



مشتق‌گیری خودکار به‌عنوان چارچوبی عمومی برای حل رده‌ای از مسائل بهینه‌سازی

محمود امین‌طوسی*

گروه علوم کامپیوتر، دانشکده علوم ریاضی، دانشگاه فردوسی مشهد

m.amintoosi@um.ac.ir

چکیده. مشتق‌گیری خودکار یکی از ابزارهای کلیدی در پیاده‌سازی روش‌های مبتنی بر گرادیان در یادگیری ماشین است که امکان محاسبه‌ی دقیق و کارای مشتق‌ها را برای توابع پیچیده فراهم می‌کند. در این مقاله الگوریتم کلاسیک گرادیان کاهشی، در چارچوب مشتق‌گیری خودکار با نمایش چند مثال جذاب بازخوانی شده است. در این رویکرد، مسئله‌ی بهینه‌سازی به‌صورت یک تابع هدف قابل ارزیابی در برنامه‌نویسی فرموله می‌شود، حتی در مواردی که تابع هدف دارای بیان تحلیلی ساده یا متعارف در ریاضیات کلاسیک نیست، گرادیان‌ها به‌طور خودکار محاسبه می‌شوند و الگوریتم گرادیان کاهشی بدون تغییر مفهومی به‌کار گرفته می‌شود. نکته‌ی اصلی آن است که بسیاری از مسائل ظاهراً ناهمگون را می‌توان صرفاً با تعریف مناسب تابع هدف، در یک چارچوب محاسباتی مشترک حل کرد. برای تبیین تدریجی این دیدگاه، چهار مثال با پیچیدگی فزاینده بررسی شده‌اند. رگرسیون خطی به‌عنوان نخستین مثال انتخاب شده است، زیرا مسئله‌ای آشنا با فرم بسته‌ی تحلیلی است و بستری شفاف برای نمایش ارتباط میان گرادیان کاهشی، مشتق‌گیری خودکار، و الگوریتم‌های پایه‌ی یادگیری ماشین فراهم می‌کند. در ادامه، مثال‌هایی ارائه می‌شوند که اگرچه در ریاضیات کلاسیک تابع هدف صریح و تمیز ندارند، اما در قالب یک تابع هدف محاسباتی قابل بیان، و با استفاده از مشتق‌گیری خودکار قابل حل هستند. هدف مقاله، ارائه‌ی متنی آموزشی و انگیزه‌بخش است که نشان دهد چگونه مشتق‌گیری خودکار می‌تواند پلی میان بهینه‌سازی عددی کلاسیک و کاربردهای مدرن یادگیری ماشین ایجاد کند، به‌گونه‌ای که برای دانشجویان علوم ریاضی و علوم کامپیوتر قابل‌درک و الهام‌بخش باشد.

2020 Mathematics Subject Classification. 68T07, 68U10, 68U10, 68T45.

واژگان کلیدی. مشتق‌گیری خودکار، گرادیان کاهشی، یادگیری ماشین، تابع هدف، بهینه‌سازی عددی.
تاریخ: دریافت ۱۴۰۴/۱۱/۰۳ بازنگری ۱۴۰۴/۱۱/۲۳ پذیرش ۱۴۰۵/۰۳/۳۱ انتشار برخط ۱۴۰۵/۰۳/۳۱
نحوه ارجاع به این مقاله: م. امین‌طوسی، مشتق‌گیری خودکار به‌عنوان چارچوبی عمومی برای حل رده‌ای از مسائل بهینه‌سازی، ۶ (۱۴۰۵)، شماره ۱، ۱-۱۴.
©دانشگاه فردوسی مشهد.

۱. پیش‌گفتار

در بسیاری از متون کلاسیک بهینه‌سازی و یادگیری ماشین، تمرکز اصلی بر طراحی مدل‌ها و استخراج روابط تحلیلی برای گرادین‌ها است. در این رویکرد، فرم صریح تابع هدف و مشتقات آن نقش تعیین‌کننده‌ای در امکان‌پذیری حل مسئله ایفا می‌کنند. با این حال، ظهور چارچوب‌های مشتق‌گیری خودکار^۱ این نگاه را دگرگون کرده است. در این چارچوب‌ها، دیگر نیازی به محاسبه‌ی تحلیلی گرادین‌ها وجود ندارد؛ کافی است مسئله به صورت یک فرآیند/تابع محاسباتی مشتق‌پذیر فرموله شود.

هدف این مقاله تأکید بر این نکته‌ی کلیدی است که اهمیت اصلی مشتق‌گیری خودکار نه در سرعت محاسبه‌ی گرادین‌ها، بلکه در آزادی عمل برای تعریف تابع هدف/زیان نهفته است. به بیان دیگر، هر مسئله‌ای که بتوان آن را به صورت یک نگاشت محاسباتی مناسب از متغیرهای تصمیم به یک مقدار اسکالر تعریف کرد، در چارچوب بهینه‌سازی مبتنی بر گرادین قابل حل خواهد بود؛ حتی اگر این مسئله دارای ماهیتی غیرکلاسیک باشد.

در این مقاله، به جای تمرکز بر یک کاربرد خاص، یک الگوی مشترک حل مسئله مورد توجه قرار می‌گیرد و نشان داده می‌شود که چگونه مسائل ظاهراً نامرتب می‌توانند ذیل این الگو قرار گیرند. مثال‌ها به صورت هدفمند از ساده به پیچیده مرتب شده‌اند: از رگرسیون خطی به عنوان یک مسئله‌ی کلاسیک و آشنا، تا چیدمان گراف، انتقال سبک عصبی، و در نهایت نقاشی با شبکه‌های عصبی. در تمامی این مثال‌ها، الگوریتم بهینه‌سازی تغییری نمی‌کند و تنها تعریف متغیرهای تصمیم و تابع زیان متفاوت است. مخاطب اصلی مقاله دانشجویان و پژوهشگران علوم ریاضی و علوم کامپیوتر هستند که با مفاهیم پایه‌ای بهینه‌سازی و یادگیری ماشین آشنایی دارند، اما لزوماً پیش‌زمینه‌ای در حوزه‌هایی مانند گرافیک کامپیوتری، پردازش تصویر، یا فیزیک محاسباتی ندارند. تمام مثال‌ها به گونه‌ای انتخاب شده‌اند که بدون نیاز به دانش تخصصی میان‌رشته‌ای قابل فهم باشند و نقش مشتق‌گیری خودکار را به صورت شفاف برجسته کنند.

در ادامه‌ی مقاله، این چارچوب کلی از طریق مجموعه‌ای از مثال‌ها با پیچیدگی فزاینده بررسی می‌شود. ترتیب مثال‌ها به گونه‌ای انتخاب شده است که خواننده به تدریج از یک مسئله‌ی کلاسیک با ساختار تحلیلی روشن به مسائلی برسد که در آن‌ها تعریف تابع زیان پیچیده‌تر، فضای متغیرها بزرگ‌تر، و مشتق‌گیری تحلیلی عملاً غیرممکن است. هدف آن است که نشان داده شود بدون تغییر در الگوریتم گرادین کاهشی، و صرفاً با بازتعریف تابع زیان، می‌توان طیف گسترده‌ای از مسائل را در یک چارچوب محاسباتی واحد حل کرد.

۲. چارچوب کلی حل مسائل بهینه‌سازی با مشتق‌گیری خودکار

در تمامی مثال‌های ارائه شده در این مقاله، مسئله به صورت کمینه‌سازی یک تابع زیان^۲ مشتق‌پذیر بر حسب متغیرهای تصمیم فرموله می‌شود و سپس با استفاده از الگوریتم گرادین کاهشی حل می‌گردد. نکته‌ی اساسی آن است که گرادین‌ها نه به صورت تحلیلی، بلکه با استفاده از مشتق‌گیری خودکار محاسبه می‌شوند. چارچوب کلی این فرآیند در الگوریتم^۱ خلاصه شده است.

¹Automatic Differentiation

²Loss Function

الگوریتم ۱ چارچوب کلی بهینه‌سازی مبتنی بر گرادیان کاهشی با مشتق‌گیری خودکار

ورودی: تابع زیان $\mathcal{L}(x)$ ، مقدار اولیه‌ی متغیرهای تصمیم x_0 ، نرخ یادگیری $\eta > 0$ ، شرط توقف خروجی: مقدار بهینه‌شده‌ی متغیرها (پارامترها)

۱: مقداردهی اولیه: $x \leftarrow x_0$

۲: تا زمانی که شرط توقف برقرار نشده است به صورت زیر پارامترها را بروزرسانی کن:

$$x \leftarrow x - \eta \nabla_x \mathcal{L}(x)$$

▷ محاسبه‌ی گرادیان با مشتق‌گیری خودکار

۳: x را به عنوان مقدار بهینه‌ی پارامترها برگردان

مشتق‌گیری خودکار چیست؟

مشتق‌گیری خودکار روشی عددی برای محاسبه‌ی دقیق گرادیان‌هاست که بر پایه‌ی تجزیه‌ی تابع به عملیات ابتدایی و استفاده از قاعده‌ی زنجیره‌ای عمل می‌کند. برخلاف مشتق‌گیری نمادین، منجر به فرمول‌های تحلیلی نمی‌شود و برخلاف تفاضلات متناهی، دچار خطای تقریب نیست. مشتق‌گیری خودکار یک الگوریتم بهینه‌سازی نیست؛ بلکه ابزاری برای محاسبه‌ی گرادیان‌هاست. این ابزار زمانی بیشترین اهمیت را پیدا می‌کند که تابع هدف به‌صورت یک فرآیند محاسباتی پیچیده تعریف شده باشد، به‌گونه‌ای که مشتق‌گیری دستی از آن عملاً امکان‌پذیر نباشد.

در بخش‌های بعدی نشان داده می‌شود که اگرچه این الگوریتم از نظر ظاهری بسیار ساده است، اما انتخاب مناسب متغیرهای تصمیم و تعریف تابع زیان به‌صورت یک فرآیند محاسباتی مشتق‌پذیر، امکان حل طیف گسترده‌ای از مسائل را فراهم می‌کند؛ از مسائل آماری کلاسیک گرفته تا فرآیندهای تولیدی پیچیده. در چهار بخش بعد به حل چهار مسأله‌ی زیر با این روال کلی مبتنی بر مشتق‌گیری خودکار پرداخته خواهد شد:

(۱) رگرسیون خطی

(۲) ترسیم (نمایش) گراف در فضای دوبعدی

(۳) انتقال سبک عصبی

(۴) نقاشی با شبکه‌های عصبی

هر یک از این موارد در بخش مربوطه تعریف خواهد شد.

۳. رگرسیون خطی و گرادیان کاهشی: نقطه‌ی شروع

به‌عنوان نقطه‌ی شروع، مسئله‌ی رگرسیون خطی بررسی می‌شود؛ مثالی آشنا برای اغلب دانشجویان علوم ریاضی که امکان تمرکز بر چارچوب کلی بهینه‌سازی را بدون درگیر شدن با پیچیدگی‌های محاسباتی فراهم می‌کند. به فرض مجموعه‌ای از داده‌ها به‌صورت $\{(x_i, y_i)\}_{i=1}^n$ در اختیار است و هدف، تقریب این داده‌ها با یک مدل خطی از فرم $\hat{y} = w_0 + w_1 x$ می‌باشد.

۱.۳. رگرسیون به عنوان یک مسئله‌ی بهینه‌سازی. در رگرسیون خطی، برای سنجش کیفیت تقریب، تابع خطای میانگین مربعات (MSE)^۳ به‌عنوان تابع هدف انتخاب می‌شود:

$$\mathcal{L}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^n (w_0 + w_1 x_i - y_i)^2.$$

^۳Mean Squared Error (MSE)

که به صورت یک مسئله بهینه‌سازی، به عنوان یک مسئله یادگیری - پیدا کردن w_j ها - قابل بیان است:

$$\min_{w_0, w_1} \mathcal{L}(w_0, w_1).$$

برای حل این مسئله، از الگوریتم گرادیان کاهشی (الگوریتم ۱)^۴ استفاده می‌شود که در آن پارامترها به صورت تکراری و بر اساس گرادیان تابع خطا به روزرسانی می‌شوند:

$$w_j \leftarrow w_j - \eta \frac{\partial \mathcal{L}}{\partial w_j}$$

که در آن η نرخ یادگیری است.

در این مثال، مشتق‌های جزئی تابع خطا نسبت به پارامترها به صورت تحلیلی به سادگی قابل محاسبه هستند و حتی فرم حل بسته‌ی^۵ مسئله نیز وجود دارد، اما در مواردی که روش حل بسته کارا نیست و منجمله در شبکه‌های عصبی و یادگیری عمیق از الگوریتم گرادیان کاهشی برای حل آن استفاده می‌شود [۱].

۲.۳. نقش مشتق‌گیری خودکار. محاسبه مشتق تابع هدف (تابع زیان، تابع خطا یا تابع انرژی)^۶ بخشی ضروری در الگوریتم گرادیان کاهشی هست و چارچوب‌های یادگیری ماشین مانند پای‌تورچ^۷ و تانسورفلو^۸ امکان مشتق‌گیری خودکار تابع هدف را دارند و محاسبه‌ی گرادیان‌ها می‌تواند به قابلیت مشتق‌گیری خودکار این ابزارها واگذار گردد. در این حالت، تابع خطا صرفاً به عنوان یک فرآیند محاسباتی تعریف شده و گرادیان آن نسبت به پارامترها به صورت خودکار محاسبه می‌شود، بی‌آنکه نیازی به استخراج و پیاده‌سازی مشتق‌های تحلیلی وجود داشته باشد. در مسئله‌ای به سادگی رگرسیون خطی فوق‌الذکر، مشتق‌گیری خودکار مزیت بنیادینی نسبت به روش‌های کلاسیک ایجاد نمی‌کند اما پایه و اساس همه روش‌های یادگیری در شبکه‌های عصبی و مدل‌های یادگیری عمیق است [۱، ۳، ۸]. هدف از طرح این مثال، نمایش چارچوب کلی حل مسئله و آماده‌سازی ذهن خواننده برای مثال‌های بعدی است که در آن‌ها محاسبه‌ی تحلیلی مشتق‌ها عملاً امکان‌پذیر نخواهد بود.

۳.۳. نتیجه‌ی محاسباتی. اجرای الگوریتم بهینه‌سازی منجر به برازش یک خط بر روی داده‌ها می‌شود که کمینه‌کننده‌ی تابع خطای تعریف شده است. نمایش گرافیکی این خط در کنار داده‌های اولیه، به خوبی همگرایی الگوریتم و صحت فرموله‌سازی مسئله را نشان می‌دهد. در شکل ۱، نتیجه‌ی روال بهینه‌سازی الگوریتم ۱ برای برآورد رگرسیون خطی تعدادی داده مصنوعی نشان داده شده است. داده‌های واقعی با نقاط آبی نمایش داده شده‌اند. در مثال‌های بعدی، مشاهده خواهد شد که اگرچه ساختار کلی الگوریتم تغییری نمی‌کند، اما ماهیت تابع هدف و پیچیدگی محاسبه‌ی گرادیان‌ها به طور چشمگیری افزایش می‌یابد.

هدف از طرح این مثال، نه ارائه‌ی یک روش جدید، بلکه تثبیت چارچوب کلی حل مسئله است؛ چارچوبی که در مثال‌های بعدی بدون تغییر باقی می‌ماند، در حالی که ماهیت مسئله به طور چشمگیری پیچیده‌تر می‌شود. در این مثال، چارچوب کلی بهینه‌سازی کاملاً با الگوریتم ۱ منطبق است؛ تنها تابع هدف و متغیرهای تصمیم مشخص شده‌اند. در ادامه‌ی مقاله، با افزایش تدریجی پیچیدگی مسائل، نشان داده خواهد شد که مشتق‌گیری خودکار از یک ابزار کمکی به یک مؤلفه‌ی ضروری در حل مسئله تبدیل می‌شود.

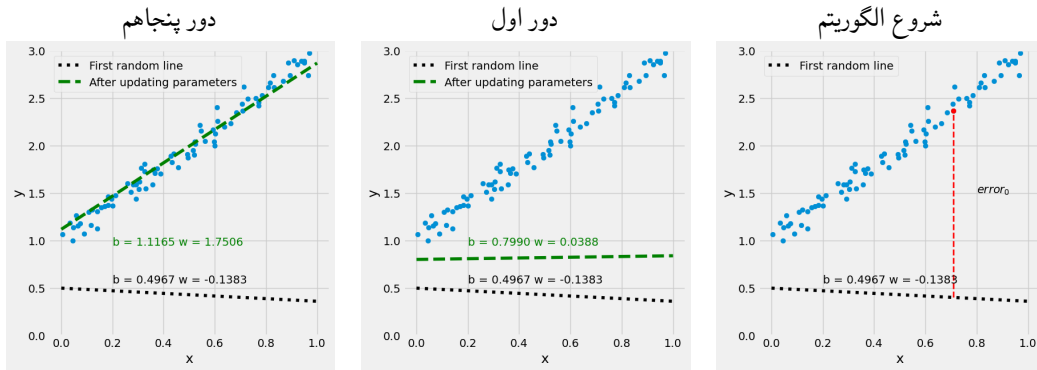
⁴Gradient Descent

⁵Closed Form Solution

⁶Objective, Loss, Error or Energy Function

⁷PyTorch

⁸TensorFlow



شکل ۱: برازش رگرسیون خطی $y = wx + b$ با استفاده از گرادینت کاهشی و مشتق‌گیری خودکار در پای‌تورچ. نقاط آبی داده‌های نویزی تولیدشده را نشان می‌دهند. خط‌چین سیاه بیانگر مقداردهی اولیه‌ی مدل و خط‌چین سبز مراحل دوم و آخر روال بهینه‌سازی است.

۴. ترسیم گراف به‌عنوان یک مسئله‌ی بهینه‌سازی

در این بخش، مسئله‌ی ترسیم گراف به‌عنوان یک مسئله‌ی بهینه‌سازی پیوسته بررسی می‌شود. این مسئله، اگرچه در ظاهر ماهیتی بصری دارد، اما در بنیان خود کاملاً ریاضی بوده و می‌توان آن را در قالب کمینه‌سازی یک تابع هدف اسکالر فرموله کرد. فرض کنید یک گراف بدون جهت $G = (V, E)$ با مجموعه‌ی رأس‌ها V و مجموعه‌ی یال‌ها E داده شده است. هدف، انتساب یک موقعیت دوبعدی به هر رأس

$$v \in V \mapsto \mathbf{p}_v = (x_v, y_v) \in \mathbb{R}^2$$

به‌گونه‌ای است که نمایش حاصل، از نظر ساختاری و بصری خوانا باشد. شکل ۲ یک چیدمان تصادفی از یک گراف مصنوعی مشبک را نشان می‌دهد. همان‌گونه که مشاهده می‌شود، با اینکه می‌دانیم گراف، یک گراف مشبک هست اما با انتخاب تصادفی مختصات رأس‌ها، نمایش نامنظمی دارد. هدف آن هست که مختصات رئوس گراف در صفحه به گونه‌ای پیدا شوند که گراف نمایش مناسبی داشته باشد. به این ترتیب، مسئله‌ی ترسیم گراف به یافتن یک ماتریس مختصات X تقلیل می‌یابد.

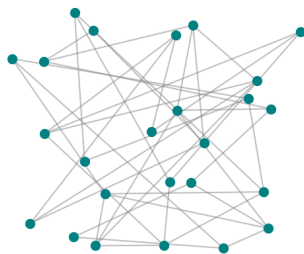
$$X = \begin{bmatrix} \mathbf{p}_1^\top \\ \vdots \\ \mathbf{p}_{|V|}^\top \end{bmatrix} \in \mathbb{R}^{|V| \times 2}.$$

۱.۴. فرم ریاضی مسئله. برای تبدیل مسئله‌ی ترسیم گراف به یک مسئله‌ی بهینه‌سازی، مجموعه‌ای از معیارهای کیفی در قالب یک تابع هدف/انرژی/زیان تعریف می‌شود. این معیارها معمولاً شامل موارد زیر هستند:

- نزدیک‌بودن رأس‌های متصل توسط یال،
- فاصله‌گرفتن رأس‌های غیرمتصل از یکدیگر،
- جلوگیری از هم‌پوشانی رأس‌ها.

این ملاحظات به‌صورت جملات ریاضی در تابع هدف $\mathcal{L}(X)$ ترکیب می‌شوند.

۲.۴. تعریف تابع هدف. تابع هدف از دو مؤلفه‌ی اصلی جاذبه و دافعه تشکیل شده است:



شکل ۲: چیدمان تصادفی رئوس گراف در فضای دو بعدی.

۱. جمله‌ی جاذبه (یال‌ها). برای هر یال $(u, v) \in E$ ، رأس‌های متناظر باید به یکدیگر نزدیک باشند. این ویژگی با جمله‌ی زیر مدل می‌شود:

$$L_{\text{attract}}(X) = \sum_{(u,v) \in E} \|\mathbf{p}_u - \mathbf{p}_v\|^2.$$

۲. جمله‌ی دافعه (رأس‌ها). برای جلوگیری از تجمع و هم‌پوشانی رأس‌ها، تمام جفت‌های متمایز رأس‌ها یک نیروی دافعه را تجربه می‌کنند:

$$L_{\text{repulse}}(X) = \sum_{u \neq v} \frac{1}{\|\mathbf{p}_u - \mathbf{p}_v\| + \varepsilon},$$

که در آن $\varepsilon > 0$ یک مقدار کوچک برای جلوگیری از تقسیم بر صفر است. تابع هدف نهایی. با ترکیب دو مؤلفه‌ی فوق، تابع هدف نهایی به‌صورت زیر تعریف می‌شود:

$$\mathcal{L}(X) = L_{\text{attract}}(X) + \lambda L_{\text{repulse}}(X),$$

که در آن $\lambda > 0$ ضریب تنظیم تعادل بین جاذبه و دافعه است. در نتیجه، مسئله‌ی ترسیم گراف به مسئله‌ی بهینه‌سازی زیر تقلیل می‌یابد:

$$\min_{X \in \mathbb{R}^{|V| \times 2}} \mathcal{L}(X).$$

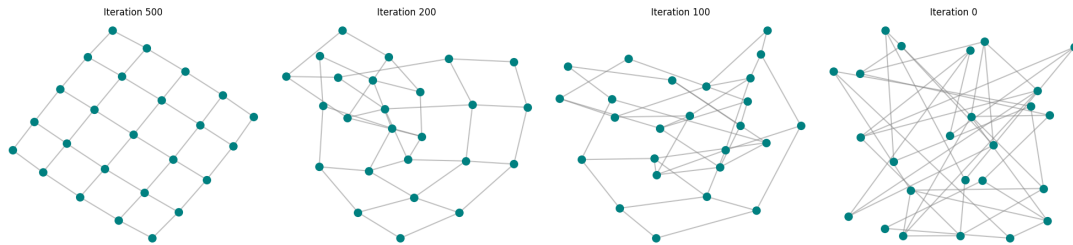
۳.۴. چالش مشتق‌گیری تحلیلی. اگرچه هر یک از جملات تابع هدف به‌تنهایی ساختاری ساده دارند، اما در عمل:

- تعداد جملات تابع هدف با اندازه‌ی گراف رشد می‌کند،
- هر متغیر مکانی به‌طور هم‌زمان در جملات متعدد ظاهر می‌شود،
- و تغییر یا افزودن معیارهای جدید مستلزم بازنویسی کامل گرادیان‌ها است.

در چنین شرایطی، مشتق‌گیری تحلیلی از تابع $\mathcal{L}(\cdot)$ فرآیندی پیچیده، مستعد خطا و فاقد انعطاف‌پذیری خواهد بود.

۴.۴. نقش مشتق‌گیری خودکار. در چارچوب مشتق‌گیری خودکار، تابع هدف $\mathcal{L}(X)$ صرفاً به‌عنوان یک نگاشت محاسباتی تعریف می‌شود و گرادیان آن نسبت به متغیرهای مکانی $-\nabla_X \mathcal{L}$ - به‌صورت خودکار محاسبه می‌گردد. در این رویکرد:

- مختصات رأس‌ها به‌عنوان متغیرهای قابل مشتق در نظر گرفته می‌شوند،
- تابع هدف می‌تواند شامل توابع غیرخطی و غیرساده باشد،
- و الگوریتم گرادیان نزولی بدون مشتق‌گیری دستی اجرا می‌شود.



شکل ۳: بهینه‌سازی چیدمان گراف با کمیینه‌سازی تابع انرژی $\mathcal{L}$. نمایش موقعیت رأس‌ها در چند تکرار مختلف از الگوریتم بهینه‌سازی.

در شکل ۳، وضعیت گراف در چند تکرار مختلف از فرایند بهینه‌سازی نمایش داده شده است. مشاهده می‌شود که ساختار منظم گراف به تدریج از مقداردهی اولیه تصادفی بازیابی می‌شود.

۵.۴. نتیجه‌ی محاسباتی. اجرای فرایند بهینه‌سازی منجر به جابه‌جایی تدریجی رأس‌ها و کاهش یکنواخت مقدار تابع هدف $\mathcal{L}(\cdot)$ می‌شود. در نتیجه، آرایشی از رأس‌ها حاصل می‌گردد که ساختار گراف را به شکل منظم‌تری بازتاب می‌دهد. نکته‌ی قابل توجه آن است که ساختار کلی الگوریتم بهینه‌سازی در این مثال هیچ تفاوت مفهومی با مثال رگرسیون خطی ندارد؛ تفاوت اصلی صرفاً در تعریف و پیچیدگی تابع هدف نهفته است. این مثال نشان می‌دهد که:

- تابع هدف می‌تواند حاصل ترکیب معیارهای متنوع و غیرساده باشد،
- یک مسئله‌ی گرافی را می‌توان به صورت یک مسئله‌ی پیوسته مدل کرد،
- و مشتق‌گیری خودکار ابزاری کلیدی برای حل چنین مسائل پیچیده‌ای است.

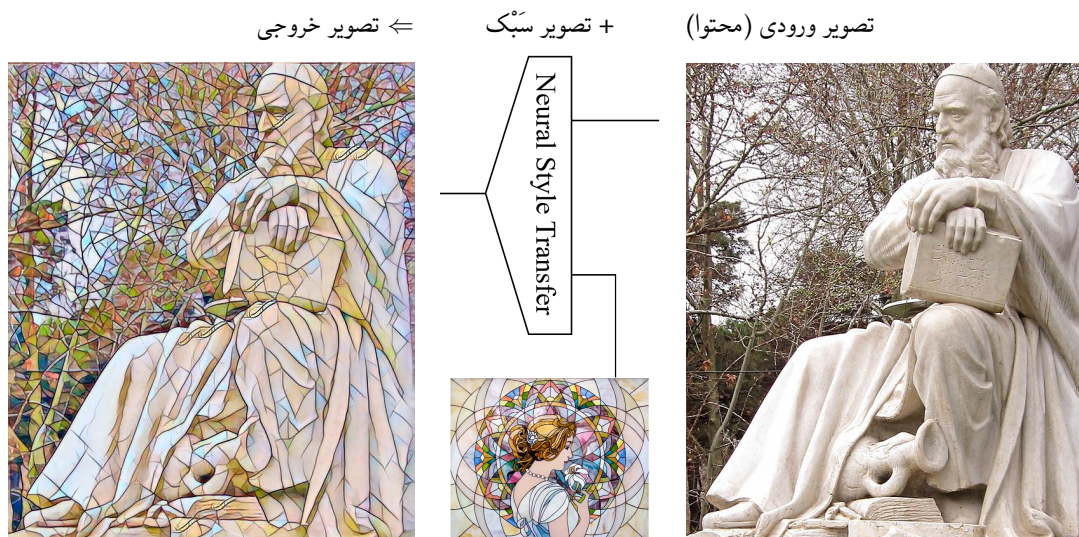
در مقایسه با مثال رگرسیون خطی، در اینجا مشتق‌گیری تحلیلی نه تنها زمان‌بر، بلکه از نظر پیاده‌سازی نیز غیرمنعطف است؛ در حالی که چارچوب مشتق‌گیری خودکار امکان آزمایش و تغییر سریع تابع هدف را فراهم می‌کند. از دیدگاه الگوریتم کلی، تفاوتی میان این مثال و رگرسیون خطی وجود ندارد؛ تنها ساختار و پیچیدگی تابع هدف تغییر کرده است. در مثال بعدی، نشان داده خواهد شد که همین ایده برای داده‌هایی با ساختار پیچیده‌تر، مانند تصاویر، نیز قابل استفاده است.

۵. انتقال سبک عصبی در چارچوب بهینه‌سازی مبتنی بر مشتق‌گیری خودکار

در این بخش، مسئله‌ای را خواهیم دید که در آن متغیر تصمیم دارای هزاران و یا میلیون‌ها مؤلفه است. مسئله‌ی انتقال سبک عصبی^۹ [۴] یک نمونه از چنین مسائلی است که در همان چارچوب کلی بهینه‌سازی معرفی شده در ابتدای مقاله تحلیل می‌شود. منظور از انتقال سبک عصبی [۵، ۶، ۷]؛ روشی است که با بهره‌گیری از شبکه‌های عصبی پیچشی^{۱۰} می‌تواند سبک بصری یک تصویر را به تصویر دیگر منتقل سازد. شکل ۴ یک نمونه انتقال سبک عصبی را نشان می‌دهد که در آن سبک (استایل) تصویر وسط بر روی تصویری از مجسمه‌ی عمر خیام اعمال شده است. هدف در این بخش آن است که نشان داده شود بدون تغییر در ساختار الگوریتم

^۹Neural Style Transfer

^{۱۰}Convolutional Neural Networks (CNNs)



شکل ۴: تصویر وسط به عنوان سبک (استایل) بر روی تصویری از عمر خیام (به عنوان ورودی) اعمال شده و تصویر خروجی به دست آمده است. برنامه مرتبط با این شکل از گیت‌هاب زیر قابل دسترس است: <https://github.com/eriklindernoren/Fast-Neural-Style-Transfer>.

بهینه‌سازی، تنها با تغییر در تعریف تابع هدف $\mathcal{L}(\cdot)$ ، می‌توان از مسائل بسیار ساده (رگرسیون خطی) به مسائلی با ابعاد بسیار بالا (اعمال سبک تصویر) گذر کرد.

۱.۵. متغیر تصمیم. برخلاف مثال‌های پیشین، در این مسئله متغیر تصمیم پارامترهای یک مدل نیست، بلکه خودِ تصویر خروجی است. به عبارت دیگر، تصویر $x \in \mathbb{R}^{H \times W \times 3}$ به عنوان بردار متغیرها در نظر گرفته می‌شود. هر پیکسل (و هر کانال رنگی) یک مؤلفه از این بردار را تشکیل می‌دهد. شبکه‌ی عصبی پیچشی از پیش آموزش‌دیده (مانند VGG-19 [۱۰]) در طول فرآیند ثابت نگه داشته می‌شود و صرفاً به عنوان نگاشتی غیرخطی برای استخراج ویژگی‌ها به کار می‌رود.

۲.۵. تعریف تابع هدف $\mathcal{L}(\cdot)$. مطابق چارچوب کلی بهینه‌سازی، تابع هدف به صورت یک تابع اسکالر از متغیر تصمیم تعریف می‌شود [۲]:

$$\mathcal{L}(x) = \alpha L_{\text{content}}(p, x) + \beta L_{\text{style}}(a, x),$$

که در آن:

- p تصویر محتوا،
- a تصویر سبک،
- α و β ضرایب موازنه‌ی اهداف هستند.

۱.۲.۵. جزء محتوایی تابع هدف. جزء محتوایی تابع هدف اختلاف میان نگاشت‌های ویژگی تصویر محتوا و تصویر متغیر در یک لایه‌ی مشخص l_c از شبکه را اندازه‌گیری می‌کند:

$$L_{\text{content}}(p, x) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^{l_c}(x) - F_{ij}^{l_c}(p) \right)^2.$$

در این عبارت، $F^l(\cdot)$ خروجی شبکه در لایه‌ی l است.

۲.۲.۵. جزء سبکی تابع هدف. جزء سبکی تابع هدف بر پایه‌ی ماتریس گرام ویژگی‌ها تعریف می‌شود. برای هر لایه‌ی l ، ماتریس گرام به‌صورت

$$G^l(x) = F^l(x) F^l(x)^\top$$

محاسبه شده و هم‌بستگی آماری بین کانال‌های ویژگی را نمایش می‌دهد. زیان سبک در لایه‌ی l برابر است با:

$$L_{\text{style}}(a, x, l) = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l(x) - G_{ij}^l(a))^2,$$

و زیان نهایی سبک با جمع وزن‌دار این مقادیر روی چند لایه به‌دست می‌آید و مسئله‌ی انتقال سبک عصبی به‌صورت زیر فرموله می‌شود:

$$\min_{x \in \mathbb{R}^{H \times W \times 3}} \mathcal{L}(x).$$

از دیدگاه الگوریتم کلی، بردار پارامترها با تصویر x جایگزین شده و تابع هدف همان $\mathcal{L}(x)$ است؛ الگوریتم بهینه‌سازی بدون تغییر اجرا می‌شود.

۳.۵. نقش مشتق‌گیری خودکار در چارچوب الگوریتم. در هر تکرار از الگوریتم:

- (۱) مقدار تابع هدف $\mathcal{L}(x)$ از طریق عبور رو به جلو در شبکه محاسبه می‌شود؛
- (۲) گرادیان تابع هدف $(\nabla_x \mathcal{L})$ با استفاده از مشتق‌گیری خودکار محاسبه می‌گردد؛
- (۳) تصویر x با یک گام گرادیانی به‌روزرسانی می‌شود.

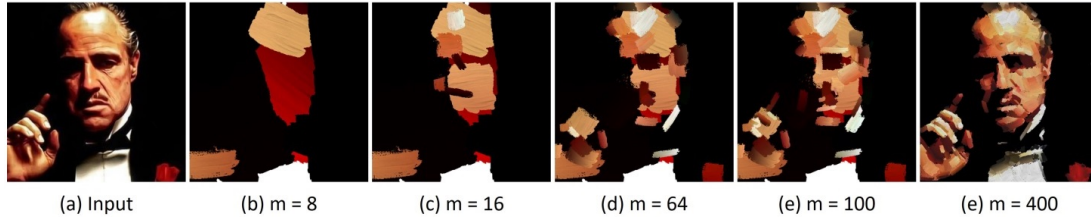
تمام این مراحل دقیقاً با همان الگوریتم کلی گرادیان نزولی انجام می‌شود، با این تفاوت که:

- گراف محاسباتی بسیار عمیق است،
- تعداد متغیرها بسیار زیاد است،
- و محاسبه‌ی دستی گرادیان‌ها عملاً غیرممکن است.

با اجرای این فرآیند، تصویر اولیه (نویز یا تصویر محتوا) به‌تدریج به تصویری تبدیل می‌شود که مقدار تابع هدف $\mathcal{L}$ را کمینه می‌کند. تصویر نهایی، هم ساختار هندسی تصویر محتوا و هم الگوهای آماری تصویر سبک را بازتاب می‌دهد (شکل ۴). این مثال، مشابه مثال قبل است:

- الگوریتم بهینه‌سازی تغییری نکرده است،
- تنها تعریف تابع هدف $\mathcal{L}(\cdot)$ و ماهیت متغیر تصمیم تغییر کرده است،
- و مشتق‌گیری خودکار امکان تعمیم مستقیم این چارچوب به مسائل با میلیون‌ها متغیر را فراهم کرده است.

از این منظر، انتقال سبک عصبی ادامه‌ی طبیعی همان مسیری است که از رگرسیون خطی آغاز شد و با ترسیم گراف گسترش یافت. در بخش بعد کاربرد جذاب دیگری را خواهیم دید.



شکل ۵: مراحل نقاشی یک چهره با روش نقاشی عصبی، بازنشر بخشی از شکل ۷ از [۱۲]. تصاویر (b) تا (e) مرحله‌ای از نقاشی چهره مرجع (a) توسط قلم مو و آبرنگ با روش نقاشی عصبی را نشان می‌دهند.

۶. نقاشی عصبی در چارچوب بهینه‌سازی مبتنی بر مشتق‌گیری خودکار

در این بخش، مسئله‌ی نقاشی با شبکه‌های عصبی که از آن با عنوان نقاشی عصبی^{۱۱} یاد می‌شود را به‌عنوان یک مسئله‌ی بهینه‌سازی پیوسته و مشتق‌پذیر فرموله می‌کنیم. هدف، تولید یک تصویر نهایی از طریق توالی‌هایی از ضربات نقاشی^{۱۲} است؛ به‌گونه‌ای که تصویر حاصل، از نظر بصری با یک تصویر مرجع داده‌شده بیشترین تطابق را داشته باشد [۹، ۱۱، ۱۲]. شکل ۵ یک نمونه را نشان می‌دهد که چگونه این روش، مراحل نقاشی شدن یک چهره روی بوم نقاشی با قلم مو و آبرنگ را می‌تواند به نمایش بگذارد. این مثال نشان می‌دهد که چگونه یک فرآیند تولیدی پیچیده و غیرپیکسلی را می‌توان در قالب یک تابع هدف قابل مشتق‌گیری تعریف کرد و با استفاده از مشتق‌گیری خودکار بهینه نمود.

۱.۶. **فرم ریاضی مسأله.** فرض کنید $I^* \in \mathbb{R}^{H \times W \times 3}$ تصویر مرجع (هدف) باشد. نقاشی نهایی I نه به‌صورت مستقیم در فضای پیکسل‌ها، بلکه از طریق پردازش زدن^{۱۳} (رندرینگ)^{۱۴} مشتق‌پذیر مجموعه‌ای از ضربات تولید می‌شود. این ضربات به‌صورت ترتیبی روی یک بوم خالی اعمال می‌شوند. اگر T تعداد کل ضربات باشد، آنگاه پارامترهای نقاشی را می‌توان به‌صورت زیر نمایش داد:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_T\},$$

که در آن هر θ_t پارامترهای ضربه‌ی t -ام را مشخص می‌کند.

۲.۶. **متغیرهای تصمیم.** هر ضربه‌ی نقاشی با یک بردار پارامتر تعریف می‌شود:

$$\theta_t = (x_t, y_t, \alpha_t, \ell_t, c_t),$$

که در آن:

$$\bullet (x_t, y_t) \in \mathbb{R}^2: \text{موقعیت مرکز ضربه روی بوم؛}$$

$$\bullet \alpha_t \in [0, 2\pi): \text{زاویه‌ی ضربه؛}$$

^{۱۱}Neural Painting

^{۱۲}Strokes

^{۱۳} پردازش زدن (رندرینگ) و یا ارائه ماشینی تصاویر، عبارت است از نمایان‌سازی فرایند تولید ماشینی تصاویر بر پایه مدل‌های محاسباتی و سپس نمایش رایانه‌ای آن‌ها بر روی نمایشگر.

^{۱۴}Rendering

• $\ell_t \in \mathbb{R}^+$: طول یا مقیاس ضربه؛

• $c_t \in \mathbb{R}^3$: رنگ ضربه است.

در نتیجه، متغیر بهینه‌سازی اصلی، کل مجموعه‌ی پارامترهاست:

$$\Theta \in \mathbb{R}^{T \times d},$$

که d تعداد پارامترهای هر ضربه است. تمام مشتق‌گیری‌ها و به‌روزرسانی‌ها نسبت به همین متغیر Θ انجام می‌شود.

۳.۶. رندرینگ مشتق‌پذیر. یک تابع رندرینگ مشتق‌پذیر را در نظر می‌گیریم:

$$I = \mathcal{R}(\Theta),$$

که با دریافت پارامترهای ضربات، تصویر نهایی را تولید می‌کند. ویژگی کلیدی $\mathcal{R}$ این است که نسبت به تمام مؤلفه‌های Θ مشتق‌پذیر است، یعنی $\partial I / \partial \Theta$ وجود دارد. این ویژگی اجازه می‌دهد گرادیان خطا از فضای تصویر نهایی، از طریق فرآیند رندرینگ، به پارامترهای هندسی و رنگی ضربات منتقل شود.

۴.۶. تعریف تابع هدف L . هدف، یافتن پارامترهای Θ به‌گونه‌ای است که تصویر رندر شده I بیشترین شباهت را به تصویر مرجع I^* داشته باشد. تابع هدف کلی به‌صورت ترکیبی از دو جمله تعریف شده است:

$$\mathcal{L}(\Theta) = L_{\text{recon}}(\mathcal{R}(\Theta), I^*) + \lambda L_{\text{reg}}(\Theta).$$

۱. جمله‌ی بازسازی (Reconstruction Loss). این جمله میزان اختلاف بین تصویر تولیدشده و تصویر هدف را اندازه‌گیری می‌کند. یک انتخاب متداول، نرم دوی اختلاف بین تصویر حاصل از مدل و تصویر مرجع است:

$$L_{\text{recon}} = \|\mathcal{R}(\Theta) - I^*\|_2^2.$$

این جمله مستقیماً کیفیت بصری نقاشی نهایی را هدف قرار می‌دهد و گرادیان آن نسبت به پارامترهای ضربات از طریق مشتق‌گیری زنجیره‌ای محاسبه می‌شود:

$$\frac{\partial L_{\text{recon}}}{\partial \Theta} = \frac{\partial L_{\text{recon}}}{\partial I} \cdot \frac{\partial I}{\partial \Theta}.$$

۲. جمله‌ی منظم‌سازی (Regularization). برای جلوگیری از تولید ضربات غیرطبیعی یا ناپایدار، از جملات منظم‌سازی استفاده می‌شود. به‌عنوان مثال:

$$L_{\text{reg}}(\Theta) = \sum_{t=1}^T (\|\ell_t\|^2 + \|\nabla c_t\|^2),$$

که در آن:

- جمله‌ی اول از بزرگ شدن بیش از حد ضربات جلوگیری می‌کند؛
- جمله‌ی دوم تغییرات شدید رنگ را جریمه می‌کند.

این جملات باعث می‌شوند فضای جست‌وجوی بهینه‌سازی هموارتر شده و گرادیان‌ها رفتار پایدارتری داشته باشند.

۵.۶. دیدگاه بهینه‌سازی و مشتق‌گیری. مسئله‌ی نهایی به صورت زیر خلاصه می‌شود:

$$\min_{\Theta} \mathcal{L}(\Theta).$$

بهینه‌سازی با استفاده از الگوریتم‌های مبتنی بر گرادین (مانند SGD یا Adam) انجام می‌شود:

$$\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla_{\Theta} L(\Theta^{(k)}).$$

نکته‌ی کلیدی این است که:

- مشتق‌گیری نه نسبت به پیکسل‌ها،
- بلکه مستقیماً نسبت به پارامترهای هندسی و رنگی ضربات انجام می‌شود؛
- و کل فرآیند رندرینگ، بخشی از گراف محاسباتی قابل مشتق‌گیری است.

به این ترتیب، نقاشی عصبی نمونه‌ای گویا از کاربرد مشتق‌گیری خودکار در یک مسئله‌ی تولیدی غیرمتعارف است، که در آن متغیرهای تصمیم، ساختاری هندسی و معنایی دارند، نه صرفاً مقادیر عددی در یک شبکه‌ی کلاسیک. از این منظر، نقاشی عصبی نقطه‌ی پایانی مسیر ارائه‌شده در این مقاله است: جایی که یک فرآیند تولیدی پیچیده، با متغیرهایی دارای معنا و ساختار هندسی، در همان چارچوب ساده‌ی بهینه‌سازی مبتنی بر مشتق‌گیری خودکار و بدون هیچ تغییری در الگوریتم اصلی حل می‌شود.

۷. جمع‌بندی و نتیجه‌گیری

در این مقاله تلاش شد تا مشتق‌گیری خودکار نه صرفاً به عنوان یک ابزار پیاده‌سازی در یادگیری ماشین، بلکه به عنوان یک چارچوب مفهومی برای حل مسائل بهینه‌سازی پیوسته معرفی شود. ایده‌ی محوری مقاله آن است که با تعریف مناسب یک تابع هدف مشتق‌پذیر، می‌توان مسائل بسیار متنوعی را با استفاده از یک الگوریتم بهینه‌سازی مبتنی بر گرادین حل کرد. چهار مثال ارائه‌شده نشان دادند که این چارچوب از مسائل کلاسیک و کم‌بعد مانند رگرسیون خطی تا مسائل پیچیده و تولیدی مانند انتقال سبک و نقاشی عصبی قابل تعمیم است. در تمامی این موارد، تفاوت اصلی در ماهیت متغیرهای تصمیم و تابع هدف است، نه در الگوریتم بهینه‌سازی. این مشاهده، دیدگاهی یکپارچه نسبت به مسائل بهینه‌سازی فراهم می‌کند و مرز میان مدل‌های «ساده» و «پیچیده» را کمرنگ می‌سازد. از دیدگاه آموزشی، این چارچوب نشان می‌دهد مفاهیم کلاسیکی مانند گرادین، تابع هدف و بهینه‌سازی چگونه مستقیماً در کاربردهای مدرن و خلاقانه ظاهر می‌شوند. گرچه که بیان کامل دو مثال آخر فراتر از این نوشتار است، اما امید است این مقاله گامی کوچک در جهت پیوند عمیق‌تر میان آموزش ریاضیات، بهینه‌سازی عددی، و روش‌های نوین یادگیری ماشین باشد.

مراجع

- [۱] محمود امین طوسی، کاربرد بسط تیلور در کاهش حجم شبکه‌های عصبی پیچشی برای طبقه‌بندی نقاشی‌های سبک امپرسیونیسم و مینیاتور، نشریه ریاضی و جامعه ۵ (۱۳۹۹)، شماره ۱، صص. ۱-۱۶.
- [۲] محمود امین طوسی، انتقال سبک برای افزایش داده‌های آموزشی شبکه‌های کانولوشنی در شناسایی شعله آتش، هوش محاسباتی در مهندسی برق ۱۳ (۱۴۰۱)، شماره ۴، صص. ۹۷-۱۱۴.
- [۳] محمود امین طوسی، تمام متصل به تمام پیچشی: پلی به گذشته، رایانش نرم و فناوری اطلاعات ۱۱ (۱۴۰۱)، شماره ۱، صص. ۶۰-۷۲.

[4] A. Leon, A. Gatys, S. Ecker, and M. Bethge, *Image style transfer using convolutional neural networks*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 2414-2423.

- [5] Xun Huang and Serge Belongie, *Arbitrary style transfer in real-time with adaptive instance normalization*, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, 1510–1519.
- [6] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song, *Neural Style Transfer: A Review*, IEEE Transactions on Visualization & Computer Graphics **26** (2020), no. 11, 3365–3385.
- [7] Yujun Jing, Yezhou Yang, Zunlei Feng, Jingdong Ye, Yingyu Yu, and Mingli Song, *Neural style transfer: A review*, IEEE Transactions on Visualization and Computer Graphics **26** (2020), no. 11, 3365–3385.
- [8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), Curran Associates, Inc., 2012, 1097–1105.
- [9] Reiichiro Nakano, *Neural painters: A learned differentiable constraint for generating brushstroke paintings*, arXiv preprint arXiv:1904.08410 (2019), Available at <https://arxiv.org/abs/1904.08410>.
- [10] Karen Simonyan and Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*, International Conference on Learning Representations, 2015.
- [11] Ning Xie, Hirotaka Hachiya, and Masashi Sugiyama, *Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting*, IEICE Transactions on Information and Systems **96** (2013), no. 5, 1134–1144.
- [12] Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi, *Stylized Neural Painting*, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Los Alamitos, CA, USA), IEEE Computer Society, June 2021, 15684–15693.



AUTOMATIC DIFFERENTIATION AS A UNIFYING FRAMEWORK FOR A BROAD CLASS OF OPTIMIZATION PROBLEMS

MAHMOOD AMINTOOSI

Computer Science Dept. Faculty of Mathematical Sciences, Ferdowsi University of Mashhad
m.amintoosi@um.ac.ir

Abstract.

Automatic differentiation is a fundamental tool in gradient-based methods for machine learning, enabling efficient and accurate derivative computation for complex functions. This paper revisits classical gradient descent within automatic differentiation, illustrating it through increasingly intricate examples. Optimization problems are formulated as computable objective functions, even when lacking a closed-form analytical expression in classical mathematics. Gradients are automatically computed, and gradient descent is applied without conceptual modification. A central insight is that heterogeneous problems can be unified under a single computational framework via appropriate objective formulation. Four examples of increasing complexity are presented: linear regression as a familiar baseline, then cases where classical mathematics lacks explicit formulations yet become tractable through automatic differentiation. The goal is an accessible pedagogical exposition showing how automatic differentiation bridges classical numerical optimization and modern machine learning, making these concepts accessible for students in mathematics and computer science.

2020 Mathematics Subject Classification. 68T07, 68U10, 68U10, 68T45

Keywords. Automatic differentiation, Gradient descent, Optimization in machine learning, Objective function, Numerical optimization

Date: Received 23-01-2026 Revised 12-02-2026 Accepted 21-06-2026 Available Online 21-06-2026

©Ferdowsi University of Mashhad.