



## پروژه پنجم

**هدف:** آشنایی با شبکه‌های عصبی بازخوردی<sup>۱</sup> و بهینه‌سازی.

**کد:** پیاده‌سازی این پروژه را به زبان پایتون انجام دهید؛ در این فعالیت مجاز به استفاده از tensorflow یا pytorch یا jax می‌باشید. فایل‌های کد خود را بر اساس شماره سوال و زیر قسمت خواسته شده‌ی آن نام گذاری کنید (برای مثال می‌توان نام گذاری قسمت اول برای سوال سوم تمرین را بصورت `P3_a_preprocessing.py` در نظر گرفت). فایل‌های ارسالی‌تان بایستی با فرمت `py` یا `ipynb` (با حفظ خروجی هر سلول) باشد.

**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 که دارای فهرست بوده و پاسخ‌ها بترتیب در آن قرار گرفته‌اند و نام، نام خانوادگی و شماره دانشجویی‌تان در قسمت چپ سربرگ تمامی صفحات تکرار شده است. علاوه بر خواسته‌ی مستقیم هر سوال، مقتضی است که نمودارهای خطا (loss) و صحت (accuracy) را به ازای مجموعه داده‌های آموزش و اعتبارسنجی رسم نمایید. همچنین در صورت امکان ماتریس درهم‌ریختگی را بصورت رنگ‌آمیزی شده به همراه اعداد متناظر برای مجموعه داده‌های آموزش، آزمون و اعتبارسنجی نیز تولید نمایید. لازم به ذکر است که در هر آموزش بایستی موارد مهم تنظیم شده نظیر تابع خطا، بهینه‌ساز (به همراه پارامترهای تنظیم شده‌ی آن مانند نرخ یادگیری)، معماری شبکه‌ی آموزشی (کتابخانه‌ها و ابزارهایی برای بصری‌سازی موجود است)، تعداد گام آموزشی، اندازه دسته (Batch Size)، آمارگان تفکیک مجموعه داده (به آموزش، آزمون و اعتبارسنجی)، پیش‌پردازش‌های اعمالی بروی داده‌گان ورودی و... ذکر گردد.

**تذکره:** مطابق قوانین دانشگاه هر نوع کپی برداری و اشتراک کار دانشجویان غیر مجاز بوده و با تمامی طرفین برخورد خواهد شد. استفاده از کدها و توضیحات اینترنت به منظور یادگیری صرفاً با ارجاع به آن بلامانع است، اما کپی کردن آن 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](#))

Email: [ann.ceit.aut@gmail.com](mailto:ann.ceit.aut@gmail.com) CC: [m.ebadpour@aut.ac.ir](mailto:m.ebadpour@aut.ac.ir)

**توجه:** می‌توانید از منابع و بسترهای سخت افزاری برخط رایگان نظیر Google Colab یا Kaggle استفاده نمایید.

**تاخیر مجاز:** در طول ترم، ده روز زمان مجاز تاخیر برای ارسال پروژه‌ها در اختیار دارید (بدون کسر نمره). این تاخیر را می‌توانید بر حسب نیاز بین پروژه‌ها مختلف تقسیم کنید که مجموع آن نباید بیشتر از ده روز شود. پس از استفاده از این تاخیر مجاز، هر روز تاخیر باعث کسر ۱۰٪ نمره‌ی کسب شده‌ی آن تمرین خواهد شد.

**ارسال:** فایل‌های کد و گزارش خود را در قالب یک فایل فشرده با فرمت StudentID\_HW05.zip تا تاریخ ۱۴۰۳/۴/۴ صرفاً از طریق سایت کورسز ارسال نمایید. ارسال از طریق تلگرام، ایمیل و سایر راه‌های ارتباطی مجاز نبوده و تصحیح صورت نخواهد گرفت.

<sup>1</sup> Recurrent Neural Networks

۱. به سوالات زیر بصورت خلاصه و برای هر یک حداکثر در سه بند پاسخ دهید:

- i. بطور کلی بهینه‌سازها<sup>۲</sup> (نظیر ADAM) به دنبال یافتن وزن‌های شبکه‌های عصبی هستند بطوریکه که توابع هزینه (Loss Functions) کمینه شود. مشتق‌پذیر بودن توابع یاد شده چه تاثیری در بهینه‌ساز دارد؟ اگر مشتق‌پذیر نباشد، چه رویکردهایی برای بهینه‌سازی آن وجود دارد؟ یک مورد را به دلخواه توضیح دهید. (۷ امتیاز)
  - ii. محدب<sup>۳</sup> بودن توابع به چه معناست و چرا مطلوب است که در بهینه‌سازی، توابع هزینه محدب باشد؟ اگر محدب نباشد، چگونه میتوان آن را بهینه نمود؟ (۹ امتیاز)
  - iii. الگوریتم بهینه‌سازی نیوتن را مطالعه کرده و آن را با نزول در راستای گرادیان<sup>۴</sup> مقایسه کنید. در چه نوع مسائلی استفاده از الگوریتم نیوتن ارجحیت دارد؟ (۱۰ امتیاز)
  - iv. ضمن مطالعه کلی الگوریتم AdaGrad، بیان کنید که چگونه می‌توان از آن برای بهینه ساختن نرخ یادگیری (Learning Rate) بهره گرفت. (۶ امتیاز)
- فرض کنید مسئله‌ی دسته بندی دودویی بحرانی بودن/نبودن شرایط یک کارگاه صنعتی بر اساس اطلاعاتی محیطی آن را در اختیار دارید که داده‌های دما، رطوبت، فشار و ذرات معلق بر اساس سنسورهای نصب شده در هر یک ثانیه ارسال می‌گردد. شما بایستی با در نظر گرفتن دنباله‌ای از داده‌های ارسالی بتوانید تشخیص دهید که شرایط بحرانی است یا خیر.
- v. یک شبکه‌ی بازخردادی Elman که با دولاپی‌ی مخفی که به ترتیب سه و دو نورون تعبیه شده است، طراحی نمایید و تعداد وزن‌های مورد نیاز برای یادگیری در این شبکه را با بیان علت محاسبه نموده و ابعاد تمامی بردارهای (Vectors & Tensors) مشاهده شده در شبکه (ورودی‌ها/میانی‌ها/خروجی‌ها) را با محاسبات و استدلال نمایش دهید. انتظار می‌رود که شما بتوانید سیر تغییرات ابعاد بردارها و چگونگی آن را نشان دهید؛ مثلاً شکل بردار ورودی برای یک دسته (batch) چگونه تعیین می‌شود و تا رسیدن به خروجی شکل آن چرا و چگونه تغییر پیدا کرده است و با چه وزن‌هایی متأثر شده است. (۱۳ امتیاز)
- آیا تا کنون به روند عملکرد بهینه‌سازها فکر کرده‌اید؟ آیا می‌توان آن را یک شبکه‌ی بازخردادی در نظر گرفت؟ در این پروژه هدف طراحی و پیاده‌سازی یک بهینه‌ساز می‌باشد. برای درک بهتر، توضیحات ریاضیاتی زیر داده می‌شود.

---

<sup>2</sup> Optimizers

<sup>3</sup> Convex

<sup>4</sup> Gradient Descent

روند یادگیری پارامترهای ( $\theta$ ) یک شبکه عمیق ( $f$ ) با الگوریتم های مرسوم نزول در راستای گرادیان (نظیر SGD) را می توان به ازای ورودی های آموزشی  $x$  بصورت رابطه ی (۱) در نظر گرفت:

$$\theta_{i+1} = \theta_i - \alpha \nabla f(x; \theta_i) \quad (۱)$$

حال اگر فرض شود که به جای نرخ یادگیری ثابت  $\alpha$  از یک تابع (شبکه ی عمیق) نظیر  $g$  با پارامترهای قابل یادگیری  $\phi$  استفاده کنیم، می توان رابطه ی (۱) را بصورت رابطه ی (۲) بازنویسی نمود:

$$\theta_{i+1} = \theta_i + g(\nabla f(x; \theta_i); \phi) \quad (۲)$$

در نهایت می توان پارامتر  $\theta_i$  را نیز به عنوان یک ورودی دیگر به  $g$  در نظر گرفت و رابطه ی (۲) را نیز بصورت زیر بازنویسی کرد:

$$\theta_{i+1} = g(\nabla f(x; \theta_i), \theta_i; \phi) \quad (۳)$$

حال می توان نتیجه گرفت که اگر تابع  $g$  را یک شبکه ی بازخردادی (نظیر LSTM یا GRU) در نظر گرفت، امکان ارائه ی یک بهینه ساز وجود دارد. کل فرآیند یاد شده را می توان با دو حلقه (بیرونی<sup>۵</sup> و درونی<sup>۶</sup>) انجام داد که معماری کلی آن در ادامه ضمیمه شده است. پیمایش یکبار حلقه ی بیرونی معادل است با یک تکرار (Epoch) برای آموزش شبکه ی  $g$  و پیمایش یکبار حلقه درونی معادل است با تولید یک داده ی آموزشی برای شبکه ی  $g$ .

۲. در این سوال هدف طراحی و پیاده سازی یک بهینه ساز بر اساس توضیحات فوق می باشد:

- i. مجموعه داده اول را به عنوان مجموعه داده آموزشی در نظر بگیرید. آن را نمایش داده و پس درهم سازی به ۵۰ زیر مجموعه تقسیم کنید بطوریکه در هر مجموعه داده از هر کلاس به تعداد برابر نمونه موجود باشد و انتخاب نمونه ها نیز با احتمال یکنواخت صورت گرفته باشد. حال، یک شبکه ی  $MLP$  دلخواه طراحی نمایید و آن را  $f$  بنامید که  $f_1, f_2, \dots$  شبکه های  $MLP$  با معماری یکسان هستند و صرفاً مقداردی اولیه ی آن ها برابر متفاوت است. (۱۵ امتیاز)
- ii. یک شبکه ی بازخردادی مبتنی بر  $GRU$  طراحی نمایید و آن را  $g$  بنامید که وظیفه ی آن بهینه سازی وزن های قابل یادگیری معماری  $f$  برای هدف مورد نظر می باشد. با استفاده از ۵۰ زیر مجموعه ی ایجاد شده در قسمت قبل، شبکه ی  $g$  را آموزش دهید. روند پیاده سازی آموزش، معماری طراحی شده و سایر جزئیات مورد نظر را در گزارش درج نمایید. توجه داشته باشید به ازای هر حلقه ی درونی (در هر حلقه ی بیرونی)، یک مقداردی کاملاً جدید برای شبکه ی  $f$  صورت میگیرد. در هر زیر مجموعه، نسبت آموزش به آزمون را ۸:۲ در نظر بگیرید. (۴۲ امتیاز)

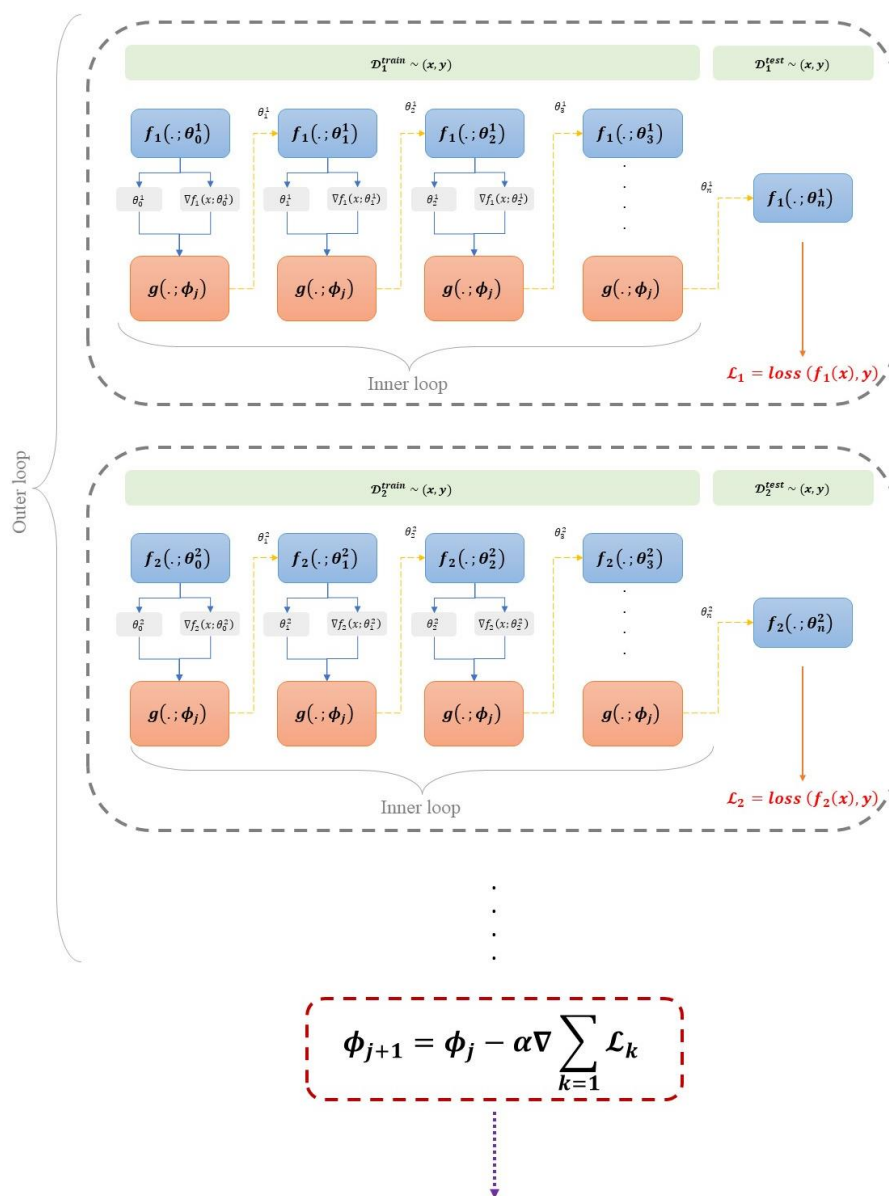
<sup>5</sup> Outer Loop

<sup>6</sup> Inner Loop

.iii مجموعه داده‌ی دوم را نیز بارگذاری کرده و آن را نمایش دهید و تفاوت های آن را با مجموعه داده‌ی اول بیان کنید. حال آن را به ۳۰ زیر مجموعه همانند توضیحات قسمت (i) تقسیم کنید. در نهایت، به ازای هر مجموعه داده یک شبکه با معماری  $f$  در نظر گرفته و با شبکه‌ی  $g$  آن را بهینه‌سازی کنید. میانگین دقت و خطا را گزارش نمایید. (۲۸ امتیاز)

.iv (اختیاری) با مطالعه و تحقیق روشی ارائه دهید تا بتوان عملکرد بهینه‌ساز  $g$  را بصورت کمی و کیفی ارزیابی نموده و همچنین بتوان آن را با بهینه‌ساز  $ADAM$  مقایسه نمود. (۱۰ امتیاز)

ضمیمه: (فایل با کیفیت نیز قرار شده است)



Learning  $\phi$  parameters of  $g$  network can be handled by any predefined optimizer like ADAM, SGD, SAG.