



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

هوش مصنوعی

نیم‌سال دوم ۹۸-۹۹

استاد: محمدحسین رهبان

تمرین دوم جست‌وجوی محلی و جست‌وجو در فضای پیوسته مهلت ارسال: ۲ فروردین

لطفا به موارد زیر توجه داشته باشید:

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص شده است.
- همکاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد.
- امکان ارسال با تاخیر وجود ندارد بنابراین بهتر است انجام تمرین را به روزهای پایانی موکول نکنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.

سوالات نظری (۶۲ نمره)

۱. (۶ نمره) عملکرد الگوریتم Simulated Annealing با تغییرات دمایی زیر چگونه است؟

(آ) (۲ نمره) دما را خیلی سریع کاهش دهیم.

(ب) (۲ نمره) دما را مقداری ثابت و مثبت در نظر بگیریم.

(ج) (۲ نمره) دما را صفر در نظر گرفته و تغییر ندهیم.

۲. (۱۱ نمره) می‌خواهیم با استفاده از الگوریتم Hill-Climbing مسئله Boolean Satisfiability Problem را حل

کنیم. در این مسئله عبارتی به فرم Conjunction of Disjunctions با تعداد متغیر Boolean داده می‌شود. هدف پیدا کردن یک مقداردهی است که مقدار عبارت را صحیح کند.

اگر تابع Successor حالت‌های همسایه را با تغییر یک متغیر تولید کند، به سوالات زیر پاسخ دهید.

(آ) (۲ نمره) فضای جستجو با n متغیر چه اندازه‌ای دارد؟

(ب) (۲ نمره) با داشتن n متغیر، تابع Successor چند همسایه برای حالت فعلی تولید می‌کند؟

(ج) (۲ نمره) یک تابع هدف (Objective Function) برای این مسئله پیشنهاد دهید.

(د) (۲ نمره) عبارت زیر را در نظر بگیرید که در آن تنها مقدار متغیر B صحیح باشد. حالت بعدی تولید شده توسط الگوریتم چه خواهد بود؟ توضیح دهید.

$$(A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge (B \vee C) \wedge (B \vee \neg C) \wedge (\neg B \vee D)$$

(ه) (۳ نمره) برای عبارت زیر، یک بهینه محلی بیابید که حاصل عبارت را صحیح نکند.

$$(A \vee B \vee \neg C) \wedge (\neg A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C) \wedge (\neg A \vee B \vee \neg C)$$

۳. (۸ نمره) مسئله Vertex Cover مسئله‌ای NP-Complete است که هدف آن انتخاب کوچکترین زیرگراف ممکن

از یک گراف است به طوری که حداقل یک سر از هر یال گراف، در زیرگراف باشد.

روش حل این مسئله را با الگوریتم ژنتیک توضیح داده و نحوه‌ی مدل‌سازی ژنوم‌ها، تابع fitness، selection، cross-over و mutation را مشخص کنید.

۴. (۱۰ نمره) مسئله جمع زیرمجموعه‌ها مسئله‌ای است که در آن تلاش می‌کنیم زیرمجموعه‌ای ناتهی از یک مجموعه

را بیابیم که جمع اعضای آن صفر باشد. از آنجا که این مسئله NP-Complete است، برای حل آن از جستجو محلی استفاده می‌کنیم.

(آ) (۸ نمره) مسئله را بگونه‌ای مدل کنید که با استفاده از الگوریتم Hill-Climbing قابل حل شود. فرض

کنید تعداد اعضای مجموعه اصلی n باشد. تعریف هر حالت، تعریف و تعداد همسایه‌های آن و تابع هدف را مشخص کنید.

(ب) (۲ نمره) در مجموعه زیر یک بهینه محلی مثال بزنید که بهینه سراسری نباشد. در صورتی که مدل شما چنین حالتی ندارد، آن را اثبات کنید.

$$\{-3, -2, 5, 6, 9\}$$

۵. (۸ نمره) فرض کنید تابع $f: \mathbb{R} \rightarrow \mathbb{R}$ ، تابعی پیوسته مشتق‌پذیر باشد که مشتق آن نیز پیوسته است. همچنین فرض کنید تابع f دارای کمینه سراسری $x^* \in (-\infty, \infty)$ است که می‌خواهیم آن را با استفاده از الگوریتم گرادیان کاهشی بیابیم. برای این کار از $x^{(0)} \in \mathbb{R}$ و $\epsilon > 0$ ثابت استفاده کرده و بروزرسانی $x^{(t)} = x^{(t-1)} - \epsilon f'(x^{(t-1)})$ را تکرار می‌کنیم. درستی یا نادرستی عبارات زیر را با ذکر توضیحات کافی مشخص نمایید.

(آ) (۲ نمره) الگوریتم به ازای هر طول گام $\epsilon > 0$ همگرا می‌شود.

(ب) (۲ نمره) اگر تابع f دارای یک کمینه محلی x^* متفاوت با x^* بوده و به ازای مقداری از زمان $x^{(t)} = x^*$ برقرار شود، الگوریتم به x^* همگرا نمی‌شود.

(ج) (۲ نمره) در صورت همگرایی الگوریتم، به x^* همگرا می‌شود، اگر و تنها اگر تابع f محدب باشد.

(د) (۲ نمره) اگر $f(x) = (y - wx)^2$ و الگوریتم همگرا شود، آنگاه به x^* همگرا شده است.

۶. (۹ نمره) تابع $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ را به صورت زیر تعریف می‌کنیم. می‌خواهیم از نقطه اولیه $w_0 = (10, 5, 5)$ شروع کنیم و به مقدار مینیمم تابع نزدیک شویم.

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5x_2^3 + (x_2 - 2x_3)^4 + 3x_1x_4^2$$

(آ) (۴ نمره) با استفاده از Gradient Descent تا دو مرحله جلو بروید. مقدار α را برابر 0.1 در نظر بگیرید.

(ب) (۳ نمره) به طور کلی در روش Gradient Descent اگر مقدار α را خیلی زیاد در نظر بگیریم چه مشکلی پیش می‌آید؟ مقدار خیلی کم چطور؟ برای پیدا کردن مقدار مناسب α روشی ارائه دهید.

(ج) (۲ نمره) فرض کنید به دلایلی مثل تعداد زیاد متغیرها، نمی‌توانید مقدار مناسبی برای α پیدا کنید. در این حالت چطور می‌توان مطمئن شد که Gradient Descent نقطه مینیمم را پیدا می‌کند؟ با فرض این که قرار دادن α بسیار کوچک زمان بر و غیر ممکن است.

۷. (۱۰ نمره) می‌دانیم تابع $f: \mathbb{R} \rightarrow \mathbb{R}$ محدب است اگر و تنها اگر:

$$\forall x, y \in \mathbb{R}, 0 \leq \alpha \leq 1: f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

با توجه به این تعریف، گزاره‌های زیر را ثابت کنید.

(آ) (۲ نمره) جمع دو تابع محدب، محدب است.

(ب) (۳ نمره) اگر دو تابع f و g محدب باشند، آنگاه تابع $\max\{f(x), g(x)\}$ محدب است.

(ج) (۳ نمره) اگر دو تابع f و g مثبت، محدب و هر دو صعودی (یا هر دو نزولی) باشند، آنگاه تابع $f(x)g(x)$ محدب است.

(د) (۲ نمره) اگر تابع f مثبت، محدب و صعودی بوده و تابع g مثبت و محدب و نزولی باشد، آنگاه تابع $\frac{f(x)}{g(x)}$ محدب است.

سوالات عملی (۳۸ نمره)

۱. (۲۰ نمره) در این سوال، می‌خواهیم مسئله graph bisection را با استفاده از الگوریتم‌های Local Search حل کنیم.

تعریف مسئله

به شما یک گراف با n راس داده می‌شود که n زوج است. شما باید راس‌های این گراف را طوری به دو بخش تقسیم کنید که:

۱. اندازه هر بخش برابر $\frac{n}{2}$ باشد.

۲. تعداد یال‌های بین دو بخش کمینه شود.

برای این کار یک بار باید از الگوریتم Simulated Annealing و بار دیگر از الگوریتم Hill Climbing کمک بگیرید.

ورودی:

در اولین خط ورودی n (تعداد راس‌ها) و در خط بعدی m (تعداد یال‌ها) داده می‌شود. در هر یک از m خط بعدی، دو سر هر یال به شکل $v_1 v_2$ می‌آید.

خروجی:

در اولین خط خروجی، تعداد یال‌های بین دو بخش را چاپ کنید. در خط بعدی، شماره راس‌های بخش اول را با فاصله چاپ کنید. در آخرین خط، شماره راس‌های بخش دوم را با فاصله چاپ کنید.

ورودی نمونه:

6
4
1 4
2 5
4 5
4 6

خروجی نمونه:

1
2 3 5
1 4 6

۲. (۱۸ نمره) هدف از این تمرین، یادگیری مدل کردن و حل یک مسئله جدید با استفاده از local search است. مدل کردن یک مسئله جدید به یک مسئله search مهارتی اساسی در هوش مصنوعی است که در مواجهه با بسیاری از مشکلات کاربرد دارد. به طور کلی، local search ممکن است قادر به حل کامل مسئله نباشد اما می تواند پاسخ مناسبی برای مسئله پیدا کند.

شرح مسئله:

فرض کنید یک متخصص ژنتیک هستید که روی ژن های تعدادی موجود عجیب کار می کنید. می خواهید نشان دهید بعضی از این موجودات نسبت به سایرین شباهت بیشتری به هم دارند. به این ترتیب تصمیم می گیرید دنباله ژن های این موجودات را روی هم map کنید و یک امتیاز شباهت بر اساس تعداد ژن های برابر محاسبه کنید. در نتیجه موجودات با امتیاز شباهت کمتر احتمالاً شباهت بیشتری به هم دارند. به همین شکل در ادامه می خواهید شباهت بین مجموعه ای از موجودات را مشخص کنید.

تعریف مسئله:

به شما k رشته با نام X_i از مجموعه حروف V داده می شود. طول هر رشته X_i برابر N_i است. شما می خواهید این رشته ها را روی هم map کنید و به این وسیله آن ها را با هم مقایسه کنید. یک راه ساده برای این کار این است که فرض کنید این map کردن در دو مرحله انجام می شود: مرحله conversion و مرحله matching. مرحله conversion در واقع یک تابع F است که یک رشته ورودی می گیرد و یک رشته خروجی می دهد. تمام $F(X_i)$ ها طولی برابر N دارند. N بزرگ تر مساوی تمام N_i هاست. تابع F تنها می تواند به رشته ورودی تعدادی dash اضافه کند. هزینه تبدیل X به $F(X)$ برابر است با $CC \times \text{number of dashes}$ که در آن CC یک مقدار ثابت است. وقتی تمام رشته ها از مرحله conversion گذشتند، مرحله matching رشته ها را در هر اندیس با هم مقایسه می کند. هزینه matching دو کاراکتر، توسط تابع مقارن $MC(c_1, c_2)$ تعیین می شود که در آن c_1 و c_2 دو کاراکتر عضو $V \cup \{-\}$ هستند. هزینه matching دو رشته برابر مجموع هزینه matching کاراکترهای متناظر آن دو رشته است. هزینه matching تعدادی رشته، برابر مجموع هزینه matching هر جفت از آن هاست.

مثال:

فرض کنید $k = 3$ نشان دهنده تعداد رشته ها باشد. همچنین مجموعه حروف برابر $V = \{A, C, T, G\}$ است. فرض کنید ۳ رشته داده شده به شکل زیر باشند:

X_1 : ACTGTGA

X_2 : TACTGC

X_3 : ACTGA

پس برای این مثال، N_1 و N_2 و N_3 به ترتیب برابر ۷، ۶ و ۵ هستند. همچنین فرض کنید متغیرهای مربوط به هزینه ها به شکل زیر باشد:

$$CC = 3$$

و

$$MC(x, y) = 2$$

اگر x و y عضو V باشند و $x \neq y$.
به همین ترتیب $MC(x, -) = 1$ و $MC(x, x) = 0$.
می‌توانیم تبدیل‌ها را به شکل زیر تعریف کنیم:

$$F(X_1) : -ACTGTGA$$

$$F(X_2) : TACT - -GC$$

$$F(X_3) : -ACTG - -A$$

با این تبدیل‌ها $N = 8$ خواهد بود. هزینه تبدیل‌ها به ترتیب برابر ۳، ۶ و ۹ است. هزینه matching بین $F(X_1)$ و $F(X_2)$ برابر ۵ است. به طور مشابه هزینه matching بین $F(X_2)$ و $F(X_3)$ برابر ۵ است. در نتیجه هزینه کل matching برابر است با: $5 + 5 + 2 = 12$.
هزینه کلی این مثال برابر مجموع هزینه conversion و matching است: $3 + 6 + 9 + 12 = 30$.

خواسته مسئله:

شما باید الگوریتم ژنتیکی طراحی کنید که با دریافت اطلاعات گفته شده، تبدیلی (conversion) از رشته‌ها را پیدا کند که کمترین هزینه نهایی را داشته باشد. تبدیل نشان داده شده در مثال بالا تبدیل بهینه نیست.

ورودی:

برای تست کدهای ژنتیک شما، تعدادی تست کیس طراحی شده که فرمت آن‌ها در ادامه توضیح داده می‌شود:
در خط اول ورودی تعداد حروف موجود در مجموعه حروف ($|V|$) داده می‌شود. در خط بعدی مجموعه حروف می‌آید. خط بعد عدد K می‌آید که نشان‌دهنده تعداد رشته هاست. در k خط بعدی، رشته‌های X_1 تا X_k می‌آید. در خط بعد، ثابت CC برای هزینه‌های مربوط به Conversion و پس از آن، تابع MC می‌آید. تابع MC به شکل یک ماتریس $(|V| + 1) \times (|V| + 1)$ داده می‌شود. $MC[i][j]$ هزینه matching کاراکتر i ام و کاراکتر j ام از مجموعه حروف را نشان می‌دهد. سطر و ستون آخر، مربوط به کاراکتر dash است.

خروجی:

در اولین خط خروجی کمترین هزینه‌ای که الگوریتم شما محاسبه کرده است را چاپ کنید. در k خط بعدی، رشته‌های تبدیل شده را چاپ کنید. توجه کنید که هر چقدر الگوریتم شما در بازه زمانی منطقی بتواند تبدیل با هزینه کمتری را پیدا کند، نمره بیشتری دریافت خواهید کرد.

ورودی نمونه:

4
A, C, T, G
3
ACTGTGA
TACTGC
ACTGA
3
0 2 2 2 1
2 0 2 2 1
2 2 0 2 1
2 2 2 0 1
1 1 1 1 0

خروجی نمونه:

27
ACTGTGA
TA-CTGC
AC- -TGA