

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده فنی و مهندسی علامه طباطبائی

گروه مهندسی کامپیوتر

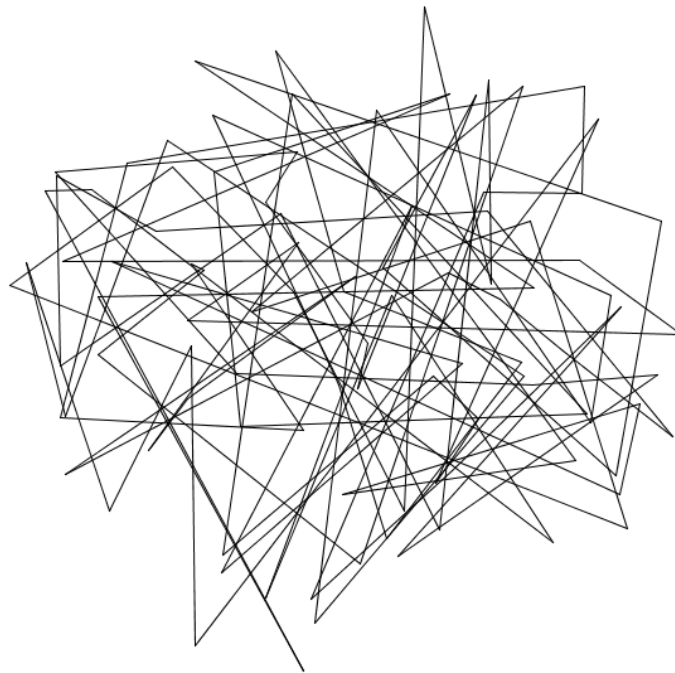
نام درس: روش پژوهش و ارائه

موضوع ارائه: الگوریتم فراابتکاری کلونی مورچگان و حل مسائل حوزه یادگیری ماشین

نام استاد: سرکار خانم جلوداری

نام دانشجو: مهدی فرساد خلیلی

بهار ۱۴۰۳



Meta Heuristic Algorithms – Ant Colony Optimization Algorithm & Solving Machine Learning Problems



الگوریتم های فراابتکاری، الگوریتم بهینه ساز کلونی مورچگان

و

حل مسائل یادگیری ماشین

فهرست مطالب

- ❖ معرفی اجمالی و کلی الگوریتم های فراابتکاری
- ❖ بررسی و معرفی برخی از الگوریتم های فراابتکاری
- ❖ معرفی و بررسی الگوریتم بهینه ساز کلونی مورچگان
- ❖ معرفی و بررسی نحوه عملکرد الگوریتم کلونی مورچگان در دنیای انسان و کامپیوتر
- ❖ نحوه حل مسائل یادگیری ماشین با استفاده از الگوریتم های فراابتکاری
- ❖ حل مسئله فروشنده دوره گرد

الگوریتم های ابتکاری و فراابتکاری – (Heuristic and Meta heuristic Algorithms)



- پایه ریاضی قوی ندارند
- بیشتر ماهیت تصادفی دارند
- راه حل بهینه دقیق را تضمین نمیکنند
- برای راه حل های تقریبی خوب هستند
- نیازی به اثبات همگرایی ریاضی ندارند
- جذابیت عملی به همراه سهولت پیاده سازی
- از نظر محاسبتی سریعتر از جست و جوی جامع هستند
- از یک اصل طبیعی، فیزیکی یا بیولوژیکی شبیه به جست و جو یا بهینه ساز تقلید میکنند

روش ها و الگوریتم های بهینه سازی به دو دسته تقسیم میشوند

دقیق:

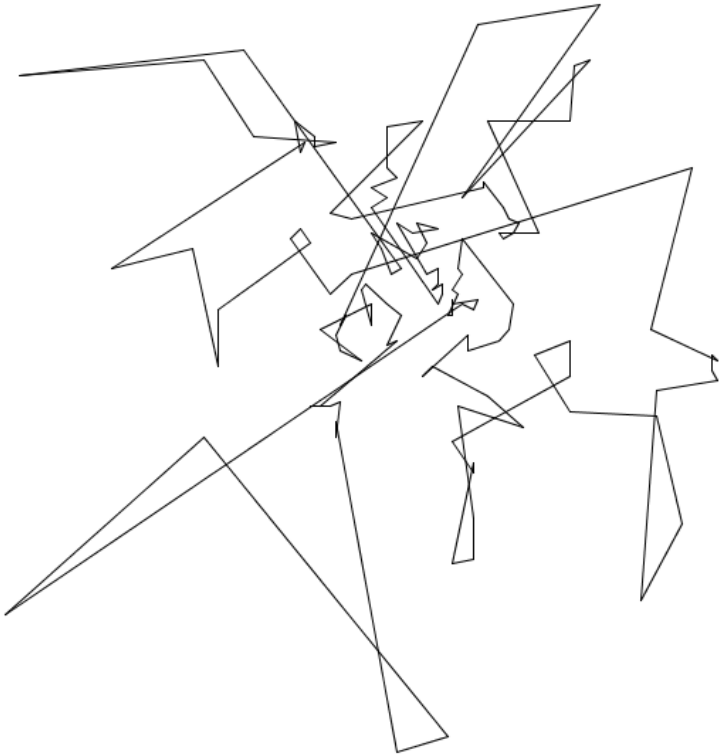
- جواب بهینه به صورت دقیق
- عدم کارایی در شرایط محاسباتی سخت و گسترده
- زمان محاسباتی بالا

تقریبی:

- جواب نزدیک به بهینه
- حل محاسبات سخت تر در زمان کوتاه تر



انواع حالت های الگوریتم های بهینه سازی تقریبی



ابتکاری heuristic

فراابتکاری Meta heuristic

فوق ابتکاری Hyper heuristic

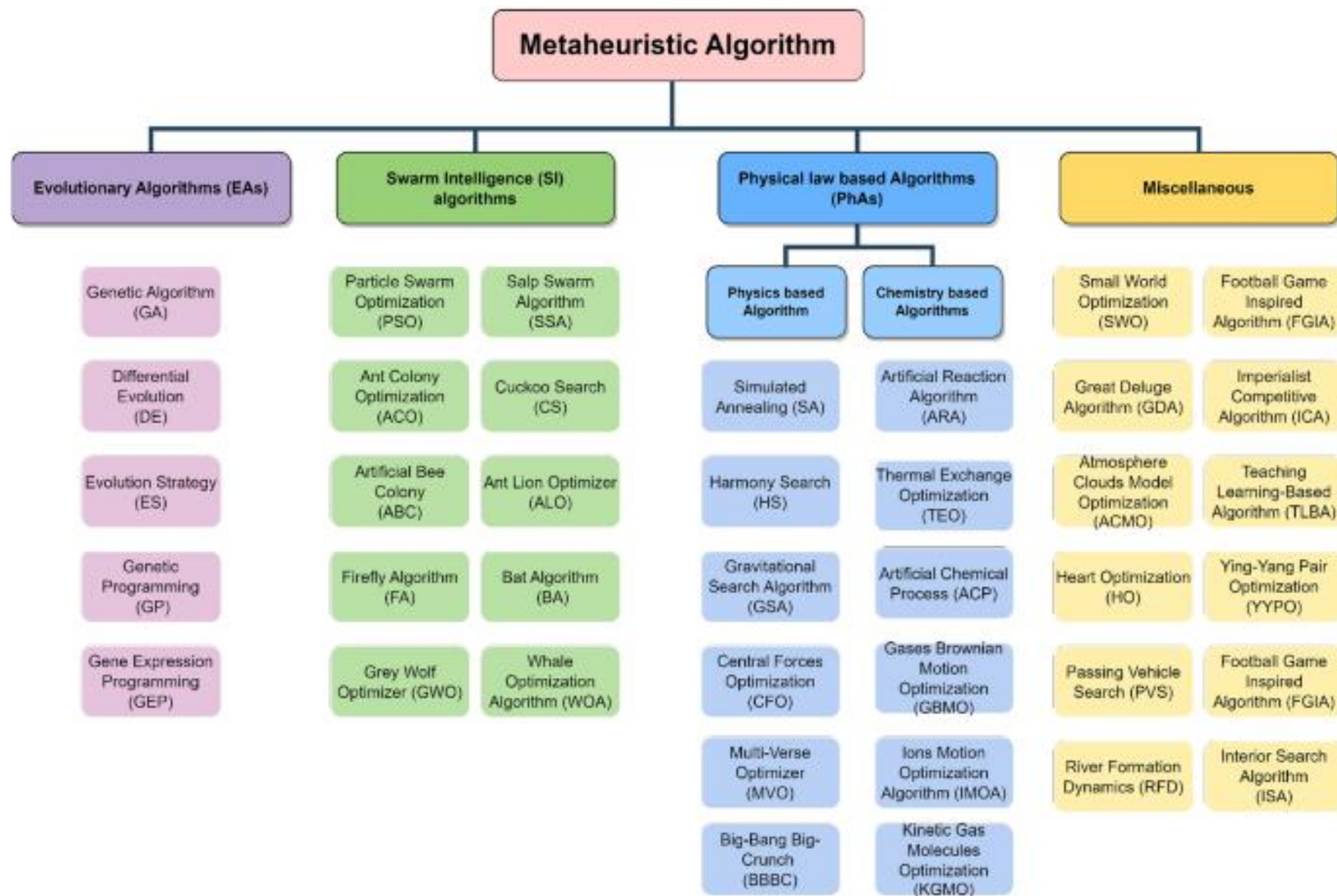
✓ معمولاً طبقه بندی الگوریتم ها به (تعداد راه حل اولیه) و (دامنه تقلید) بستگی دارند.

الگوریتم های فراابتکاری

فراابتکاری Meta heuristic

- ❖ مبتنی بر یک جواب و مبتنی بر جمعیت
- ❖ الهام گرفته شده از طبیعت و بدون الهام از طبیعت
- ❖ با حافظه و بدون حافظه
- ❖ قطعی و احتمالی

✓ مبتنی جمعیت شامل: الگوریتم تکامل، الگوریتم ژنتیک، بهینه سازی کلونی مورچگان، کلونی زنبورها و...



الگوریتم بهینه سازی کلونی مورچگان

Ant colony optimization algorithm (ACO)

- اولین بار در ۱۹۹۲ توسط مارکو دوریگو در پایان نامه دکترایش مطرح شد.
- در این روش می توان توسط مسائل محاسباتی- عددی بر مبنای علم احتمالات بهترین مسیر را در یک نمودار یافت.
- الهام گرفته شده از رفتار مورچه ها در جستجوی غذا
- مسئله یافتن کوتاه ترین مسیر و یا فروشنده دوره گرد را میتوان با این الگوریتم حل کرد.



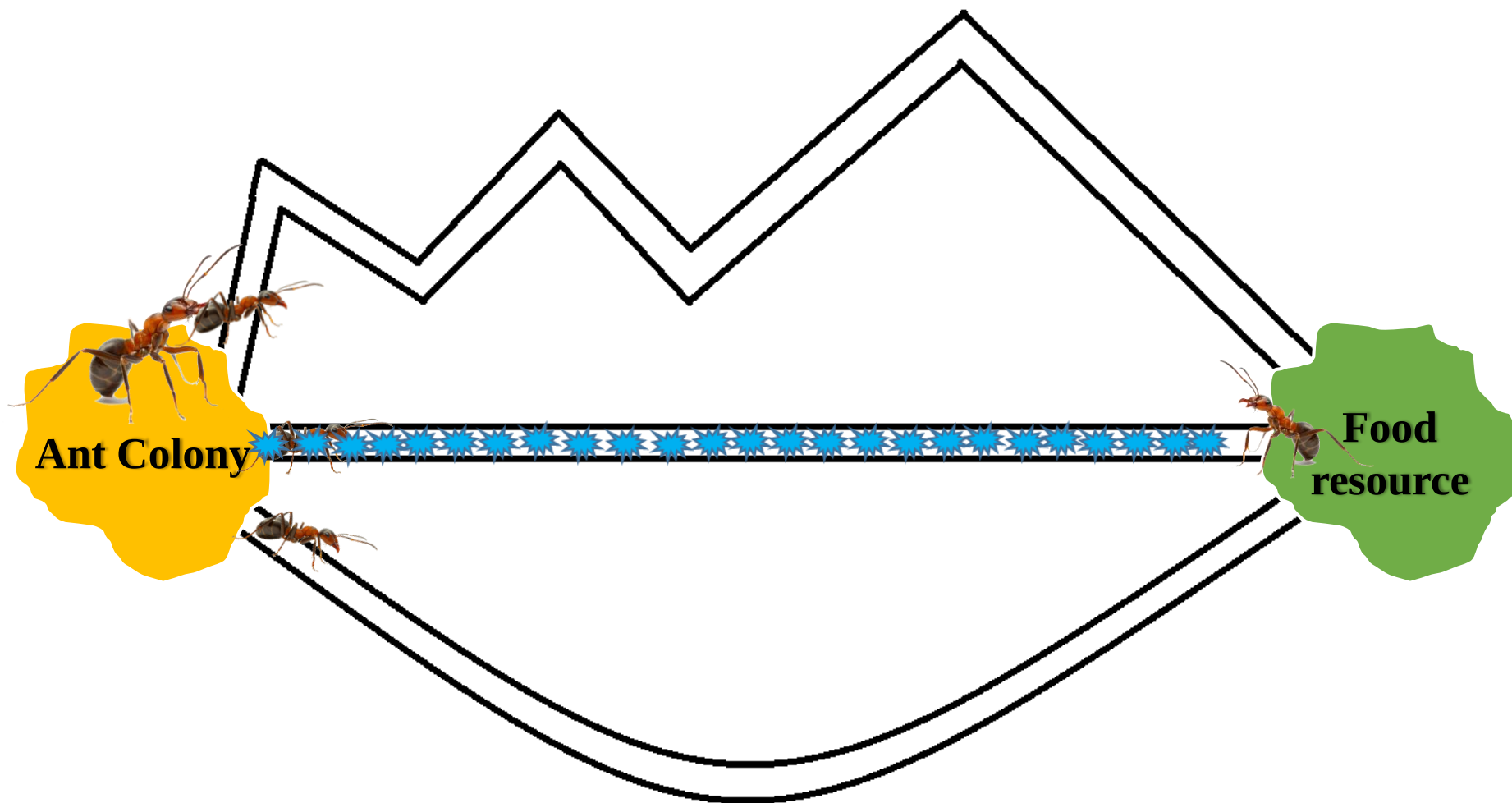
الگوریتم بهینه سازی کلونی مورچگان

Ant colony optimization algorithm (ACO)

- حرکت مورچه های مصنوعی بر روی نمودار باعث فراهم شدن راه حل بهتری از جانب سایر مورچگان مصنوعی میشود.
- یکی از مهم ترین و جالبترین رفتار مورچه ها، رفتار آنها برای یافتن غذا است و به ویژه چگونگی پیدا کردن کوتاهترین مسیر میان منابع غذایی و آشیانه.
- رفتار مورچه ها دارای نوعی هوشمندی توده ای است که ردی از فرومون (Pheromone) به جا می گذارند.



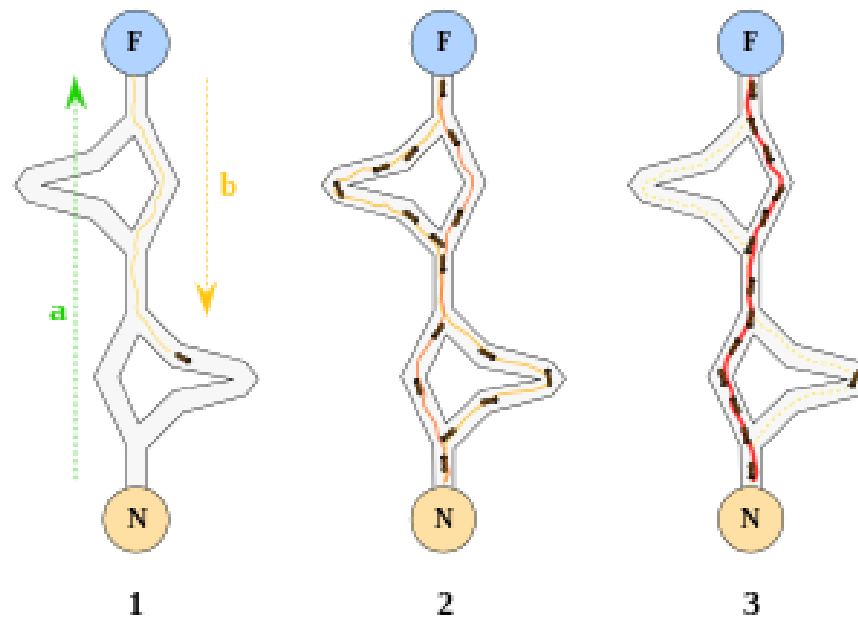
نحوه مسیریابی مورچگان بین کلونی و منبع غذا



❖ ماده شیمیایی ترشح شده از مورچگان فرومون (Pheromone) نام دارد که به مرور تبخیر می شود که فواید بسیاری دارد.

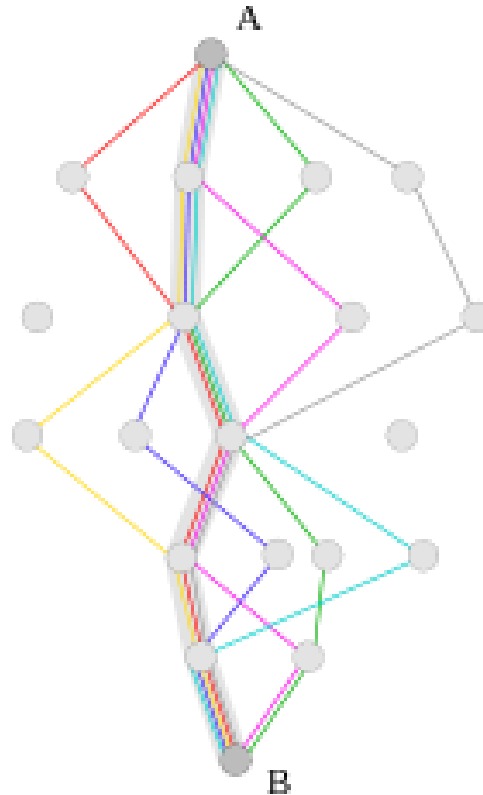
تأثیر فرومون در عملکرد مورچگان

وقتی یک مورچه مسیر کوتاهی (خوبی) را از کلونی تا غذا بیابد بقیه مورچه‌ها به احتمال زیادی همان مسیر را دنبال می‌کنند و با تقویت مداوم آن مسیر و تبخیر ردهای دیگر، به مرور همه مورچه‌ها هم مسیر می‌شوند. هدف الگوریتم مورچه‌ها تقلید این رفتار توسط مورچه‌هایی مصنوعی است که روی نمودار در حال حرکت اند. مسئله یافتن کوتاه‌ترین مسیر است و حل‌الش این مورچه‌های مصنوعی اند.



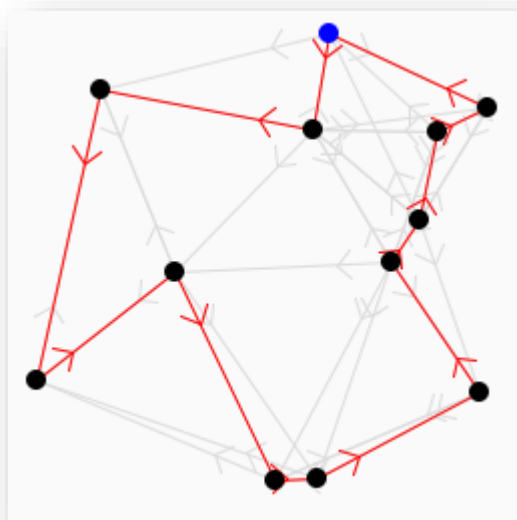
مزیت‌های کلونی مورچگان (ACO) Ant Colony Optimization

(تبخیر شدن فرومون) و (احتمال-تصادف) به مورچه‌ها امکان پیدا کردن کوتاه‌ترین مسیر را می‌دهد. این دو ویژگی باعث ایجاد انعطاف در حل هرگونه مسئله بهینه‌سازی می‌شوند.



کاربرد بهینه سازی الگوریتم کلونی مورچگان

از کاربردهای ACO می توان به بهینه کردن هر مسئله ای که نیاز به یافتن کوتاهترین مسیر دارد، اشاره نمود:



۱. مسیر یابی داخل شهری و بین شهری

۲. مسیر یابی بین پست های شبکه های توزیع برق ولتاژ بالا

۳. مسیر یابی شبکه های کامپیوتری

۴. استفاده از وب ، خوشه بندی داده ها

۵. استفاده از ACO در بهینه سازی شبکه های توزیع آب

۶. لبه یابی تصاویر و پردازش تصویر

۷. بیوانفورماتیک

۸. تجارت مالی



نحوه عملکرد الگوریتم



پروسه پیدا کردن کوتاه‌ترین مسیر توسط مورچه‌ها، ویژگی‌های بسیار جالبی دارد، که در ادامه با آن آشنا خواهیم شد.

ادامه بررسی عملکرد الگوریتم

- خود سازمان بودن
- عدم وجود مکانیزم کنترل مرکزی
- قدرت زیاد
- حتی تلفات هم تأثیر زیادی روی کارایی سیستم ندارد.
- پروسه یک فرایند تطبیقی است از آنجا که رفتار هیچ کدام از مورچه‌ها معین نیست و تعدادی از مورچه‌ها همچنان مسیر طولانی‌تر را انتخاب می‌کنند، سیستم می‌تواند خود را با تغییرات محیط منطبق کند.
- پروسه قابل توسعه است و می‌تواند به اندازه دلخواه بزرگ شود.



یادگیری ماشین (Machine learning)

توانایی ماشین ها برای یادگیری الگو ها و پیش بینی بر اساس داده ها

انواع مختلف الگوریتم های ML

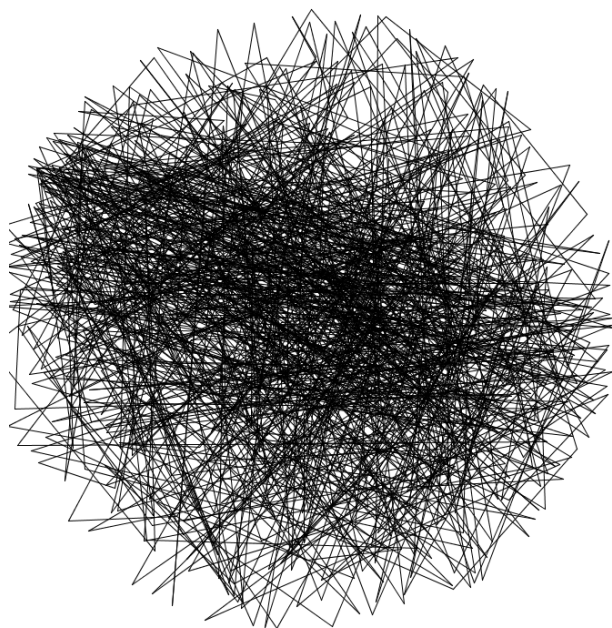
✓ تحت نظارت

- استفاده از داده های برچسب گذاری شده برای آموزش

✓ بدون نظارت

- تمرکز بر یافتن الگو ها و داده های بدون برچسب

✓ یادگیری تقویتی



نمونه هایی از نحوه استفاده از ML در حوزه های مختلف، مانند تشخیص تصویر، پردازش زبان طبیعی و تصمیم گیری و...

یادگیری ماشین (Machine learning)

پرکاربردترین زبان های برنامه نویسی در پروژه ها و مشکلات یادگیری ماشین پایتون است.

□ دلایل اصلی محبوبیت پایتون:

○ کتابخانه های گسترده، مانند Tensorflow, PyTorch, scikit-learn, keras

○ سادگی و خوانایی

○ سازگاری با پلتفرم های مختلف

○ پشتیبانی انجمن

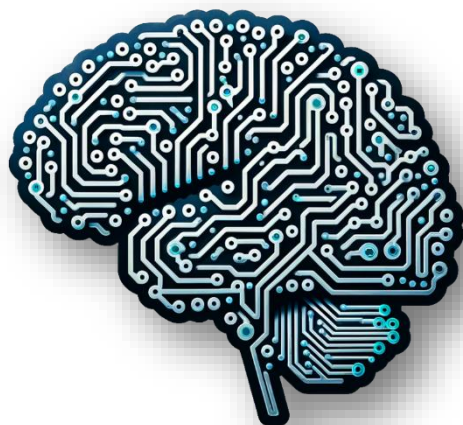
○ سرعت و عملکرد



□ سایر زبان های مورد استفاده در یادگیری ماشین عبارتند از:

Julia ، Java ، C++ ، R

نحوه حل مسائل یادگیری ماشین با الگوریتم های فراابتکاری



۱- تعریف مشکل (Define the problem)

۲- جمع آوری داده ها (Collect data)

۳- آماده سازی داده ها (Prepare the data)

۴- انتخاب الگوریتم مناسب (Select an algorithm)

۵- پیاده سازی الگوریتم (Implement the algorithm)

۶- آموزش مدل (Train the model)

۷- ارزیابی مدل (Evaluate the model)

۸- استقرار مدل (Deploy the model)

۹- تنظیم الگوریتم (Tune the algorithm)

۱۰- تجزیه و تحلیل نتیجه (Analyze results)

پیاده سازی الگوریتم (Implement the algorithm)

مرحله پیاده سازی در پروژه یادگیری ماشین جایی است که مفاهیم نظری و الگوریتم ها به کد تبدیل میشوند که شامل ترجمه مدل یادگیری ماشین انتخاب شده به یک زبان برنامه نویسی، آماده سازی داده ها، آموزش مدل و ارزیابی عملکرد آن است.

❖ نمونه مثال هایی از این مرحله مانند؛ (حدس آب و هوا) یا (تشخیص ایمیل هرزنامه بر اساس متن ایمیل)

```
11 # Preprocess the data
12 df['text'] = df['text'].str.lower() # Convert text to lowercase
13 df['text'] = df['text'].str.replace('[^\w\s]', '') # Remove punctuation
14
15 # Split the data into training and testing sets
16 X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.2, random_state=42)
17
18 # Vectorize the text data
19 vectorizer = TfidfVectorizer(stop_words='english')
20 X_train = vectorizer.fit_transform(X_train)
21 X_test = vectorizer.transform(X_test)
```

آموزش مدل (Train the model)

آموزش مدل، شامل یافتن مقادیر بهینه برای پارامترهای مدل است که یک تابع ضرر مشخص را به حداقل می‌رساند. تابع ضرر خطای بین پیش‌بینی‌های مدل و برچسب‌های واقعی در مجموعه داده را اندازه‌گیری می‌کند.

❖ برای مثال میتوان برنامه ای نوشت که قیمت خانه را با توجه به اندازه و متراژ حدس بزند؛

توضیحات لازم:

- برنامه ابتدا با داده‌ای در اختیارش قرار داده و سپس با فراخوانی متد `fit()` به مدل آموزش می‌دهیم که مدل بتواند در دفعات بعدی با توجه به ویژگی‌ها (اندازه خانه) بتواند هدف (قیمت خانه) را یاد بگیرد.

```
23 # Create and train a logistic regression model
24 model = LogisticRegression()
25 model.fit(X_train, y_train)
```

ارزیابی مدل (Evaluate the model)

بخش ارزیابی بخش مهمی از یک پروژه یادگیری ماشینی است که در آن عملکرد مدل آموزش دیده ارزیابی و تجزیه و تحلیل می شود.

❖ در نمونه مثال زیر (تشخیص ایمیل هرزنامه) سعی بر ارزیابی و بهبود عملکرد مدل داریم.

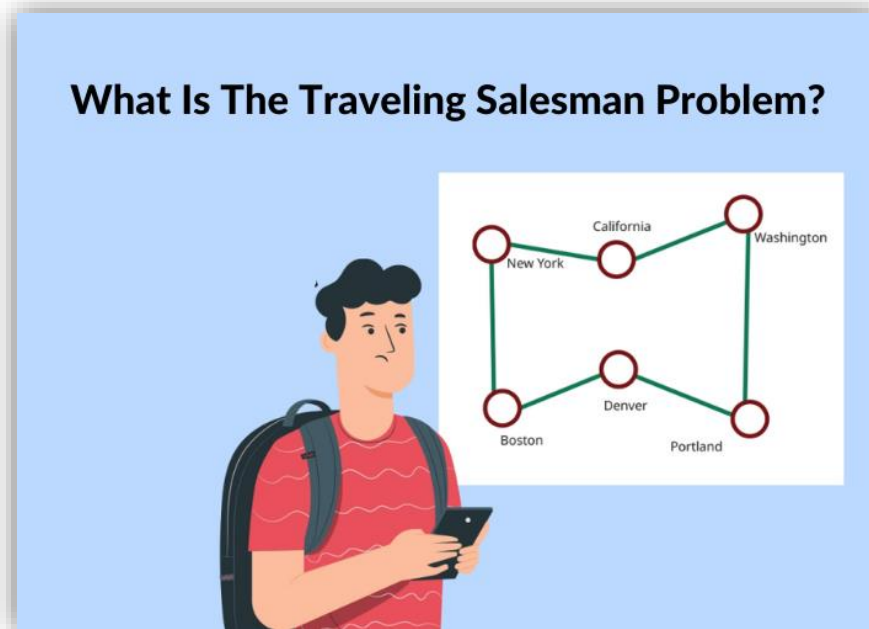
```
27 # Evaluate the model on the test set
28 y_pred = model.predict(X_test)
29 accuracy = accuracy_score(y_test, y_pred)
30 print('Accuracy:', accuracy)
31
32 # Save the trained model for future use
33 import pickle
34 filename = 'spam_classifier.sav'
35 pickle.dump(model, open(filename, 'wb'))
```

مرور کلی بر روی تمامی مراحل

```
1  # Import necessary libraries
2  import pandas as pd
3  from sklearn.model_selection import train_test_split
4  from sklearn.feature_extraction.text import TfidfVectorizer
5  from sklearn.linear_model import LogisticRegression
6  from sklearn.metrics import accuracy_score
7
8  # Load the dataset
9  df = pd.read_csv('spam.csv')
10
11 # Preprocess the data
12 df['text'] = df['text'].str.lower() # Convert text to lowercase
13 df['text'] = df['text'].str.replace('[^\w\s]', '') # Remove punctuation
14
15 # Split the data into training and testing sets
16 X_train, X_test, y_train, y_test = train_test_split(df['text'], df['label'], test_size=0.2, random_state=42)
17
18 # Vectorize the text data
19 vectorizer = TfidfVectorizer(stop_words='english')
20 X_train = vectorizer.fit_transform(X_train)
21 X_test = vectorizer.transform(X_test)
22
23 # Create and train a logistic regression model
24 model = LogisticRegression()
25 model.fit(X_train, y_train)
26
27 # Evaluate the model on the test set
28 y_pred = model.predict(X_test)
29 accuracy = accuracy_score(y_test, y_pred)
30 print('Accuracy:', accuracy)
31
32 # Save the trained model for future use
33 import pickle
34 filename = 'spam_classifier.sav'
35 pickle.dump(model, open(filename, 'wb'))
```

حل مسئله فروشنده دوره گرد (TSP)

مسئله فروشنده دوره گرد یک مسئله بهینه سازی کلاسیک است که در آن فروشنده باید کوتاه ترین مسیر ممکن را برای بازدید از مجموعه ای از شهرها و بازگشت به شهر شروع پیدا کند، در حالی که تنها یک بار از هر شهر بازدید می کند.



❖ مسئله ی فروشنده دوره گرد را میتوان از طریق الگوریتم بهینه ساز کلونی مورچگان و با کمک چند کتابخانه یا فریموورک های پایتون به جواب رساند، که در ادامه نحوه حل آن آشنا میشویم.

حل مسئله فروشنده دوره گرد (TSP)

با استفاده از الگوریتم بهینه سازی کلونی مورچگان و Tensorflow

```
1 import tensorflow as tf
2
3 # Initialize pheromone matrix
4 pheromone_matrix = tf.Variable(tf.random.uniform((num_cities, num_cities), 0.1, 1.0))
5
6 # Define training step
7 @tf.function
8 def train_step():
9     # Create ant population
10    ants = [tf.random.shuffle(tf.range(num_cities)) for _ in range(num_ants)]
11
12    # Ant construction
13    for ant in ants:
14        ant_path = []
15        current_city = ant[0]
16        while len(ant_path) < num_cities:
17            # Get probabilities from pheromone matrix
18            probs = tf.nn.softmax(pheromone_matrix[current_city])
19            # Select next city
20            next_city = tf.random.categorical(tf.math.log(probs), 1)[0]
21            # Add city to path
22            ant_path.append(next_city)
23            # Update current city
24            current_city = next_city
25
26    # Pheromone update
27    for ant_path in ant_paths:
28        pheromone_matrix = (1 - rho) * pheromone_matrix + rho * (1 / tf.reduce_sum(tf.gather(pheromone_matrix, ant_path)))
29
30    # Train for specified number of iterations
31    for _ in range(num_iterations):
32        train_step()
33
34    # Get solution
35    shortest_path = tf.argmax(pheromone_matrix, axis=1)
```

حل مسئله فروشنده دوره گرد (TSP)

با استفاده از الگوریتم بهینه سازی کلونی مورچگان و PyTorch

```
1 import torch
2
3 # Initialize pheromone matrix
4 pheromone_matrix = torch.nn.Parameter(torch.rand(num_cities, num_cities))
5
6 # Define training step
7 def train_step():
8     # Create ant population
9     ants = [torch.randperm(num_cities) for _ in range(num_ants)]
10
11     # Ant construction
12     for ant in ants:
13         ant_path = []
14         current_city = ant[0]
15         while len(ant_path) < num_cities:
16             # Get probabilities from pheromone matrix
17             probs = torch.softmax(pheromone_matrix[current_city], dim=0)
18             # Select next city
19             next_city = torch.multinomial(probs, 1)[0]
20             # Add city to path
21             ant_path.append(next_city)
22             # Update current city
23             current_city = next_city
24
25     # Pheromone update
26     for ant_path in ant_paths:
27         pheromone_matrix = (1 - rho) * pheromone_matrix + rho * (1 / torch.sum(torch.gather(pheromone_matrix, 1, ant_path)))
28
29 # Train for specified number of epochs
30 optimizer = torch.optim.Adam([pheromone_matrix])
31 for epoch in range(num_epochs):
32     train_step()
33     optimizer.step()
34
35 # Get solution
36 shortest_path = torch.argmax(pheromone_matrix, dim=1)
```



با تشکر از توجه شما

Any
Question

