پاسخ تمرین دوم هوش مصنوعی

سوال ۱)

الف. در الگوریتم Simulated Annealing اگر دما به اندازه کافی کند کاهش نیابد، شانس پیدا کردن بهینه سراسری کم می‌شود.
ب. در این حالت، احتمال رفتن به یک همسایه بدتر تنها به ΔE بستگی دارد. همچنین چون دما با گذر زمان کاهش نمی‌یابد، احتمال خروج از یک نقطه بهینه نیز کاهش نیافته و شانس رسیدن به بهینه سراسری کمتر است.
ج. با فرض آنکه شرط پایان الگوریتم $T = 0$ نباشد، احتمال رفتن به یک همسایه بدتر صفر خواهد بود. به این ترتیب الگوریتم در هر حالت، همسایه‌ها را با ترتیب تصادفی انتخاب می‌کند تا جایی که به اولین همسایه بهتر برسد. پس الگوریتم مانند First Choice Hill-climbing عمل می‌کند.

سوال ۲)

سوال ۲)
آ) فضای جستجو برابر است با:

2^n

ب) n همسایه
ج) بیشینه کردن تعداد عبارات disjunction که true می‌شوند. (که این مقدار برابر تعداد اعضای conjunction است)
د) در این حالت تعداد عبارات disjunction که true می‌شوند برابر است با 3. با تغییر هریک از متغیرها این تعداد به صورت زیر تغییر می‌کند که الگوریتم بهترین آن‌ها را انتخاب می‌کند.

تغییر A <- 4
تغییر B <- 4
تغییر C <- 3
تغییر D <- 4

بنابراین یکی از تغییرات A، B، یا D به صورت رندوم انتخاب می‌شود.
ه) مقداردهی A(true)، B(false)، C(true) یک بهینه محلی تولید می‌کند چرا که در این حالت تعداد عبارات true برابر 3 است و در همه حالات همسایه هم این تعداد برابر 3 است.

سوال ۳)

Date.

۳- رئوس گراف را از ۱ تا n شماره‌گذاری می‌کنیم. یک رشته به طول تعداد یال‌ها در نظر می‌گیریم

و برای هر یال شماره‌ی رأسی که می‌خواهیم در این رشته باشد را می‌نویسیم. تابع fitness برای

این مدل تعداد اعداد متمایز درون این رشته است. هر کروموزوم برابر با

تعداد کل رأس‌ها منهای تابع fitness تقسیم بر کل احتمال انتخاب دارد.

تابع cross-over به این صورت است که دو کروموزوم را از یک قسمت جدا کرده

و به صورت دو به دو با هم ترکیب می‌کنیم. برای mutation هم تعدادی از حروف را انتخاب

کرده و بر اساس گراف اصلی به جای عدد فعلی، عدد آن یکی رأس یال را می‌گذاریم.

چون از هر یال یک رأس در این رشته قرار دارد، پس هر رشته همواره در حالت Vertex Cover

قرار دارد.

مدلسازی مسئله با گرفتن رشته‌ای دودویی به طول $|V|$ برای نشان دادن حضور یا عدم حضور هر راس نیز صحیح است. اما لطفا به منفی نشدن تابع Fitness و همچنین درستی آن در مقایسه رشته‌ها دقت کنید. شرط پوشاندن تمام یال‌های گراف، یک شرط ضروری است و پاسخ باید این ویژگی را داشته باشد. به عنوان مثال اگر بین دو راس a و b با درجه یک، دقیقا یال باشد، تابع Fitness شما نباید برنداشتن دو راس a و b را ترجیح دهد.

سوال ۴)

سوال ۴: به هر زیرمجموعه یک رشته باینری n حرفی (n: اندازه مجموعه) متناظر می‌کنیم. عضو iام در زیرمجموعه وجود دارد اگر و تنها اگر کاراکتر iام رشته ۱ باشد.

همسایه‌ی هر رشته را رشته‌هایی در نظر می‌گیریم که با حذف یا اضافه کردن یک عضو (طوری که نقض نشود) به وجود می‌آیند. یعنی یک بیت را انتخاب کرده و آن را نات کنیم. تعداد همسایه‌ها برای مجموعه‌های یک عضوی n-1 تا (چون نمی‌توانیم یک عضو را حذف کنیم) و برای مابقی n تا (تعداد حالت‌های انتخاب یک بیت) است.

تابع هدف را قدر مطلق مجموع اعضای زیرمجموعه در نظر می‌گیریم و هدفمان مینیمم کردن این تابع است.

ب) زیرمجموعه {۶، ۲-، ۳-} یک بهینه محلی است چون با اضافه کردن هر کدام از ۵ و ۹ یا حذف هر کدام از اعضا |مجموع اعضا| از ۱ بیشتر می‌شود. و بهینه کلی‌مان هم {۵، ۲-، ۳-} است.

اگر به روش‌های پیدا کردن همسایه‌های هر حالت حرکت جابه‌جا کردن یک عضو درون مجموعه با یک عضو بیرون مجموعه (جابه‌جا کردن یک صفر و یک در صورت وجود) را هم اضافه کنیم در مثال داده شده بهینه محلی نخواهیم داشت. اما مثلاً در {۸، ۸-، ۳، ۳، ۳} زیرمجموعه {۸-، ۳، ۳، ۳} بهینه کلی می‌شود.

سوال ۵)

الف) نادرست! فرض کنید $f(x) = x^2$ است و در حالت اولیه از $x^{(0)} = 1$ شروع کرده‌باشیم و $\epsilon = 1$ در اینصورت $x^{(2i)} = 1,\ x^{(2i+1)} = -1$ و همیشه بین دو مقدار -1 و 1 این مقدار جابه‌جا می‌شود و در نتیجه هیچ‌گاه همگرا نخواهد شد.

ب) درست! اگر در مرحله‌ای به کمینه محلی برسیم مشتق مورد نظر در آن نقطه صفر می‌شود و عبارت:
$x^{(t)} = x^{(t-1)} - \epsilon f'(x^{(t-1)}) = x^{t-1}$

پس در نتیجه همیشه در این نقطه که نقطه x^* نیست می‌مانیم.

ج) نادرست! مثال نقض چنین چیزی تابعی مثل

$$f(x) = \frac{1}{4}x^4 + \frac{1}{3}x^3 - x$$

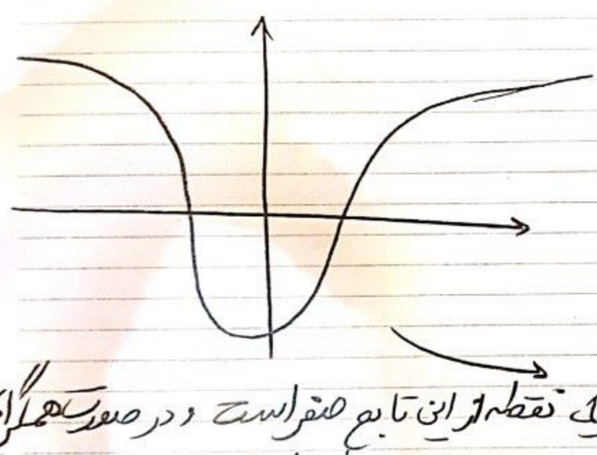
است. این تابع محدب نیست! برای بررسی این موضوع کافیست دو بار مشتق بگیریم:

$$f'(x) = x^3 + x^2 - 1$$

$$f''(x) = 3x^2 + 2x$$

در نقطه -0.5 این مقدار منفی است!

مثال:



تنها مشتق یک نقطه از این تابع صفر است و در صورت همگرایی به آن می‌رسیم که همان مینیمم مطلق است

اما تابع محدب نیست.

د) درست! ابتدا ثابت می‌کنیم اگر تابع f محدب باشد و الگوریتم همگرا شود به x^* همگرا می‌شود. در صورت محدب بودن می‌دانیم f' یک تابع صعودی است چون در x^* مینیمم است می‌دانیم تا قبل از x^* مقدار f' منفی است و پس از آن مثبت است. بنابراین اگر $x^{(t)}$ پس از x^* باشد کم می‌شود و اگر قبل از x^* باشد زیاد می‌شود. بنابراین در صورت همگرایی یکسری کم و زیاد شدن داریم و می‌دانیم x^* در بین این کم و زیاد شدن هاست پس همیشه فاصله $x^{(t)}$ از x^* یعنی $|x^{(t)} - x^*|$ در حال کم شدن است. لذا به همین مقدار همگرا می‌شود.

حالا کافیست بگوییم این تابع محدب است تا ثابت شود به x^* همگرا شود. برای اینکار ثابت می‌کنیم مشتق دوم مثبت است:

$$f(x) = (y-wx)^2 \rightarrow f'(x) = 2(y-wx)(-w) = w^2x-2yw \rightarrow f''(x) = w^2 > 0$$

سوال ۶)

آ)

$$\nabla f(x_1, x_2, x_3, x_4) = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \frac{\partial f}{\partial x_4})$$

$$\frac{\partial f}{\partial x_1} = 2 \times 1 \times (x_1 + 10x_2) + 3x_4^2$$

$$\frac{\partial f}{\partial x_2} = 2 \times 10 \times (x_1 + 10x_2) + 5 \times 3 \times x_2^2 + 4 \times 1 \times (x_2 - 2x_3)^3$$

$$\frac{\partial f}{\partial x_3} = 4 \times (x_2 - 2x_3)^3 \times (-2)$$

$$\frac{\partial f}{\partial x_4} = 2 \times 3x_1 x_4$$

$$\nabla f(x_1, x_2, x_3, x_4) =$$
$$(2x_1 + 20x_2 + 3x_4^2,\ 20(x_1 + 10x_2) + 15x_2^2 + 4(x_2 - 2x_3)^3, -8(x_2 - 2x_3)^3, 6x_1x_4)$$

بنابراین داریم:

$$(10, 5, 5, 4) - 0.1 \times \nabla f(10, 5, 5, 4) = (-6.8, -102.5, -95, -20)$$

$$(-6.8, -102.5, -95, -20) - 0.1 \times \nabla f(-6.8, -102.5, -95, -20)$$
$$= (79.56, -281767.025, 535842.5, -101.6)$$

برای قسمت ب، به ازای learning rate بزرگ ممکن است الگوریتم همگرا نشود یا واگرا شود. به ازای learning rate بسیار کوچک هم زمان زیادی طول می کشد تا الگوریتم همگرا شود. راهکارهای متفاوتی برای تعیین learning rate را میتوانید در لینک زیر مشاهده کنید:

https://automaticaddison.com/how-to-choose-an-optimal-learning-rate-for-gradient-descent/

اگر جواب دیگری نوشتید که منطقی و درست است نمره سوال را می گیرید. توجه کنید که فرق قسمت ب با ج در این سوال در این بود که در قسمت ب امکان محاسبه دقیق گرادیان به صورت ریاضی وجود داشت و میشد با بررسی گرادیان تابع loss و تغییرات آن بر حسب متغیرها مقدار مناسبی برای learning rate محاسبه کرد.

ج) در توابع غیرمحدب (non-convex) نمی‌توان مطمئن شد که به نقطه‌ی کمینه‌ی سراسری رسیده‌ایم یا خیر. زیرا علاوه‌بر عوامل دیگر بسیار بستگی به نقطه‌ی شروع دارد.

در توابع محدب (convex) با انتخاب مقدار مناسب (نه لزوما کوچک‌ترین مقدار) برای Learning Rate می‌توان مطمئن شد که الگوریتم نقطه‌ی کمینه‌ی سراسری را پیدا می‌کند. به طور دقیق‌تر، اگر اندازه‌ی گام‌ها بسیار بزرگ نباشد، تابع به یک مقدار با کمی خطا هم‌گرا می‌شود و چون تابع محدب است، کمینه‌ی سراسری خواهد بود. یعنی مقدار گرادیان در آن نقطه بسیار نزدیک به صفر خواهد بود.

به طور کلی می‌توان رفته رفته مقدار Learning Rate را کاهش داد تا به نقطه‌ی مناسب با خطای کم‌تر رسید.

سوال ۷)

سؤال ۷. باید ثابت کنیم: $\forall x,y\in R,\ 0\leqslant\alpha\leqslant 1:\ (f+g)(\alpha x+(1-\alpha)y)\leqslant\alpha(f+g)(x)+(1-\alpha)(f+g)(y)$

f,g محدب اند. $(f+g)(\alpha x+(1-\alpha)y)=f(\alpha x+(1-\alpha)y)+g(\alpha x+(1-\alpha)y)$ داریم:

$$\leqslant \alpha f(x)+(1-\alpha)f(y)+\alpha g(x)+(1-\alpha)g(y)=\alpha(f(x)+g(x))+(1-\alpha)(f(y)+g(y))$$

$$=\alpha((f+g)(x))+(1-\alpha)((f+g)(y)) \longrightarrow$$ در قسمت (آ) حکم ثابت شد

(ب) قرار می دهیم $h=Max\{f,g\}$ و حال باید ثابت کنیم:

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

اما توجه کنید به ازای هر $x\in\mathbb{R}$ داریم: $h(x)\geq f(x),\ h(x)\geq g(x)$ و از آن جایی که $h(x)=g(x)$ یا $h(x)=f(x)$

لذا حکم معادل اثبات این دو حکم است:

$$f(\alpha x+(1-\alpha)y)\leqslant\alpha h(x)+(1-\alpha)h(y) \quad (۱)$$

$$g(\alpha x+(1-\alpha)y)\leqslant\alpha h(x)+(1-\alpha)h(y) \quad (۲)$$

از آن جایی که f، g محدب اند داریم:

$$f(\alpha x+(1-\alpha)y)\leqslant\alpha f(x)+(1-\alpha)f(y)\leqslant\alpha h(x)+(1-\alpha)h(y)$$

$$g(\alpha x+(1-\alpha)y)\leqslant\alpha g(x)+(1-\alpha)g(y)\leqslant\alpha h(x)+(1-\alpha)h(y)$$

و این گونه ثابت می شود که $Max\{f(x),g(x)\}$ نیز در صورت محدب بودن f و g محدب اند.

ج. f and g are positive and convex, hence for $0 \leq \theta \leq 1$,

$$\begin{aligned} f(\theta x+(1-\theta)y)\,g(\theta x+(1-\theta)y) &\leq (\theta f(x)+(1-\theta)f(y))\,(\theta g(x)+(1-\theta)g(y)) \\ &= \theta f(x)g(x)+(1-\theta)f(y)g(y) \\ &\quad +\theta(1-\theta)(f(y)-f(x))(g(x)-g(y)). \end{aligned}$$

The third term is less than or equal to zero if f and g are both increasing or both decreasing. Therefore

$$f(\theta x+(1-\theta)y)\,g(\theta x+(1-\theta)y) \leq \theta f(x)g(x)+(1-\theta)f(y)g(y).$$

د. تابع $g(x)$ نزولی و مقعر است. بنابراین تابع $\frac{1}{g(x)}$ صعودی و محدب خواهد بود. با استفاده از قسمت ج، این قسمت ثابت می‌شود.