

(۱) از آنجا که وزن های نرون های جدید در شبکه GSOM به صورت منظم با وزنهی قبلی ایجاد می شوند در این شبکه نیاز به فاز Ordering نیست. با ورود هر داده جدید، پارامتر همسایگی به یک مقدار اولیه برمیگردد. سپس وزن ها به صورت تدریجی و با کاهش شعاع همسایگی بروز می شوند. بنابراین به طور خلاصه نیاز به شعاع همسایگی بزرگی نیست.

در فاز smoothing همسایگی و ضریب یادگیری یک مقدار کوچک می شوند تا تنظیم دقیق صورت بگیرد.

(۲) این رابطه فرمول انتشار خطا در یک شبکه دارای فیدبک تماما متصل را نشان می دهد. به طور دقیق تر، ارور در زمان  $t$  در یک واحد (نرون)  $u$  با رابطه فوق به یک نرون  $v$  در  $t$  مرحله زمانی قبل تر منتشر می شود. طبق رابطه فوق به وضوح خطاها در طول زمان در هم ضرب شده و به زمان های قبل تر منتشر می شود. بنابراین اگر عبارت درونی بیشتر از یک باشد گرادیان منفجر شونده و اگر کمتر از یک باشد گرادیان محو شونده بوجود می آید.

(۳) به طور کلی مشکل روش هایی که از انحنای تابع خطا (مشتق درجه ۲) استفاده می کنند عبارتند از:

(۱) هزینه بر بودن محاسبه واران ماتریس  $H$

(۲) نا پایدار بودن عملیات واران کردن ماتریس  $H$  (این ماتریس ممکن است واران نداشته باشد)

(۳) در صورت کوچک بودن مشتق مرتبه دوم، ممکن است دچار اشتباه شوند. (مخرج کسر افزایش می یابد)

علاوه بر موارد فوق الگوریتم LM نیاز به یک پارامتر اضافه دارد که اگرچه به صورت اتوماتیک تعیین میشود اما سرعت آموزش را کاهش میدهد.

(۴) داده ها باید تعداد مناسبی داشته باشند و در فضای حالت مسئله به خوبی توزیع شده باشند.

داده ها نباید مرتب شده باشند. متغیرهای آن ها تا حد ممکن نباید همبستگی داشته باشند. متغیرهای زائد باید از داده ها حذف شده اند. متغیرها باید تا حد ممکن اسکیل یکسانی داشته باشند و بهتر است بین  $-1$  و  $1$  نرمال شوند. داده های پرت باید با شیوه مناسبی تشخیص داده شوند با روش های مناسب به آنها رسیدگی شود (حذف کل داده، حذف متغیر پرت، جاگذاری متغیر پرت و...)

کل دادگان باید به صورت مناسبی به چند مجموعه برای آموزش، تست و ارزیابی شبکه تقسیم شود.

(۵) این لایه به عنوان حد واسط بین دو بلاک متوالی استفاده میشود و دارای نقش های زیر است:

الف) با اجرای batch normalization باعث می شود توزیع داده های ورودی به بلاک بعد حدودا ثابت بماند و یادگیری کند نشود.

ب) حجم فیچر کمپ ها را به شیوه کاهش میدهد:

۱) با اجرای کانولوشن  $1$  در  $1$  تعداد فیچر کمپ ها را کنترل و تعیین می کند.

۲) با اجرا pooling ابعاد فیچر کمپ ها را کاهش و میدان دید را افزایش می دهد

۶) این شبکه یکی از کاربردهای معماری کدکننده کدگشا برای توصیف ویدیو است.

قسمت بالایی شبکه بخش کدکننده است که در آن CNN ها ویژگی های هر فریم را استخراج می کنند و 3d cnn ها ارتباط بین فریم های متوالی را استخراج می کند. سپس نتایج هر دو ترکیب شده و به وسیله شبکه کدگشا توصیف ویدیو ایجاد میشود. شبکه کدگشا یک شبکه بازگشتی چند لایه است که در چند مرحله توصیف کدی را که بخش کدکننده ایجاد کرده بود استخراج می کند.

۷) اندازه (عمق) آن بستگی به تعداد فیچر مپ هایی دارد که قرار است روی آن اعمال شود.  $(1 \times 1 \times \text{تعداد فیچر مپ های ورودی})$

تعداد کانولشن ها بستگی به کاربرد مسئله و جایگاه لایه کانولوشنی ۱ در ۱ شبکه دارد. به عنوان مثال اگر در لایه بعد به  $m$  فیچر مپ نیاز داشته باشیم، تعداد کانولوشن ها  $m$  خواهد بود.

۸) تابع هدف در این شبکه به صورت یک بازی minMax تعریف شده است که بخش جداکننده میخواهد آن را ماکس کند (دقت تشخیص نمونه های واقعی را افزایش دهد) و بخش مولد میخواهد با تولید نمونه های غیر واقعی، سعی کند دقت بخش جداکننده را کاهش دهد. این کار به صورت آموزش بخش مولد به گونه ای است که پارامتر هایش در جهتی تغییر یابند که امید ریاضی خطای بخش جداکننده را افزایش دهد. بنابراین این شبکه باید برای افزایش دادن تابعی از خطای بخش جداکننده آموزش ببیند.

۹) ابتدا میانگین و انحراف معیار همه فیچر مپ ها محاسبه می شود. سپس برای داشتن توزیع نرمال کل فیچر مپ ها منهای میانگین و تقسیم بر انحراف معیار میشوند (به طور دقیق تر تقسیم بر جذر واریانس به علاوه یک مقدار اسیلون)

سپس کلیه فیچر مپ ها که دارای توزیع نرمال با میانگین صفر و واریانس یک شده اند با استفاده از دو پارامتر  $a$  و  $b$  به صورت  $ax+b$  به توزیع جدید منتقل می شوند. (پارامتر های  $a$  و  $b$  در شبکه یاد گرفته می شوند)

نرمال سازی به این دلیل انجام میشود که با تغییر وزن های لایه های قبل ممکن است ورودی های لایه فعلی در هر مرحله دارای توزیع متفاوتی شود و در نتیجه یادگیری سخت تر میشود (Internal covariate shift = باعث سخت تر شدن تعیین ضریب یادگیری و پارامترهای اولیه نیز میشود)

منظور از دسته ای این است که کلیه فیچر مپ ها با هم نرمال میشوند و در واقع از میانگین و واریانس کل آن ها برای نرمال سازی استفاده می شود.

چون کلیه فیچر مپ ها با هم در لایه بعدی پردازش می شوند این کار باید به صورت دسته ای انجام شود، در واقع هر درایه از فیچر مپ ها به صورت یک متغیر تصادفی است که برای نرمال سازی باید چندین مقدار از آن را داشته باشیم. یعنی باید چند فیچر مپ با هم نرمال شوند.

۱۰) خودکدگذار های نوین گیر سعی میکنند خروجی (بازسازی) decoder نسبت به نویز حساس نباشد و در نتیجه لزوماً بردار بازنمایی  $h=f(x)$  نسبت به نویز مقاوم نیست. بلکه بازسازی آن  $g(f(x))=g(h)$  نسبت به نویز مقاوم است. از طرف مقابل خودکدگذار انقباضی سعی می کند خروجی encoder یا همان بردار  $h$  نسبت به نویز مقاوم باشد (با استفاده از تابع هدف متفاوت)

بنابراین خودکدگذار انقباضی یک همسایگی کوچک در فضای  $x$  را به یک نقطه فشرده در فضای  $h$  نگاشت می کند.

(۱۱) بدین منظور باید متغیرهای مولفه پنهان از هم مستقل باشند یعنی اعضای غیر قطری ماتریس کوواریانس آن صفر باشند. (مانند توزیع نرمال با واریانس قطری).

به بیان دیگر بردارهای پایه توصیف کننده فضای حالت  $z$  orthonormal باشند. (مانند بردارهای پایه  $pca$ )

(۱۲) در صورت تغییرات زیاد در بازه های زمانی مختلف، از خوشه بندی بازه های زمانی استفاده شود. (به عنوان مثال در مسئله پیش بینی مصرف برق، الگوی مصرف در بازه های زمانی مختلف بسیار متفاوت است)

استفاده از کل دیتاست جمع آوری شده باعث افزایش دقت می شود.

تا حد امکان نمونه های آموزشی نزدیک به نمونه هایی باشد که در آینده قرار است پیش بینی شوند.

استفاده از شبکه های غیر خطی به جز در مواردی که پیش بینی یک قدم در مجموعه ای متفاوت از مجموعه آموزش است، ارجحیت دارد.

استفاده از مدل هایی که خطا نیز به عنوان ورودی دارند (MA) ارجحیت دارد.