

بهینه سازی

به دنبال پیدا کردن متغیرهای مستقل مطلوب خودمون هستیم.

مسئله^۱: نگاشت کردن متغیرهای مستقل بر متغیرهای وابسته در غالب یک الگوریتم.

• مثال:

پیدا کردن ابعاد بهینه پنجره یک اتاق

سوال ۱: ابعاد بهینه رو بر اساس چه چیزی میخوای؟ تابع هدف!

جواب ۱: بر اساس نور بهینه

سوال ۲: نور بهینه یعنی چی؟

جواب ۲: مثلاً در یک اقلیم کویری لازمه که دریافت تابش / نور محدود بشه و از طرف دیگه باید فضا مون بر اساس فلان استاندارد حداقل میزان نور رو دریافت کنه.

در هر مسئله بهینه سازی متغیرهای هدف و مستقل میتونن متفاوت باشن.

متغیر هدف (وابسته)^۲: پس مقدار نوری که وارد فضا میشه. (در بهینه سازی یک هدفه، یک متغیره)

متغیر مستقل^۳: با فرض ثابت ماندن مساحت پنجره، مثلاً ارتفاع و عرض پنجره، یا عمق سایه بان یا باید محدوده تغییرات رو برای یافتن جواب بهینه مشخص کنیم. (میتونه بیش از یک متغیر باشه)

عملیات نگاشت^۴: فرایند اتصال بین متغیرهای مستقل و وابسته

$$f(\vec{X})$$

$$\vec{X} = \{x_1, x_2, \dots, x_n\}$$

¹ problem

² Dependent variables

³ Independent variables / features / attributes / dimensions

⁴ mapping

مرور : حداقل مقدار تابع $f(x) = x^2$ رو پیدا کنید.

با دیدن نمودار متوجه میشیم که در زمانی که $x = 0$ باشد، مقدار $f(x)$ حداقل خواهد بود. به دلیل وجود رابطه ریاضی روشن و مشخص در این مسئله، یافتن جواب کار راحتی خواهد بود.

روش حل ریاضی:

$$f'(x) = 2x \rightarrow 2x = 0 \rightarrow x = 0$$

در نقطه $x = 0$ ، شیب خط مماس برابر با صفره. از اینجا میتونیم نقاط اکسترمم رو پیدا کنیم. پس برای پیدا کردن اکسترمم های یک معادله، لازمه که مشتقش رو برابر صفر قرار بدیم تا اکسترمم هاش بدست بیاد.

بهینه سازی در مباحث هوش مصنوعی gradient descent می نامند. یعنی گرادیان کاهش. با حرکت از هر نقطه ای به اطراف بر اساس شیب خط میتونیم متوجه بشیم که داریم به سمت مینیموم حرکت میکنیم یا ماکسیمم.

در موضوعات معماری، تبدیل هر مسئله به یک رابطه روشن ریاضی کار دشواریه.

مثال : میزان دید از یک پنجره به بیرون رو بر اساس یک معادله ریاضی بنویسید.

$$X_1 = \text{عرض پنجره}$$

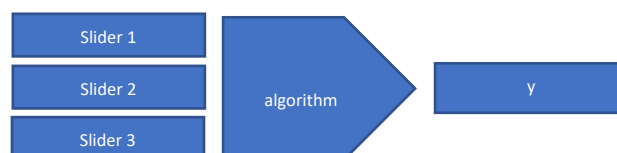
$$X_2 = \text{طول پنجره}$$

$$f(x_1, x_2) = \dots$$

اگر من بتونم رابطه ریاضی بین متغیرهای مستقل و وابسته رو در قالب یک فرمول ریاضی پیدا کنم، میتونم ارزش مشتق بگیرم و به بر اساس گرادیان و شیب خط به سمت اکسترمم ها حرکت کنم و جواب مطلوبمو پیدا کنم.

اما الزاما همیشه نمیشه این رابطه رو پیدا کرد. پس ما تلاش میکنیم از این فرایند یک الگوریتم بسازیم.

توی گرس هاپر ما یک سری اسلایدر داریم که میتونه نقش x ها رو برای ما پیدا کنه.



در موضوعات این چینی به جای حل مسئله به روش ریاضی، از روش های فراابتکاری، مسئله رو حل میکنیم.

بهینه سازی: به دنبال مینیمایز/ماکسیمایز کردن یک تابع $f(\vec{X})$ هستیم.

تابع $f(\vec{X})$ نام های دیگری مانند تابع هدف^۵ یا تابع هزینه^۶ هم دارد. نام های دیگری نیز دارد.^{۸۷}

^۵ Objective function مینیمایز باشد یا ماکزیمایز

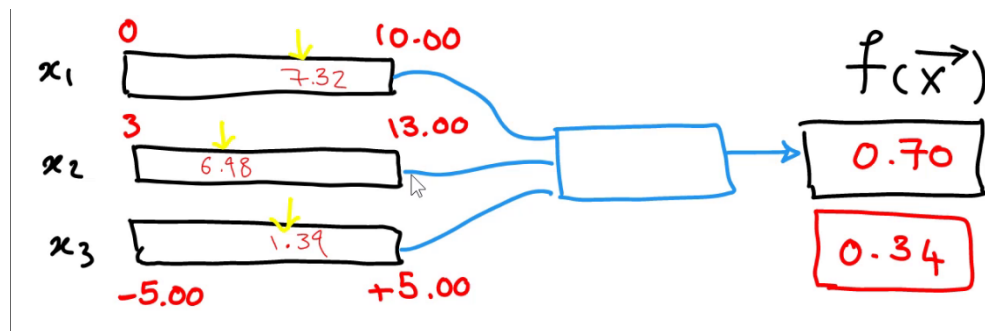
^۶ Cost function همیشه دنبال کاهش هستیم

^۷ Fit function

^۸ Loss function

تعیین دامنه متغیرهای مستقل:

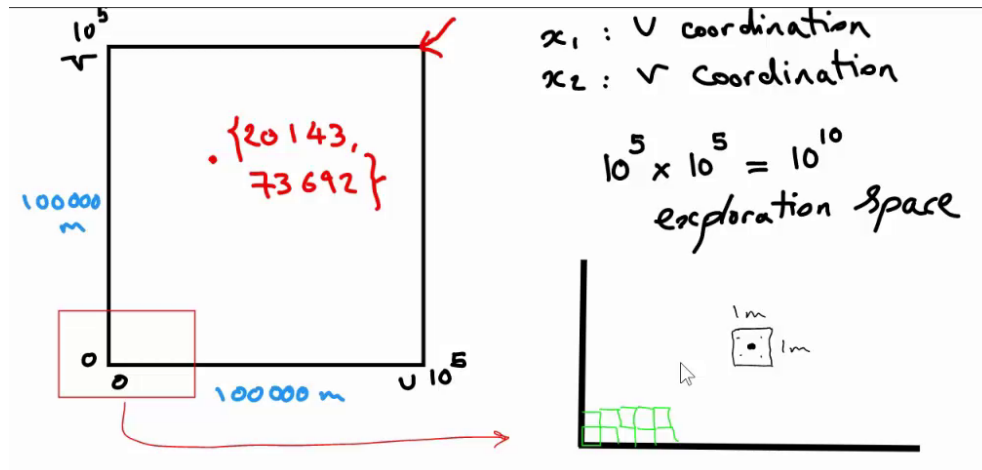
برای اسلایدرها باید بازه تعریف کنیم



اول باید ببینیم که هر اسلایدری چندتا حالت رو ممکنه به خودش بگیره. برای فهمیدنش لازمه که حداقل، حداکثر و گام تغییرات رو متوجه باشیم.

الگوریتم ژنتیک

فرض کنیم، می‌خواهیم نقطه‌ای که کمترین ارتفاع رو از سطح دریا داره توی یک کویر پیدا کنیم. این کار رو با برداشت انسانی انجام میدیم. کویر همون فضای جستجو^۹ی ماست.



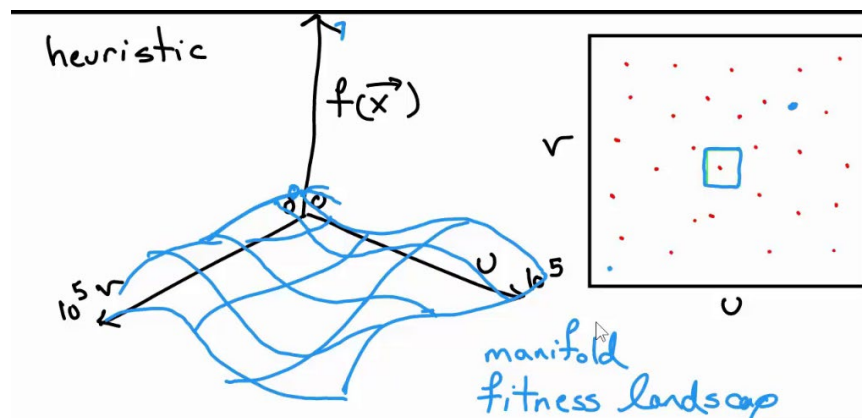
در این مسئله، وابسته به دقت گام‌های هر متغیر جواب‌ها تصاعدی زیاد میشه و ما نمیتونیم همه جواب‌ها رو محاسبه کنیم و اندازه‌گیری کنیم. در این موارد، از الگوریتم‌های هیوریستیک استفاده می‌کنیم. الگوریتم‌هایی که با اونها به دنبال کشف فضا هستیم.

مثلا ممکنه ما ۵۰ نفر رو ببریم کویر و بگیم رندوم پخش بشن و ارتفاع نقطه‌ای که توش هستن رو گزارش کنن. ایجنتی که بهترین جواب رو داشت، بقیه یک گام بهش نزدیک شن و باز گزارش کنن و باز... .

سوال اینه که آیا جوابی که بهش میرسیم مینیموم گلوبال هست؟ نه الزاما! ممکنه کلا جواب جای دیگه‌ای باشه.

این مثال، مفهوم کشف کردن فضا. یه تعدادی ایجنت داریم، یه تعدادی هم لوپ داریم. که باهاش یه جوابی رو گزارش میکنیم با تعداد خیلی کمتر که یک منطقی داره ولی الزاما ایده‌آل نیست.

اگه این مسئله رو ببریم در فضای کارتزینی، خواهیم داشت :



⁹ Exploration / search space

توی مثال کویر، با بردن اطلاعات به فضای کارترینی، شاید شکل گیری یک fitness landscape یا manifold هستیم. از اونجایی که ما فرم این manifold رو نمیدونیم، همیشه ترسیمش کرد. اگه میدونستیم که دیگه نیازی به بهینه سازی نبود. به راحتی نقطه مینیموم رو پیدا میکردیم.

اگه توی مسئله ای بیش از دو متغیر مستقل داشتیم، فضای fitness landscape رو نمی‌تونیم نمایش بدیم، اما از نظر ریاضی کاملاً وجود داره.

از اونجایی که ما به دنبال کشف فضای fitness هستیم، مسئلهمون مربوط به کشف فضاست و در دسته بندی متاهوریستیک قرار میگیره.

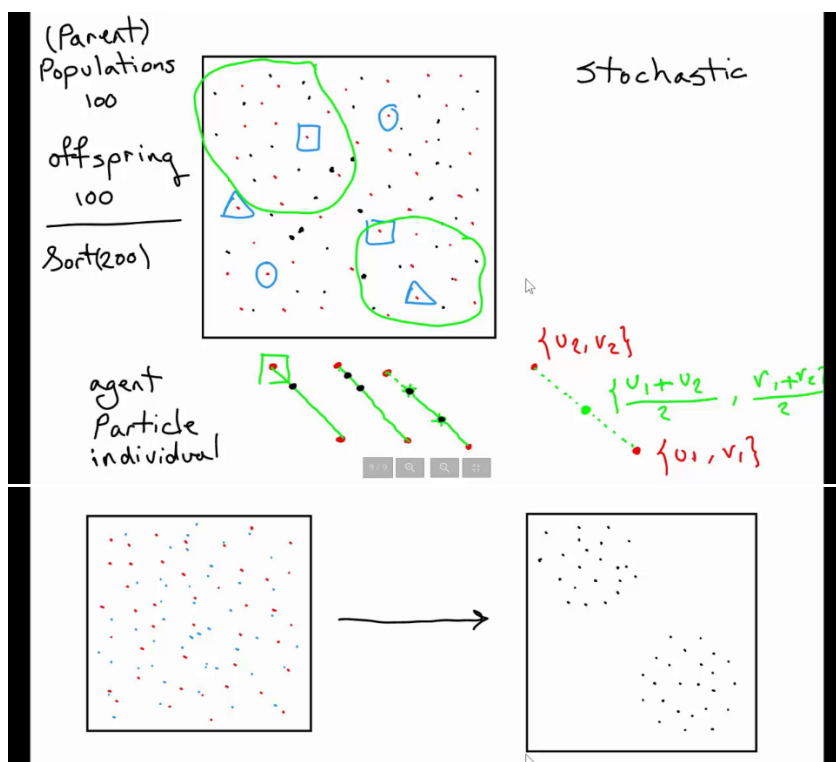
روش کار ژنتیک

یه فضایی داریم به اسم فضای جستجو

یه تعدادی ایجنت^{۱۰} داریم. مثلاً صد تا داریم که این رو بصورت random uniform پخش میکنیم.

شروع الگوریتم ژنتیک رندوم هست به همین دلیل بهشون میگن Stochastic. یعنی شروع از رندوم ولی یونیفرم.

بعد از اینکه رندوم شروع شد، از هر دو ایجنت، یک کاپل میسازیم که در اینصورت ۵۰ تا کاپل داریم. هدف اینه که فرزند تولید کنن. یعنی نقطه جدید سمپلینگ درست کنن که این نقطه یه جایی بین دوتا والد هست. برای ایجاد فرزندا از روش ها و حالت های مختلفی استفاده میشه کرد. به جمعیت والدها^{۱۱}، فرزندا^{۱۲} هم اضافه میشن. این جامعه رو سورت میکنیم و یه تعدادی رو از بهترین ها انتخاب میکنیم و باز فرایند رو انجام رو ادامه میدیم. به این فرایند^{۱۳} (حذف) گفته میشه.

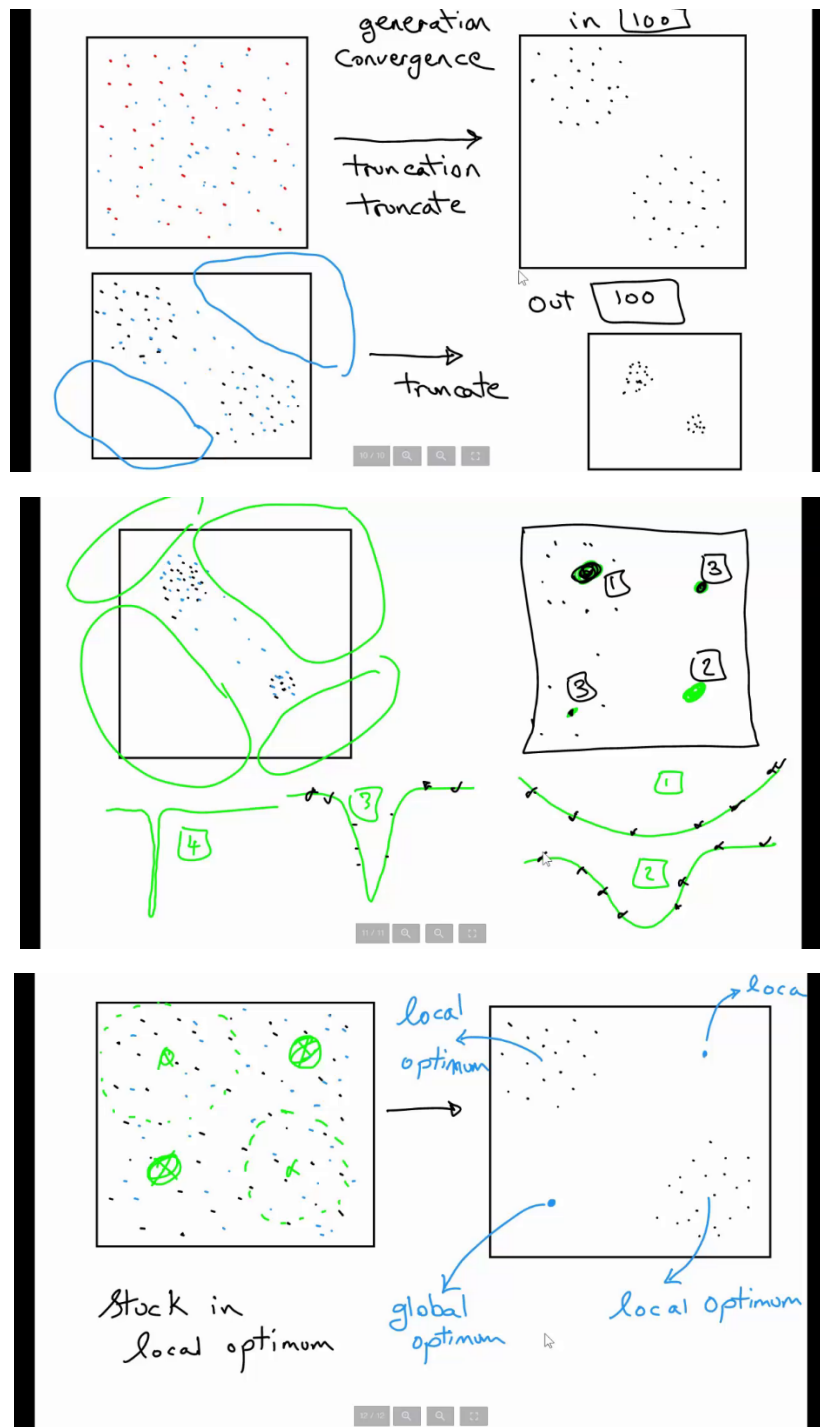


¹⁰ Agent / particle / individual / population

¹¹ parents

¹² offspring

¹³ Truncation



برای اینکه از local optimum ها فرار کنیم، یا احتمال پیدا کردن global optimum ها بیشتر بشه، باید از جهش استفاده کنیم.

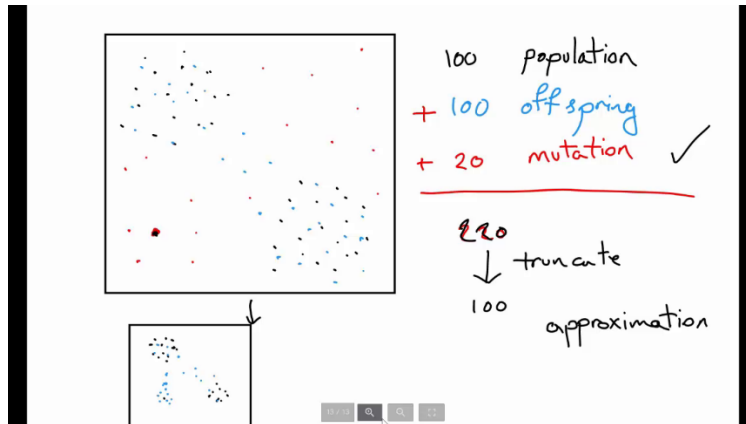
جهش^{۱۴}:

اتفاقی خارج از قاعده تکرار شونده موجود

جهش می‌تونه خودش قاعده ای مختلفی داشته باشه.

1. یکی از بُدهای یکی از والد‌ها رو مثلاً عوض کنیم
2. کلاً رندوم باشه

با هدف رسیدن به global optimum . با همه این توضیحات، همچنان امکان نرسیدن به global optimum وجود دارد. به همین دلیل الگوریتم ژنتیک، یک الگوریتم approximate است.



ما نیاز به شرط خروج^{۱۵} داریم :

1. شرط تعداد نسل : به تعداد مشخصی نسل پیش بره و بعدش تموم شه.
2. شرط عدم بهبود پاسخ : یه وقت فلان نسل گذشت و جواب بهتر نشد تموم شه.

Exploration VS exploitation

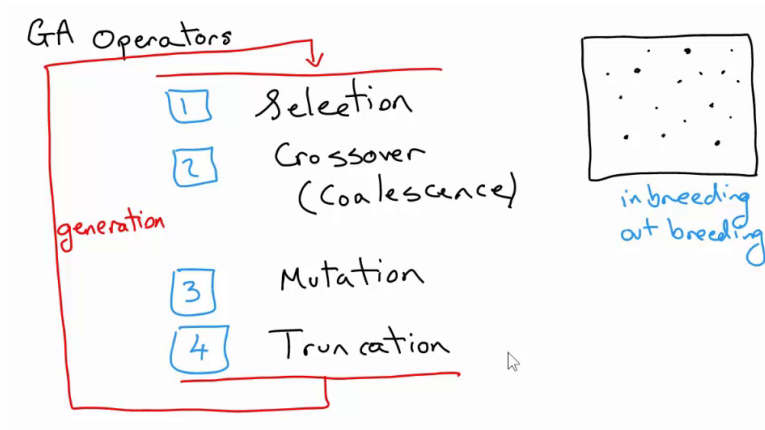
Exploration: دوست داریم دنبال جواب های بیشتری باشیم که نکته چیزی رو از دست داده باشیم.

Exploitation: دوست داریم تمرکزمون رو بذاریم روی یک ناحیه و با دقت اونجا رو بگردیم.

در نسل های اول بیشتر explore کن و در نسل های پایانی بیشتر exploit کن.

جمع بندی فرایند الگوریتم ژنتیک

1. الگوریتم ژنتیک مستقل از خود مسئله کار می‌کند و به همین دلیل بهش می‌گیم فرا ابتکاری^{۱۶}
2. الگوریتم یک فرایند تکرار شونده رو طی می‌کنه پس یک الگوریتم تکاملی^{۱۷} هست
3. الگوریتم شروع رندوم داره^{۱۸}
4. برای موضوعات پیوسته بهتره.
5. اوپرتورها:
 - a. انتخاب^{۱۹}: کاپل ساختن ها با چه قاعده ای انجام بشه. مثلاً بهترین جواب ها با هم و بدترین جواب ها با هم یا برعکس^{۲۰}
 - b. ترکیب^{۲۱}:
 - c. جهش:
 - d. ترانکیت^{۲۲}:



¹⁶ Meta-heuristic

¹⁷ Evolutionary

¹⁸ Stochastic

¹⁹ selection

²⁰ Inbreeding / outbreeding

²¹ Crossover / coalescence

²² Truncate