|  |  |  |
| --- | --- | --- |
|  | به نام خدا |  |
| **دانشگاه تهران**  **دانشکده‌ مهندسی برق و کامپیوتر**  **هوش مصنوعی**  **پروژه پنجم فاز دوم** | | |

|  |  |
| --- | --- |
| محمدعلی شاکردرگاه | نام و نام خانوادگی |
| 81019487 | شماره‌ دانشجویی |
| 06/04/1400 | تاریخ ارسال گزارش |

­

**فهرست گزارش**

[سوال 1 – فاز سوم: بخش اول 3](#_Toc32834855)

[سوال ۲ – امتیازی 6](#_Toc32834856)

# سوال 1 – optimizer

راه حل های مشکلات گرادیان کاهشی

1- Momentum:

در روش گرادیان کاهشی، برای بروزرسانی وزن ها به صورت بود

حال با روش Momentum به شکل زیر تغییر میکند.

در ابتدا عبارت جدیدی با نام Velocity(t) که نرخ رشد سرعت(مشتق تابع Loss) معرفی

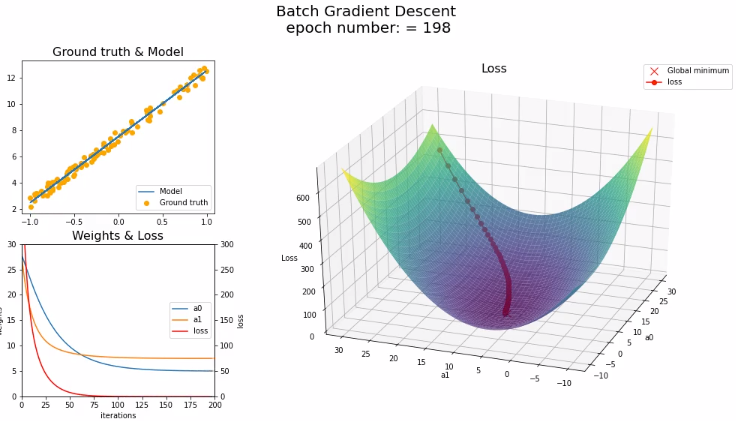
میشود و با فرمول زیر بروز رسانی میشود

و با استفاده از آن میتوان وزن را به شکل زیر به روز رسانی کرد:

این به ما تبیین میکند که با چه نرخی (قدرت و سرعتی) به چه سمتی حرکت کنیم که تا به نقطه بهینه میل کنیم و باعث میشود سریع تر به سمت نقطه بهینه میل کنیم.

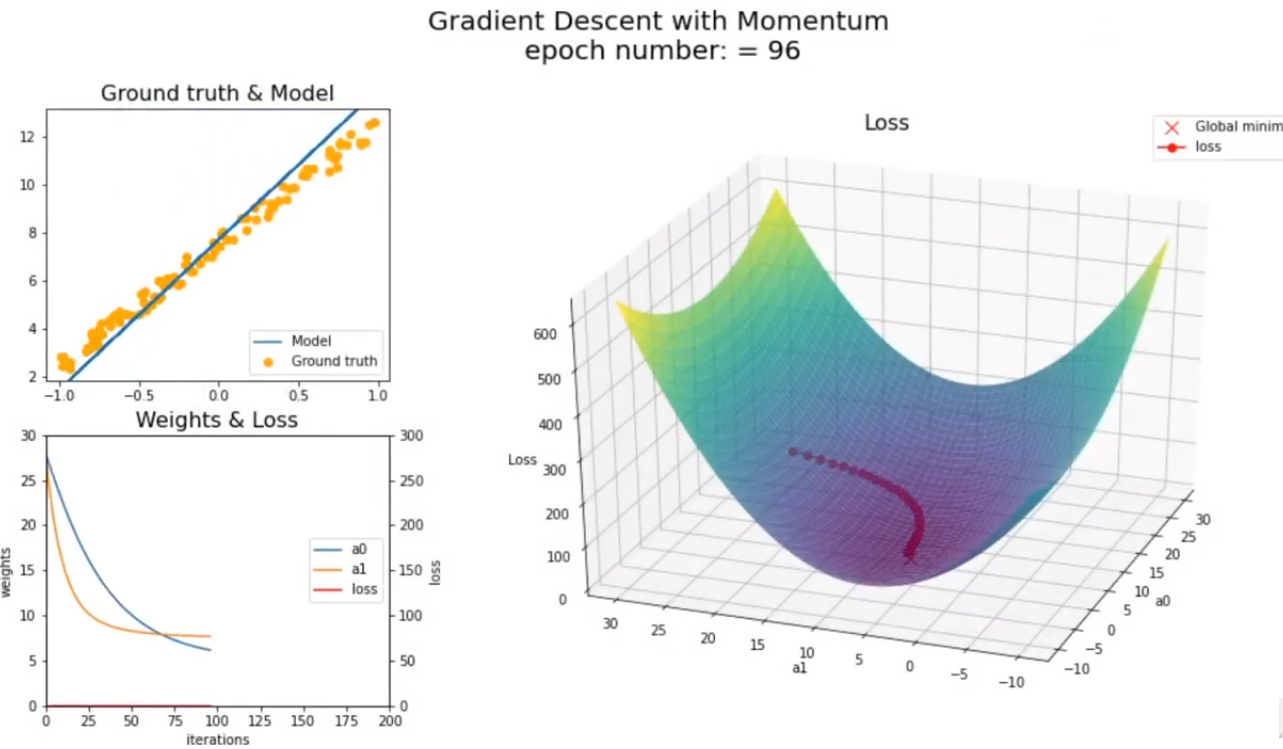
حال سعی میشود با استفاده از شبیه سازی این مساله تبیین شود.

* در شکل 1-1 با روش گرادیان کاهشی سعی شده به جواب optimal برسیم، مشاهده میشود بعد از 198 ایپاک Regression به دقت خوبی رسیده و مقدار تابع Loss کمینه شده است.



شکل 1-1 مینیمم کردن تابع Loss با گرادیان کاهشی

* در شکل 2-2 با روش گرادیان کاهشی با Momentum سعی شده به جواب optimal برسیم، مشاهده میشود بعد از 96 ایپاک Regression به دقت خوبی رسیده و مقدار تابع Loss کمینه شده است که دیده میشود **بسیار سریع تر شد و اثر Vanishing تقریبا از بین رفت**.

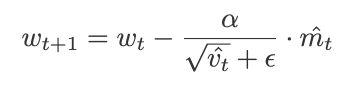


شکل 1-2 مینیمم کردن تابع Loss با گرادیان کاهشی با Momentum

2- Adam:

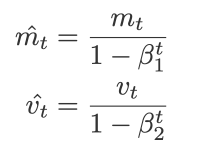
در عمل Adam optimization algorithm یکی از Extension های الگوریتم گرادیان کاهشی میباشد.

در این روش از دو مزیت AdaGrad و RMSProb استفاده میشود و Adam از میانگین ممان دوم گرادیان استفاده میکند و تغییرات وزن را به مانند زیر میکند:

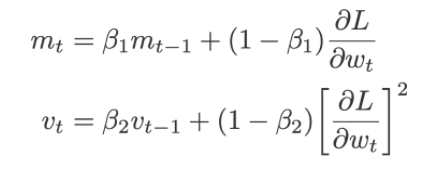


شکل 1-3 فرمول به روز رسانی Adam

که در آن پارامتر های بایاس و نرخ یادگیری به مانند زیر هستند:



1-4-1 4 پارامتر های بایاس و نرخ به روز رسانی Adam



شکل 1-4-2 پارامتر های بایاس و نرخ به روز رسانی Adam

روش بهینه سازی Adam در مسائل Non-Convex مزیت هایی دارد که در زیر به ترتیب آورده شده اند:

1. پیاده سازی مستقیم و Straight-forward دارد
2. به لحاظ محاسباتی بیهینه است
3. به لحاظ Memory Usage حافظه کمی اشغال میکند
4. در برابر Diagonal rescaling of the gradient تغییر ناپذیز است
5. برای مسائل با ابعاد بالا پرکاربرد میباشد.
6. برای مسائل Non-Stationary Objective مناسب میباشد
7. برای مسئله های Noisy بسیار مناسب میباشند

# سوال 2 – امتیازی

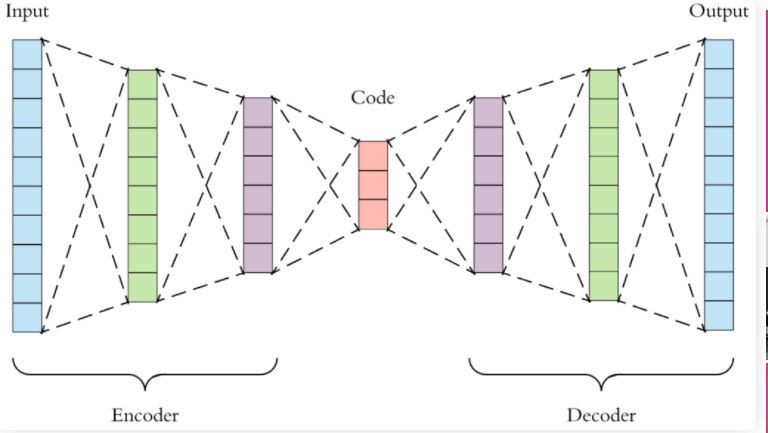
## ه: **Dimension reduction: Auto-Encoder**

در این سوال هدف آن است که با استفاده از Auto Encoder ابعاد داده های خود را کم کنیم.

دلیل این امر: داده های زیاد یعنی پردازشش زیاد و این امر به افزایش زمان برنامه ما می انجامد، پس سعی میکنیم از این الگوریتم استفاده کرده و ابعاد مسئله خود را کاهش داده تا نه تنها دقت خوبی داشته باشیم بلکه در زمان و فضا نیز صرفه جویی کرده باشیم

بهترین پیاده سازی خود را که 4 لایه ای بود را انتخاب میکنیم و روی آن Auto Encoder میزنیم.

نکته مهم، تفهیم موضوع Auto Encoder است که مانند شکل زیر عمل میکنند:



شکل 3و 1 ساختار یک AutoEncoder

عملیات Auto Encoder به این صورت است که یک قسمت انکودر دارد و یک قسمت دیکودر دارد و سعی میکند لایه به لایه ابعاد را کوچکتر کند و ابعاد کوچکتر را دیکود کند، تا ببیند با داده های اصلی Correlation به چه صورت است.اگر Loss آن قابل قبول بود، از آن تعداد ابعاد استفاده میشود.