CSE 543: Homework I

Corbin Charpentier (Student ID: TODO)

Q1

We have $g:[0,1]\rightarrow \mathbb{R}$ and is twice-differentiable with $g(0)=g'(0)=0$.

Q1.1

By the Fundamental Theorem of Calculus, we have:

$$ \begin{aligned} g'(x) &= g'(0)+\int_0^x g''(b)db \ g(x) &= g(0) + \int_0^x \left (g'(0)+\int_0^x g''(b)db \right)db \

\end{aligned} $$

Now plugging in what was given

$$ \begin{align} g(x) &= 0 + \int_0^1 \left (0 + \int_0^1 \sigma(x-b) g''(b)db \right)db \ &= \int_0^1 \int_0^1 \sigma(x-b) g''(b)db db \ &= \int_0^1 \sigma(x-b) g''(b)db \ \end{align} $$ Step (3) gets us back to the original proposition.

Q1.2

Suppose $\left|g^{\prime \prime}\right| \leq \beta$ over $[0,1]$ for some $\beta>0$, and let $\epsilon>0$ be given. Prove that there exists a ReLU network $f(x) \triangleq \sum_{i=1}^{m} a_{i} \sigma\left(x-b_{i}\right)$ with $m \leq\left\lceil\frac{\beta}{\epsilon}\right\rceil$ and $|f-g|_{\infty} \leq \epsilon$

g' is by definition β-Lipschitz. (Wikipedia)

By 1D Approximation theorem in the lecture notes, a 2-layer neural network f exists with the following attributes:

The neural network as $\left\lceil\frac{\beta}{\epsilon}\right\rceil$ nodes

The neural network can be expressed with: $f(x)=\sum_{i=1}^{m} a_{i} \mathbb{1}*{x-x*{i} \geq 0}$

$\left|f-g^{\prime}\right|_{\infty}\leq\epsilon$

We let

$x_{i} \triangleq \frac{(i-1) \epsilon}{\beta}$

$m \triangleq\left\lceil\frac{\beta}{\epsilon}\right\rceil$

$a_{i} \triangleq |g^{\prime}\left(x_{i}\right)-g^{\prime}\left(x_{i-1}\right)|$

Now

||$f-g^{\prime} |_{\infty} \leq \epsilon \implies \left|f(x)-g^{\prime}(x)\right| \leq \epsilon$ (1D proof)

Therefore

$||\int(f-g^{\prime}) |_{\infty} \leq \epsilon \implies \left|\int (f(x)-g^{\prime}(x))\right| \leq \epsilon$

Which implies $f(x)=x\sum_{i=1}^{m} a_{i} \mathbb{1}{x-x{i} \geq 0} = \sum_{i=1}^{m} a_{i} \sigma\left(x-b_{i}\right)$ for all $b_i=0$

Since $\sigma(x) = x * threshold(x)$

$$ \begin{aligned} \end{aligned} $$

Q2

Suppose $f: \mathbb{R}^{d} \rightarrow R$ is locally Lipschitz and positively homogeneity of degree L (a function g is positive homogeneous of degree L if $g(\alpha x)=\alpha^{L} g(x)$ for any $\alpha \geq 0$). We will prove that for any given $x \in \mathbb{R}^{d}$, for $s \in \partial f(x)$, we have $\langle s, x\rangle=L f(x)$. Here $\partial f(x)$ is Clarke Differential.

Q2.1

Show that when $x=0$, and $s \in \partial f(x)$, we have $\langle s, x\rangle=L f(x)$.

By definition of homogeneity, we have: $g(\alpha x)=\alpha^{L} g(x)$. Plugging in $x=0$, we get: $$ \begin{aligned} &f(\alpha*0) = \alpha^{L} f(0) \ &\implies f(0) = \alpha^{L} f(0) \ &\implies 1 = \alpha^{L} \ &\implies L = log_\alpha{1} = 0 \ & \text{where } f(0) \neq 0 \end{aligned} $$

And trivially, since $x=0$: $$ \begin{aligned} \langle s, 0\rangle= 0 \end{aligned} $$

And finally: $$ \begin{aligned} \langle s, x\rangle &=L f(x) \ \langle s, 0\rangle &=0 * f(x) \ 0&=0 \end{aligned} $$

Q2.2

Show for all $x \neq 0$ such that $\nabla f(x)$ exists, $\langle\nabla f(x), x\rangle=L f(x)$. Hint: You can use the following basic property about gradient: $$ \lim _{\delta \rightarrow 0} \frac{f(x+\delta x)-f(x)-\langle\nabla f(x), \delta x\rangle}{\delta}=0 . $$ Q2.3 (3 Points) Using the definition of Clarke Differential to show that for any given $x \in \mathbb{R}^{d}$, for $s \in \partial f(x)$, we have $\langle s, x\rangle=L f(x)$.

$$ \frac{d z_{t}}{d w_{1}}=\sum_{\tau<t} \frac{d z_{t}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_{1}} $$

Using f's homogeneity property (and some algebra), we manimuplate the limit as follows:

$$ \begin{aligned} &\lim _{\delta \rightarrow 0} \frac {f(x+\delta x)-f(x)-\langle\nabla f(x), \delta x\rangle} {\delta}=0 \

&\implies \lim _{\delta \rightarrow 0} \frac {(1+\delta)^Lf(x) - f(x) - \delta\langle\nabla f(x), x\rangle} {\delta}=0 \

&\implies \lim _{\delta \rightarrow 0} \frac {f(x)((1+\delta)^L - 1) - \delta\langle\nabla f(x), x\rangle} {\delta}=0 \

&\implies \lim _{\delta \rightarrow 0} \frac {f(x)(1+\delta+\delta^2+...+\delta^L - 1) - \delta\langle\nabla f(x), x\rangle} {\delta}=0 \

&\implies \lim _{\delta \rightarrow 0} \frac {f(x)(\delta+\delta^2...+\delta^L) - \delta\langle\nabla f(x), x\rangle} {\delta}=0 \

&\implies \lim _{\delta \rightarrow 0} f(x)(1 + \delta+\delta^2...+\delta^{L-1}) = \lim _{\delta \rightarrow 0} \langle\nabla f(x), x\rangle\

&\implies f(x) * \lim _{\delta \rightarrow 0} (1 + \delta)^{l-1} = \langle\nabla f(x), x\rangle\

&\implies f(x)*1 = \langle\nabla f(x), x\rangle\

\end{aligned} $$

And indeed, it can be observed that the only value of L that satisfies L-homogeneity ($Lf(x)=f(xlog_{\alpha}(L))$) is 1.

Q2.3

Using the definition of Clarke Differential to show that for any given $x \in \mathbb{R}^{d}$, for $s \in \partial f(x)$, we have $\langle s, x\rangle=L f(x)$.

We only need to show that $\langle s, x\rangle=L f(x)$ is true everywhere that $x \ne 0$ and $\nabla f(x)$ does not exist, since these two cases have already been proven, and because f is given to be locally ρ-Lipschitz, we know that $\forall x \in S$, where S is a neighborhood of f, $\partial f(x)$ exists.

Note the definition of the Clarke Differential: $$ \partial f(x):=\operatorname{conv}\left(\left{s \in \mathbb{R}^{d}: \exists\left{x_{i}\right}*{i=1}^{\infty} \text { s.t. } x*{i} \rightarrow x, \nabla f\left(x_{i}\right) \rightarrow s\right}\right) $$

Therefore, using the proof from Q2.2, there is guaranteed to be an $s \in \partial f(x)$ that satifies $\langle s, x\rangle=L f(x)$.

Q3

Q3.1

$$ f\left(w_{1}, w_{2}\right)=\left(\sin \left(2 \pi w_{1} / w_{2}\right)+3 w_{1} / w_{2}-\exp \left(2 w_{2}\right)\right) \cdot\left(3 w_{1} / w_{2}-\exp \left(2 w_{2}\right)\right) $$

1. $z_{1}=w_{1} / w_{2}$
2. $z_{2}=\sin \left(2 \pi z_{1}\right)$
3. $z_{3}=\exp \left(2 w_{2}\right)$
4. $z_{4}=3 z_{1}-z_{3}$
5. $z_{5}=z_{2}+z_{4}$
6. $z_{6}=z_{4} z_{5}$
7. $y=z_6$

Compute $\frac{df}{dw} = \frac{dz_6}{TODO}$ asdf

1. $\bar{w_1} = \bar{z_1}\frac{dz_1}{dw_1} = \frac{1}{\bar{w_2}}$

   $\bar{w_2} = \bar{z_1}\frac{dz_1}{dw_2} = -\frac{\bar{w_1}}{(\bar{w_2})^2}$

2. $\bar{z_1} = \bar{z_2}\frac{dz_2}{dz_1} = 2 \pi \cos \left(2 \pi z_{1}\right)$

3. $\bar{w_2} = \bar{z_3}\frac{dz_3}{dw_2} = 2 \exp \left(2 w_{2}\right)$

4. $\bar{z_1} = \bar{z_4}\frac{dz_4}{dz_1} = 3\bar{z_4}$

   $\bar{z_3} = \bar{z_4}\frac{dz_4}{dz_3} = -\bar{z_4}$

5. $\bar{z_2} = \bar{z_5}\frac{dz_5}{dz_2} = \bar{z_5}$

   $\bar{z_4} = \bar{z_5}\frac{dz_5}{dz_4} = \bar{z_5}$

6. $\bar{z_4} = \bar{z_6}\frac{dz_6}{dz_4} = \bar{z_6}\bar{z_5}$

   $\bar{z_5} = \bar{z_6}\frac{dz_6}{dz_5} = \bar{z_6}\bar{z_4}$

7. $\bar{z_6} = \bar{y} = 1$

TODO: pseudocode? Is this sufficient?

Q3.2

We use the following formula to calculate the forward mode auto-differentiation for w_1 and w_2: $$ \frac{d z_{6}}{d w_{k}}=\sum_{\tau<6} \frac{d z_{6}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_{k}} \text{where } k \in {1, 2} $$

Q3.2.1

Assume we have already evaluated $(z_1, ...,z_6)$ identically to question Q3.1 and have the result stored in memory.

Next, we compute compute all the derivatives using the chain rule, first respect to w_1, then w_2:

When this pseudo code is actually implemented, the two snippets below will be abstracted into a common function that computes the forward auto-differnetiation.

Without further ado, compute $\frac{dz_6}{dw_2}$

1. $\bar{z_1} = \frac{dz_1}{dw_1} = \frac{1}{w_2}$

2. $\bar{z_2} = \sum_{\tau<2} \frac{d z_{2}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_1} = 2\pi \cos(2\pi \bar{z_1}) + \bar{z_1}$

3. $\bar{z_3} = \sum_{\tau<3} \frac{d z_{3}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_1} = 0 + 0 = 0$

4. $\bar{z_4} = \sum_{\tau<4} \frac{d z_{4}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_1} = 3\bar{z_1} + 0 + -\bar{z_3} = 3\bar{z_1} -\bar{z_3}$

5. $\bar{z_5} = \sum_{\tau<5} \frac{d z_{5}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_1} = 0 + \bar{z_2} + 0 + \bar{z_4} = \bar{z_2} + \bar{z_4}$

6. $\bar{z_6} = \sum_{\tau<6} \frac{d z_{6}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_1} = 0 + 0 + 0 + (\bar{z_4}z_5) + (z_4\bar{z_5}) = \bar{z_4}z_5 + z_4\bar{z_5} $

Finally: $\frac{dz_6}{dw_1} = \bar{z_6}$

Now compute $\frac{dz_6}{dw_1}$

1. $\bar{z_1} = \frac{dz_1}{dw_2} = \frac{w_2}{w_2^2}$
2. $\bar{z_2} = \sum_{\tau<2} \frac{d z_{2}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_2} = 0$
3. $\bar{z_3} = \sum_{\tau<3} \frac{d z_{3}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_2} = 2\exp(2w_2)$
4. $\bar{z_4} = \sum_{\tau<4} \frac{d z_{4}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_2} = \bar{z_1} + 0 + -\bar{z_3} = 3\bar{z_1} -\bar{z_3}$
5. $\bar{z_5} = \sum_{\tau<5} \frac{d z_{5}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_1} = 0 + \bar{z_2} + 0 + \bar{z_4} = \bar{z_2} + \bar{z_4}$
6. $\bar{z_6} = \sum_{\tau<6} \frac{d z_{6}}{\partial z_{\tau}} \frac{d z_{\tau}}{\partial w_1} = 0 + 0 + 0 + (\bar{z_4}z_5) + (z_4\bar{z_5}) = \bar{z_4}z_5 + z_4\bar{z_5} $

Finally: $\frac{dz_6}{dw_2} = \bar{z_6}$

Q3.2.2

We'd need to compute chain of derivatives with respect to each dimension in the input vector $w \in \mathbb{R}^d$.

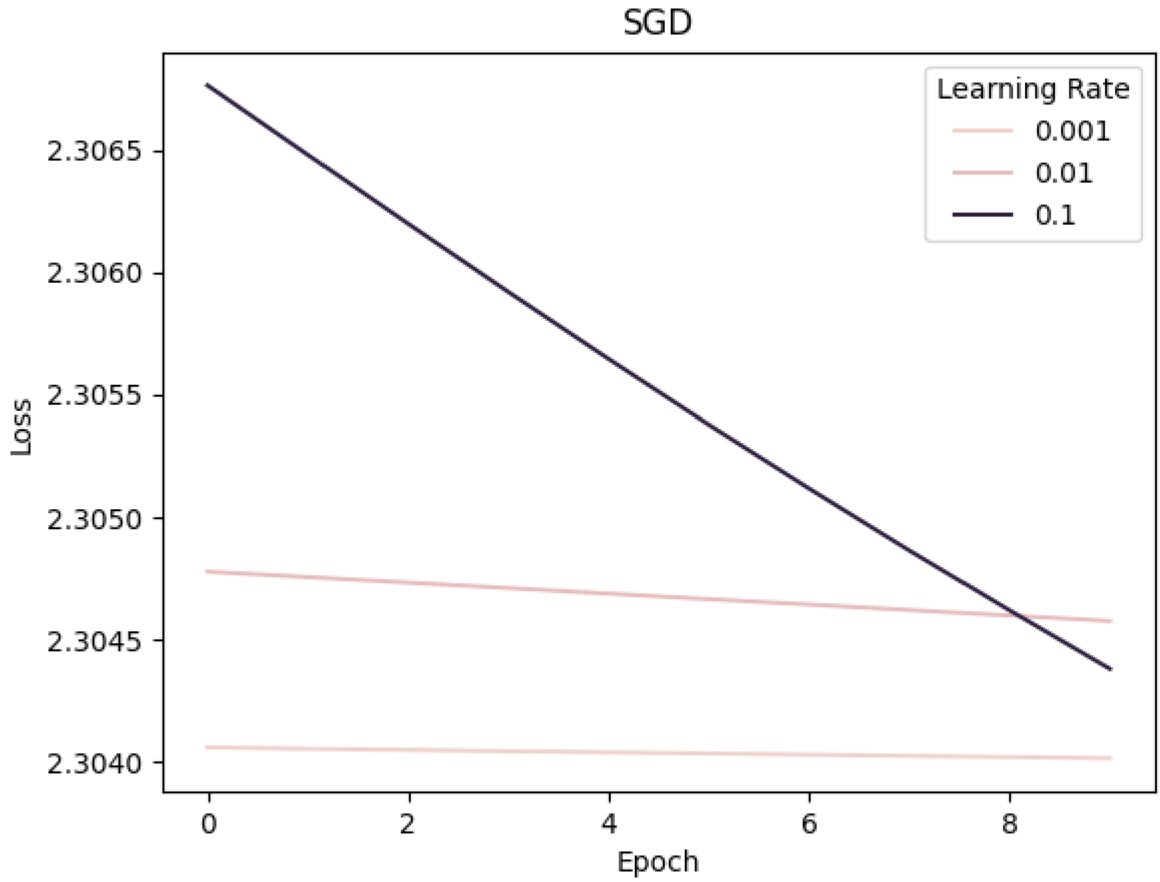
Q3.2.3

Since T computations are required for reach input variable d , the upper bound is: $$O(dT)$$

Q4

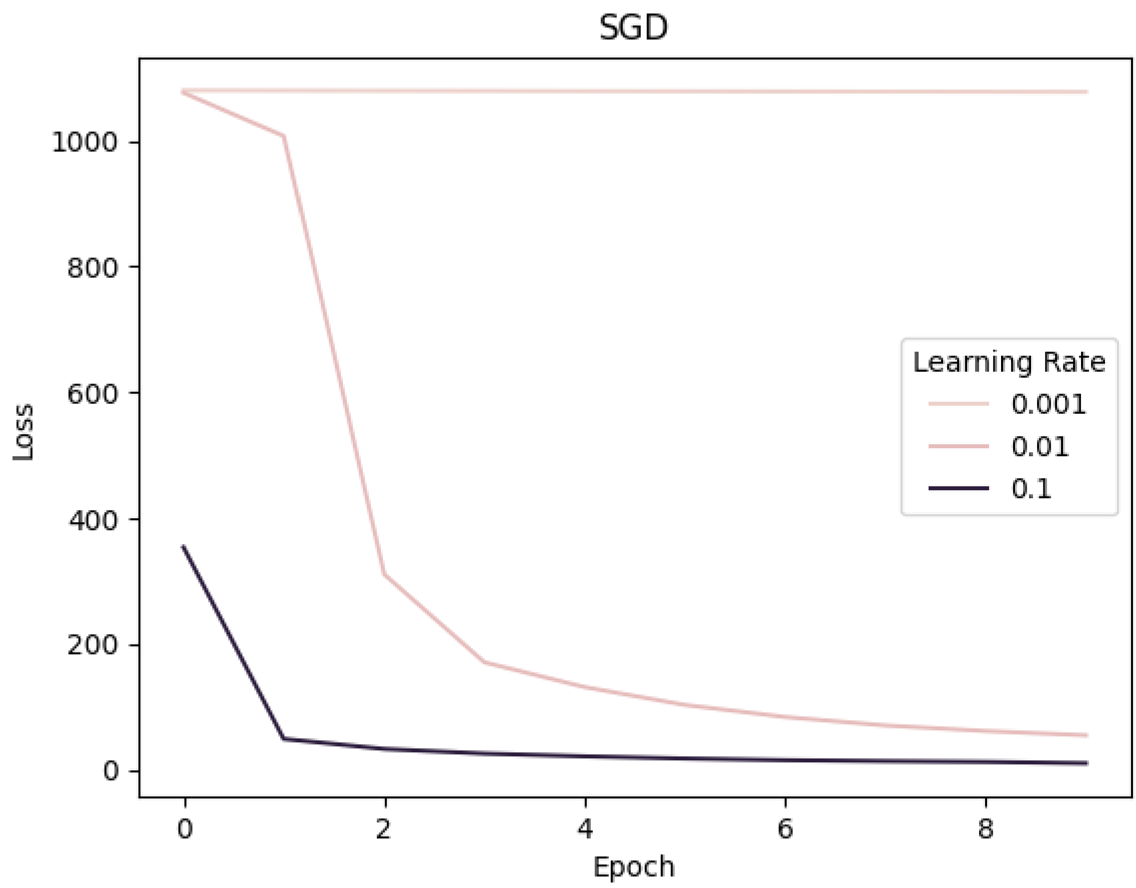
Q4.1

Gradient Descent

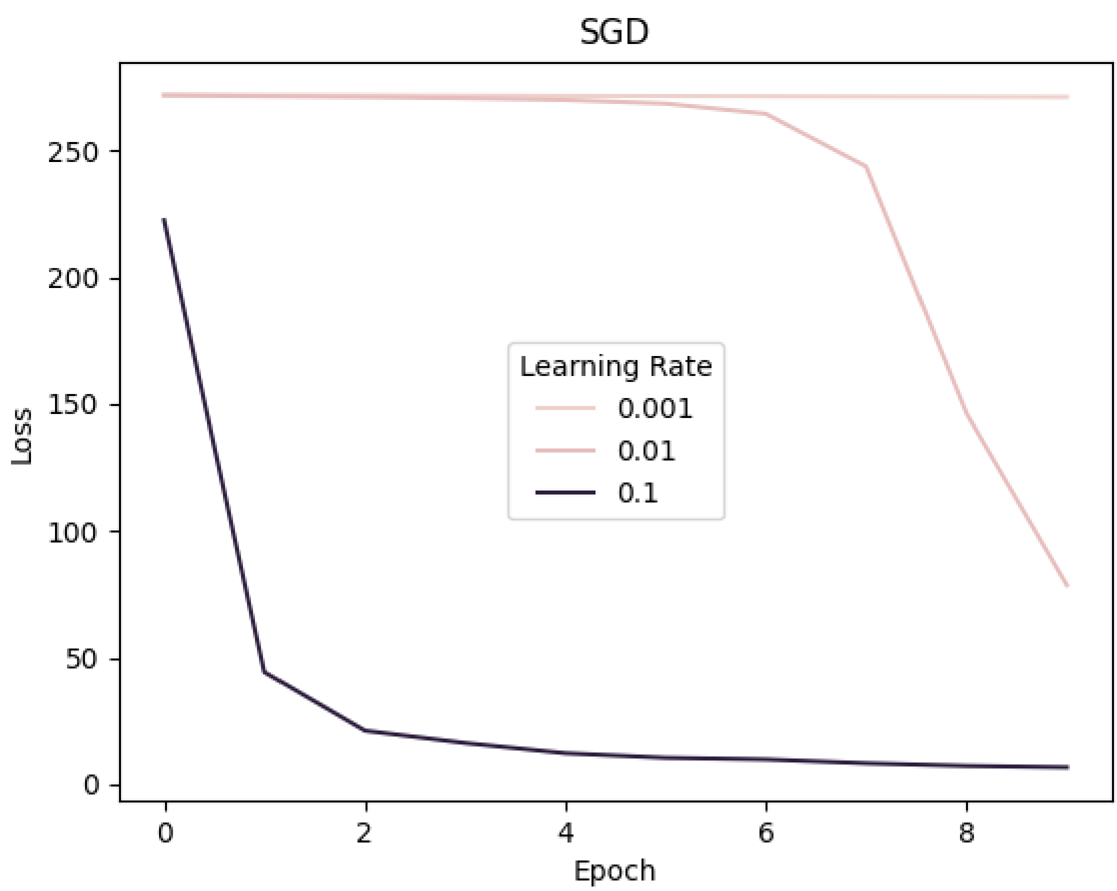


Batch size: all data; Number of epochs: 10

Minibatch SGD

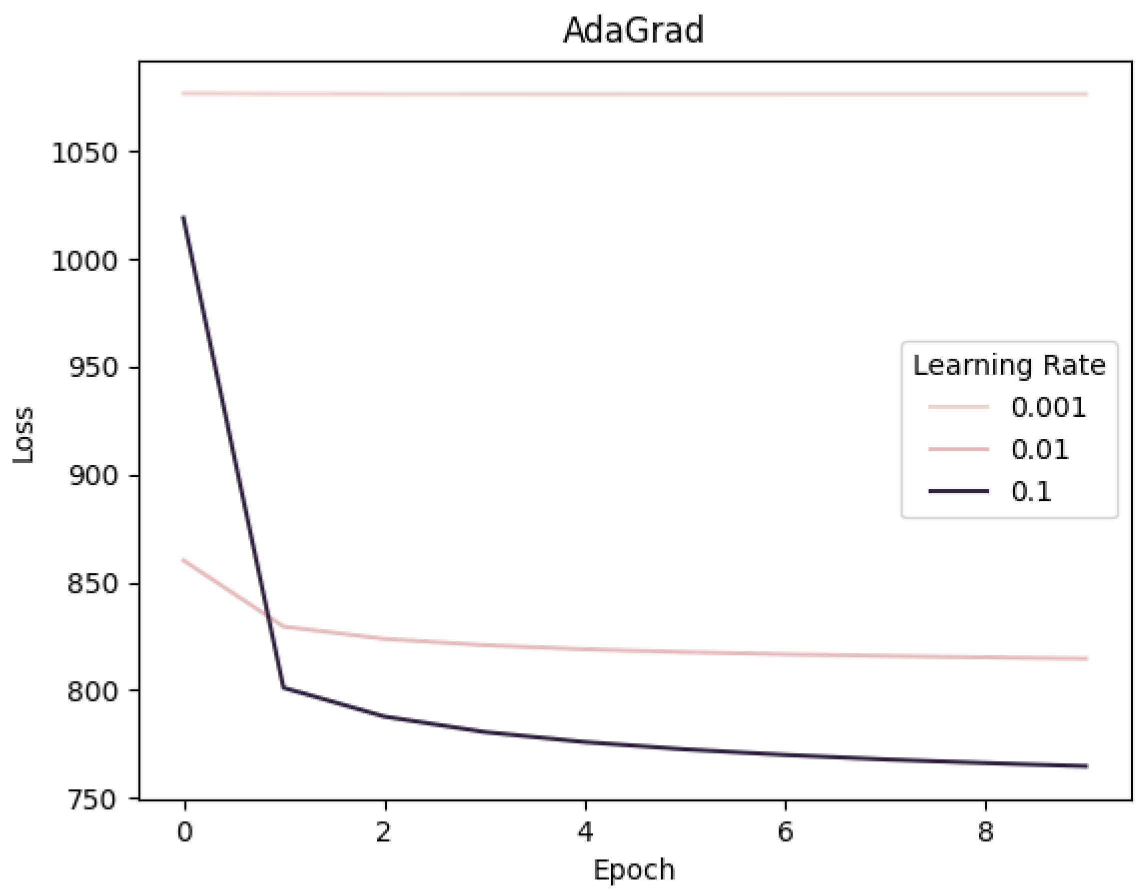


Batch size: 128; Number of epochs: 10

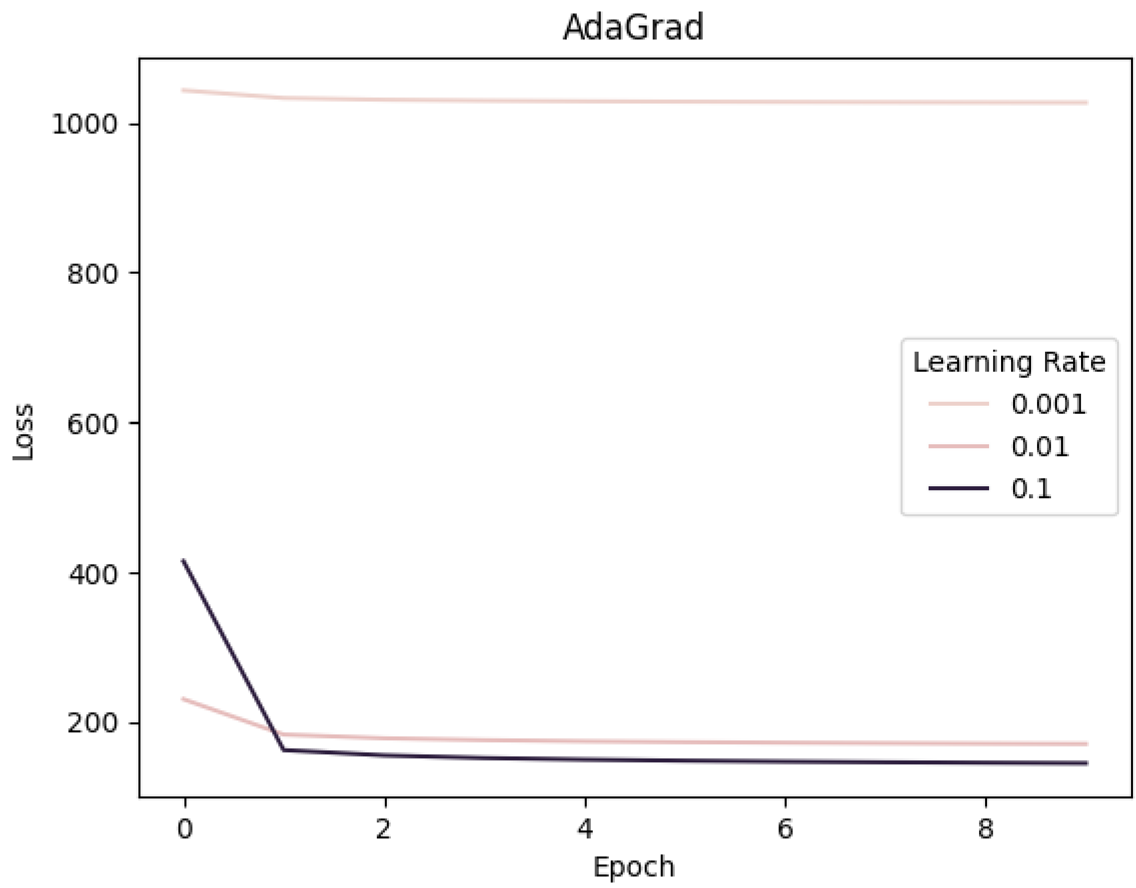


Batch size: 512; Number of epochs: 10

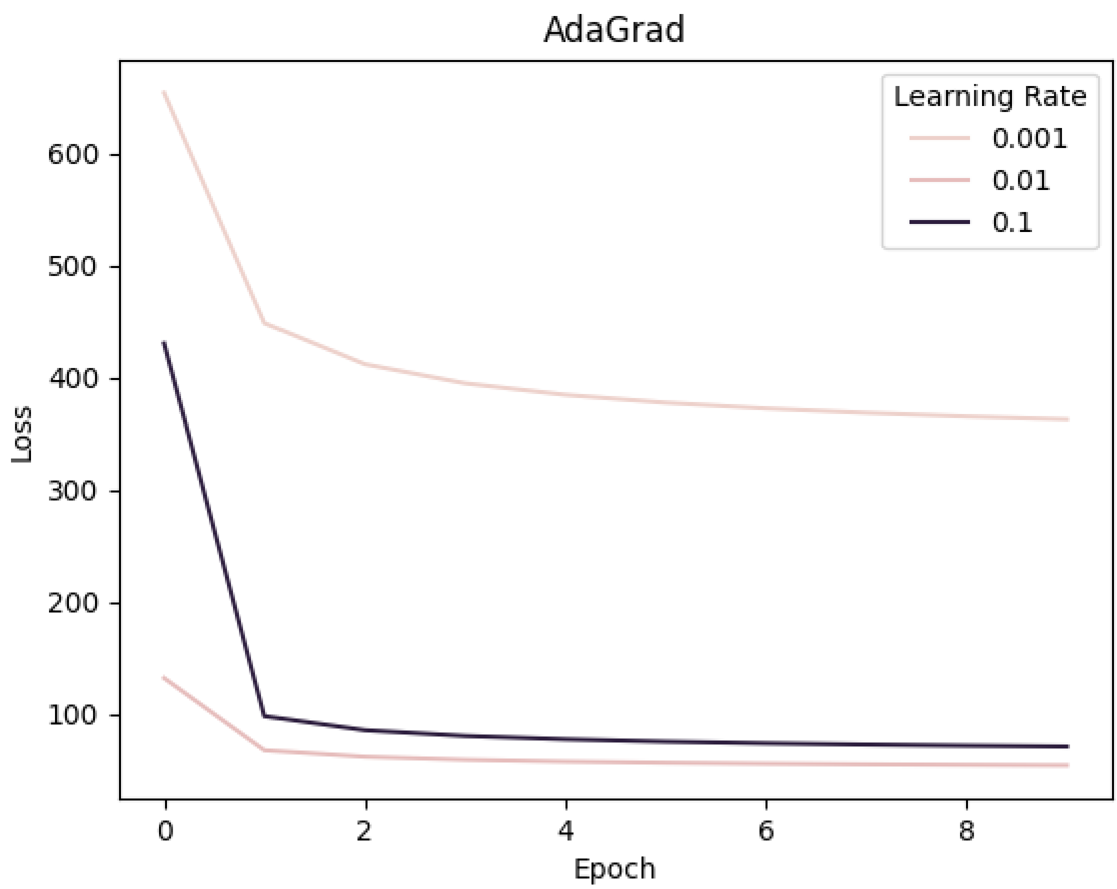
AdaGrad



Batch size: 128; Learning decay: 0.9; Number of epochs: 10

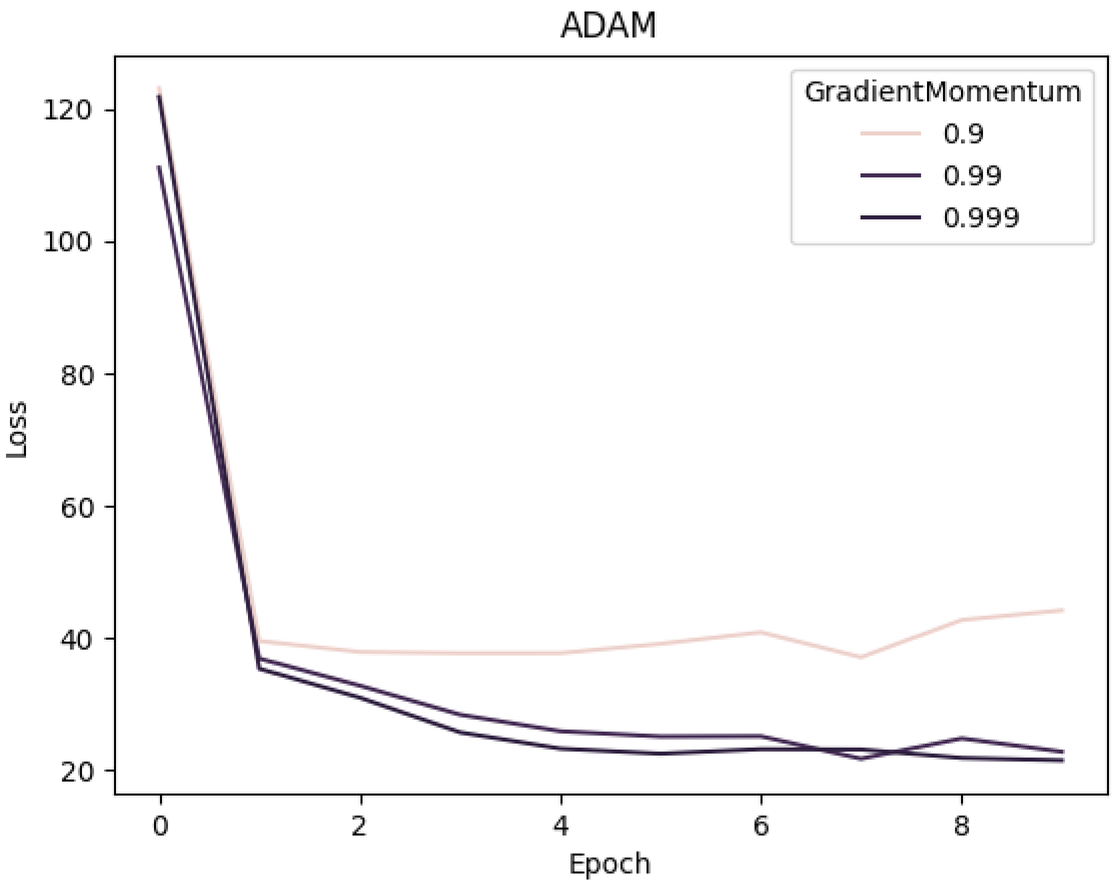


Batch size: 128; Learning decay: 0.09; Number of epochs: 10

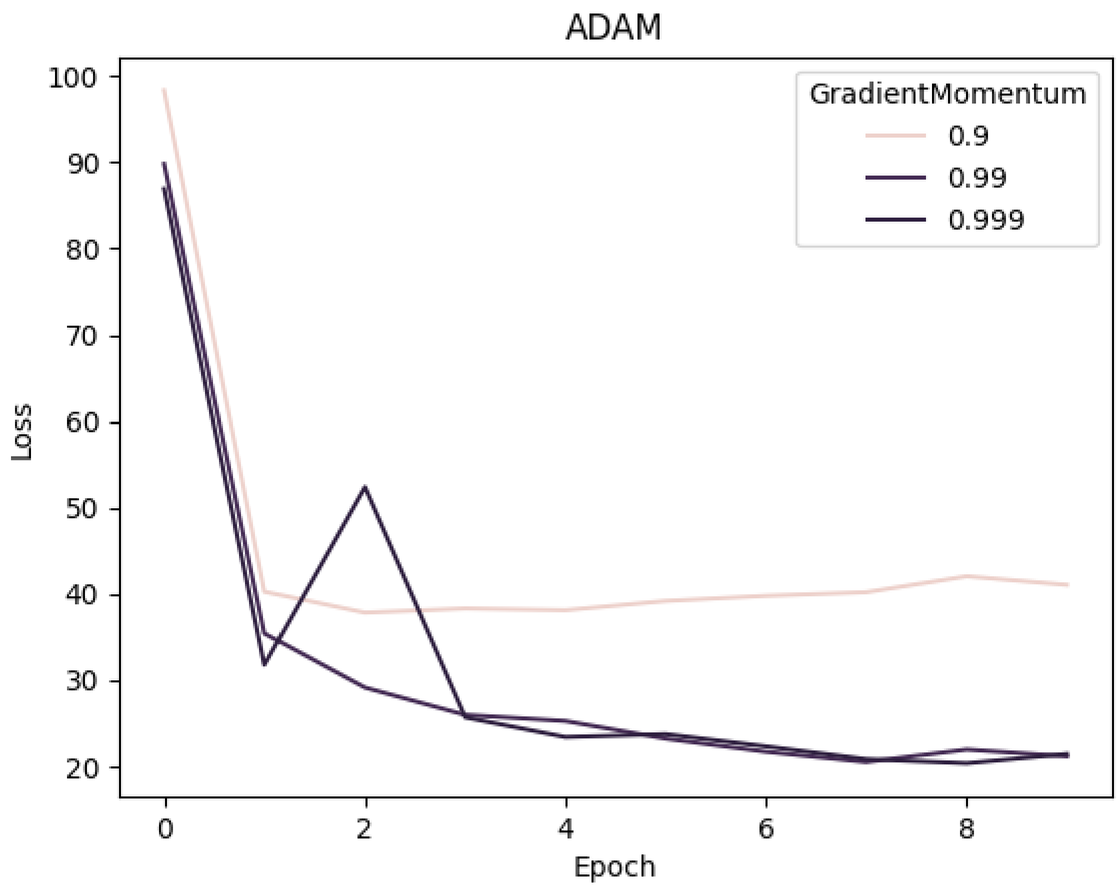


Batch size: 128; Learning decay: 0.009; Number of epochs: 10

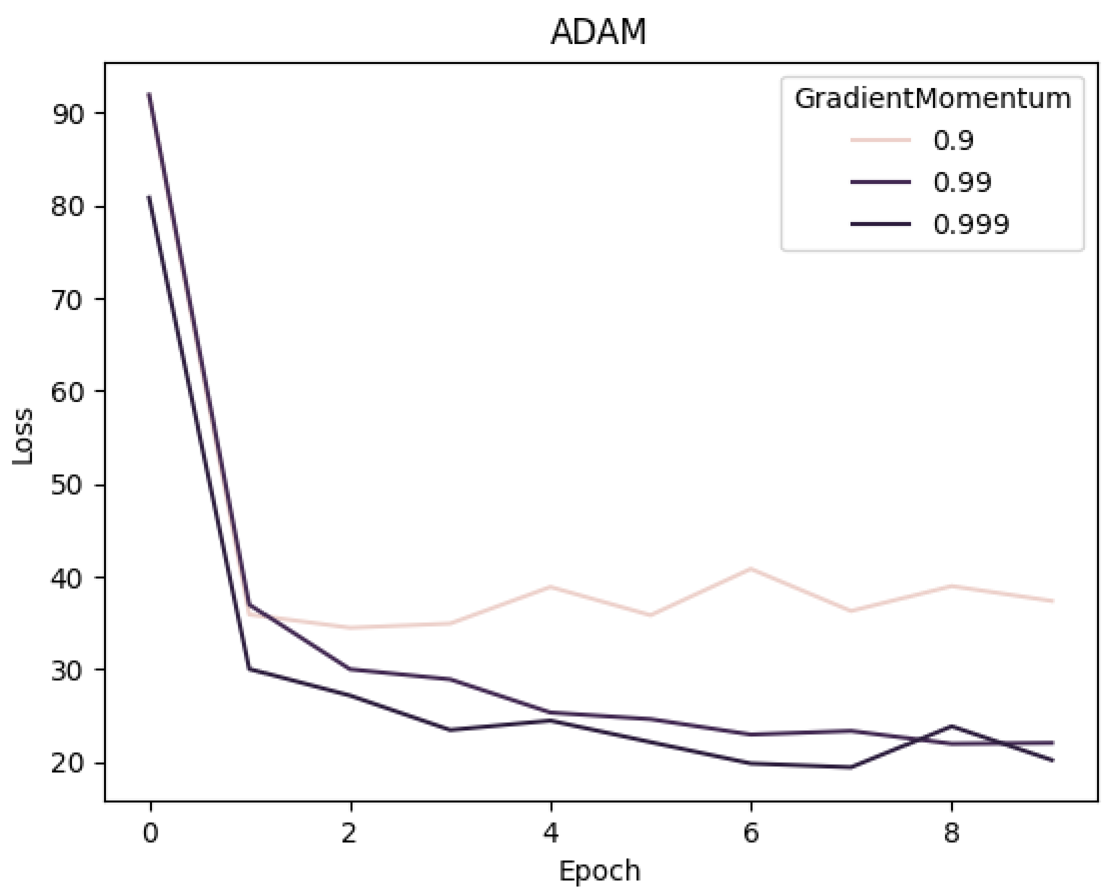
Adam



Batch size: 128; β_1: 0.9; β_2: 0.999; Number of epochs: 10



Batch size: 128; β_1: 0.9; β_2: 0.999; Number of epochs: 10



Batch size: 128; β_1: 0.9; β_2: 0.999; Number of epochs: 10

4.2

While adaptive optimization algorithms (e.g. Adam and AdaGrad) seem to converge faster, they don't seem to generalize as well as simple SGD (this is not a conclusion drawn from the tests here). However, recent research suggests might be simply a matter of hyper parameter tuning (adaptive strategies have more hyperparemeters).

Now pivoting to vanilla gradient descent vs mini-batch SGD, the principle advantage of mini-batch memory usage; it does not require the entire dataset to be in memory like VGD, which is intractible for large datasets.