

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

مبانی و کاربردهای هوش مصنوعی

پروژه‌ی اول

نیمسال دوم سال تحصیلی ۹۸-۹۷

---

مواردی که در انجام این پروژه برنامه نویسی باید رعایت کنید:

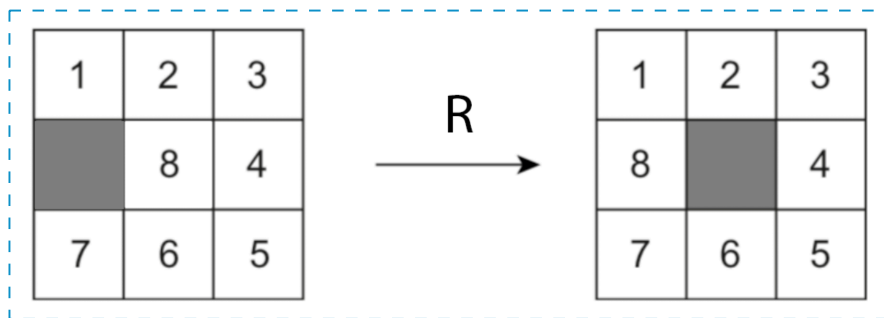
- پروژه را با یکی از زبانهای **Java**, **C++** یا **Python** انجام دهید.
- برای این پروژه علاوه بر کد پیاده‌سازی شده، گزارشی تهیه کنید که در آن نحوه مدل سازی مساله و خروجی‌های به دست آمده از هر روش جستجو ارائه شود. این گزارش بخشی از نمره‌ی پروژه را تشکیل می‌دهد.
- پروژه باید تک نفری انجام شود و با هر گونه مشابهت در کدها برخورد خواهد شد.
- پروژه تحویل حضوری خواهد داشت. تاریخ تحویل حضوری متعاقبا اعلام خواهد شد.
- کدها و مستندات خود را در یک فایل **zip** ذخیره کرده و نام آن را به **project1\_studentID.zip** تغییر دهید. سپس آن را در مدل بارگذاری نمایید .
- در صورت داشتن هرگونه سوال به **sare.soltani74@gmail.com** یا **n3urotransmitter@gmail.com** ایمیل دهید.

## سوال اول

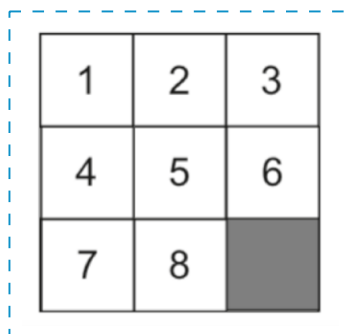
در این سوال هر دانشجو می‌بایست مجموعه‌ای از الگوریتم‌های جستجوی کلاسیک که فهرست آن‌ها در ادامه ذکر شده است را پیاده‌سازی و از این الگوریتم‌ها برای حل مساله‌ی داده شده استفاده کند. پیاده‌سازی الگوریتم‌ها و مساله‌ی مورد نظر می‌بایست به صورت مستقل از یکدیگر باشند، به گونه‌ای که برای حل یک مساله‌ی جدید نیازمند هیچگونه تغییری در پیاده‌سازی الگوریتم‌ها نباشند و بنابراین، کلاس‌های **problem** و **algorithm** باید جداگانه پیاده شوند.

## مسئله‌ی Sliding puzzle

مسئله‌ی sliding puzzle را در نظر بگیرید، در این مسئله یک بورد  $3 \times 3$  پازل در اختیار داریم که یکی از خانه‌های آن خالی است. هدف این است که مسیر رسیدن به بورد با اعداد مرتب را از هر ورودی با قرارگیری رندوم پازل‌ها را پیدا کنیم. یک مثال حرکت در بورد به صورت زیر می‌باشد :



و پاسخ نهایی به صورت مقابل می‌باشد:



ورودی این برنامه یک ماتریس از نحوه قرارگیری اعداد جدول می‌باشد (به جای خانه خالی، عدد صفر قرار دهید)، به عنوان نمونه:

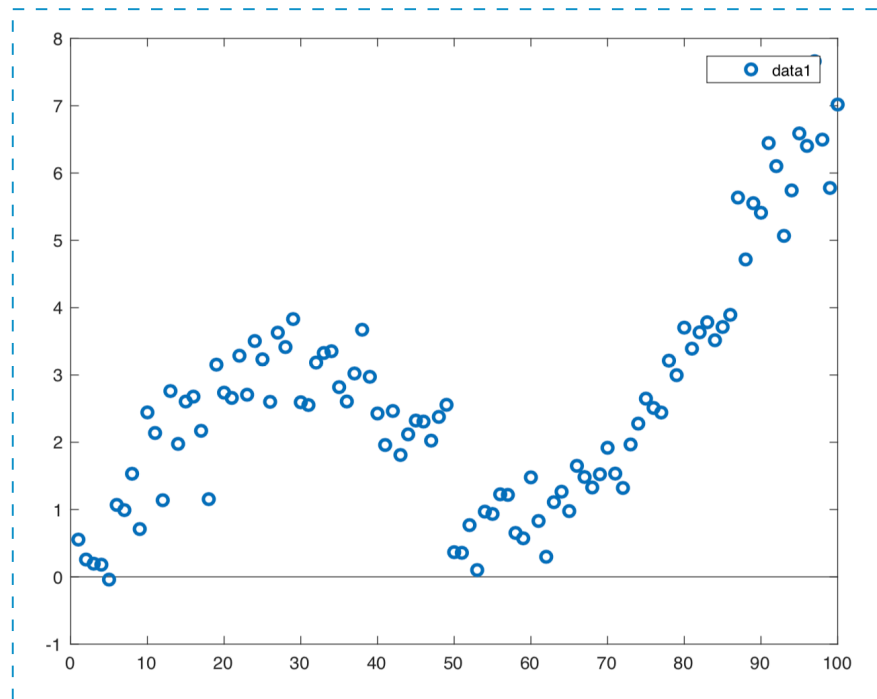
```
# sample input file(8-puzzle)
4 5 2
1 7 3
0 6 8
```

خروجی مساله دنباله‌ای از حرکت‌ها جهت رسیدن به حالت هدف می‌باشد. این مسئله را با الگوریتم‌های زیر حل کرده و آن‌ها را از نظر تعداد گره‌های گسترش داده شده، تعداد گره‌های تولید شده، عمق جواب و حداکثر تعداد گره‌هایی که در طی الگوریتم در حافظه قرار می‌گیرند، مقایسه کنید.

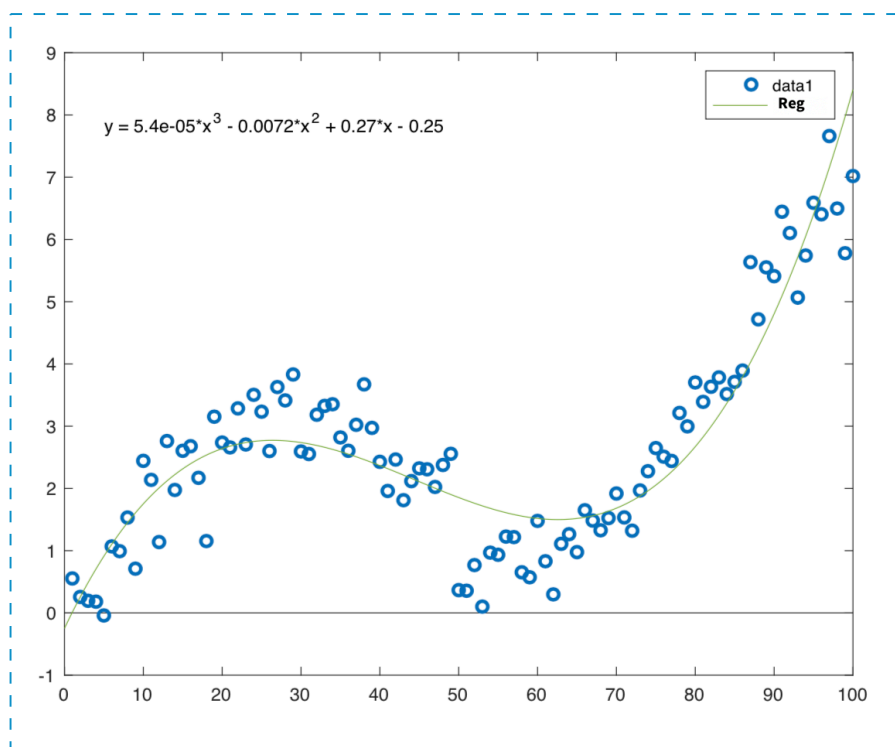
- a. سطح اول
- b. عمق اول (در سه حالت نامحدود، با عمق محدود و با افزایش تدریجی عمق)
- c. دو جهته
- d. هزینه یکنواخت
- e. A\* با تابع شهودی مجموع فاصله‌ی منتهن هر خانه با عددی که باید در آن خانه قرار بگیرد.

## سوال دوم

هدف از این بخش، حل مسئله‌ی رگرسیون چندجمله‌ای درجه سه به کمک الگوریتم ژنتیک می‌باشد. جهت درک مسئله، شکل زیر را در نظر بگیرید:



می‌خواهیم بهترین نمودار چندجمله‌ای که از تمام این نقاط رد شود را پیدا کنیم. این نمودار تابعی به فرم استاندارد  $y_p = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  می‌باشد که  $n$ : درجه‌ی این چندجمله‌ای است. برای مثال برای نقاط بالا و به ازای  $n = 3$ ، پاسخ به صورت زیر می‌باشد:



برای حل مسئله، باید چندجمله‌ای را به فرم استاندارد تعریف کرده، یک تابع خطا برای آن در نظر گرفته و آن را کمینه کنیم. در این مسئله تابع خطای Mean Square Error (MSE) را در نظر می‌گیریم که به صورت زیر تعریف می‌شود :

$$MSE = \frac{1}{n} \sum_{i=1}^n \left( y_i - y_{p,i} \right)^2$$

جهت حل مسئله‌ی این مسئله، مراحل زیر را دنبال کنید :

#### مرحله ۱ (تولید جمعیت اولیه)

جمعیت اولیه را population نام‌گذاری می‌کنیم. توجه کنید که تعداد اعضای جمعیت اولیه (populationSize) از پارامترهای قابل تنظیم مسئله می‌باشد. هریک از اعضای population، یک individual نام دارد و هر individual دارای متغیرهای قابل تغییری می‌باشد که ژن نامیده می‌شوند. در این مسئله هر individual، چهار ژن دارد که شامل ضرایب تابع چندجمله‌ای مورد نظر می‌باشند. ابتدا تمام ژن‌های هریک از individualها را به صورت تصادفی، مقداردهی کنید.

#### مرحله ۲ (تعیین میزان شایستگی هر individual)

تابع fitness function، بیانگر میزان شایستگی هر individual در مسیر رسیدن به پاسخ مسئله می‌باشد که از آن برای تعیین والدین جهت تولید نسل بعدی استفاده می‌کنیم.

برای هر individual، مقدار fitness را به صورت  $\frac{1}{1 + MSE(individual)}$  محاسبه کنید.

#### مرحله ۳ (انتخاب والدین)

در این مرحله از روش tournament selection استفاده می‌کنیم. در این روش تعداد k عضو را به صورت تصادفی انتخاب کرده و بهترین آن‌ها را برمی‌گزینیم. توجه کنید که k از پارامترهای قابل تنظیم می‌باشد و tournamentSize نام دارد. جمعیت برگزیده، والدین نامیده می‌شوند.

#### مرحله ۴ (تولید نسل جدید)

در مرحله‌ی بعد، باید تولید نسل جدید انجام گیرد. به این مرحله crossover گفته می‌شود. در این مرحله باید نسل جدید از طریق ترکیب هر ۲ والد برگزیده از مرحله‌ی قبل صورت گیرد. الگوریتم زیر، حالت ساده‌ای از تولید فرزند را نشان می‌دهد :

$$parent_1 = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$parent_2 = b_3x^3 + b_2x^2 + b_1x + b_0$$

$$child = a_3x^3 + b_2x^2 + b_1x + a_0$$

توجه کنید که برای تولید فرزندان، نیمی از ضرایب (در اینجا دو) از هر والد به صورت تصادفی به فرزند انتقال داده می‌شود. تولید فرزندان جدید تا رسیدن تعداد فرزندان جدید به جمعیت اولیه، ادامه دهید.

### مرحله ۵ (جهش)

در این مرحله، تابع `mutation` را تعریف می‌کنیم. پارامتر `mutationRate` را به عنوان یک پارامتر قابل تنظیم، تعریف می‌کنیم. در هر نسل تولید شده، ژن‌های هریک از `individual`های تولید شده را با احتمال `mutationRate` تغییر می‌دهیم. مثلاً اگر `mutationRate` را ۰.۰۱ در نظر بگیریم، در هر نسل پس از تولید فرزندان، هریک از ژن‌های هریک از فرزندان با احتمال ۰.۰۱ جهش می‌یابد. برای ایجاد جهش در این مسئله، به هریک از ژن‌های جهش‌یافته، مقداری نویز گوسی با میانگین صفر و واریانس  $\sigma$  اضافه می‌کنیم. مقدار  $\sigma$  از پارامترهای قابل تنظیم مسئله می‌باشد.

### مرحله ۶ (بازگشت)

در این مرحله، فرزندان تولید شده را به همراه جمعیت قبلی بر حسب `fitness` مرتب کرده و از بین آن‌ها، به تعداد `populationSize` بهترین‌ها را انتخاب کرده و به عنوان جمعیت جدید در نظر می‌گیریم. کلیه مراحل را به اندازه‌ی متغیر `numberOfGenerations` که از پارامترهای قابل تنظیم مسئله که بیانگر تعداد نسل‌ها می‌باشد، تکرار می‌کنیم.

از این الگوریتم جهت حل مسئله‌ی رگرسیون چندجمله‌ای درجه‌ی ۳، استفاده کنید.

مسئله را با تمام پارامترهای زیر، حل کنید و در نهایت درمورد تاثیر هر پارامتر در همگرایی پاسخ، بحث کنید.

`numberOfGenerations = 5000`

`populationSize = 50`

`tournamentSize = 2`

`mutationRate = 0.01, 0.02, 0.05, 0.1`

$\sigma = 1E-3, 1E-2, 1E-1, 1, 10$

در هر `generation`، بهترین، بدترین و میانگین تابع شایستگی را ذخیره کرده و در نهایت نمودار میزان این ۳ متغیر در طول `generations` را نمایش دهید.

ورودی مسئله، یک فایل متشکل از ۱۰۰ عدد (از نوع CSV)، می‌باشد که بیانگر مقادیر  $y$  نقاط اولیه روی فضای بین  $[0 : 0.1 : 9.9]$  می‌باشد و خروجی باید بهترین ضرایب را برای نمودار گذرنده‌ی درجه سه چاپ کند.