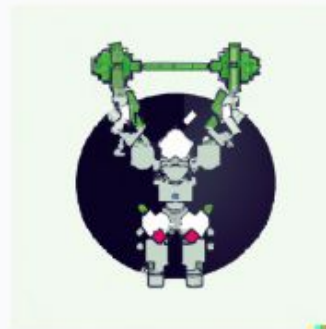
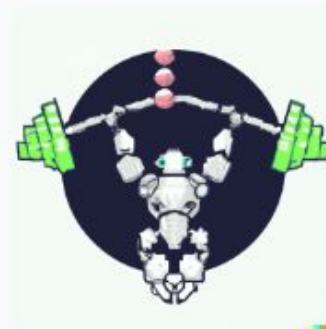
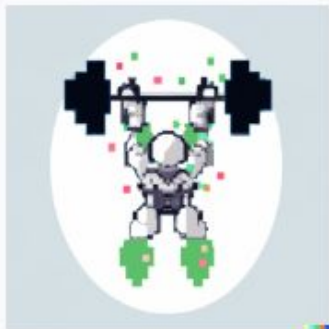
**a humanoid robot lifting a barbell, no human, cartoon, designer logo,
vector logo, award winning design, masterpiece, strong brand**



ORIGINAL



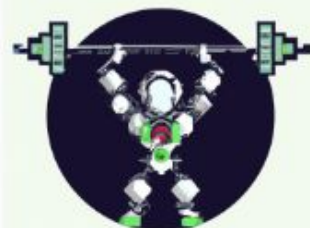
ORIGINAL



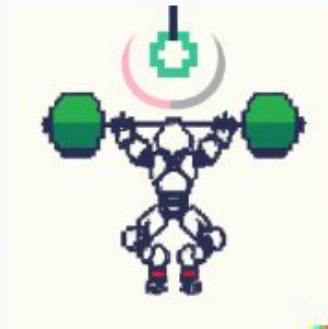
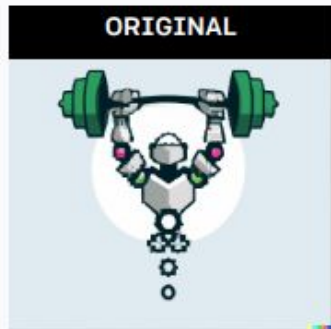
ORIGINAL



ORIGINAL

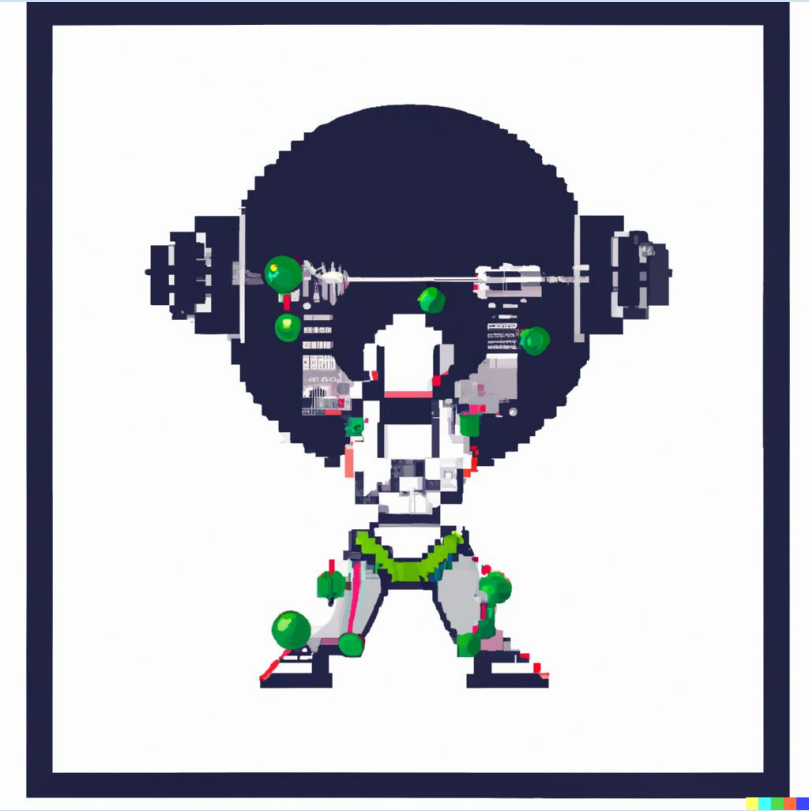


ORIGINAL



ORIGINAL





FRASE 2:

a humanoid robot lifting a barbell, cartoon, designer logo, vector logo, award winning design, masterpiece, strong brand



ORIGINAL



ORIGINAL



ORIGINAL



ORIGINAL



ORIGINAL

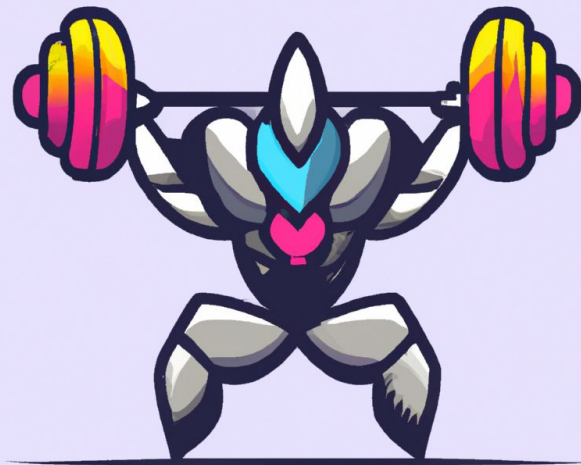
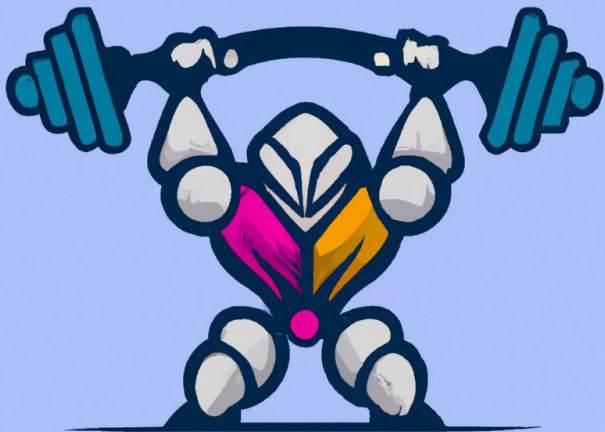


ORIGINAL



ORIGINAL





1) Ricerca tradizionale

Intendono risolvere il problema individuando lungo l'albero una sequenza di azioni che formano il cammino che porta allo stato obiettivo, ma...

...risultava complicato individuare una corretta sequenza. Analizzando il nostro problema, non siamo riusciti a definire la soluzione come un cammino verso l'obiettivo.

2) Ricerca con avversari

L'idea era quella di creare il gioco della *"battaglia all'ultimo muscolo"*: due fasce muscolari adatte ad essere allenate insieme diventano i nostri due giocatori, ma...

...la difficoltà che abbiamo riscontrato è stata non essere riusciti a trovare delle regole al gioco. Inoltre non erano ben specificate delle mosse killer che avrebbero indirizzato il gioco verso lo stato terminale e quindi alla sua fine.

3) Classificazione

Si intendeva realizzare un classificatore che sulla base dei valori delle variabili indipendenti *peso* e *altezza*, avrebbe stabilito il valore corretto della variabile dipendente categorica *Indice di Massa Corporea*.

4) Clustering

Si voleva tentare di raggruppare gli esercizi su una base di caratteristiche con valori tra di loro simili. Sulla base delle informazioni inserite dall'utente, sarebbero stati scelti gli esercizi del relativo cluster ottenuto precedentemente, ma...

... i costi di tempo, la necessità di costruire un dataset e l'intero processo di ingegnerizzazione hanno impedito l'approfondimento di queste soluzioni.