

STDP-based spiking deep convolutional neural networks for object recognition

چکیده

مطالعات قبلی نشان داده است STDP می تواند در یک شبکه نورونی اسپایکی (SNN) استفاده شود تا ویژگی های بصری با پیچیدگی کم یا متوسط را بدون نظارت استخراج کند. البته در این مطالعات از معماری های نسبتاً کم عمق و تنها با یک لایه آموزش پذیر استفاده شده است. تحقیق دیگری نشان داده است که داشتن تعداد لایه زیاد می تواند قدرت تشخیص را افزایش دهد، رویکردی که به آن یادگیری عمیق می گویند. در نتیجه مدلی که در این مقاله طراحی شده است، یک SNN عمیق که شامل چندین لایه پیچشی^۱ (قابل آموزش با STDP) و استخراجی^۲ است. از یک برنامه کدگذاری زمانی نیز استفاده شده است که در آن ابتدا فعال ترین نورون ها و سپس نورون هایی که فعالیت کمتر دارد فایر می کنند یا در کل فایر نمی کنند. در این مقاله همچنین برتری STDP نسبت به سایر روش های یادگیری نظارت نشده نمایش داده می شود. روی هم رفته نتایج ما پیشنهاد می کند که ترکیبی از STDP و برنامه نویسی latency می تواند کلیدی باشد برای درک روشی که سیستم بینایی پستانداران با سرعت بالا و مصرف انرژی کم، می آموزد. این مکانیسم ها در سیستم دید مصنوعی مورد توجه به ویژه برای دستیابی به راه حل های سخت افزاری.

مقدمه

سیستم بینایی پستانداران، وظیفه تشخیص شی را از طریق پردازش سلسله مراتبی در امتداد مسیر شکمی قشر بینایی^۳ حل می کند.

در طی دهه های گذشته، مدل های محاسباتی مختلفی برای تقلید از پردازش سلسله مراتبی feed-forward پیشنهاد شده اند. با وجود موفقیت های محدود مدل های اولیه، پیشرفت های اخیر در DCNN منجر به مدل هایی با عملکرد بالا شده است.

^۱Convolutional
^۲Pooling

^۳Visual Cortex

اگر چه معماری DCNN ها از به نوعی از سیستم بصری پستانداران الهام گیری شده است اما آنها کاملاً از پردازش عصبی و مکانیسم های یادگیری واقعی در قشر^۱ غافل می شوند.

واحد های محاسباتی DCNN ها مقادیر اعشاری به یکدیگر ارسال می کنند که با سطح فعالیت آنها تطابق دارد؛ درحالی که نورون های بیولوژیکی با ارسال ضربه های الکتریکی (مثل اسپایک) با یکدیگر ارتباط برقرار می کنند. دامنه و مدت زمان تمام این اسپایک ها تقریباً یکسان است؛ بنابراین آنها با زمان انتشارشان به طور کامل توصیف می شوند. جالب توجه است که میانگین نرخ اسپایک زدن در سیستم بصری پستانداران بسیار پایین است؛ بنابراین به نظر می رسد نورون ها تنها زمانی فایر می کنند که می خواهند پیام های مهمی را ارسال کنند و بعضی از این اطلاعات را نیز می توان در زمان اسپایک آنها رمز گذاری کرد. چنین برنامه نویسی **spiketime** منجر به یک محاسبات عصبی سریع و پربازده در مغز می شود (کل مغز انسان حدوداً ۱۰ الی ۲۰ وات انرژی مصرف می کند).

DCNN های اخیر که عملکرد بالایی دارند، با الگوریتم های نظارت شده پس انتشار^۲ آموزش داده شده اند که این الگوریتم ها ریشه بیولوژیکی ندارند. اگرچه با دقت خوبی کار می کنند، اما همگرایی در آنها به دلیل^۳ CAP، کند تر است. به علاوه، با توجه به اینکه این DCNN ها معمولاً میلیونها پارامتر آزاد دارند، برای جلوگیری از **overfitting** به میلیون ها مثال برچسب خورده نیاز است. با این حال، پستانداران به ویژه انسان ها می توانند با استفاده از مثال های بسیار کمتری یاد بگیرند و آن هم در حالی که بیشتر اوقات برچسبی در کار نیست. آنها ممکن است بتوانند چنین کاری را به لطف **STDP** انجام دهند، که یک روش یادگیری بدون نظارت است که در قشر بینایی پستانداران رخ می دهد. با توجه به **STDP**، سیناپس هایی که از طریق آنها اسپایک های پیش سیناپسی قبل از/بعد از اسپایک های پس سیناپسی رخ می دهد، تقویت/تضعیف می شوند.

تا به امروز **SNN** های مختلفی برای حل مسئله تشخیص شی ارائه شده است. گروهی از این شبکه ها در واقع نسخه تبدیل شده DCNN های سنتی هستند. ایده اصلی این است که هر واحد محاسباتی DCNN را با یک نورون **spiking** که نرخ فایر کردن اش، با خروجی آن واحد ارتباط دارد جایگزین می کنیم. هدف از این شبکه ها کاهش مصرف انرژی در DCNN ها است. با این حال، مشکلات اجتناب ناپذیر چنین کدنویسی **spike-rate** ای، نیاز به تعداد زیادی اسپایک برای هر تصویر و زمان پردازش زیاد است. علاوه بر این، به کارگیری الگوریتم پس انتشار و استفاده از هردو سیناپس خروجی تحریکی و مهاری در یک نورون از نظر زیست شناسی مورد قبول نیست. از طرف دیگر، **SNN** هایی وجود دارند که در اصل شبکه **spiking** هستند و الگوهای اسپایک زدن را

^۱Cortex

^۲Back-Propagation

^۳Credit Assignment Problem is the problem of determining the actions that lead to a certain outcome.

^۴Labeled

یاد می گیرند. گروه اول این شبکه ها از روش های یادگیری مانند پس انتشار اتوانکدر^۱ استفاده می کنند که از نظر زیست شناسی قابل قبول نیستند. گروه دوم متشکل از SNN با الهام گیری از قوانین یادگیری زیستی هستند که دارای معماری کم عمق یا تنها یک لایه آموزش پذیر می باشند.

در این مقاله یک شبکه SDNN (شبکه نورونی عمیق اسپایکینگ مبتنی بر STDP) همراه با یک کدنویسی عصبی spike time ارائه شده است. این شبکه از یک لایه کدنویسی زمانی تشکیل شده است که به دنبال آن یکسری لایه های کانولوشنال استخراج کننده ویژگی و لایه های پولینگ قرار دارند. لایه اول تصویر ورودی را به یک spike train نامتقارن تبدیل می کند؛ جایی که اطلاعات بصری به ترتیب زمانی اسپایک ها کدگذاری^۲ می شوند. نورون های موجود در لایه پیچشی، اسپایک های ورودی را ادغام می کنند و پس از رسیدن به آستانه^۳ اسپایک می زنند. این لایه ها برای یادگیری ویژگی های بصری به STDP مجهزند. از طریق شبکه، جایی که نورون ها در آخرین لایه کانولوشنال نمونه های اولیه جسم را یاد می گیرند و تشخیص می دهند، ویژگی های بصری بزرگتر و پیچیده تر می شوند. در پایان کلسیفایر، کتگوری تصاویر ورودی را بر اساس فعالیت نورون ها در آخرین لایه پولینگ با استفاده از فیلدهای پذیرش جهانی، تشخیص می دهد.

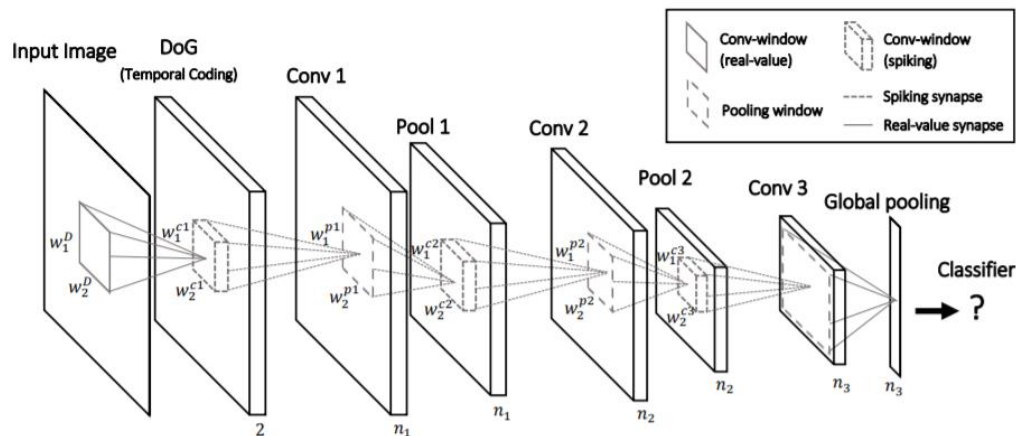
علاوه بر عملکرد بالا، SDNN پیشنهادی بسیار کم مصرف است و با تعداد کمی اسپایک در هر تصویر کار می کند که آن را برای پیاده سازی سخت افزاری نورومورفیک مناسب می کند. اگر چه پیشرفته ترین DCNN های فعلی در کار های مختلف تشخیص، نتایج خیره کننده ای کسب کرده اند، اما کار ادامه یافته روی مدل های الهام گرفته از مغز می تواند در آینده به سیستم های هوشمند قوی تری ختم شود که حتی در بهبود درک خود از مغز کمک کند.

^۱Autoencoder
^۲Encode

^۳Threshold

SDNN پیشنهادی

نمونه معماری SDNN پیشنهادی با سه لایه کانولوشنال و سه لایه پولینگ در شکل نمایش داده شده است. اولین لایه شبکه از DoG برای تشخیص کنتراست تصویر ورودی استفاده می کند.



نورون ها در لایه کانولوشنال با ترکیب اسپایک های ورودی از لایه قبلی که ویژگی های بصری ساده تری را تشخیص می دهند، ویژگی های پیچیده تری را تشخیص می دهند. نورون های کانولوشنال به محض اینکه ویژگی بصری مدنظر خود که به وزن سیناپسی آنها وابسته است را تشخیص می دهند، اسپایک می زنند. از طریق یادگیری، نورون هایی که زودتر فایر می کنند، STDP را اجرا می کنند و با استفاده از مکانیزم winner-take-all از فایر کردن سایر نورون ها جلوگیری می کنند. به این ترتیب، شبکه ویژگی های برجسته و مکرر بیشتری را می آموزد. نورون های موجود در لایه پولینگ اولین اسپایک دریافت شده از نورون های همسایه در لایه قبلی را که برای همان ویژگی انتخاب می شوند، منتشر می کنند. لایه های کانولوشنال و پولینگ نیز به ترتیب متوالی مرتب می شوند.

لازم به ذکر است که پتانسیل درونی تمام نورون ها قبل از پردازش تصویر بعدی، صفر می شود و همچنین یادگیری تنها در لایه کانولوشنال و به صورت لایه به لایه رخ می دهد. از آنجا که محاسبات هر نورون مستقل از نورون های مجاور است، برای سرعت دهی به محاسبات، هر یک از عملیات convolution، pooling و STDP همگی به صورت موازی بر روی GPU انجام می شوند.

DoG و کدنویسی زمانی

نقش مهم مرحله اول در SNN ها، کدگذاری سیگنال های ورودی به رویدادهای اسپایکی گسسته در دامنه زمانی است. این کدنویسی زمانی محتوا و میزان اطلاعاتی که هر اسپایک حمل می کند را تعیین می کند که تاثیر عمیقی بر محاسبات عصبی درون شبکه دارد. از این رو استفاده از کدنویسی کارآمد SNN ها می تواند منجر به پاسخ سریع و دقیق شود. از طرح های مختلف کدنویسی زمانی می توان در پردازش بصری استفاده کرد. در میان آنها، کدنویسی rank_order برای پردازش سریع (حتی احتمالاً در سلول های گانگلیونی شبکه) کارآمد نشان داده شده است.

سلولهای موجود در لایه اول شبکه، یک فیلتر DoG را روی فیلدهای دریافتی اعمال می کنند تا کنتراست + و - را روی تصویر ورودی تشخیص دهد. DoG به خوبی ویژگی های فراگیر مرکز سلولهای گانگلیونی شبکه را تقریب می زند. زمانی که یک تصویر به سلول های DoG داده می شود آنها کنتراست را تشخیص می دهند و اسپایک می زنند؛ هر چه یک سلول قوی تر فعال شود (کنتراست بالاتر)، زودتر اسپایک می زند. به عبارت دیگر ترتیب اسپایک ها، به ترتیب کنتراست ها بستگی دارد.

سلولهای DoG از نظر رتینوئوپیک در یکی از دو دسته ON-center و OFF-center قرار دارند که به ترتیب به کنتراست مثبت و منفی حساس اند. اگر activation یک سلول DoG از یک آستانه مشخصی بالاتر باشد، می تواند فایر کند. توجه داشته باشید که این طرح تضمین می کند که حداکثر یکی از دو سلول (مثبت یا منفی) می تواند در هر موقعیتی فایر کند. همانطور که گفته شد زمان فایر کردن یک سلول DoG با مقدار فعالسازی آن متناسب است. برای محاسبات موازی کارآمد مبتنی بر GPU، اسپایک های ورودی در بسته های متوالی با اندازه برابر گروه بندی می شوند. در هر گام زمانی، اسپایک های هر بسته به طور همزمان منتشر می شوند. به این ترتیب یک بسته از اسپایک ها با نرخ های نزدیک (که اطلاعات بصری مشابه به هم را حمل می کنند) به صورت موازی منتشر می شوند. در حالی که بسته اسپایک بعدی در گام زمانی بعدی پردازش می شود.

لایه های پیچشی

یک لایه پیچشی شامل چندین نقشه عصبی است. نورون ها در یک نقشه خاص ویژگی های بصری یکسان اما در مکان های مختلف را تشخیص می دهند. برای این منظور وزن سیناپسی نورون های متعلق به یک نقشه باید همیشه یکسان باشد. هر نورون اسپایک های ورودی را از نورون های واقع در یک پنجره مشخص در تمام نقشه های عصبی لایه قبلی دریافت می کند. از این رو، ویژگی های بصری در لایه کانولوشنال، ترکیبی از چندین

ویژگی ساده تر است که از لایه های قبلی استخراج شده است. توجه داشته باشید که پنجره های ورودی دو نورون مجاور همپوشانی زیادی دارند. از این رو شبکه می تواند ظاهر ویژگی های بصری را در هر مکان تشخیص دهد.

نورون ها در تمامی لایه های کانولوشنال، نورون های **non-leaky integrate-and-fire** هستند که اسپایک های ورودی را از نورون های پیش سیناپسی جمع آوری می کنند و زمانی که پتانسیل داخلی آنها به یک آستانه مشخصی می رسد، یک اسپایک می زنند. هر اسپایک پیش سیناپسی، پتانسیل نورون را با استفاده از وزن سیناپسی اش افزایش می دهد.

لایه های پولینگ محلی

لایه های پولینگ با استفاده از انجام یک **max pooling** غیر خطی روی نورون های همسایه با استفاده از همان ویژگی های مدنظر به شبکه کمک می کنند تا پایدار شود. برخی شواهد نشان می دهند که چنین عملیات حداکثری در سلولهای پیچیده در قشر بینایی رخ می دهد. به لطف کدنویسی **rank_order** که در شبکه ارائه شده استفاده شده است، حداکثر عملیات لایه پولینگ شامل انتشار اولین اسپایکی است که توسط حسگرها منتشر شده است.

یک نورون در نقشه عصبی یک لایه پولینگ، حداکثر عملیات را روی پنجره ای مربوط به نقشه عصبی لایه قبلی اجرا می کند. نورون های پولینگ، نورونهای **integrate-and-fire** هستند که وزن سیناپسی ورودی و آستانه آنها همگی برابر مقدار ۱ تنظیم شده اند. بنابراین اولین اسپایک ورودی آنها را فعال می کند و منجر به اسپایک خروجی می شود. با توجه به کدنویسی **rank-order** هر نورون پولینگ حداکثر یکبار می تواند فایر کند. لازم به ذکر است که هیچگونه یادگیری در لایه های پولینگ رخ نمی دهد.

نقش مهم دیگر لایه های پولینگ، فشردن سازی اطلاعات بصری است. با توجه به حداکثر عملیات انجام شده در لایه های پولینگ، نورون های مجاور با ورودی های همپوشان اطلاعات اضافی را حمل می کنند (هر اسپایک به تعداد زیادی از نورون های پولینگ همسایه فرستاده می شود). از این رو، در شبکه پیشنهادی همپوشانی بین پنجره ورودی دو نورون پولینگ مجاور (متعلق به همان نقشه)، بسیار کم تنظیم شده است. این کمک می کند تا اطلاعات بصری را با از بین بردن چیزهای اضافه، فشردن کنید و همچنین اندازه لایه های بعدی را کاهش دهید.

یادگیری بر مبنای STDP

همانطور که در بالا گفته شد، یادگیری تنها در لایه های کانولوشنال صورت می گیرد که باید بیاموزند تا ویژگی های بصری را با ترکیب ویژگی های ساده تر استخراج شده از لایه قبلی تشخیص دهند. این یادگیری به صورت لایه به لایه انجام می شود یعنی یادگیری در یک لایه کانولوشنال با نهایی شدن یادگیری در لایه کانولوشنال قبلی شروع می شود. هنگامی که یک تصویر جدید ارائه می شود، نورون های لایه کانولوشنال با یکدیگر رقابت می کنند و آنهایی که زودتر فایر می کنند، STDP را آغاز می کنند و الگوی یادگیری را می آموزند.

در طول یادگیری لایه کانولوشنال، نورون ها در نقشه یکسان که ویژگی های یکسان در مکان های مختلف را تشخیص می دهند، اسپایک های ورودی را ادغام می کنند و با یکدیگر برای انجام STDP رقابت می کنند. اولین نورونی که به آستانه برسد و فایر کند، برنده است. برنده STDP را آغاز کرده و سپس وزن های سیناپسی آن را آپدیت می کند. همانطور که قبلاً گفته شد، نورون هایی که در مکان های مختلف یک نقشه قرار دارند، وزن سیناپسی یکسانی دارند. بنابراین نورون برنده از سایر نورون ها برای انجام STDP در نقشه خود جلوگیری می کند و وزن سیناپسی آپدیت شده خود را در آنها نیز کپی می کند. همچنین برای STDP یک رقابت inner-map محلی وجود دارد. وقتی به یک نورون اجازه انجام STDP داده می شود، آن نورون از انجام STDP توسط نورون هایی که در نقشه هایی در همسایگی نزدیک آن قرار دارند جلوگیری می کند. این رقابت برای ترقیب یادگیری ویژگی های مختلف توسط نورون هایی که در نقشه های مختلف قرار دارند، امری حیاتی است.

به دلیل discretized time variable در مدل ارائه شده، این مکان وجود دارد که برخی از نورون های رقیب با هم در یک گام زمانی فایر کنند. یک سناریو ممکن این است که به صورت تصادفی نورون ار انتخاب کرده و به آن اجازه دهید STDP انجام دهد. اما گزینه بهتر انتخاب نورونی است که پتانسیل بیشتری برای نشان دادن شباهت بیشتر بین ویژگی های یادگرفته شده و الگوی ورودی دارد.

لایه پولینگ جهانی و کلاس بندی^۱

لایه پولینگ جهانی تنها در مرحله کلاس بندی استفاده می شود. نورون های لایه آخر، یک global max pooling روی آخرین لایه کانولوشنال نقشه های عصبی مربوط به خود انجام می دهند. خروجی لایه پولینگ جهانی روی تصاویر آموزشی، برای آموزش یک کلسیفایر خطی SVM استفاده می شود. در مرحله تست، تصویر

^۱Classification

شی آزمایشی توسط شبکه پردازش می شود و خروجی لایه جهانی پولینگ برای تعیین کتگوری آن به کلسیفایر داده می شود.

برای محاسبه ی خروجی لایه پولینگ جهانی، ابتدا آستانه نوروں ها در آخرین لایه کانولوشنال به مقدار بی نهایت تنظیم می شود و پتانسیل نهایی آنها (بعد از انتشار کل spike train ایجاد شده توسط تصویر ورودی) اندازه گیری می شود. این پتانسیل های نهایی را می توان تعداد اسپایک های اولیه مشترک بین ورودی اخیر و نمونه های اولیه ذخیره شده در آخرین لایه کانولوشنال مشاهده کرد. سرانجام نوروں های پولینگ جهانی، حداکثر پتانسیل را در نقشه نوروںی مربوط به خود به عنوان مقدار خروجی محاسبه می کنند.

بحث

DCNN های تحت نظارت اخیر در اکثر دیتاست های چالش برانگیز تشخیص شی مثل ImageNet به دقت بالایی رسیده اند. معماری این شبکه ها به طور گسترده ای از پردازش سلسله مراتبی عمیق در قشر بینایی الهام گرفته شده است. با این حال یادگیری مکانیزم پردازش نوروںی اعمال شده در DCNN ها با قشر بینایی در جایی که نوروں ها با اسپایک زدن ارتباط برقرار می کنند و الگو های اسپایکی را غالباً به صورت نظارت نشده یاد می گیرند، در تناقض است. به کارگیری چنین مکانیزم هایی در DCNN ها می تواند مصرف انرژی آنها را بهبود ببخشد و نیاز آنها به یادگیری نظارت شده که با استفاده از میلیون ها تصویر بر چسب زده صورت می گیرد را کاهش دهد.

در اینجا ما یک SDNN مبتنی بر STDP با کدنویسی spike-time پیشنهاد کردیم. هر نوروں حداکثر یکبار اجازه فایر کردن داشت، جایی که spike-time، اهمیت ورودی بصری آن را نشان می دهد. بنابراین نوروں هایی که زودتر فایر می کنند، اطلاعات بصری برجسته تری را حمل می کنند و از این جهت اجازه انجام STDP و یادگیری الگو های ورودی را دارند. با پیشرفت یادگیری، هر لایه به مجموعه ای از ویژگی های متنوع اما آموزنده همگرا می شود. SDNN پیشنهادی در چندین دیتاست تصویر مورد ارزیابی قرار گرفت و به دقت بالایی رسید. این نشان می دهد چگونه کدنویسی زمانی پیشنهادی و مکانیزم یادگیری (STDP و رقابت یادگیری) منجر به discriminative object representation می شود.

SDNN های پیشنهادی چندین مزیت برای همتایان خود دارند. اول، SDNN پیشنهادی ما اولین شبکه عصبی اسپایکی با بیشتر از یک لایه آموزش پذیر است که می تواند تصاویر را در مقیاس بزرگ پردازش کند. دوم، به دلیل استفاده از یک کدنویسی کار آمد زمانی، که در زمان اولین اسپایک اطلاعات بصری را کدگذاری می کند،

می توانن تصویر ورودی را با تعداد کمی اسپایک و تنها در چند گام زمانی پردازش کرد. سوم، SDNN پیشنهادی از زیست الهام گرفته و از قانون کاملاً بدون نظارت STDP بهره می برد که می تواند ویژگی های تشخیصی شی را بیاموزد و ویژگی بی ربط را نادیده بگیرد.

ما SDNN پیشنهادی را با چندین شبکه دیگر از جمله روش های نظارت نشده مانند HMAX و شبکه اتوانکدر پیچشی و همچنین روش های نظارت شده مثل DCNN ها مقایسه کردیم. SDNN پیشنهادی با استفاده از روش های بدون نظارت که برتری خود را در استخراج ویژگی های اطلاعاتی بیشتر از تصاویر آموزشی نشان می دهد، عملکرد بهتری داشت. همچنین از شبکه های عمیق نظارت شده که به شدت از overfitting و نبودن دیتای آموزشی کافی رنج می بردند نیز بهتر بود.

همچنین SDNN پیشنهادی می تواند اصلاح شود تا از مدل های اسپایکی شبکه چشم به عنوان ورودی استفاده کند. این مدل ها از فیلترینگ سلولهای گانگلیونی شبکه تقلید می کنند.

از این رو ، مطالعات آینده باید به encoding این اطلاعات اضافه در لایه ورودی تمرکز کنند .

شواهد بیولوژیکی نشان می دهند که علاوه بر مکانیزم های یادگیری بدون نظارت (مثل STDP) ، در مغز استراتژی یادگیری تقویتی مثل STDP مبتنی بر دوپامین وجود دارد. علاوه بر این، اگر چه هنوز نحوه یادگیری نظارت شده در شبکه های عصبی بیولوژیکی مشخص نیست، به نظر می رسد برای برخی از کارها (به عنوان مثال پیش بینی ورودی های حسی)، مغز باید دائماً پویایی زمانی را بر اساس بازخورد خطا یاد بگیرد. استفاده از چنین استراتژی های یادگیری تقویتی و تحت نظارت می تواند SDNN پیشنهادی را از جنبه های مختلف که با روش های یادگیری بدون نظارت اجتناب ناپذیر است، بهبود ببخشد. به ویژه آنها می توانند به کاهش تعداد ویژگی های مورد نیاز کمک کنند.