

کنترل فازی پاداش یادگیری تقویتی برای تشخیص ارقام دست‌نویس

صابر ملک زاده¹

1- گروه علوم کامپیوتر، دانشگاه ولیعصر (عج) رفسنجان، رفسنجان، ایران

خلاصه

تشخیص محیط‌های انسانی توسط سامانه‌های رایانه‌ای، همواره یکی از مهم‌ترین موضوعات در دانش هوش مصنوعی بوده است. در این میان تشخیص دست‌خط انسان و مفهوم‌سازی آن برای رایانه به عنوان یکی از مباحث در این عرصه مطرح شده است. در سال‌های اخیر با گسترش روش‌های یادگیری ماشین در هوش مصنوعی، تلاش برای استفاده از این روش برای حل مسائل مختلف مطرح در هوش مصنوعی افزایش چشمگیری داشته است. در این مقاله سعی شده با استفاده از کنترل‌کننده‌ی فازی، میزان پاداش یادگیری تقویتی، جهت تشخیص ارقام دست‌نویس بهینه شود. برای این منظور ابتدا یک نمونه از هر رقم با 10 فونت رایانه‌ای معیار، به عامل داده‌شده و سپس به‌صورت مرحله به مرحله به آموزش عامل پرداخته شده است. در مرحله‌ی بعد به آزمایش نمونه‌ها به‌وسیله داده‌های آموزشی پرداخته شده و نتایج به‌دست‌آمده حاکی از تأثیر مستقیم بهبود عملکرد تشخیص، در زمان کنترل فازی میزان پاداش یادگیری تقویتی بوده است.

واژه‌های کلیدی: یادگیری تقویتی، کنترل‌کننده‌ی فازی، تشخیص رقم، دست‌نویس

1. مقدمه

یادگیری تقویتی^{*} [1] در زمره‌ی روش‌های یادگیری ماشین است که با روش پاداش‌دهی[†] و تقویت عامل با آموزش خوب و بد به آن، عامل را به سمت هدف خوب رهنمون می‌شود. پاداش در هر مرحله به ازای انتخاب درست توسط عامل، می‌تواند به‌صورت دودویی[‡] یا به‌صورت فازی[§] باشد. در سامانه‌ی دودویی به ازای هر عمل مثبت، عامل یک امتیاز مثبت دریافت می‌کند، اما در سامانه‌ی فازی، کیفیت عمل خوب نیز مهم است و عددی بین 0 و 1 به‌عنوان پاداش اختصاص می‌یابد.

در اوایل قرن 20 میلادی ثورندایک^{**} [1]، برآن شد تا با استفاده از آزمایشی، اثر یادگیری تقویتی بر روی مغز انسان را اثبات کند. در این آزمایش گربه‌ای در قفسی کارتونی قرار داده شد. هر بار که این گربه برای بیرون رفتن از قفس تلاش‌های بهتری انجام می‌داد، غذا دریافت می‌کرد. بدین ترتیب گربه (عامل^{††}) تلاش‌های خود را به سمت بهترشدن متمرکز کرد تا غذا دریافت کند و پس از چند تلاش که در اواخر، همگی تلاش‌های خوب بودند، توانست از قفس نجات

* Reinforcement Learning

† Reward giving

‡ Binary

§ Fuzzy

** Edward Thorndike

†† Actor

پیدا کند. این همان مکانیسمی* بود که در مغز انسان نیز وجود داشت. در اوایل دهه‌ی 60 میلادی دانشمندان علوم کامپیوتر سعی نمودند تا این مکانیسم را وارد هوش مصنوعی سازند و با پاداش و جزا، عامل هوش مصنوعی را تقویت کنند تا نتایج را بهینه کرده و سریعتر به دست آورند.

در سال 1965 لطفی‌زاده [1,2] مجموعه‌های فازی را معرفی نمود. با ورود منطق فازی به علوم کامپیوتر دانشمندان سعی کردند تا از منطق فازی و در نتیجه‌ی آن کنترل‌کننده‌های فازی در کنترل نتایج و همچنین به دست آوردن نتایج از سامانه‌ی دودویی جهت دقت بیشتر استفاده کنند.

در این مقاله به بررسی کنترل فازی پاداش‌های یادگیری تقویتی، زمانی که عامل به سمت انتخاب‌های درست‌تر پیش می‌رود، پرداخته شده است. در الگوریتم یادگیری تقویتی نیز برای تعیین تأثیر نمونه‌های آموزشی، از منطق فازی استفاده شده است.

2. کارهای مشابه

در گذشته نیز فعالیت‌هایی در این زمینه بوسیله‌ی شبکه‌های عصبی مصنوعی انجام شده است. در این پژوهش‌ها شبکه‌های عصبی که به طور مناسب تنظیم شده بودند توانستند، با درصد بسیار بالایی تشخیص دهند. همچنین از یادگیری تقویتی برای بهینه نمودن تشخیص استفاده شده است که صرفاً محدود به پاداش‌هایی با مقادیر ثابت بوده و یا کلاسیک[†] (غیرفازی) بوده است.

3. نمونه‌های آموزشی

جهت آموزش دادن[‡] عامل برای یادگیری بهتر و مناسب‌تر نیاز به یک نمونه‌ی آموزشی مناسب بود که تصاویر آن، هم از لحاظ وضوح مناسب باشند، هم از لحاظ کیفیت نمونه‌ها و شباهت‌شان به نمونه‌های واقعی در سطح قابل قبولی باشند و هم تعداد پیکسل‌های کمتری داشته باشند تا عملیات پردازش بر روی ساختار آن‌ها به سرعت انجام شود. برای این منظور نمونه‌هایی از ارقام پایگاه داده‌ی MNIST [3] استخراج شدند. این نمونه‌ها، تصاویری با پس‌زمینه‌ی سیاه و نوشته‌های سفید به صورت دست‌نویس از ارقام بودند. این پایگاه داده به عنوان استاندارد از نمونه‌های آموزشی و آزمایشی از دست‌خط انگلیسی شناخته می‌شود.

4. یادگیری

جهت استفاده از یادگیری تقویتی نیاز بود که یادگیری عامل در مراحل مختلف و گام‌به‌گام صورت گیرد. به همین منظور برای هر یک از ارقام، چند نمونه فونت‌های[§] معیار رایانه‌ای در سایز ۲۸ در ۲۸ پیکسل** (به اندازه نمونه‌های آموزشی) به همان شکل نمونه‌های آموزشی (با پس‌زمینه سیاه و نوشته سفید) ساخته شد. سپس ۸۰ درصد (به صورت تصادفی) نمونه‌های آموزشی که شامل ۸۰۰ نمونه برای هر رقم بودند، با نمونه فونت‌های معیار رایانه‌ای، به وسیله‌ی تابع^{††} SSim [1,4] در نرم‌افزار متلب مقایسه شدند.

* Mechanisms

† Classic

‡ Train

§ Font

** Pixel

†† Structural Similarity

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

در این فرمول X و Y تصاویر ورودی به تابع جهت بررسی شباهت، μ میانگین مقادیر پیکسل های تصاویر و σ انحراف معیار مقادیر پیکسل های تصاویر است. همچنین ضریب σ_{xy} نیز انحراف معیار پیکسل های تصاویر X و Y نسبت به یکدیگر است. این معیار مشخص می کند که پیکسل در حال بررسی در تصویر X چقدر متفاوت تر از همان پیکسل در تصویر Y است. مقدار C نیز، ضرایبی جهت ایجاد ثبات در بخش های مختلف فرمول هستند که به ایجاد ثبات در فرمول، به هنگام ظهور مخرج ضعیف کمک می کنند. این مقادیر به محدوده ی مقادیر پیکسل ها بستگی دارند.

تابع $SSim$ تشابه ساختاری میان دو تصویر را به وسیله ی فرمول بالا سنجیده و نتیجه ی تشابه را در قالب عددی بین ۱- و ۱ ارائه می دهد که به هر میزان این عدد بیشتر باشد، تصاویر داده شده به هم شبیه ترند. این تابع همان طور که در فرمول بالا نیز مشاهده می شود، از مقایسه ی میان میانگین مقادیر پیکسل های دو تصویر با یکدیگر، جهت تشخیص میزان شباهت، استفاده می کند. همچنین در نمونه های تازه منتشر شده برای این تابع، تفاوت میان دو تصویر با قطعه بندی دو تصویر و شباهت سنجی قطعه های آن ها با یکدیگر نیز سنجیده می شود. این تابع در سال ۲۰۱۵ برای اولین بار وارد ساختار نرم افزار متلب* شد.

در نتیجه ی مقایسه ها، مشاهده شد که تابع $SSim$ برای ارقام مختلف ماکزیمم های متفاوتی ارائه می دهد. برای مثال تعداد نمونه های رقم ۲ که درصد تشابه شان با نمونه فونت های معیار رایانه ای بیشتر از ۴۰ درصد بود، ۱۰ نمونه بودند، در حالی که همین مثال در رقم ۷ تنها ۱ نمونه ی نتیجه داشت که نشان از نامتوازن بودن پایگاه داده ی آموزشی نسبت به تابع $SSim$ بود. بنابراین در آموزش اولیه، ابتدا نمونه هایی که درصد تشابه آن ها با نمونه فونت های رایانه ای بیشتر از ۴۰ درصد بودند، انتخاب شدند. سپس همین عملیات با نمونه های آموزشی سابق (بدون نمونه های انتخاب شده به عنوان نمونه ی خوب در مرحله قبل) انجام شد. در این مرحله نمونه های آموزشی با نمونه های فونت رایانه ای و دست نویس هایی که به عنوان نمونه انتخاب شده بودند، مقایسه شدند و از تشابه نمونه ی آموزشی با نمونه های خوب میانگین گرفته شد و باز نمونه هایی که درصد تشابه آن ها با تمامی نمونه های خوب (نمونه ی فونت های رایانه ای و نمونه های خوب مراحل قبل) بیشتر از ۴۰ درصد بود انتخاب شدند. این تکرار تا ۵ مرحله ادامه یافت. زیرا کمتر از آن باعث انتخاب کم نمونه های خوب و عدم وجود اطلاعات کافی برای تشخیص مناسب و بیشتر از آن علاوه بر فزونی نمونه های خوب و کم شدن سرعت نرم افزار در هنگام آزمایش، باعث افزایش نمونه های بد در میان نمونه های خوب ارقام دارای نمونه های خوب بیشتر و در نتیجه باعث تشخیص نامناسب می شد.

در تمامی مراحل میزان شباهت نمونه ی انتخاب شده (تنها) با نمونه فونت های رایانه ای (حتی آن نمونه هایی که در مرحله پنجم انتخاب شدند برای دریافت میزان شباهت، جهت ذخیره، تنها با نمونه فونت های رایانه ای مقایسه می شدند.) به ترتیب شماره ی ذخیره ی آن در میان نمونه های خوب، در فایل هایی که برای هر رقم مجزا بودند، ذخیره شدند تا در مرحله ی آزمایش میزان شباهت آن ها به فونت های رایانه ای در میزان تأثیر گذاری آن ها در پاداش یادگیری تقویتی، مؤثر باشد.

5. آزمایش

جهت انجام مرحله ی آزمایش نیاز بود، برای ۲۰ درصد باقی مانده ی نمونه ها، عملیات شباهت سنجی $SSim$ با نمونه های خوب انجام پذیرد. در این میان، میزان شباهت هر نمونه ی خوب، به نمونه های فونت استاندارد رایانه ای

* Matlab

مورد بررسی قرار گرفت تا میزان استاندارد بودن آن نسبت به فونت های معیار رایانه ای در نتیجه ی نهایی مؤثر باشد و هر چقدر نمونه ی دست نوشته ی خوب به فونت های رایانه ای نزدیک تر باشد، اثر آن بر نتیجه ی تشخیص نیز بیشتر باشد. میانگین شباهت نمونه های خوب به فونت های معیار رایانه ای متفاوت بود و این بر نتیجه ی تشخیص تأثیر می گذاشت، به گونه ای که هر میزان این میانگین برای رقمی بیشتر بود، تمامی نمونه های آزمایشی، تمایل بیشتری را به انتخاب آن رقم به عنوان رقم نتیجه داشتند. کمترین میزان شباهت نمونه های خوب به فونت های معیار در هر رقم، از بیشترین میزان آن کم شد و میزان نزدیکی هر عدد به بزرگترین یا کوچکترین مقدار به عنوان عددی بین ۰ و ۱ انتخاب شد. برای مثال اگر بیشترین میزان میانگین شباهت فونت خوب به فونت های معیار ۰.۳۶ و میزان کمترین شباهت ۰.۳۲ بود، عدد اختصاص یافته به نمونه ی دارای میزان تشابه ۰.۳۲ که کوچکترین مقدار بازه بود ۰.۲۵، برای ۰.۳۶ که بیشترین مقدار بود، ۰.۷۵ و برای ۰.۳۴ که در میانه ی بازه بود ۰.۵ اختصاص یافت. در صورتی که کمترین مقدار را برابر صفر و بیشترین مقدار برابر ۱ قرار می گرفت، نمونه ی خوب دارای میزان تشابه ۰.۳۲ بی تأثیر می شد و همچنین نمونه ی خوب دارای میزان تشابه ۰.۳۶ کاملاً مؤثر بود. در حالی که میزان شباهت ۱ تنها مختص نمونه ی فونت های رایانه ای (به طور کامل شبیه به خودش) بود. این بخش به نوعی پیاده سازی کنترل کننده ی فازی بود. چرا که هر زمان میزان تشابه نمونه ی خوب به نمونه ی معیار رایانه ای بیشتر بود، میزان اثرگذاری آن نیز کنترل شده و به طبع میزان تشابه نمونه ی آزمایشی به رقم مورد بررسی نیز بیشتر بود. این عمل باعث می شد تا میزان اثرگذاری نمونه های خوب که به عنوان پاداش یادگیری تقویتی در نظر گرفته شده بودند، به جای ۰ یا ۱ بودن، عددی میان ۰ و ۱ بنا به نتیجه ی کنترل کننده ی فازی باشند. همچنین با توجه به فرمول یادگیری تقویتی:

$$R = \sum_{t=0}^{\infty} \gamma^t r_{t+1},$$

که در این فرمول R میزان پاداش کل، γ عامل تخفیف و r_t میزان پاداش در مرحله ی $t+1$ از یادگیری است، عامل تخفیف نیز بر میزان اثرگذاری پاداش ها مؤثر بود. این میزان اثرگذاری همان میزان تشابه نمونه ی خوب به نمونه های معیار بود.

در صورتی که میزان تشابه نمونه های خوب در هر مرحله از یادگیری نسبت به تمامی نمونه های خوب انتخاب شده در مراحل قبلی به عنوان پاداش نمونه ی خوب مورد بررسی ثبت می شد، محتمل بود نمونه ای که در مرحله ی آخر یادگیری انتخاب شده بود میزان پاداش بیشتری نسبت به نمونه ی بهتری که در مرحله نخست یادگیری انتخاب شده بود، داشته باشد. اما در صورتی که پاداش میزان شباهت نمونه ی خوب در حال انتخاب، تنها با نمونه ی معیار تعیین می شد، نمونه هایی که در مراحل بالاتر یادگیری انتخاب می شدند، نسبت به نمونه هایی که در مرحله نخست یادگیری انتخاب شده بودند، دارای پاداش کمتری بودند و (در این مثال خاص) این همان نرخ یادگیری و همچنین عامل تخفیف مورد استفاده در یادگیری تقویتی است. نرخ یادگیری تعیین می کند که نمونه های قدیمی به چه میزان به نمونه های جدید ترجیح داده شوند و عامل تخفیف نیز تعیین کننده ی میزان اثرگذاری هر نمونه است. با توجه به لزوم بیشتر بودن تأثیر نمونه های انتخاب شده در مراحل جدید نسبت به نمونه های انتخاب شده در مراحل قبلی، نرخ یادگیری و عامل تخفیف هر دو یکسان هستند.

یکی دیگر از مزیت های کنترل فازی پاداش یادگیری تقویتی، تعیین میزان خوب یا بد بودن تشخیص است. برای مثال اگر تفاوت میان رقمی که دارای بیشینه مقدار به دست آمده در تشخیص یک نمونه ی آزمایشی است و نمونه ی بیشینه ی بعد از آن بسیار کم باشد، می توان حدس زد به احتمال زیاد تشخیص صحیح نبوده است. زیرا مقدار شباهت نمونه ی آزمایشی نزدیک به هر دو رقم دارای بیشینه مقدار است. در این صورت می دانیم که انتخاب صحیح به طور حتم یکی از این دو رقم خواهد بود ولی انتخاب اینکه کدام یک از آن ارقام، انتخاب صحیح هستند، کمی مشکل است و این مشکل در تشخیص را می توان به کاربر استفاده کننده، اعلام نمود.

6. نتایج

- در آزمایشی که بر روی ۲۰ درصد نمونه‌های پایگاه داده (۲۰۰ نمونه) از هر کدام از ارقام انجام شد، نتایج زیر به دست آمد:
- بدون استفاده از یادگیری تقویتی و کنترل‌کننده‌ی فازی (تنها با استفاده از تابع $SSim$) نتیجه‌ی به دست آمده، تشابه ۴۸ درصدی بود و باز از نظر زمانی نیز قابل عملیاتی شدن در نمونه‌های تجاری نبود.
- بدون استفاده از یادگیری تقویتی و با استفاده از کنترل‌کننده‌ی فازی (مقایسه با تمام نمونه‌های یادگیری بدون انتخاب بهترین آن‌ها با استفاده از یادگیری تقویتی و ضرب تشابه به دست آمده در تشابه نمونه‌ی یادگیری نسبت به نمونه‌ی معیار) نتیجه‌ی تشخیص ۵۳ درصد بود. با توجه به زمان زیاد لازم برای مقایسه، از نظر زمانی نیز قابل عملیاتی شدن در نمونه‌های تجاری نبود.
- بدون استفاده از کنترل‌کننده‌ی فازی (بدون استفاده از ضریب تشابه میان نمونه‌های خوب و نمونه‌های معیار رایانه‌ای) و با استفاده از یادگیری تقویتی، این درصد تشخیص به ۷۲ درصد کاهش پیدا نمود.
- به‌وسیله‌ی شیوه‌ی ارائه شده در این مقاله، نتایج به دست آمده به طور میانگین، تشخیص ۸۸ درصدی را نشان می‌دادند.

از مهم‌ترین روش‌های تشخیص ارقام دست‌نویس، استفاده از شبکه‌های عصبی مصنوعی عمیق است. در این روش توابعی برای تشخیص ساختار ارقام دست‌نویس در لایه‌ی پنهان شبکه‌ی عصبی قرار داده می‌شود تا ساختار نمونه را در سطوح مختلف (از جزئیات تا کلیات) بررسی کرده و میزان تشابه ساختاری دو نمونه را بازگرداند. این روش به تازگی نتایج خوبی به دست داده و توانسته تشخیص بالای ۹۵ درصد را نیز ارائه کند.

هرچند شبکه‌های عصبی مصنوعی درصد تشخیص بالایی را ارائه می‌دهند [5]، اما در این مقاله سعی شده است تا از تابع $SSim$ جهت تشخیص میزان تشابه استفاده شود. با توجه به این که استفاده از شبکه‌های عصبی مصنوعی موجب ایجاد تفاوت در موضوع مقاله نمی‌شد، به دلیل سادگی بیان مسئله از تابع $SSim$ که یکی از بهترین توابع تشخیص آماری شباهت دو تصویر نیز بود، استفاده شد. با استفاده از این تابع، کنترل فازی پاداش یادگیری تقویتی، باعث ایجاد بهبود محسوس‌تری در نتایج حاصله، نسبت به استفاده از شبکه‌های عصبی مصنوعی می‌شد که می‌توانست بهبود درصد تشخیص را به خوبی نشان دهد. البته این بدان معنا نیست که در صورت استفاده از شبکه‌های عصبی مصنوعی به جای $SSim$ و کنترل فازی پاداش یادگیری تقویتی، بهبودی در نتایج حاصل نشد، بلکه تنها میزان بهبود درصد تشخیص ارقام، کمتر بود.

7. نتیجه گیری

نتایج به دست آمده حاکی از آن بودند که استفاده از یادگیری تقویتی تأثیر بیشتری نسبت به کنترل‌کننده‌ی فازی در نتیجه به دست آمده دارد، اما این بدین معنی نیست که کنترل‌کننده‌ی فازی باعث بهبود نتایج به دست آمده نمی‌شود. استفاده از ترکیب هر دو روش به وضوح نشان‌دهنده‌ی بهبود قابل توجهی در نتایج الگوریتم $SSim$ بود و این همان هدف مقاله است.

شاید مهم‌ترین کاربرد این روش را بتوان در تشخیص دست‌خط در زبان‌های مختلف و تبدیل آن‌ها به نوشته‌های رایانه‌ای تعیین کرد. البته با توسعه‌ی این ترکیب بر روی نمونه‌هایی از جنس‌های مختلف، می‌توان در بسیاری از موارد به بهبود نتایج تشخیص یا انتخاب، کمک کرد.

از جمله کارهایی که در این زمینه در آینده قابل اجرا است، می‌توان به استفاده از ترکیب یادگیری تقویتی و منطق فازی در بهبود نتایج الگوریتم شبکه‌های عصبی بر روی تشخیص دست‌خط پرداخت. همچنین در سال‌های اخیر استفاده از روش‌های یادگیری عمیق بر بستر شبکه‌های عصبی مصنوعی نیز گسترش یافته‌اند که با در نظر گرفتن این موضوع تلاش

برای بهبود الگوریتم‌های تشخیص دست‌خط به‌وسیله تشخیص ساختاری تصاویر در لایه‌های پنهان و عمیق شبکه‌های عصبی مصنوعی و ترکیب با یادگیری تقویتی و تعیین پاداش‌های یادگیری به‌وسیله کنترل‌کننده‌های فازی می‌تواند به بهبود هر چه بیشتر نتایج کمک شایانی کند.

مراجع

1. دانشنامه‌ی آزاد ویکی‌پدیا، صفحات منطق فازی، کنترل‌کننده‌ی فازی، ادوارد ثورندایک، لطفی زاده، Reinforcement Learning و تشخیص الگو
2. Zadeh, Lotfi A. "FUZZY LOGIC SYSTEMS: ORIGIN, CONCEPTS, AND TRENDS." (2004).
3. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998
4. Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "[Image quality assessment: From error visibility to structural similarity](#)," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, Apr. 2004.
5. <http://www.codeproject.com/Articles/16650/Neural-Network-for-Recognition-of-Handwritten-Digit>