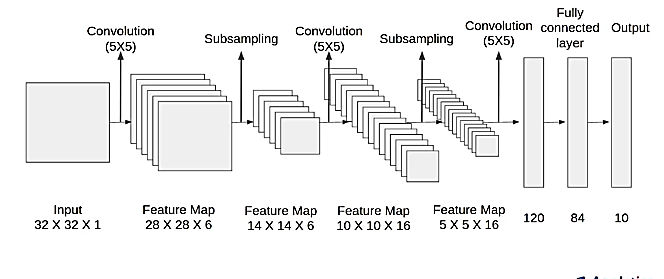
Part1-Model architecture:

اولین و مهم ترین بخش طراحی یک شبکه عصبی انتخاب یک معماری مناسب است. در صورت انتخاب اشتباه معماری نه تنها باعث میشود لایه ها به طرز مورد نظر به خوبی کار نکنند بلکه باعث کاهش دقت چه در آموزش و چه در ارزیابی می شوند. یک راه که اول به ذهن میرسد آن است که چندین مدل طراحی کرده و در نهایت خروجی آنها را با یکدیگر مقایسه کنیم. اما این راه چندین مشکل دارد اولین مشکل و واضح ترین آن این است که ما زمان کافی برای تست چندین مدل را نداریم در صورتی که این کار نیاز به زمان زیاد دارد از مشکلات دیگر می توان به عدم تجربه کافی در طراحی معماری شبکه عصبی اشاره کرد.

حال سوالی که پیش می آید این است که برای معماری مدل باید چه کنیم؟ پاسخ ما استفاده از معماری هایی است که برای مسئله ای مشابه با مسئله ما طراحی شده اند. از جمله این معماری ها میتوان به معماری های Lenet 5، Alexnet و VGGnet اشاره کرد که مورد استفاده ما در این پروژه معماری Lenet 5 میباشد. شکل زیر این معماری تعداد لایه ها و اندازه آنها را نمایش میدهد.



که همانطور که مورد انتظار ماست در نهایت به ما یک بردار ده تایی که نماینده بردار مشخص کننده رقم عدد ماست ارائه میدهد. که برای توضیحات بیشتر میتوانید به (<https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/>) مراجعه کنید.

Part2-Tuning Parameters

حال که معماری شبکه مشخص شده است باید پارامترهای یادگیری آن را تنظیم کنیم.

ضریب یادگیری (Learning Rate):

learning rate یک پارامتر است که تعیین می‌کند که مقدار چقدر از گام‌های آموزش برای بهینه‌سازی وزن‌ها و بایاس‌ها استفاده شود. با افزایش learning rate، مقدار تغییرات وزن‌ها و بایاس‌ها در هر گام آموزش نیز افزایش می‌یابد.

1. کم شدن learning rate: با کاهش learning rate، هر گام آموزش کوچکتر می‌شود و بهینگی آموزش افزایش می‌یابد. این باعث می‌شود که الگوریتم بهینه سازی به صورت دقیق تر به جوانب جانبی منحنی هسته رسیدگی کند و به نتایج بهتر برسد. با این حال، استفاده از learning rate خیلی کوچک منجر به سرعت پیدا نکردن حلول مناسب و گیر کردن در نقطه محلی مینیمم محلی می‌شود.

2. زیاد شدن learning rate: با افزایش learning rate، هر گام آموزش بزرگتر می‌شود و سرعت آموزش افزایش می‌یابد. این باعث می‌شود که الگوریتم بهینه سازی به سرعت به جوانب جانبی منحنی هسته رسیدگی کند و در نقاط بهینه قابل قبول توقف کند. با این حال، استفاده از learning rate خیلی بزرگ منجر به عبور از نقاط بهینه قابل قبول و دست رس نکردن به حلول مناسب می‌شود.

اندازه بسته ها(Batch size):

در شبکه‌های عصبی، batch size به تعداد نمونه‌هایی از داده‌ها اشاره دارد که در هر مرحله از آموزش به شبکه وارد می‌شوند. به عبارت دیگر، batch size تعداد داده‌هایی است که در هر مرحله از آموزش برای به‌روزرسانی وزن‌ها استفاده می‌شود.

1. کم شدن batch size:

سرعت آموزش: با کاهش batch size، تعداد عملیات مورد نظر برای به‌روزرسانی وزن‌ها کاهش پیدا می‌کند. این باعث می‌شود که سرعت آموزش شبکه افزایش یابد.

حافظه: با کاهش batch size، حافظه لازم برای نگاشت دادگان نسخه بزرگتر شبکه کاهش می‌یابد.

تنوع: با کاهش batch size، تنوع داده‌های ورودی به شبکه افزایش می‌یابد. این ممکن است منجر به یادگیری بهتر و عمومی‌تر شود.

2. زیاد شدن batch size:

سرعت آموزش: با افزایش batch size، تعداد عملیات مورد نظر برای به‌روزرسانی وزن‌ها افزایش پیدا می‌کند. این باعث می‌شود که سرعت آموزش شبکه کاهش یابد.

حافظه: با افزایش batch size، حافظه لازم برای نگاشت دادگان نسخه بزرگتر شبکه افزایش می‌یابد.

تنوع: با افزایش batch size، تنوع داده‌های ورودی به شبکه کاهش می‌یابد. این ممکن است منجر به چسبیدگی به دادگان خاص و در نتیجه چالش بالاتر در تطبیق با دادگان جدید شود.

لاندا:

این پارامتر تعیین کننده تاثیر تریپلت لاس بر روی شبکه است که کاهش شدید آن باعث کاهش دقت در لایه های ماقبل آخر گشته اما تاثیر زیادی بر روی دقت نهایی نخواهد داشت در عین حال افزایش بی رویه آن باعث کاهش تاثیر cross entropy می‌شود، که با این اتفاق دقت شبکه در تشخیص ارقام به شدت افت خواهد کرد.

Part3-Missing Data handling

برای پیش بینی داده ها بدون بردار فیچر سه راهکار وجود دارد:

1. کم کردن تاثیر: یک راه حل ساده و ابتدایی برای آنکه بتوانیم داده های بدون بردار فیچر را در مدل خود استفاده کنیم این است که هر داده ترین را یک بار با بردار فیچر و بار دیگر با بردار فیچر صفر آموزش دهیم نتیجه این روند آن خواهد شد که وزن ورودی به نورون های متصل به بردار فیچر کاهش خواهد داشت و شبکه در واقع به این بردار به چشم یک کمک اضافه نگاه خواهد کرد.
2. نادیده گرفتن: راه حل نهایی و منتخب ما نادیده گرفتن این فیچرها است همانطور که در خیلی از موارد سعی میشود بدون در نظر گرفتن داده ناموجود و فقط با داشتن اطلاعات موجود در مسئله به حل مشکل پرداخته میشود. این راهکار درعین سادگی و شفاف بودن تا حدی پارامترهای مدل را نسبت به داده های موجود کاهش میدهد که این در مواردی که داده های در دسترس کم هستند می تواند تا حد قابل توجهی جلوی اورفیت شدن شبکه را بالا ببرد.