



تشخیص کاراکترهای فارسی تایپ شده با الگوریتم گاز عصبی رشد یابنده (GNG)

کیوان برنا^۱، زهرا نیلفروشان^۲، پدram پاشائیان^۳ و فاطمه رضازاده بیدگلی^۴

Persian Typed Character Recognition with the Growing Neural Gas Algorithm (GNG)

Keivan Borna, Zahra Nilforoushan, Pedram Pashaeian, Fatemeh RezaZadeh Bidgoli

Email: borna@khu.ac.ir

چکیده

در این مقاله با استفاده از الگوریتم گاز عصبی رشد یابنده (GNG)، گراف مرتبط با یک کاراکتر فارسی تایپ شده را از روی تصویر آن کاراکتر استخراج می‌کنیم. این گراف به عنوان یک ویژگی که ریخت کاراکتر را کاملاً یاد می‌گیرد، برای تشخیص کاراکتر به کار می‌رود. GNG یکی از انواع الگوریتم‌های خودسامان‌ده است و نوعی یادگیری رقابتی به حساب می‌آید. مهم‌ترین مزیت این روش علاوه بر سرعت و دقت قابل قبول، حل چالش‌های مهمی مانند چرخش، نویز، تغییر شکل است. این الگوریتم تصویر کاراکتر را به عنوان ورودی می‌گیرد و یک گراف که ریخت کاراکتر در آن حفظ شده است را به عنوان خروجی برمی‌گرداند.

کلمات کلیدی

پردازش تصویر، تشخیص کاراکتر، گاز عصبی رشد یابنده، نقشه‌های خودسامان‌ده، هندسه محاسباتی.

۱. مقدمه

یکی از مهم‌ترین وظایف پردازش تصویر کمک به دیجیتال کردن متون چاپی موجود است. این کار نیاز به تشخیص کاراکتر با روش‌های بینایی ماشین دارد. نقطه مشترک تمامی روش‌هایی که تا کنون به کار رفته‌اند، استخراج ویژگی‌هایی از کاراکتر مورد نظر و سپس کلاس‌بندی به کمک این ویژگی‌هاست. تشخیص کاراکتر در زبان‌های مختلف چالش‌های متفاوتی دارد. بعضی چالش‌ها از جمله نویز، چرخش، تغییر شکل و ... در همه زبان‌ها وجود دارند و برخی دیگر از جمله نقطه، پیوسته‌نویسی، شکل‌های مختلف از یک حرف و ... فقط در زبان‌هایی مانند فارسی وجود دارند. در کنار روش‌های پردازش تصویر که برای این کار وجود دارند، هندسه محاسباتی نیز پای خود را به این زمینه

^۱ عضو هیأت علمی دانشکده علوم ریاضی و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.

^۲ عضو هیأت علمی دانشکده فنی و مهندسی، دانشگاه خوارزمی، تهران، ایران.

^۳ دانشجوی کارشناسی ارشد دانشکده فنی و مهندسی، دانشگاه خوارزمی، تهران، ایران.

^۴ دانش آموخته کارشناسی ارشد دانشکده فنی و مهندسی، دانشگاه خوارزمی، تهران، ایران.

* نویسنده مسؤل



باز کرده است. یکی از این روش‌ها استخراج گراف از تصویر کاراکتر و سپس مقایسه ریخت‌شناسانه گراف‌ها با هم به منظور دسته‌بندی و تشخیص کاراکترهاست. روش‌های یادگیری رقابتی و نقشه‌های خودسازمان‌ده انواعی از شبکه‌های عصبی هستند که با تکرارهای زیاد، در نهایت شبکه‌ای از گره‌ها و یال‌ها را نتیجه می‌دهند. این شبکه همان گرافی است که ریخت تصویر مورد نظر (که در اینجا تصویر یک کاراکتر است) را در خود حفظ کرده است. هدف این مقاله بکارگیری روش GNG برای تشخیص کاراکترهای فارسی تایپ شده است.

۲. نتایج اصلی: الگوریتم گاز عصبی رشد یابنده (GNG)

این الگوریتم برای اولین بار در سال ۱۹۹۵ توسط Bernd Fritzke ارائه شد. الگوریتم GNG یکی از انواع SOM [۱] است که ریخت را استخراج می‌کند یا داده‌ها را کلاس‌بندی می‌کند. این الگوریتم برای بهبود روشی که از ترکیب گاز عصبی (Neural Gas) و یادگیری هب رقابتی (CHL) استفاده می‌کرد به وجود آمد و آن روش را بهبود داد و منعطف‌تر کرد [۲].

در این قسمت، مراحل مختلف الگوریتم GNG بر روی تصویر کاراکتر را شرح می‌دهیم:

۱. الگوریتم با دو گره اولیه S_1 و S_2 آغاز می‌شود. این گره‌ها به صورت تصادفی انتخاب می‌شوند و در ابتدا الزامی وجود ندارد که در داخل فضای داده‌های مورد نظر (در اینجا داده‌های مورد نظر پیکسل‌های کاراکتر هستند) قرار داشته باشند. برای هر گره متغیری به نام خطا (Error) برابر با صفر مقاردهی اولیه می‌شود.

مجموعه یال‌ها نیز شامل یال بین این دو گره است و سن این یال برابر با صفر مقاردهی می‌شود. این دو گره و یال مورد اشاره، شبکه‌ای را شکل می‌دهند که باید در پایان به داده مورد نظر همگرا شود.

۲. یکی از پیکسل‌های داده مورد نظر، به صورت تصادفی، به عنوان ورودی (Input) به شبکه داده می‌شود.

۳. بر اساس مختصات پیکسل ورودی جدید، یک گره از شبکه گاز عصبی به عنوان گره برنده انتخاب می‌شود. این گره که با اندیس S_1 نام‌گذاری می‌شود نزدیک‌ترین گره به پیکسل ورودی است. اندیس‌گذاری گره برنده به این دلیل است که به همین ترتیب، بر اساس کاربردهای الگوریتم، می‌توان تعداد گره‌های برنده را بیش از یک هم در نظر گرفت. در اینجا از فاصله اقلیدسی به عنوان معیاری از نزدیکی استفاده شده است:

$$s_1 = \arg \min_i \|w_i - x\| \quad (1)$$

که در آن x مختصات پیکسل ورودی و w_i مختصات گره i ام از شبکه گاز عصبی هستند.

۴. سن همه یال‌های متصل به S_1 افزایش می‌یابد.

۵. خطای گره S_1 به اندازه مربع فاصله بین پیکسل ورودی و S_1 افزایش می‌یابد.

$$\Delta Error_{s_1} = \|w_{s_1} - x\|^2 \quad (2)$$

۶. گره S_1 با ضریبی به نام شدت یادگیری، به سمت پیکسل ورودی میل می‌کند.

$$w_{s_1}(t+1) = w_{s_1}(t) + \lambda(t)(x - w_{s_1}) \quad (3)$$

در اینجا t و $t+1$ شماره تکرار الگوریتم را نشان می‌دهند. شدت یادگیری که در اینجا با λ و به صورت متغیری وابسته به تعداد تکرار نشان داده شده، می‌تواند ثابت باشد یا با توجه به تعداد تکرارها کاهش یابد.

۷. همه همسایه‌های مستقیم S_1 نیز به سمت پیکسل ورودی میل می‌کنند.

$$w_n(t+1) = w_n(t) + \lambda(t)(x - w_n) \quad (4)$$

رابطه شماره (۴) برای همه همسایه‌های مستقیم S_1 اجرا می‌شود.



۸. یک یال با سن صفر بین S_1 و S_2 (دومین گره نزدیک به پیکسل ورودی) اضافه می‌شود. اگر این دو گره از قبل متصل باشند سن یال بین آن دو صفر می‌شود.

۹. یال‌هایی که سن آن‌ها بیشتر از حداکثر سن مجاز (Max Age) باشد، حذف می‌شوند. این حداکثر سن مجاز بر اساس کاربرد تعیین می‌شود.

۱۰. اگر گره‌هایی تنها باقی مانده باشند که یال ندارند (Isolate Nodes)، آن گره‌ها نیز حذف می‌شوند.

۱۱. بعد از هر تعداد مشخصی از اجرای مراحل ۱ تا ۹، یک گره به شبکه اضافه می‌شود. تعداد گره‌های شبکه نیز محدودیت دارد (Max Nodes). اضافه کردن گره به این صورت انجام می‌شود:

۱-۱۱. گره با بیشترین مقدار خطا در شبکه را تعیین می‌کنیم و نام آن را q می‌گذاریم.

۲-۱۱. اگر مقدار خطای q از حداکثر خطا (Max Error) که در ابتدای الگوریتم تعیین می‌شود، بیشتر

باشد، گره جدید r را بین q و دورترین همسایه مستقیم آن یعنی f اضافه می‌کنیم:

$$w_r = \frac{w_q + w_f}{2} \quad (1)$$

۱۱-۳. مقادیر حداکثر تعداد گره و حداکثر خطا، بر ایجاد گره‌ها و یال‌های مستقل و همچنین دور در گراف

تاثیر می‌گذارند و انتخاب آن‌ها با توجه به کاربرد مورد نظر، در به جواب رسیدن الگوریتم تاثیرگذار است.

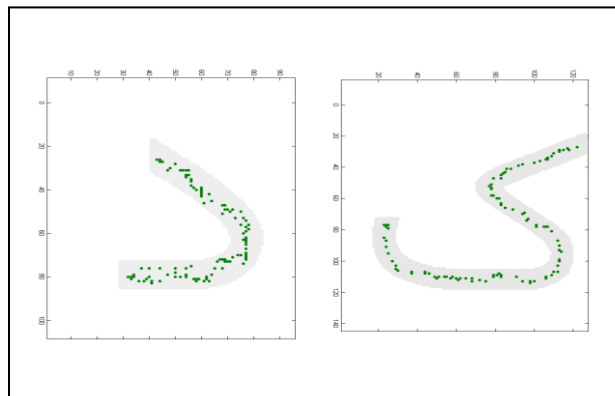
۱۲. بعد از هر تعداد مشخصی از پیکسل‌های ورودی، مقدار خطای همه گره‌ها به صفر تغییر می‌کند.

۱۳. اگر هیچکدام از شرط‌های پایان الگوریتم برقرار نبود به شماره ۲ برمی‌گردد. شرط‌های پایان الگوریتم می‌توانند عبارت باشند از: اجرای الگوریتم با تعداد تکرار مشخص، رسیدن به خطای قابل قبول، همگرا نشدن برنامه با توجه به روند تغییر خطا [۳].

۳. نتایج اجرا و شبیه سازی الگوریتم

با توجه به اینکه خروجی الگوریتم گاز عصبی رشد یابنده، مجموعه‌ای از نقاط شبکه هستند که ساختار اصلی تصویر ورودی را به ما می‌دهد، می‌توان گفت که خروجی الگوریتم GNG، از نوع ویژگی (feature) است.

در ادامه، در شکل ۱، به عنوان نمونه، خروجی الگوریتم بر روی حروف فارسی «ک» و «د» آمده است:

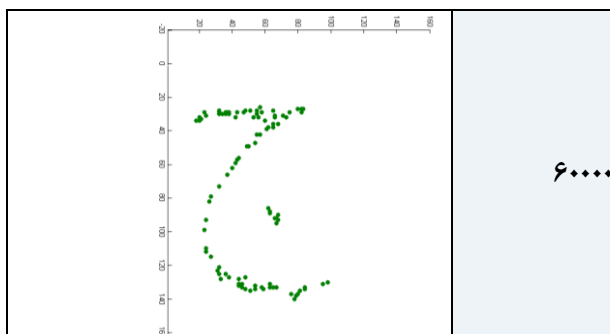


شکل ۱: خروجی حاصل از الگوریتم GNG روی حروف «د» و «ک»

مراحل مختلف اجرای الگوریتم روی حرف «ج» در شکل ۲ و گراف خروجی حاصل از اجرای GNG در شکل ۳ به تصویر آمده است:

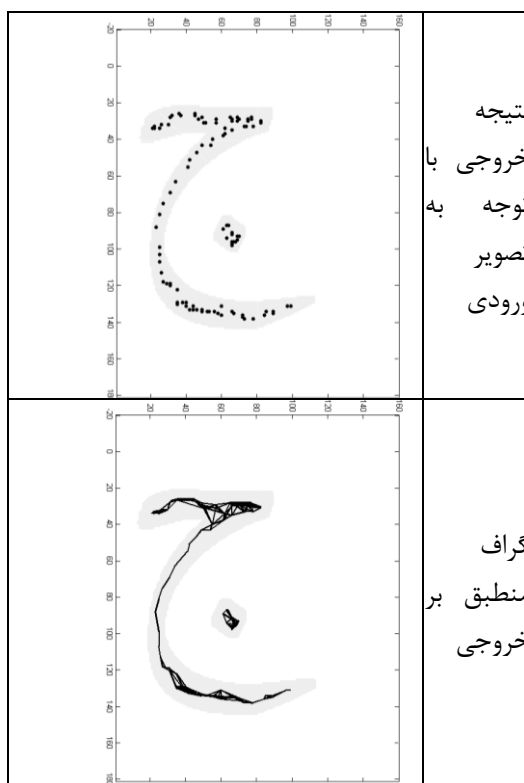


نتیجه	تعداد اجرا
	۱۰۰۰
	۱۰۰۰۰
	۲۰۰۰۰
	۳۰۰۰۰



شکل ۲: نتایج خروجی حاصل از اجرای الگوریتم GNG روی تصویر حرف «ج» در تعداد اجراهای مختلف

عوامل مهمی که در به نتیجه رسیدن الگوریتم بسیار تاثیرگذار هستند مانند تعداد تکرارها، حداکثر سن یال‌ها و حداکثر تعداد گره‌ها، با سعی و خطا به دست می‌آیند و برای هر کاراکتر متفاوت هستند.



شکل ۳: خروجی و گراف، منطبق بر تصویر ورودی

یکی دیگر از نکات مهم، ذات پرتکرار این الگوریتم است که لزوم اجرای آن بر روی تعداد زیادی از پیکسل‌ها را سبب می‌شود. به عبارت دیگر، این الگوریتم بر روی تعداد داده‌های کم و با تکرار کم جواب درستی نمی‌دهد.



۴. ارزیابی کارایی الگوریتم پیشنهادی

با توجه به اینکه خروجی روش GNG روی حروف فارسی، مجموعه‌ای از نقاط ساختار اصلی حروف می‌باشد، این روش را با سایر روش‌های مشابه، مقایسه نموده‌ایم. در واقع با روش‌هایی که خروجی حاصل از اجرای آن‌ها روی تصاویر، مجموعه‌ای از نقاط یا ویژگی‌های ساختاری تصویر می‌باشند. از جمله این روش‌ها، اجرای چند موجک (Wavelet) مختلف روی تصویر، استخراج اسکلت (Skeletonization) تصویر و نازک‌سازی (Thinning) و مقایسه ابعاد خروجی آن‌ها با روش پیشنهادی است. در جدول ۱، نتایج ابعاد خروجی روی سه کاراکتر "ج"، "ک" و "د" با بکارگیری ۷ روش ارائه شده است. همان طور که مشاهده می‌شود، الگوریتم پیشنهادی مقاله، GNG، به نسبت سایر روش‌ها، تعداد نقاط کمتری برای عرضه خروجی بکار گرفته است.

جدول ۱: مقایسه نتایج ابعاد خروجی بر حسب پیکسل

الگوریتم	ج	ک	د
Biorthogonal	206	177	95
Coiflets	1332	1101	616
Daubechies	272	247	125
Haar	92	88	51
Skeleton	239	296	115
Thin	240	221	96
GNG	54	36	11

در جدول ۲ نسبت ابعاد تصویر اولیه به ابعاد خروجی در ۷ الگوریتم عرضه شده است که مجدداً برتری روش پیشنهادی این مقاله تأیید می‌گردد.

جدول ۲: مقایسه نسبت ابعاد تصویر اولیه به ابعاد خروجی

الگوریتم	ج	ک	د
Biorthogonal	0.0107	0.0095	0.0099
Coiflets	0.0694	0.0589	0.0642
Daubechies	0.0142	0.0132	0.0130
Haar	0.0048	0.0047	0.0053
Skeleton	0.0124	0.0158	0.0120
Thin	0.0125	0.0118	0.0100
GNG	0.0028	0.0019	0.0011

۵. نتیجه گیری و کارهای آینده

این مقاله گامی در جهت بکارگیری الگوریتم GNG برای تشخیص کاراکترهای فارسی تایپ شده است. این روش در کنار حل چند مسئله مهم در زمینه تشخیص کاراکتر، مشکلاتی هم دارد. از قبیل مشکل در شناسایی حروفی که از نظر نقطه تفاوت دارند، مانند «ج» و «چ». به هر حال خروجی GNG که ریخت کاراکتر را بدون حساسیت به چرخش یا تغییر شکل یاد می‌گیرد و در خود حفظ می‌کند می‌تواند کاربردهای زیادی چه در حوزه تشخیص کاراکتر یا سایر موضوعات پردازش تصویر داشته باشد.



پس از حذف دورها و حلقه‌های اضافی از گراف به کمک الگوریتم‌هایی مانند گراف همسایگی نسبی [۴] و استخراج ریخت از روی آن، می‌توان با آزمایش، کلاس‌بند مناسب برای این خروجی را یافت و به نتیجه نهایی یعنی تشخیص کاراکتر به هر صورت (تایپی یا دست‌نویس) رسید.

۶. منابع و مراجع

- [1] T.Kohonen, "The self-Organizing map" Proceeding of the IEEE, 78, 1990, pp. 1464-1480.
- [2] B.Fritzke, "A Growing Neural Gas network learns topologies", Advances in Neural Information Processing Systems, Vol. 7, No. 1, 1995, pp. 625-632.
- [3] Kazuhisa Fujita, "Extract an Essential Skeleton of a Character as a Graph from a Character Image" IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 5, No. 1, 2013, pp. 35-39.
- [4] Godfried T. Toussaint, "The Relative Neighborhood Graph of a finite planar set" Pattern Recognition, Vol. 12, 1980, pp. 261-268.