



## Assignment NO.5 Solutions

Deep learning | Spring 1401 | Dr.Mohammadi

Teacher Assistant:

Shabnam Ezatzadeh

Fatemeh Anvari

---

Student name : **Amin Fathi**

Student id : **400722102**

## Problem1

در این مقاله نگارندن علاوه بر دو روش معروف قبلی برای بهبود sgd یعنی ۱- روش های استفاده از نرخ یادگیری تطبیقی (adaptive learning rate) مانند Adam و AdaGrad ، و ۲- روش های استفاده از سرعت مانند Nesterov Momentum ؛ روش جدیدی برای بهینه سازی sgd ارایه میدهند به نام Lookahead که در نهایت هم نشان می دهند سر بار محاسبات و حافظه را کاهش می دهد و فرایند یادگیری پایدار تر و با واریانس کمتری را هم به ارمغان می آورد .

در این روش ما با دو مجموعه وزن رو به رو هستیم ، وزن های سریع یا (فی) و وزن های کند یا همان (تتا) ، در این روش با استفاده از اپتیمایزر های استاندارد ابتدا مجموعه وزن های سریع آپدیت می شوند و سپس در هر  $k$  سری آپدیت این مجموعه وزن های سریع ، مقدار نهایی آن ها به عنوان مقدار جدید برای مجموعه وزن های کند ( توسط یک تطبیق خطی بین این دو سری وزن ) انتخاب خواهد شد . الگوریتم را در شکل زیر مشاهده می کنید که در آن آلفا یک نرخ یادگیری برای وزن های کند است :

---

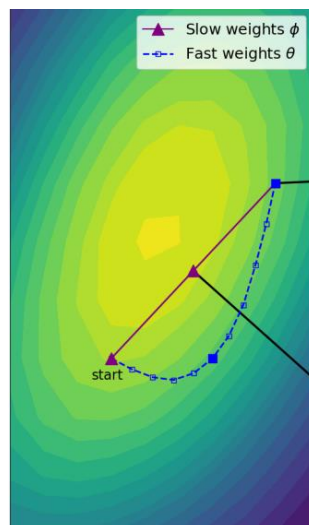
### Algorithm 1 Lookahead Optimizer:

---

**Require:** Initial parameters  $\phi_0$ , objective function  $L$   
**Require:** Synchronization period  $k$ , slow weights step size  $\alpha$ , optimizer  $A$   
**for**  $t = 1, 2, \dots$  **do**  
    Synchronize parameters  $\theta_{t,0} \leftarrow \phi_{t-1}$   
    **for**  $i = 1, 2, \dots, k$  **do**  
        sample minibatch of data  $d \sim \mathcal{D}$   
         $\theta_{t,i} \leftarrow \theta_{t,i-1} + A(L, \theta_{t,i-1}, d)$   
    **end for**  
    Perform outer update  $\phi_t \leftarrow \phi_{t-1} + \alpha(\theta_{t,k} - \phi_{t-1})$   
**end for**  
**return** parameters  $\phi$

---

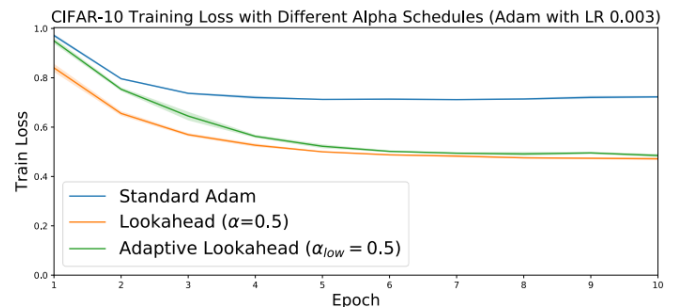
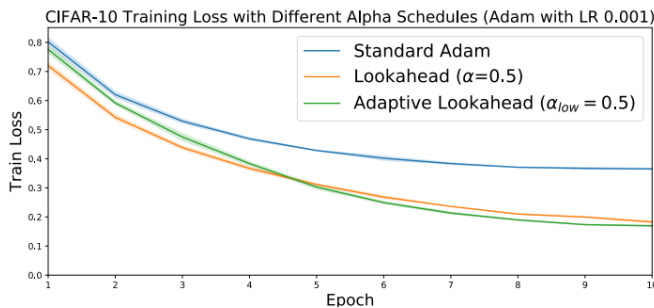
که در شکل زیر هم به صورت شهودی مشاهده می کنید



طبق آنچه در شبه کد این الگوریتم آمده و در شکل صفحه قبل هم مشاهده شد در واقع وزن های کند در به روز رسانی شان در هر حلقه داخلی که شامل  $k$  بار اجرای فرایند آپدیت کردن وزن های سریع است آپدیت می شوند و فرمول آن به شکل زیر است :

$$\begin{aligned}\phi_{t+1} &= \phi_t + \alpha(\theta_{t,k} - \phi_t) \\ &= \alpha[\theta_{t,k} + (1 - \alpha)\theta_{t-1,k} + \dots + (1 - \alpha)^{t-1}\theta_{0,k}] + (1 - \alpha)^t\phi_0\end{aligned}$$

نگارندان مقاله ادعا کرده اند انتخاب مقدار آلفا به صورت فیکس شده از اکثر مزایای انتخاب تطبیقی آن برخوردار است فلذا مقدار آن را در روند نگارش مقاله ثابت در نظر گرفته اند . در مورد تاثیر انتخاب آلفا در فرمول بالا می توان نتایج زیر را هم مورد بررسی قرار داد :

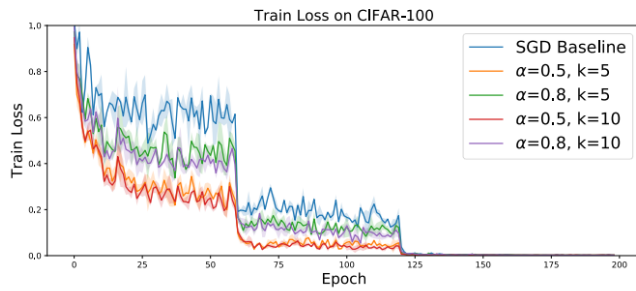


وزن های سریع هم طبق الگوریتم زیر به روز رسانی می شوند :

$$\theta_{t,i+1} = \theta_{t,i} + A(L, \theta_{t,i-1}, d).$$

که در آن  $A$  اپتیمایزری است که برای به روز رسانی انتخاب شده است ، و  $L$  تابع ضرر یا objective function و  $d$  هم تعداد نمونه ها در mini batch

در ادامه ، نگارندگان این الگوریتم را در مورد مسایل CIFAR10 و CIFAR100 و ImageNet و پردازش زبان طبیعی به کار برده اند و نتایج آن را ثبت و تحلیل کرده اند . به طور مثلاً مقدار loss و accuracy برای آلفا ها  $k$  های متفاوت در CIFAR-100 را در شکل زیر مشاهده می کنید :



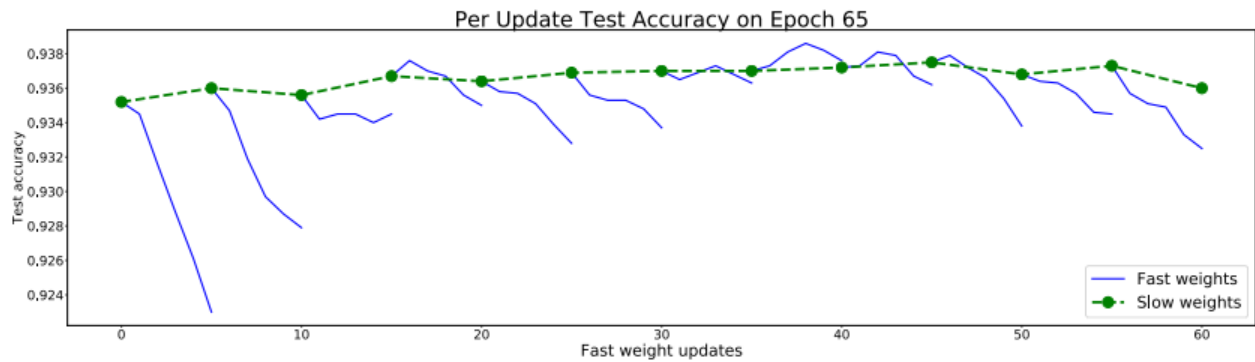
$\alpha \backslash k$	0.5	0.8
5	$78.24 \pm .02$	$78.27 \pm .04$
10	$78.19 \pm .22$	$77.94 \pm .22$

Table 5: All settings have higher validation accuracy than SGD (77.72%)

Figure 9: CIFAR-100 train loss and final test accuracy with various  $k$  and  $\alpha$ .

که نشان دهنده نتایج نسبتاً مشابه و مقاوم بودن این الگوریتم نسبت به آلفا ها و  $k$  های متفاوت است .

از مزایای این الگوریتم توانایی تطبیقش با اپتیمایزر های مختلف است ، و از آن جا که واریانس را کاهش می دهد موجب همگرایی سریع تر می شود و نتایج قابل قبولی هم ارایه می دهد که در صفحه قبل مشاهده کردیم .



در تصویر بالا هم مشاهده می شود که در به روز رسانی های مربوط به وزن های سریع چطور عملکرد مدل تخریب می شود و این تخریب توسط به روز رسانی وزن های کند کاور می شود و واریانس پایینی را مدل تجربه می کند .

## Problem 2

مشکلی که با نرخ آموزش داریم اندازه ان است چنانچه که کوچک باشد با سرعت کمی به همگرایی می رود و چنانچه بزرگ باشد ممکن است واگرا شود و نقطه بهینه را هم رد کند بنابراین بعضی مواقع بهتر است از روش های دیگری استفاده کنیم و بسط تیلور را ادامه دهیم تا به انحنای منحنی هم توجه کرده باشیم .

چنانچه بسط تیلور را تا مشتق دوم ادامه دهیم به رابطه زیر می رسم :

$$f(\mathbf{x} + \epsilon) = f(\mathbf{x}) + \epsilon^T \nabla f(\mathbf{x}) + \frac{1}{2} \epsilon^T \nabla^2 f(\mathbf{x}) \epsilon + \mathcal{O}(\|\epsilon\|^3).$$

مشتق دوم را جهت آسانتر شدن خوانش فرمول ماتریس هسین یا  $H$  می نامیم که یک ماتریس  $d \times d$  است (  $d$  همان ابعاد فضای ویژگی تابع است ) فرمول حاصل به شکل زیر است :

$$f(\mathbf{x} + \epsilon) = f(\mathbf{x}) + \epsilon^T \nabla f(\mathbf{x}) + \frac{1}{2} \epsilon^T \mathbf{H} \epsilon + \mathcal{O}(\|\epsilon\|^3)$$

برای یافتن مقدار بهینه این تابع تقریبی ( تقریب مرتبه دو دارایه بهینه سراسری است ) ، می توان مشتق آن نسبت به  $\epsilon$  را برابر با صفر قرار داد که در این صورت داریم :

$$\nabla f(\mathbf{x}) + \mathbf{H} \epsilon = 0 \Rightarrow \epsilon = -\mathbf{H}^{-1} \nabla f(\mathbf{x})$$

همانطور که مشاهده می شود بجای نرخ آموزش ( در هنگام استفاده از فقط ترم اول بسط تیلور ) از معکوس ماتریس Hessian استفاده شده است ، چون در واقع داریم از یک ماتریس استفاده می کنیم ممکن است نرخ آموزش های متفاوتی در جهت های مختلف داشته باشیم ، همچنین استفاده از این ماتریس باعث می شود که دیگر مجبور به تنظیم و ثابت سازی نرخ یادگیری

نباشیم . روش نیوتن از آنجا که با یک ماتریس  $d*d$  کار می کند ممکن است با افزایش  $d$  حجم محاسبات بسیار بیشتر شود و همچنین با توجه به حضور مشتق دوم در فرمول ، چنانچه مقدار آن منفی باشد موجب افزایش مقدار loss می شود و نه کاهش آن که برای حل آن می توان از قدر مطلق تابع تقریب استفاده کرد .

منبع : صحبت های کلاس درس

### Problem 3

در حالت with replacement احتمال انتخاب یک داده از مجموعه داده  $n$  عضوری برای train مدل به صورت تصادفی  $1/n$  است ، احتمال انتخاب یک داده منحصر به فرد حداقل برای یک بار برابر است با :

$$P(\text{choose } i) = 1 - P(\text{omit } i) = 1 - (1 - 1/n)^n \approx 1 - e^{-1} \approx 0.63.$$

احتمال انتخاب یک داده دقیقا یک بار هم برابر است با :

$$\binom{n}{1} \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} = \frac{n}{n-1} \left(1 - \frac{1}{n}\right)^n \approx e^{-1} \approx 0.37.$$

همانطور که مشاهده می شود در این حالت همه داده ها لزوما انتخاب نمی شوند و چه بسا برخی داده ها هم بیش از یک بار انتخاب شوند که این باعث کاهش کارایی مدل می شود ولی در حال without با توجه به این که همه داده ها استفاده می شوند میتوان عملکرد بهتری را انتظار داشت .

منابع :

صفحه ۴۶۶ کتاب رفرنس

[\[1903.01463\] SGD without Replacement: Sharper Rates for General Smooth Convex Functions \(arxiv.org\)](https://arxiv.org/abs/1903.01463)

#### Problem 4

پیاده سازی این تمرین با توجه به فرمول های مشخص و توضیح داده شده در کلاس و نمونه کد های موجود در اینترنت چندان کار سختی نبود ، صرفا تنها جای مبهم آن محاسبه گرادیان نسبت به وزن ها بود که در شکل زیر توضیح داده شده است و در نهایت هم در جداول نتایج مقایسه شده است . a در اینجا همان خروجی سیگموئید است .

$$\alpha(z) = \frac{1}{1+e^{-z}} \quad \left. \begin{array}{l} L(\alpha, y) = -y \log(\alpha) \\ - (1-y) \log(1-\alpha) \end{array} \right\} \quad z = wx + b$$

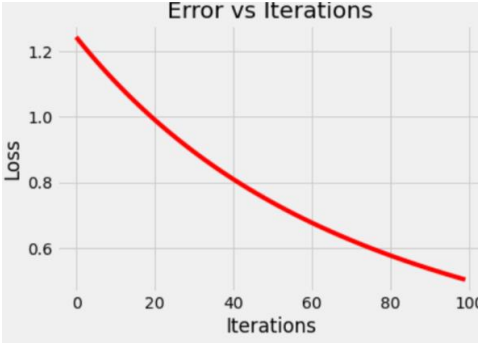
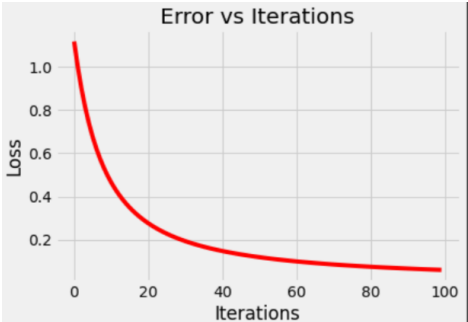
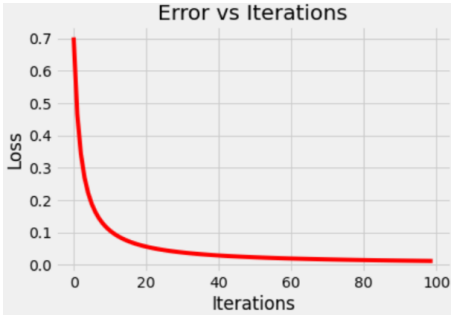

$$\Rightarrow \frac{\partial L}{\partial w} = \frac{\partial L}{\partial \alpha} \frac{\partial \alpha}{\partial z} \frac{\partial z}{\partial w}$$

$$= -\frac{y}{\alpha} \times \alpha(1-\alpha) \times x$$

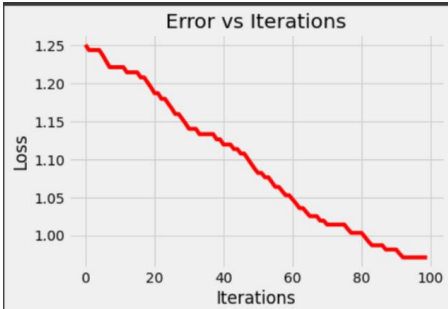
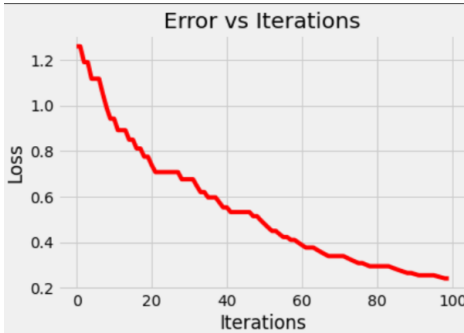
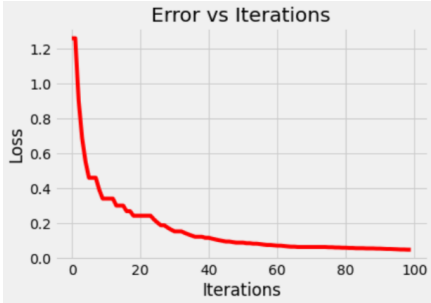
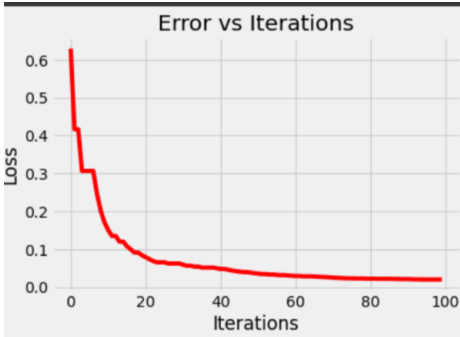
$$= -y(1-\alpha)x$$

: GD

در جدول زیر مشاهده می شود که کمترین خطا و بهترین همگرایی مربوط به نرخ یادگیری ۱ می باشد و بدترین مربوط به نرخ یادگیری 0.01



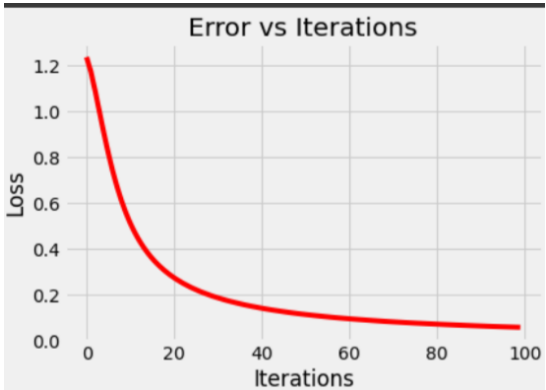
LOSS VS ITERATION	LOSS	LEARNING RATE
 <p>Error vs Iterations</p> <p>Loss</p> <p>Iterations</p>	0.5	0.01
 <p>Error vs Iterations</p> <p>Loss</p> <p>Iterations</p>	0.06	0.1
 <p>Error vs Iterations</p> <p>Loss</p> <p>Iterations</p>	0.012	0.5
 <p>Error vs Iterations</p> <p>Loss</p> <p>Iterations</p>	0.006	1


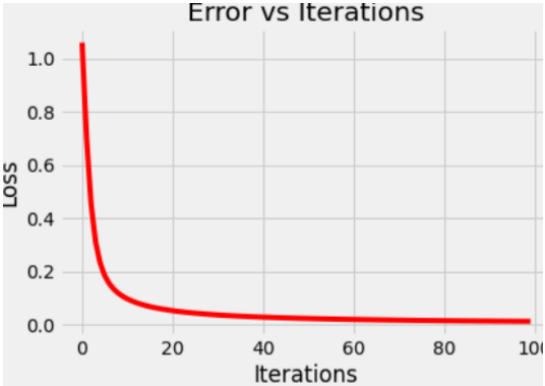
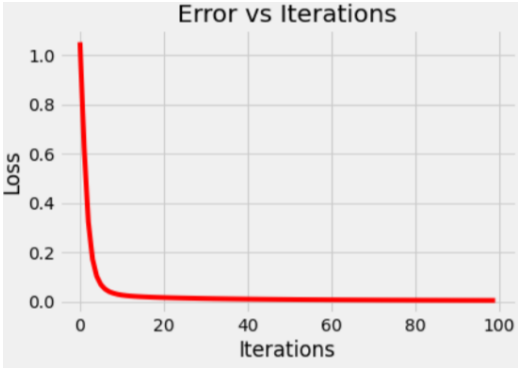
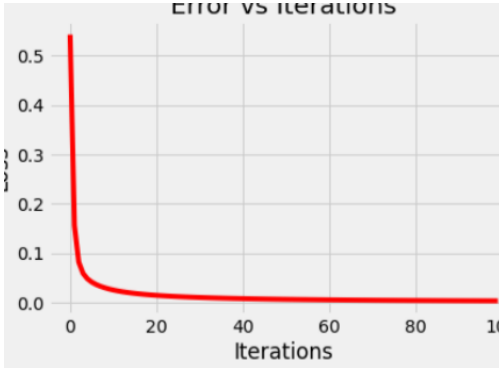



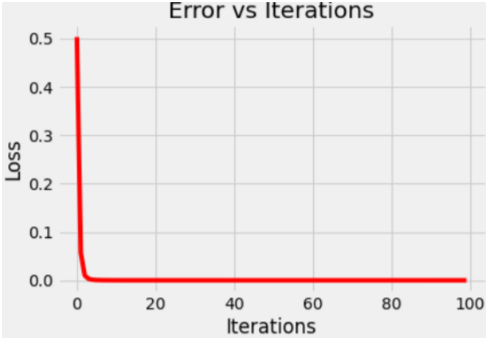


LOSS VS ITERATION	LOSS	LEARNING RATE
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 1.00 to 1.25. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 1.25 and decreases steadily to about 0.97 by iteration 100.</p>	0.97	0.01
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.2 to 1.2. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 1.2 and decreases to about 0.24 by iteration 100.</p>	0.24	0.1
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.2. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 1.2 and decreases rapidly to about 0.046 by iteration 100.</p>	0.046	0.5
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 0.6. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 0.6 and decreases rapidly to about 0.019 by iteration 100.</p>	0.019	1

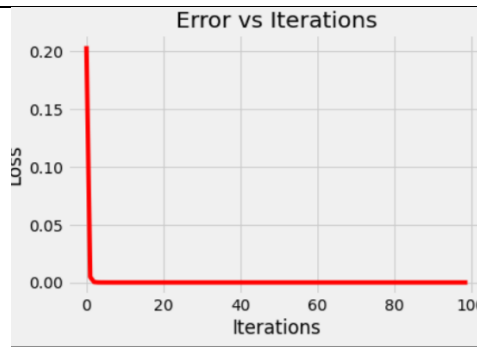
مشاهده می شود که در SGD عملکرد با لرنینگ ریت یکسان نسبت به GD ضعیف تر است ولی همچنان نرخ یادگیری ۱ بهترین و نرخ یادگیری 0.01 بدترین عملکرد را بین ۴ نرخ یادگیری دارد .

### GD + Momentum

Loss vs iteration	loss	beta	Learning rate
	0.18	0.7	0.01
	0.12	0.8	0.01
	0.059	0.9	0.01

 <p>The plot shows a red line representing the error over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.0. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The error starts at approximately 1.05 at iteration 0 and decreases rapidly, reaching near zero by iteration 20, and remains stable until iteration 100.</p>	0.017	0.7	0.1
 <p>The plot shows a red line representing the error over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.0. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The error starts at approximately 1.05 at iteration 0 and decreases rapidly, reaching near zero by iteration 20, and remains stable until iteration 100.</p>	0.011	0.8	0.1
 <p>The plot shows a red line representing the error over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.0. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The error starts at approximately 1.05 at iteration 0 and decreases rapidly, reaching near zero by iteration 10, and remains stable until iteration 100.</p>	0.005	0.9	0.1
 <p>The plot shows a red line representing the error over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 0.5. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The error starts at approximately 0.55 at iteration 0 and decreases rapidly, reaching near zero by iteration 20, and remains stable until iteration 100.</p>	0.003	0.7	0.5

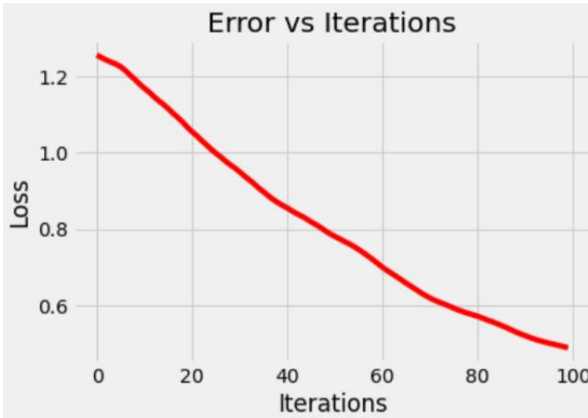
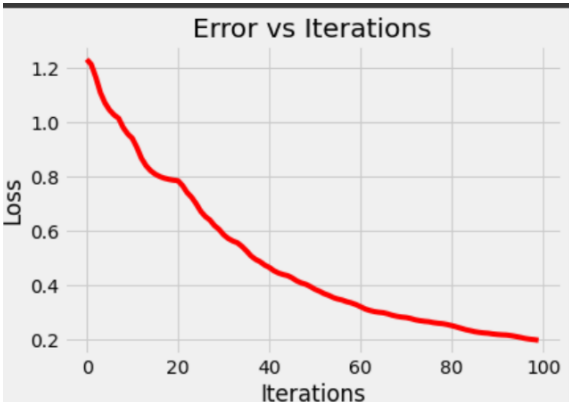
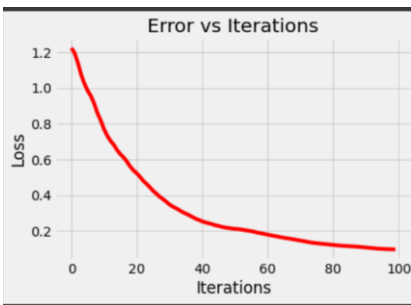
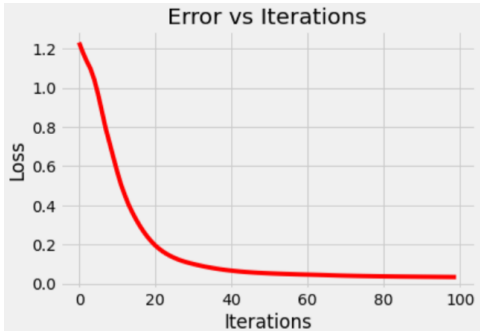
	0.002	0.8	0.5
	0.0002	0.9	0.5
	0.0016	0.7	1
	0.0007	0.8	1
	0 تقريبا	0.9	1

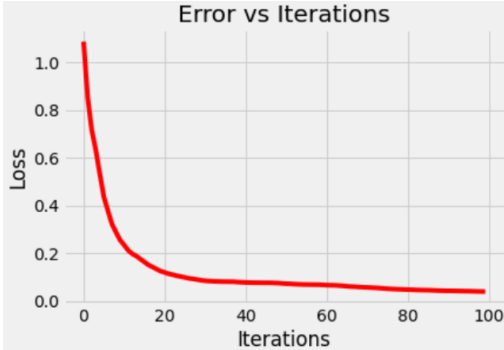
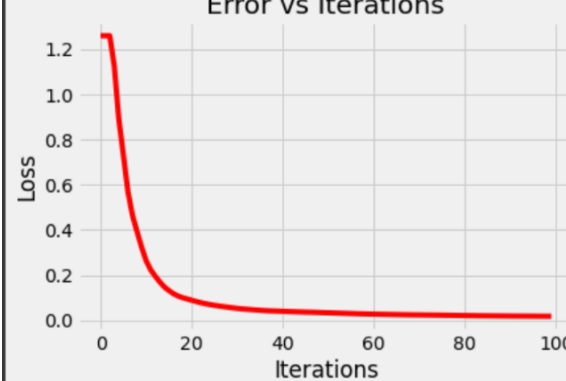

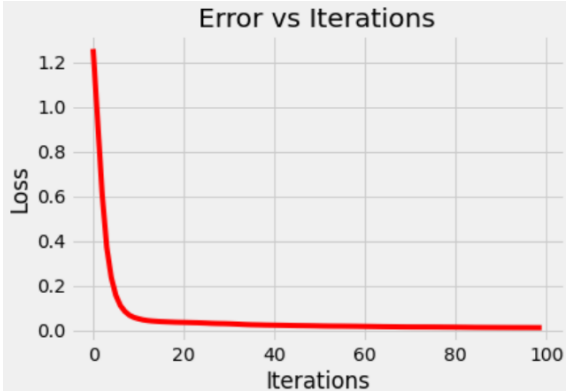



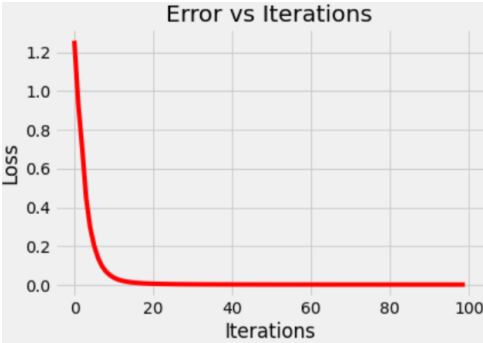
همانطور که مشاهده می شود بهترین عملکرد مربوط به نرخ یادگیری ۱ و بتای 0.9 و بدترین هم مربوط به نرخ یادگیری 0.01 و 0.7، همچنین عملکرد gd+momentum از gd عملکرد بهتری دارد.

### SGD + Momentum

Loss vs iteration	loss	beta	Learning rate
	0.87	0.7	0.01
	0.73	0.8	0.01

 <p>The graph shows a smooth, gradual decrease in loss from approximately 1.25 at iteration 0 to about 0.5 at iteration 100. The curve is continuous and has a consistent downward slope.</p> <table><tr><th>Iterations</th><th>Loss</th></tr><tr><td>0</td><td>1.25</td></tr><tr><td>20</td><td>1.05</td></tr><tr><td>40</td><td>0.85</td></tr><tr><td>60</td><td>0.70</td></tr><tr><td>80</td><td>0.60</td></tr><tr><td>100</td><td>0.50</td></tr></table>	Iterations	Loss	0	1.25	20	1.05	40	0.85	60	0.70	80	0.60	100	0.50	0.049	0.9	0.01
Iterations	Loss																
0	1.25																
20	1.05																
40	0.85																
60	0.70																
80	0.60																
100	0.50																
 <p>The graph shows a smooth decrease in loss from approximately 1.25 at iteration 0 to about 0.2 at iteration 100. The curve is continuous and has a consistent downward slope.</p> <table><tr><th>Iterations</th><th>Loss</th></tr><tr><td>0</td><td>1.25</td></tr><tr><td>20</td><td>0.80</td></tr><tr><td>40</td><td>0.45</td></tr><tr><td>60</td><td>0.30</td></tr><tr><td>80</td><td>0.25</td></tr><tr><td>100</td><td>0.20</td></tr></table>	Iterations	Loss	0	1.25	20	0.80	40	0.45	60	0.30	80	0.25	100	0.20	0.19	0.7	0.1
Iterations	Loss																
0	1.25																
20	0.80																
40	0.45																
60	0.30																
80	0.25																
100	0.20																
 <p>The graph shows a smooth decrease in loss from approximately 1.25 at iteration 0 to about 0.1 at iteration 100. The curve is continuous and has a consistent downward slope.</p> <table><tr><th>Iterations</th><th>Loss</th></tr><tr><td>0</td><td>1.25</td></tr><tr><td>20</td><td>0.50</td></tr><tr><td>40</td><td>0.25</td></tr><tr><td>60</td><td>0.15</td></tr><tr><td>80</td><td>0.10</td></tr><tr><td>100</td><td>0.05</td></tr></table>	Iterations	Loss	0	1.25	20	0.50	40	0.25	60	0.15	80	0.10	100	0.05	0.097	0.8	0.1
Iterations	Loss																
0	1.25																
20	0.50																
40	0.25																
60	0.15																
80	0.10																
100	0.05																
 <p>The graph shows a smooth decrease in loss from approximately 1.25 at iteration 0 to about 0.05 at iteration 100. The curve is continuous and has a consistent downward slope.</p> <table><tr><th>Iterations</th><th>Loss</th></tr><tr><td>0</td><td>1.25</td></tr><tr><td>20</td><td>0.20</td></tr><tr><td>40</td><td>0.08</td></tr><tr><td>60</td><td>0.05</td></tr><tr><td>80</td><td>0.04</td></tr><tr><td>100</td><td>0.03</td></tr></table>	Iterations	Loss	0	1.25	20	0.20	40	0.08	60	0.05	80	0.04	100	0.03	0.033	0.9	0.1
Iterations	Loss																
0	1.25																
20	0.20																
40	0.08																
60	0.05																
80	0.04																
100	0.03																


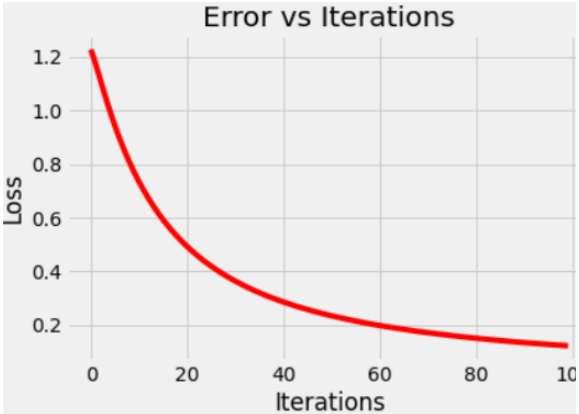

	0.04	0.7	0.5
	0.017	0.8	0.5
	0.004	0.9	0.5
	0.011	0.7	1




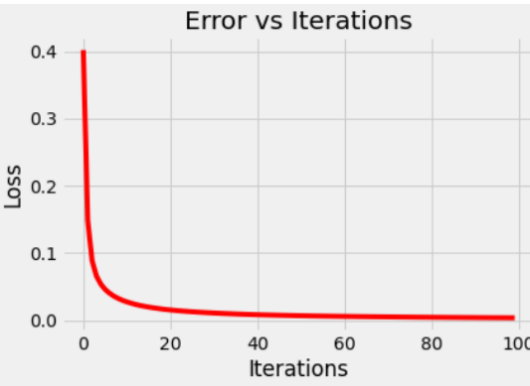
	0.0065	0.8	1
	0.0018	0.9	1


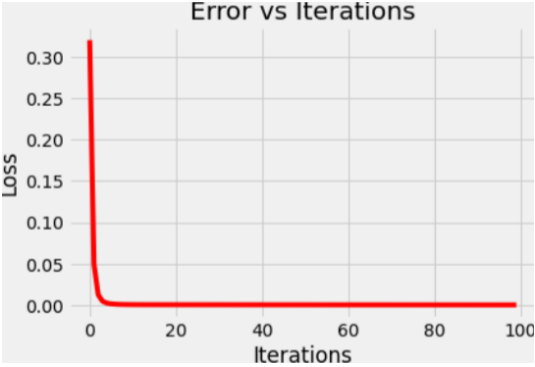

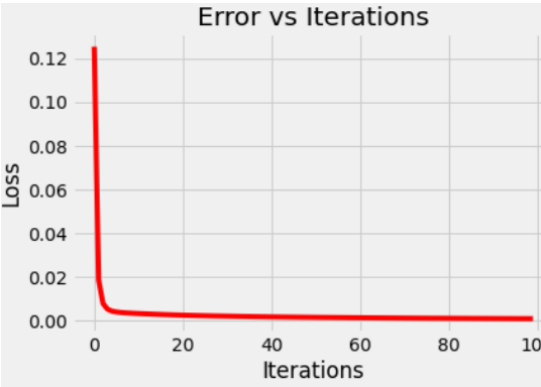
مشاهده می شود که  $gd + momentum$  عملکرد بهتری نسبت به  $sgd + momentum$  دارد ولی  $sgd + momentum$  نسبت به  $sgd$  عملکرد بهتری دارد و بهترین نرخ یادگیری هم برابر ۱ و بهترین بتا برابر 0.9 است.

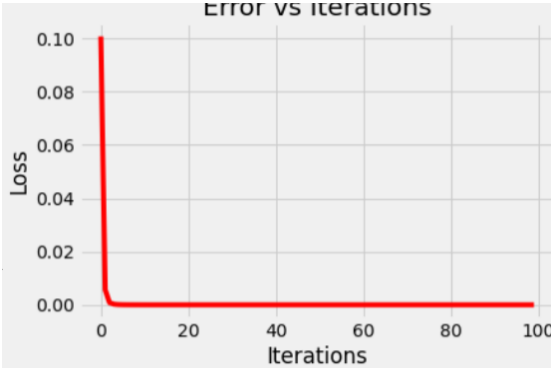


GD + Nesterov Momentum

Loss vs iteration	loss	beta	Learning rate
	0.18	0.7	0.01
	0.12	0.8	0.01
	0.058	0.9	0.01



 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.0. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at 1.0 and decreases rapidly, reaching approximately 0.1 by iteration 20, and then continues to decrease slowly towards 0.0 by iteration 100.</p>	0.017	0.7	0.1
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.0. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at 1.0 and decreases rapidly, reaching approximately 0.1 by iteration 20, and then continues to decrease slowly towards 0.0 by iteration 100.</p>	0.011	0.8	0.1
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.0. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at 1.0 and decreases rapidly, reaching approximately 0.1 by iteration 20, and then continues to decrease slowly towards 0.0 by iteration 100.</p>	0.005	0.9	0.1
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 0.4. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at 0.4 and decreases rapidly, reaching approximately 0.1 by iteration 20, and then continues to decrease slowly towards 0.0 by iteration 100.</p>	0.003	0.7	0.5





 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis (Loss) ranges from 0.00 to 0.35. The x-axis (Iterations) ranges from 0 to 100. The loss starts at approximately 0.35 at iteration 0 and drops sharply to near 0.00 by iteration 10, remaining stable thereafter.</p>	0.002	0.8	0.5
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis (Loss) ranges from 0.00 to 0.30. The x-axis (Iterations) ranges from 0 to 100. The loss starts at approximately 0.30 at iteration 0 and drops sharply to near 0.00 by iteration 10, remaining stable thereafter.</p>	0.0004	0.9	0.5
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis (Loss) ranges from 0.000 to 0.150. The x-axis (Iterations) ranges from 0 to 100. The loss starts at approximately 0.150 at iteration 0 and drops sharply to near 0.000 by iteration 10, remaining stable thereafter.</p>	0.0016	0.7	1
 <p>The plot shows a red line representing the loss over 100 iterations. The y-axis (Loss) ranges from 0.00 to 0.12. The x-axis (Iterations) ranges from 0 to 100. The loss starts at approximately 0.12 at iteration 0 and drops sharply to near 0.00 by iteration 10, remaining stable thereafter.</p>	0.001	0.8	1

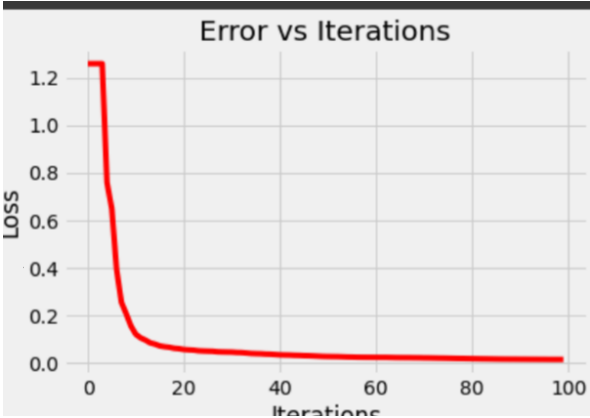



	تقریباً 0	0.9	1
---	-----------	-----	---

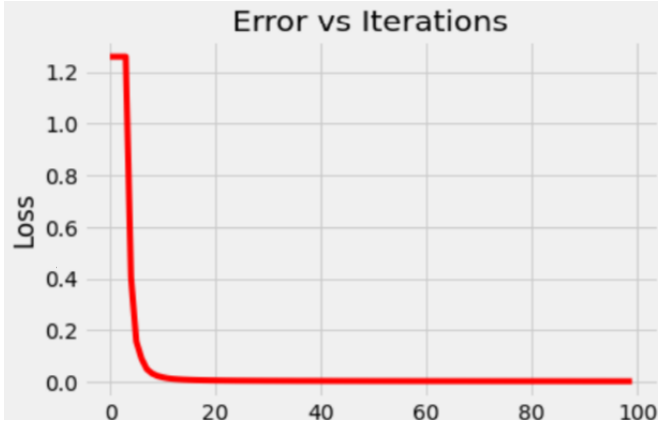
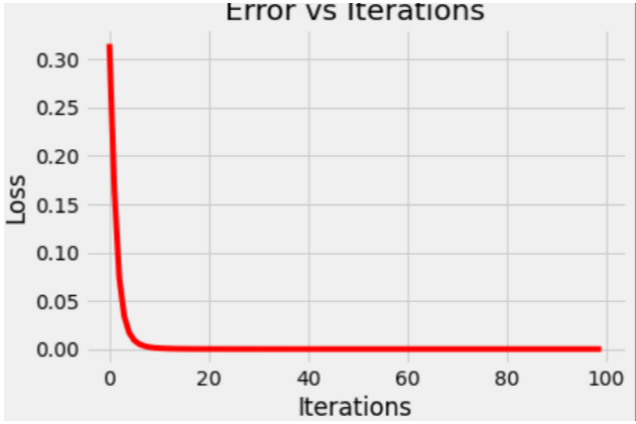
بهترین عملکرد مربوط به نرخ یادگیری ۱ و بتای 0.9 است و در کل با کمی اغماض می توان گفت Nesterov momentum از momentum بیشتر است .

### SGD + Nesterov Momentum

Loss vs iteration	loss	beta	Learning rate
	0.6	0.7	0.01
	0.4	0.8	0.01

 <p>This graph shows the training loss over 100 iterations. The loss starts at approximately 1.25 and decreases steadily, reaching about 0.25 by iteration 100. The curve is smooth and shows a consistent rate of decrease.</p>	0.21	0.9	0.01
 <p>This graph shows the training loss over 100 iterations. The loss starts at approximately 1.25 and decreases rapidly at first, then more slowly, reaching about 0.1 by iteration 100. The curve is smooth and shows a consistent rate of decrease.</p>	0.66	0.7	0.1
 <p>This graph shows the training loss over 100 iterations. The loss starts at approximately 1.25 and decreases rapidly at first, then more slowly, reaching about 0.05 by iteration 100. The curve is smooth and shows a consistent rate of decrease.</p>	0.047	0.8	0.1
 <p>This graph shows the training loss over 100 iterations. The loss starts at approximately 1.0 and decreases rapidly at first, then more slowly, reaching about 0.0 by iteration 100. The curve is smooth and shows a consistent rate of decrease.</p>	0.018	0.9	0.1

 <p>The graph shows a red line representing loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.2. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 1.25 at iteration 0 and drops sharply to about 0.1 by iteration 10, then continues to decrease slowly, reaching 0.0 by iteration 100.</p>	0.0137	0.7	0.5
 <p>The graph shows a red line representing loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 0.7. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 0.65 at iteration 0 and drops sharply to about 0.05 by iteration 10, then continues to decrease slowly, reaching 0.0 by iteration 100.</p>	0.009	0.8	0.5
 <p>The graph shows a red line representing loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 0.6. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 0.65 at iteration 0 and drops sharply to about 0.02 by iteration 10, then continues to decrease slowly, reaching 0.0 by iteration 100.</p>	0.0009	0.9	0.5
 <p>The graph shows a red line representing loss over 100 iterations. The y-axis is labeled 'Loss' and ranges from 0.0 to 1.2. The x-axis is labeled 'Iterations' and ranges from 0 to 100. The loss starts at approximately 1.25 at iteration 0 and drops sharply to about 0.05 by iteration 10, then continues to decrease slowly, reaching 0.0 by iteration 100.</p>	0.0059	0.7	1

	0.0025	0.8	1
	تقریباً 0	0.9	1

اینبار هم مشاهده می شود که بهترین نرخ یادگیری 1 و بهترین بتا 0.9 است .

منبع :