

ایمیل: banafshe.karimian@gmail.com

شماره دانشجویی: ۸۱۰۱۹۹۲۵۲

نام و نام خانوادگی: بنفشه کریمیان

در این پروژه قصد استفاده از گاسیپ ترینینگ برای دیپ لرنینگ در تسک behavioral cloning برای ماشین خودران را داریم. در ابتدا به بررسی گاسیپ ترینینگ در دیپ لرنینگ می پردازیم.

گاسیپ ترینینگ در دیپ لرنینگ

در گاسیپ ترینینگ وزن ها به عنوان پارامتری در نظر گرفته میشن که هر ایجنت با مینیمم کردن بخش خودش وزن دهی خودش رو انجام میده و در آخر وزن هایی مورد نظر ماست که برای همه داده به طور میانگین لاس ما را کمینه کند. اگر x را وزن ها و Y را (image, label) در نظر بگیریم و l برابر لاس فانکشن ما باشد آن گاه قصد مینیمم کردن تابع زیر را داریم که می توان اجماع را روی آن زد:

$$\mathcal{L}(x) = \mathbb{E}_{Y \sim \mathcal{I}}[\ell(x, Y)]$$

$$\sum_{i=1}^M \mathcal{L}(x_i) + \frac{\rho}{2} \|x_i - \bar{x}\|_2^2$$

$$\sum_{i=1}^M \mathcal{L}(x_i) + \frac{\rho}{4M} \sum_i \sum_j \|x_i - x_j\|_2^2$$

در زیر الگوریتم مربوط به گاسیپ ترینینگ در دیپ لرنینگ را می بینیم. در این الگوریتم یک بخش ارسال پیام وجود دارد که وزن خود و آلفای خود تقسیم بر دو را به یک همسایه ارسال می کنیم. همچنین یک الگوریتم پروسس کردن پیام های دریافتی را داریم. به این صورت که طبق آلفای خود و آلفای همسایه ی ارسال کننده پیام وزن خود را آپدیت و آلفای خود را افزایش می دهیم. در الگوریتم اصلی ابتدا پیام های دریافتی پردازش می شوند و یک دور یادگیری انجام می شود. سپس اگر یک برنولی موفق بود یک همسایه به طور رندوم بیدار میشود و پیام ارسال می شود.

Algorithm 1 GoSGD: workers Pseudo-code

```

1: Input:  $p$ : probability of exchange,  $M$ : number of threads,  $\eta$ : learning rate
2: Initialize:  $x$  is initialized randomly,  $x_i = x, \alpha_i = \frac{1}{M}$ 
3: repeat
4:   PROCESSMESSAGES(msgi)
5:    $x_i \leftarrow x_i - \eta^t v_i^t$ 
6:   if  $S \sim B(p)$  then
7:      $j = \text{Random}(M)$ 
8:     PUSHMESSAGE(msgj)
9:   end if
10: until Maximum iteration reached
11: return  $\frac{1}{M} \sum_{m=1}^M x_m$ 

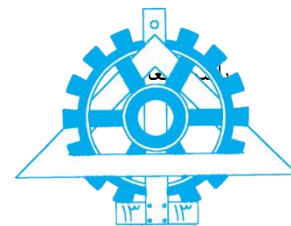
```

Algorithm 2 Gossip update functions

```

1: function PUSHMESSAGE(queue msgj)
2:    $x_i \leftarrow x_i$ 
3:    $\alpha_i \leftarrow \frac{\alpha_i}{2}$ 
4:   msgj.push(( $x_i, \alpha_i$ ))
5: end function
6: function PROCESSMESSAGES(queue msgi)
7:   repeat
8:     ( $x_j, \alpha_j$ )  $\leftarrow$  msgi.pop()
9:      $x_i \leftarrow \frac{\alpha_j}{\alpha_i + \alpha_j} x_j + \frac{\alpha_i}{\alpha_i + \alpha_j} x_i$ 
10:     $\alpha_i \leftarrow \alpha_j + \alpha_i$ 
11:   until msgi.empty()
12: end function

```



دیتاست مورد استفاده

دیتاست مورد استفاده از شبیه ساز Udacity driving simulator جمع آوری شده که در آن از سه زاویه ی مستقیم و راست و چپ زاویه دید راننده ضبط شده و زاویه فرمان مورد نظر برای آن بخش را به عنوان لیبل ذخیره می کنیم. مثالی از دیتا:



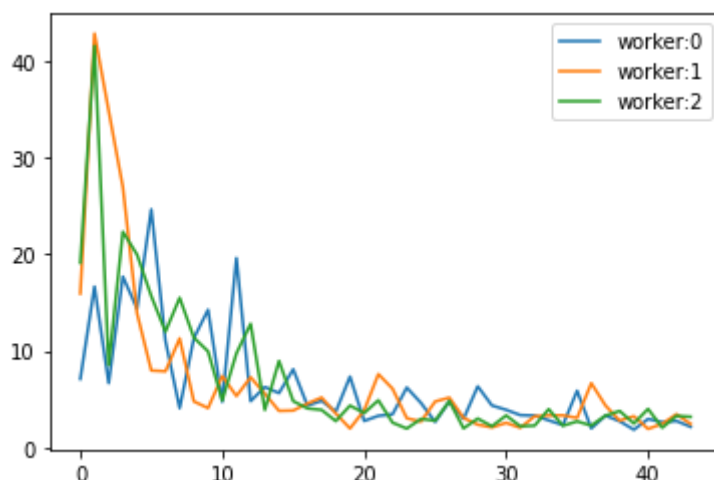
برای پردازش دیتا ابتدا گوشه ها حذف میشوند تا اطلاعات نامرتبط وارد شبکه نشود سپس داده بلرگوسی داده شده و تصویر نرمالایز میشود.

الگوریتم کلی

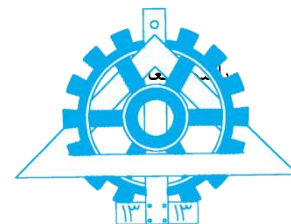
به دلیل اینکه دیتای ما شامل سه عکس از سه زاویه مختلف است، کاری که به ذهن رسید آن بود که سه ورکر مختلف داشته که هر کدام به زاویه ای خاص از دیتا دسترسی داشته باشند. طبق توضیحات قبل، هر ورکر در ابتدا مسیج های دریافتی خود را پردازش و وزن خود را در صورت وجود مسیج آپدیت می کند و یک دور یادگیری کرده و با احتمالی برنولی به یک همسایه رندوم وزن و آلفای آپدیت شده ی خود را ارسال می کند.

نتایج

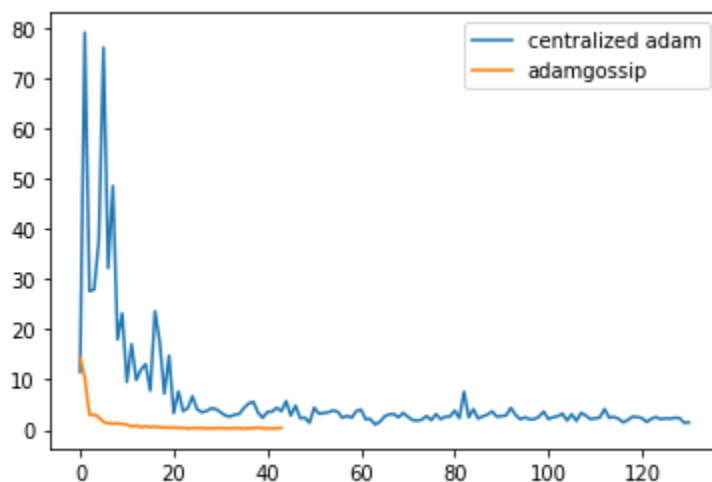
نتیجه لاس هر ورکر را در شکل زیر می توانیم مشاهده کنیم. ابتدا از sgd همانند مقاله استفاده شد اما لاس به مقدار مناسبی نرسدی به همین دلیل از آدام برای آپتیمایزر استفاده شد.



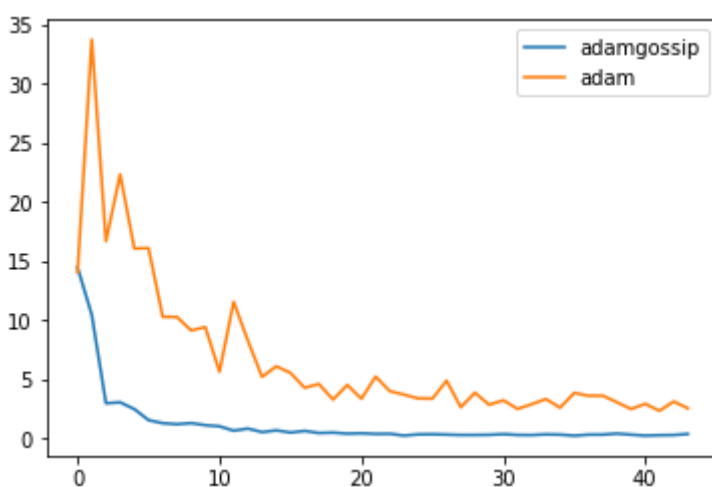
برای مقایسه با حالت سنترالایزد میانگین لاس هر سه ورکر و لاس حالت سنترالایزد را در شکل زیر مقایسه می کنیم. قابل دیدن است که روش گاسیپ آدام سریع تر به جواب رسیده.



Use gossip training in behavioral cloning for autonomous driving car

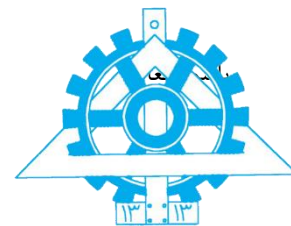


همچنین اگر هر نوع تصویر ورودی را به شبکه ای جداگانه داده و میانگین لاس ها را با حالت گاسیپ مقایسه کنیم حالت زیر بدست می آید. گاسیپ آدام در این حالت نیز از روش سنترالایز بهتر بوده.

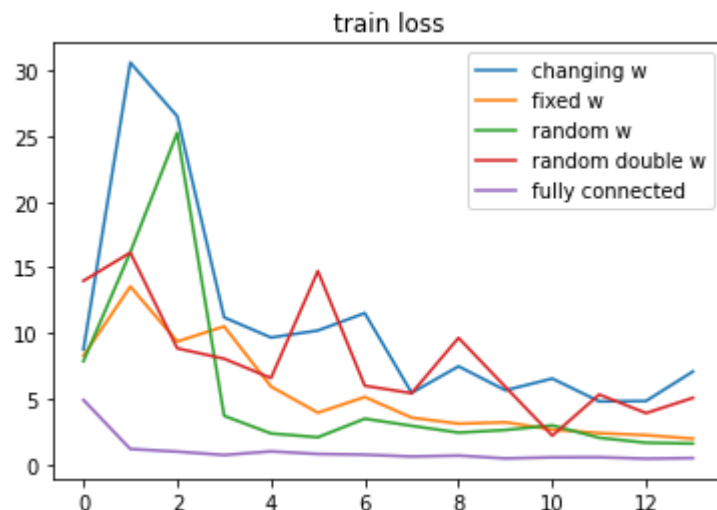


تأثیر وزن دهی مختلف

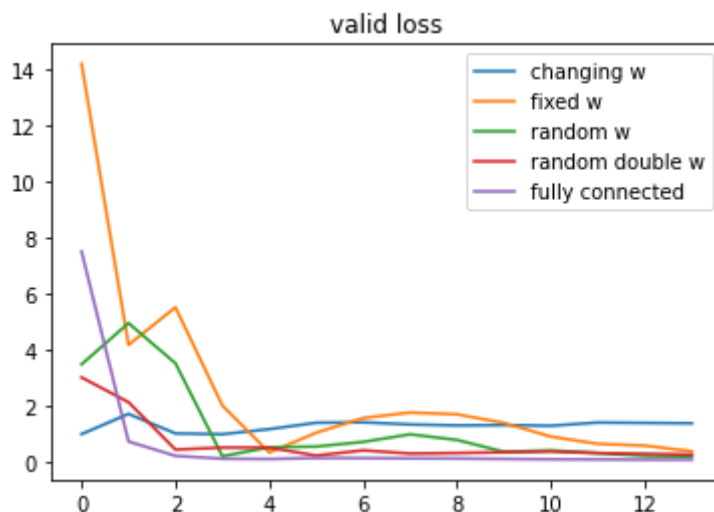
تاکنون طبق مقاله وزن ها با زمان تغییر می کردند. حال می خواهیم وزن ثابت در زمان، وزن دهی تصادفی و تصادفی دویل و حالت فولی کانتکند (دیگر گاسیپ محسوب نمیشود و تبادل همه به هم با وزن ثابت یک) را با حالت قبل مقایسه کنیم. طبق نمودار زیر برای این مسئله فولی کانتکند در لاس ترینینگ اندکی بهتر عمل کرده است. به ترتیب وزن تصادفی، وزن ثابت یک و وزن تصادفی دویل بهترین در لاس ترین بودند.



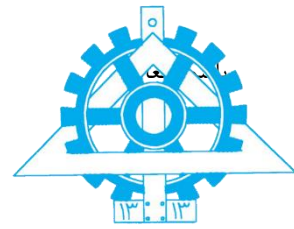
Use gossip training in behavioral cloning for autonomous driving car



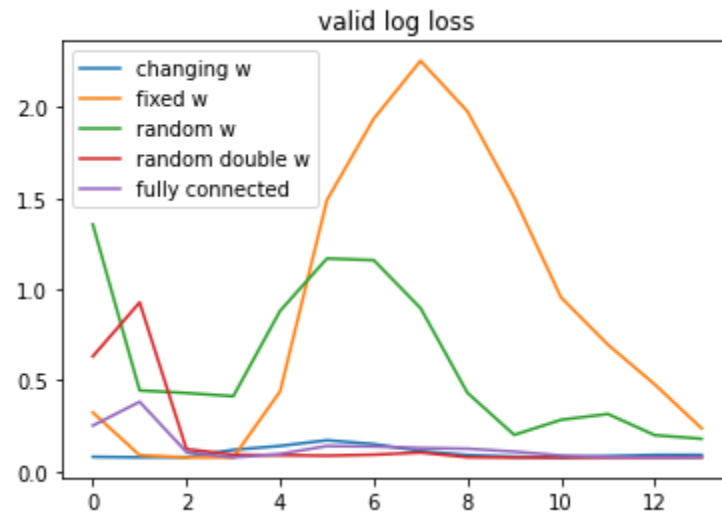
اما طبق شکل زیر لاس ولیدیشن تصادفی دوبل و فولی کانکتد بهترین بوده است. بعد از آن تصادفی و ثابت یک بهتر بوده اند.



در مقاله نوشته شده بود که یک شبکه در انتها با وزن میانگین وزن ها باید به خروجی برگردانیم. برای تست شبکه خروجی بعد هر ایتريشن یک شبکه تست با وزن میانگین ورکرها تولید و داده ولیدیشن را به آن می دهیم. در شکل زیر لاس ولیدیشن در حالت های مختلف وزن دهی حالت قبل را مشاهده می کنیم. در اینجا مشاهده می کنیم که لاس ولیدیشن روش مقاله و تصادفی دوبل و فولی کانکتد بهترین بوده اند.



Use gossip training in behavioral cloning for autonomous driving car



استفاده از الگوریتم ژنتیک برای یافتن بهترین وزن

کد روش ژنتیک برای یافتن بهترین مقدار وزن در فایل کد قرار داده شده. کروموزوم مقادیر α در مقاله و فیتنس فانکشن لاس ولیدیشن پس از یک ایتريشن است و مقادیر وزن بین ۰ تا ۳ در نظر گرفته شدند. اما برای بررسی بهتر می‌توان ماتریس وزن را نیز در نظر گرفت و تعداد دفعات تکرار بیشتری از ایتريشن‌ها گرفت و باید وزنی مانند ۰ تا ۱۰ و تعداد بیشتری ژن تولید بشه که به دلیل اینکه رم کولب پر میشه حالت ساده را در نظر گرفتیم.

||||||| 33.3% GA is running...1 gen from 3...best value = 0.08523444831371307