



پروژه درس رایانش عصبی و یادگیری عمیق

پروژه سوم

هدف: آشنایی با الگوریتم‌های رایج بهینه‌سازی شبکه‌های عصبی

کد: پیاده سازی این پروژه را به زبان پایتون انجام دهید؛ در این فعالیت مجاز به استفاده از tensorflow یا pytorch می‌باشید.

گزارش: ملاک اصلی انجام فعالیت، گزارش آن است و ارسال کد بدون گزارش فاقد ارزش است. برای این فعالیت یک فایل گزارش در قالب pdf تهیه کنید و در آن برای هر سوال، تصاویر ورودی، تصاویر خروجی و توضیحات کامل و جامعی تهیه کنید. تذکر: مطابق قوانین دانشگاه هر نوع کپی برداری و اشتراک کار دانشجویان غیر مجاز بوده و شدیداً برخورد خواهد شد. استفاده از کدها و توضیحات اینترنت به منظور یادگیری بلامانع است، اما کپی کردن غیرمجاز است.

راهنمایی: در صورت نیاز میتوانید سوالات خود را در خصوص پروژه از تدریس‌یارهای درس، از طریق ایمیل زیر یا در گروه تلگرامی بپرسید. (لینک گروه تلگرامی در سایت کورسز در دسترس بوده و قبلاً به همه‌ی دانشجویان ایمیل شده است)

Email: ann.ceit.aut@gmail.com

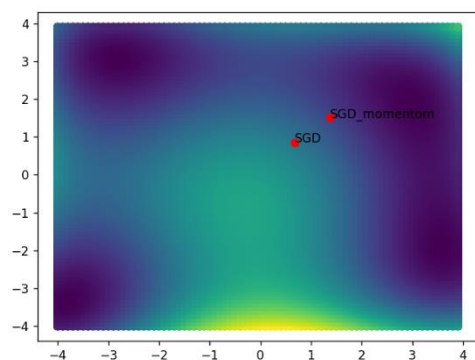
توجه: برای آموزش شبکه‌های عمیق می‌توانید از منابع و بسترهای سخت افزاری برخط رایگان نظیر Google Colab یا Kaggle استفاده نمایید.

تاخیر مجاز: در طول ترم، مجموعاً مجاز به حداکثر ده روز تاخیر برای ارسال تمرینات هستید (بدون کسر نمره). این تاخیر را می‌توانید بر حسب نیاز بین تمرینات مختلف تقسیم کنید؛ اما مجموع تاخیرات تمام تمرینات شما نباید بیشتر از ده روز شود. پس از استفاده از این تاخیر مجاز، هر روز تاخیر باعث کسر ۱۰٪ نمره‌ی آن تمرین خواهد شد.

ارسال: فایل‌های کد و گزارش خود را در قالب یک فایل فشرده با فرمت StudentID_HW02.zip تا تاریخ ۰۲/۰۱/۲۴ ارسال نمایید.

هزینه بالای آموزش شبکه‌های عصبی عمیق، همواره یکی از مهم‌ترین چالش‌های استفاده از آن‌ها بوده است. تعیین پارامترهای شبکه‌های عصبی عمیق، یک مسئله بهینه‌سازی با فضای جستجوی بسیار بزرگ است؛ بنابراین پس از تعریف یک تابع هزینه، معمولاً از روش‌های مبتنی بر نزول در راستای گرادیان برای حل آن استفاده می‌کنند. متأسفانه این روش‌ها، علاوه بر حساسیت به ابرپارامترهایی مانند ضریب یادگیری، یافتن بهینه سراسری را فقط در فضاهای جستجوی محدب^۱ تضمین می‌کنند؛ حال آنکه فضای جستجو در شبکه‌های عصبی معمولاً محدب نیستند. برای حل این مشکلات، طراحی الگوریتم‌های بهینه‌سازی مناسب، همواره یکی از زمینه‌های فعال در یادگیری عمیق بوده‌است. در این تمرین قصد داریم با حل مسائل ساده بهینه‌سازی، با برخی از الگوریتم‌های رایج بهینه‌سازی شبکه‌های عصبی عمیق آشنا شویم.

فرض کنید منحنی تابع هزینه دو شبکه عصبی به صورت‌هایی که در شکل ۱ نمایش داده شده‌اند، باشد. این منحنی‌ها توابع ارزیابی معروف برای الگوریتم‌های بهینه‌سازی هستند. برای آشنایی بیشتر با این توابع و دیدن مقادیر بهینه سراسری آن‌ها به لینک موجود در زیرنویس شکل ۱ مراجعه کنید.

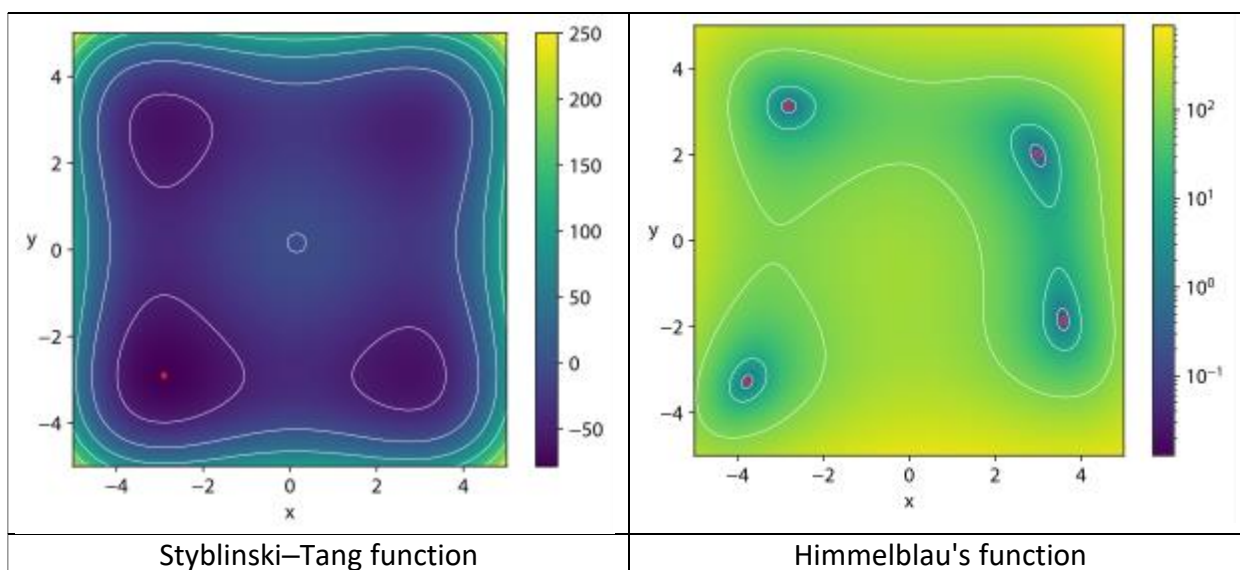


شکل ۲- یک فریم از خروجی موردنظر سوال

الف) برای تابع Himmelblau و به کمک ماژول بهینه‌سازی کتابخانه tensorflow یا pytorch، مسیر حرکت الگوریتم‌های بهینه‌سازی SGD و SGD با مومنتوم را در یک ویدیو به صورت متحرک نشان دهید(راهنمایی). خروجی شما باید یک ویدیو باشد که هر فریم از آن، یک مرحله از حرکت دو الگوریتم را نشان دهد (شکل ۲). نقطه شروع هر دو الگوریتم را (۰،۰) فرض کنید.

ب) با ضریب یادگیری یکسان، مقادیر ۰،۵ و ۰،۹ برای ضریب مومنتوم، چه تاثیری در سرعت همگرایی SGD دارد؟ این تاثیر را توجیه کنید.

ج) به نظر شما در آموزش یک شبکه عصبی، اندازه بچ چگونه در تعیین ضریب مومنتوم مناسب تاثیر گذار است؟



Styblinski-Tang function

$$f(\mathbf{x}) = \frac{\sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i}{2}$$

Himmelblau's function

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2.$$

شکل ۱- توابع مورد بررسی در این تمرین (منبع). نقاط قرمز رنگ محل بهینه(ها)ی سراسری هستند.

¹ convex

د) قسمت الف را برای تابع Styblinski–Tang با نقطه شروع (۴،۴) تکرار کنید. ضریب مومنتوم چگونه می‌تواند به فرار الگوریتم SGD از بهینه محلی کمک کند؟ (توجه: نیازی به تلاش برای رسیدن به بهینه سراسری نیست!)

ه) قسمت الف را برای الگوریتم ADAM و SGD تکرار کنید. حساسیت هر یک از الگوریتم‌ها را به مقدار ضریب یادگیری بررسی و نتایج را توضیح دهید.

و-امتیازی) به نظر شما چرا برخلاف عملکرد خوب ADAM در مسائل واقعی، در این مسئله بهینه‌سازی ساده عملکرد ضعیفی از خود نشان می‌دهد؟