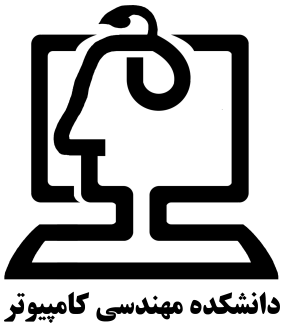
به نام خدا





دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر

استاد درس: دکتر عبادزاده

اسفند ۱۴۰۰

# درس پردازش تکاملی

**سروش مهدی**
**شماره دانشجویی: ۹۹۱۳۱۰۵۰**

**گزارش پروژه**

---

# فهرست مطالب

**۱ مقدمه** ۱

**۲ نحوه پیاده‌سازی بهینه‌ساز هیبرید** ۱
۱.۲ جزییات پیاده‌سازی . . . . . ۱

**۳ مدل کانوولوشنی استفاده شده** ۲

**۴ بررسی نتایج** ۳
۱.۴ مقایسه با دیگر بهینه‌سازها . . . . . ۴

# ۱ مقدمه

در این پروژه قصد داریم تا عملکرد یک بهینه‌ساز را که بر پایه ترکیب الگوریتم‌های PSO و نزول گرادیان میباشد را برای بهینه‌سازی وزن‌های شبکه عصبی مصنوعی بررسی کنیم و ان را با سایر بهینه‌سازهای بر پایه گرادیان برای اموزش شبکه‌های عصبی مقایسه کنیم.

# ۲ نحوه پیاده‌سازی بهینه‌ساز هیبرید

در این پروژه برای پیاده سازی بهینه‌ساز و همچنین بخش های مختلف شبکه عصبی از فریمورک pytorch استفاده کردم.

فرمول اصلی بهینه ساز پیاده شده به صورت زیر میباشد:

$$\mathrm{V}_{t+1} = WV_t - C_1 r_1 (\frac{\nabla E(X_t)}{\|\nabla E(X_t)\|^2}) + C_2 r_2 (G - X_t)$$

$$\mathrm{X}_{t+1} = X_t + V_{t+1}$$

در این مساله جمعیت ما چند شبکه عصبی میباشند که هر کدام انها متناظر با یک ذره در الگوریتم PSO هستند. در فرمول بالا $X$ متناظر با موقعیت هر ذره هست که در واقع در مساله ما همان وزن‌های شبکه عصبی میباشد. $V$ نیز برابر سرعت هر ذره میباشد. $E$ نشان دهنده تابع هزینه شبکه میباشد و $G$ نیز نشان دهنده بهترین تجربه گروه هست .

$W, C_1, C_2$ ضرایبی هستند که نشان دهنده اهمیت هر جمله در فرمول میباشند که در ادامه دقیق تر در مورد انتخاب ان ها توضیح داده خواهد شد. $r_1, r_2$ نیز دو عدد تصادفی بین صفر و یک انتخاب شده از توزیع یکنواخت هستند.

برای پیاده‌سازی این بهینه‌ساز از کلاس optimizer فریموورک پایتورچ استفاده کردم و بهینه‌ساز به نحوی طراحی شده که طبق استاندارد کتابخانه پایتورچ باشد و با هر مدل شبکه عصبی در این فریموورک سازگار هست و در واقع میتوان ان را برای بهینه‌سازی مدل های مختلف پیاده‌سازی در پایتورچ استفاده کرد.

## ۱.۲ جزییات پیاده‌سازی

در پیاده‌سازی مقدار $r_1$ را در همه حالات برابر یک گرفتم. ضریب $W$ به صورت متناسب با زمان کاهش میابد. این انتخاب به این دلیل هست که امیدواریم در ابتدا شروع الگوریتم جست و جوی عمومی بیشتر از محلی باشد و رفته رفته این نسبت برعکس شود.

برای کاهش $W$ فرمول زیر در نظر گرفته شده:

$$\mathrm{W}_t = W_{max} - \frac{W_{max} - W_{min}}{T_{max}} t$$

در این فرمول $t, T_{max}$ به ترتیب نشان دهنده شماره گامی که در ان هستیم و حداکثر تعداد گام ها میباشند. $W_{max}, W_{min}$ نیز به ترتیب نشان‌دهنده مقدار حداکثر و حداقل ضریب $W$ هستند که برای الگوریتم در نظر گرفته ایم.

ضریب $C_1$ در الگوریتم پیاده شده ثابت در نظر گرفته شده. برای این ضریب همچنین کاهش متناسب با زمان را امتحان کردم که نتیجه خوبی نداشت.
در مورد $C_2$ یک روش خودتطبیقی در نظر گرفته شده به صورت زیر:

$$\mathrm{C}_2 =: C_{2init} * (1 + \frac{GSuccessNum}{T_{max}})$$

در این فرمول $GSuccessNum$ برابر تعداد دفعاتی میباشد که بهترین تجربه گروه حال حاضر تغییر نکرده است. در واقع هر دفعه که تجربه گروه اپدیت میشود این مقدار صفر میشود و اگر تجربه گروه در هر تکرار تغییری نکند یکی به این مقدار اضافه میشود. در واقع منطق این عمل این است که اگر مدت بیشتری یک تجربه گروه ثابت بماند احتمالا موقعیت بهتری هست و باید با سرعت بیشتری به سمت ان رفت.

نکته دیگر پیاده سازی درباره نحوه استفاده از بچ ها در روند بهینه‌سازی میباشد. یکبار الگوریتم را بدون در نظر گرفتن بچ ها ران کردم و نتیجه خوبی نداشت در نتیجه تصمیم گرفتم مانند سایر بهینه‌ساز های شبکه عصبی از بچ ها استفاده کنم. مخصوصا این مورد در پیدا کردن بهترین موقعیت گروه مهم میباشد. در اموزش روی هر بچ بهترین موقعیت گروه را به این صورت اپدیت میکنم که خطای بهترین موقعیت گروهی که تا الان بدست امده را با خطای میانگین هر ذره رو بچ های ایپاک حال حاضر که تا الان محاسبه شده مقایسه میکنم.

## ۳ مدل کانوولوشنی استفاده شده

مدل استفاده شده در این پروژه یک مدل کانوولوشنی مشابه مدل LeNet با دو لایه کانوولوشنی و یک لایه fully-connected میباشد . همچنین تابع فعالسازی تمام لایه ها تابع ReLu هست.

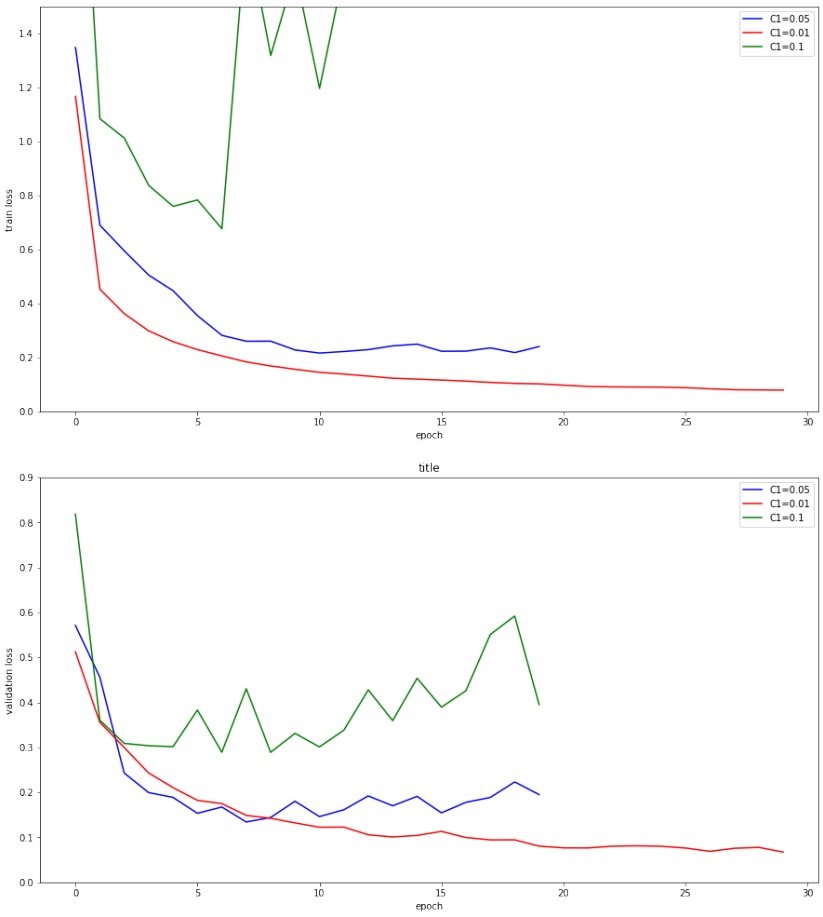
دیتاست مورد استفاده دراین پروژه دیتاست MNIST میباشد که متشکل از تصاویر سطح خاکستری از ارقام صفر تا نه به صورت دست نوشته میباشد. وظیفه ای که شبکه روی ان اموزش میبیند دسته بندی این تصاویر میباشد .

# ۴ بررسی نتایج

در ازمایش ها مختلف متوجه شدم که ضریب $C1$ تاثیر زیادی بر عملکرد بهینه‌ساز دارد. در نتیجه در ازمایش های این بخش مقادیر متغیر های دیگر به صورت زیر هستند:

$$\mathrm{W}_{max} = 0.1, W_{min} = 0.01, C2 = 0.01$$

در صفحه بعد نمودارهای مربوط به اموزش شبکه توسط این بهینه‌ساز با مقادیر مختلف امده است. مقادیر ۰/۱ و ۰/۰۵ و ۰/۰۱ برای متغیر $C1$ امتحان شده اند و در هر نسل مقدار تابع هزینه برای بهترین موجود رسم شده است. مشاهده میشود که به ازای $C1 = 0.1$ الگوریتم همگرا نمیگردد همچنین بهترین عملکرد الگوریتم مربوط به حالتی است که $C1 = 0.01$ میباشد. در مورد $C1 = 0.05$ با توجه به خطای ارزیابی مشاهده میشود که شبکه به سمت بیش برازش میرود.




شکل ۱: نمودار های مربوط به تابع هزینه شبکه روی مجموعه های اموزش و تست

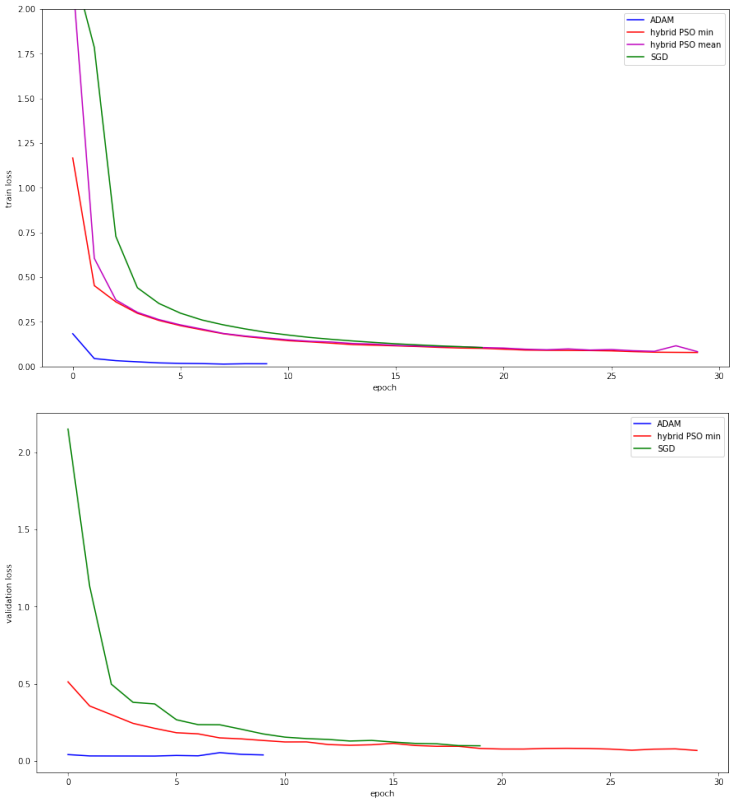
## ۱.۴ مقایسه با دیگر بهینه‌سازها

در این بخش بهترین نتیجه بخش قبل یعنی

$$\mathrm{W}_{max} = 0.1, W_{min} = 0.01, C1 = 0.01, C2 = 0.01$$

را با دو بهینه ساز ADAM و SGD مقایسه میکنم. در ادامه نمودارهای مربوط به این مقایسه را میبینیم . در این نمودار ها خطای شبکه های اموزش داده شده با بهینه‌ساز های مختلف روی مجموعه های اموزش و تست نمایش داده شده است همچنین برای بهینه ساز hybrid-PSO بهترین خطا در هر نسل و میانگین خطای هر نسل نمایش داده شده. مشاهده میشود که در این مقایسه بهینه‌ساز ADAM بهتر از دو بهینه ساز دیگر عمل میکند.

اما بهینه ساز ما سریع تر از SGD همگرا میشود و همچنین با توجه به خطای ارزیابی روی مجموعه ارزیابی نیز عملکرد خوبی دارد و بیش برازش نشده اما باید این نکته را در نظر داشته باشیم که بهینه ساز ما در هر تکرار بسته به تعداد ذرات محاسبات بیشتری از دو بهینه ساز دیگر لازم دارد. هرچند این ویژگی میتواند به پیاده سازی بهینه ساز روی سیستم های توزیع شده کمک کند




شکل ۲: نمودار های مربوط به تابع هزینه شبکه روی مجموعه های اموزش و تست

مدل اموزش دیده شده توسط بهینه ساز ما در بهترین حالت به ۹۷/۹ درصد دقت در دسته بندی روی مجموعه تست میرسد و همچنین هر ذره نیز حداقل ۹۷ درصد دقت را در این مجموعه دارد. در تمامی ازمایش ها تعداد ذرات برابر ۱۰ در نظر گرفته شده است.