COMMON OPTIMIZATION ALGORITHMS FOR DEEP LEARNING

INTRODUCTION

In this work, the most common optimization algorithms that are used to train neural networks will be presented. The algorithms will be examined in the order that they have invented. In this way, we can construct the Adam Optimizer step by step. First, a simple convex linear regression problem will be defined. Then, exponential weighted moving average operation will be examined. Afterwards, three optimization methods will be introduced and applied to obtain numerical results of the linear regression problem. Finally, a simple image classification task will be experimented with different optimization algorithms.

1.Linear Regression Problem

The linear regression problem is one of the most common problems in the optimization literature. Because of the simplicity of it, we will use linear regression problem to observe various optimization algorithms.

$$y = mx + b$$

Figure 1: Line Formula

Any line can be defined by two variables in 2-dimensional space. We will define these two variables as m and b throughout the experiments. In figure 1, 2- dimensional line formulation is shown. The objective of this problem is to obtain the most proper line which represents a given dataset. In order to find out the optimum line, we need to formulize an error function to measure the appropriateness of the line to our dataset. This error function is called the cost function in the optimization field, and the design of the cost function is not strict, i.e., there are infinitely many cost functions that can be used to solve our problem. The optimization methods that will be presented are local optimization functions. Because local optimum of the convex function is the global optimum of it, our cost function should be a convex function. In figure 2, least squares cost function which is a convex function is represented.

$$\arg_{m,b}\min\sum_{i=1}^{n}\left(y_i-(mx_i+b)\right)^2=\arg_{m,b}\min\|e\|_2^2$$

$$\cos t\,(m,b)=\sum_{i=1}^{n}\left(y_i-(mx_i+b)\right)^2$$

Figure 2: Least Square Formulation of Cost Function

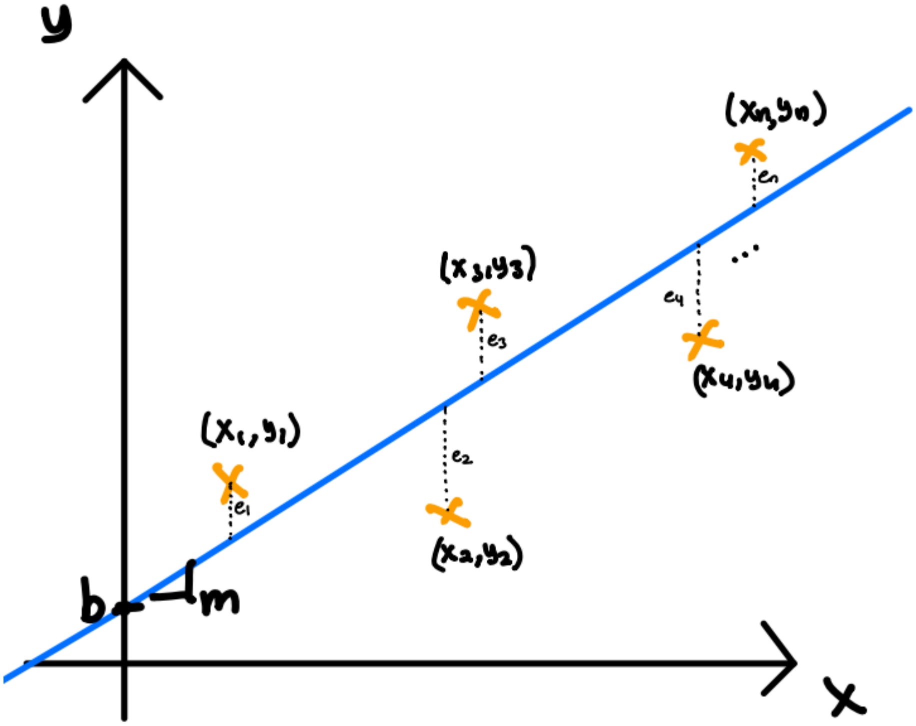


Figure 3: Visualization of Linear Regression Problem

2. Exponential Weigthed Moving Average

The exponential weighted moving average is a smoothing operation for a sequence. An entry of the output sequence depends on the entries before itself with exponentially decreasing weights. We can formulate each entry of the output sequence without applying any filter. Moreover, each entry of the output signal depends only on the current entry of the input signal and the last entry of the output signal. Therefore, exponential weighted moving average operation is an efficient algorithm both in memory cost and in compexity. There is a simple formulation of the process in figure 4. We can approximate the length of the intevral of moving average with formulation in figure 5. Therefore, we can say that if beta increases, the smoothing operation decreases its effect, and vice versa. The exponential weighted moving average method is very crucial to implement the optimization algorithms that will be introduced.

$$y\left[n\right] = \beta y\left[n-1\right] + \left(1-\beta\right) x\left[n\right], 0 \leq \beta \leq 1$$

Figure 4: Exponential weighted moving average formula

$$interval = \frac{1}{1-\beta}$$

Figure 5: Exponential weighted moving average approximate interval

Before ending up the exponential weighted moving average method, the bias correction should also be mentioned. Because the first entry of the ouput sequence is zero by algorithm, the initial period of the output sequence deviates from the expected values when it is compared to the input sequence. Therefore, a little modification on the formulation should be applied to compansate this deviation shown in the figure 5. The effect of this modification is only on the initial period of the sequence, the effect is attenuated on the following entries of the sequence.

$$y_{corr} = \frac{y}{1-\beta^{i}} \quad \text{, i is the number of iterations}$$

Figure 6: Bias correction for exponential weighted moving average

3. Gradient Descent with Momentum

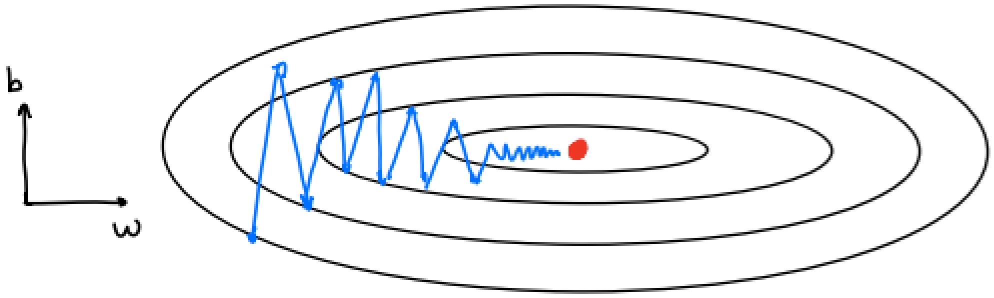


Figure6 : Level Sets of cost function and propagation visualization

The gradient descent is the first algorithm that will be introduced. This is a simple algorithm but it is a stepping stone to obtain a clear understanding of the other algorithms. The algorithm is composed of three steps:

- Compute the gradient of the cost function with respect to the variables.
- Apply exponential weighted moving average method to determine the update direction.

$$V_{dw} = \beta V_{dw} + (1-\beta)\, dw$$
$$v_{db} = \beta v_{db} + (1-\beta)\, db$$

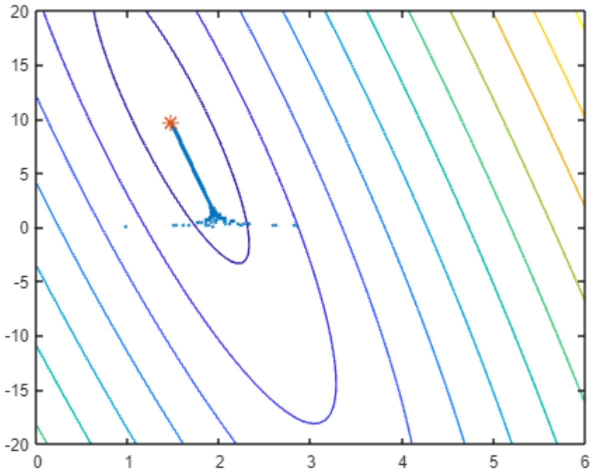
Figure 7: Formulas for step 2

- New variables are calcualted with subtraction by the learning rate times the direction.

$$w = w - \alpha v_{d\omega}$$
$$b = b - \alpha v_{db}$$

Figure 8: Formulas for step 3

Here, beta constant stands for the momentum. Momentum expression is derived directly from the exponential weighted moving average method . If the beta constant increases, the direction changes less, and vice versa. This is important to note that there is a strong analogy between the physical momentum and the momentum here. Moreover, momentum results in less oscillations in the vertical direction. Therefore, the updates becomes more efficient in the direction of the optimal point.



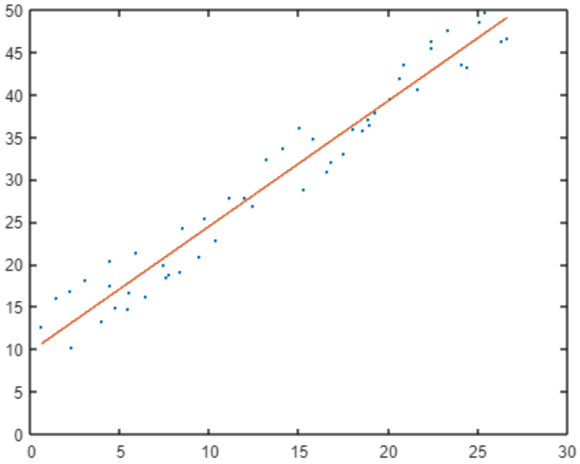


Figure 9: Propagation and level sets

Figure 10: Line estimate and data points

4. Root Mean Squared Propagation

There is two differences between the RMSPROP and the Gradient Descent with Moment:

- RMSPROP does not utilize the momentum mechanism to suppress the oscillations.
- To impede the the search in the direction of oscillations, it uses the exponential weighted moving average of the square of the gradient vector.

There are three steps to implement this algorithm at each iteration:

- Compute the gradient of the cost function with respect to the variables.
- Apply exponential weighted moving average method to square of the gradient vector.

$$S_{dw} = \beta S_{dw} + (1-\beta)\, dw^2$$
$$S_{db} = \beta S_{db} + (1-\beta)\, db^2$$

Figure 11: Formulas for step 2

- New variables are calcualted with subtraction by the learning rate times the direction. And use the coefficients obtained from the previous part to suppress the oscillations.

$$w = w - \alpha \frac{dw}{\sqrt{s_{dw}} + \varepsilon}$$

$$b = b - \alpha \frac{db}{\sqrt{s_{db}} + \varepsilon}$$

Figure 12: Formulas for step 3

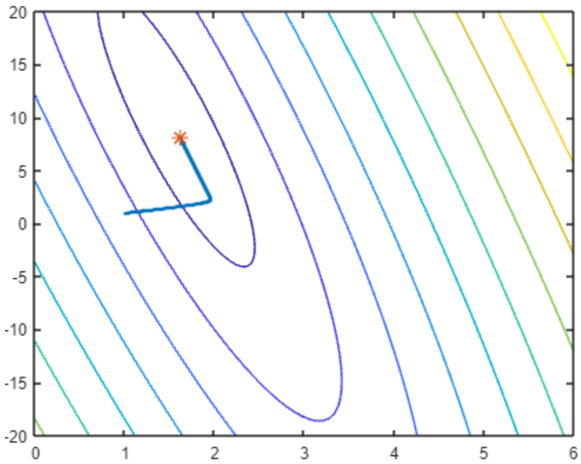


Figure 13: Propagation and level sets

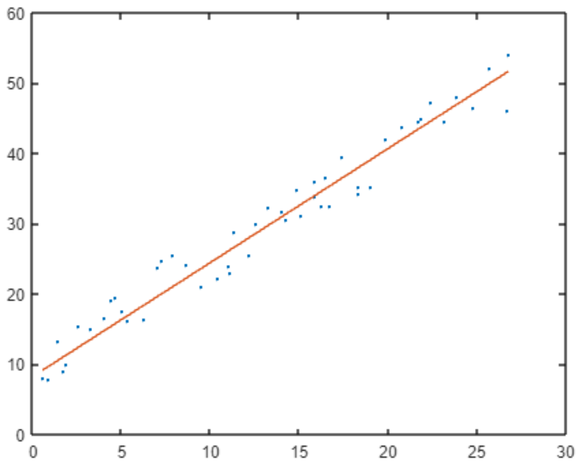


Figure 14: Line estimate and data points

5. Adam Optimizer

Adam optimizer is the combination of both the Gradient Descent with Momentum and the Root Mean Squared Propagation. Both the momentum and the exponential weighted moving average of the square of the gradient vector are used to construct this algorithm. Nowadays, Adam optimizer is one of the most common optimization algortihm in the field.

There are four steps to implement the Adam Optimizer:

- Compute the gradient of the cost function with respect to the variables.
- Apply exponential weighted moving average method to determine the update direction.

$$V_{dw} = \beta_1 V_{dw} + (1 - \beta_1)\, dw$$
$$v_{db} = \beta_1 v_{db} + (1 - \beta_1)\, db$$

Figure 15: Formulas for step 2

- Apply exponential weighted moving average method to square of the gradient vector.

$$S_{dw} = \beta_2 S_{dw} + (1 - \beta_2)\, dw^2$$
$$S_{db} = \beta_2 S_{db} + (1 - \beta_2)\, db^2$$

Figure 16: Formulas for step 3

- New variables are calcualted with subtraction by the learning rate times the direction. And use the coefficients obtained from the previous part to suppress the oscillations.

$$w = w - \alpha \frac{V_{dw}}{\sqrt{s_{dw}} + \varepsilon}$$
$$b = b - \alpha \frac{V_{db}}{\sqrt{s_{db}} + \varepsilon}$$

Figure 17: Formulas for step 4

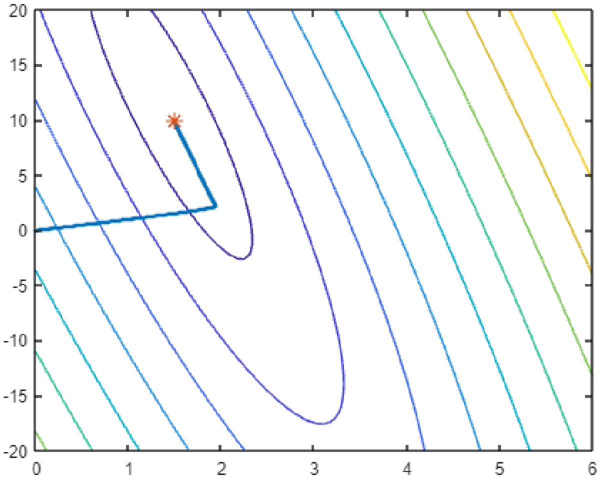


Figure 18: Propagation and level sets

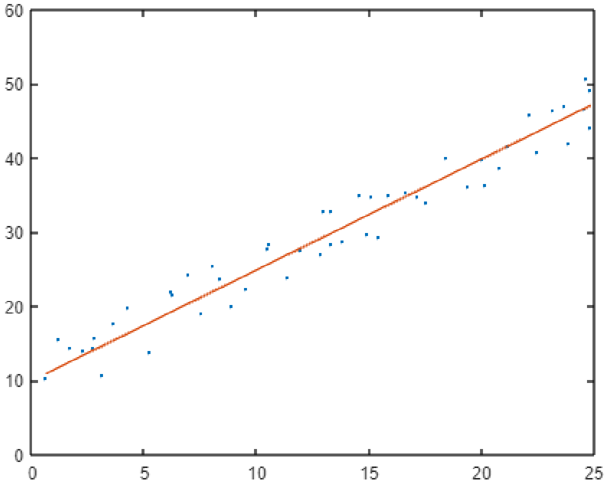


Figure 19: Line estimate and data points

6. Image Classification

To demonstrate the usage of these three optimization algorithms, the image processing functions of the MATLAB will be utilized. We will classify the Digitdataset of MATLAB in order to classify some images of handwritten numbers. Each training lasts 4 epochs and they will be conducted with distinct optimization algorithms. Finally, the graphical results are going to be examined. Training setup is created by built in MATLAB functions with appropriate parameters.

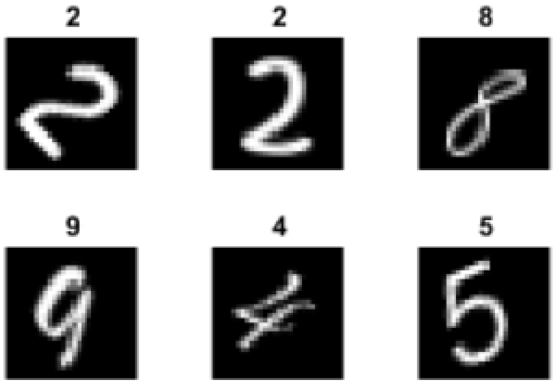


Figure 20: Some images from Digits Dataset

7. Training Results

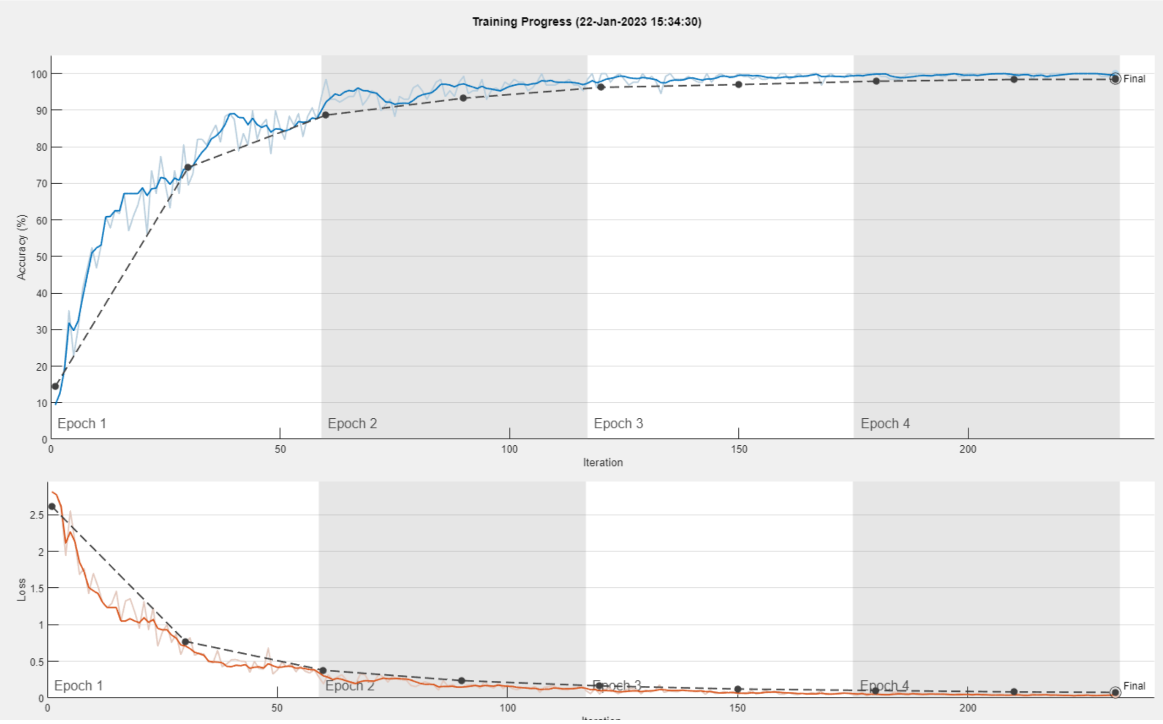
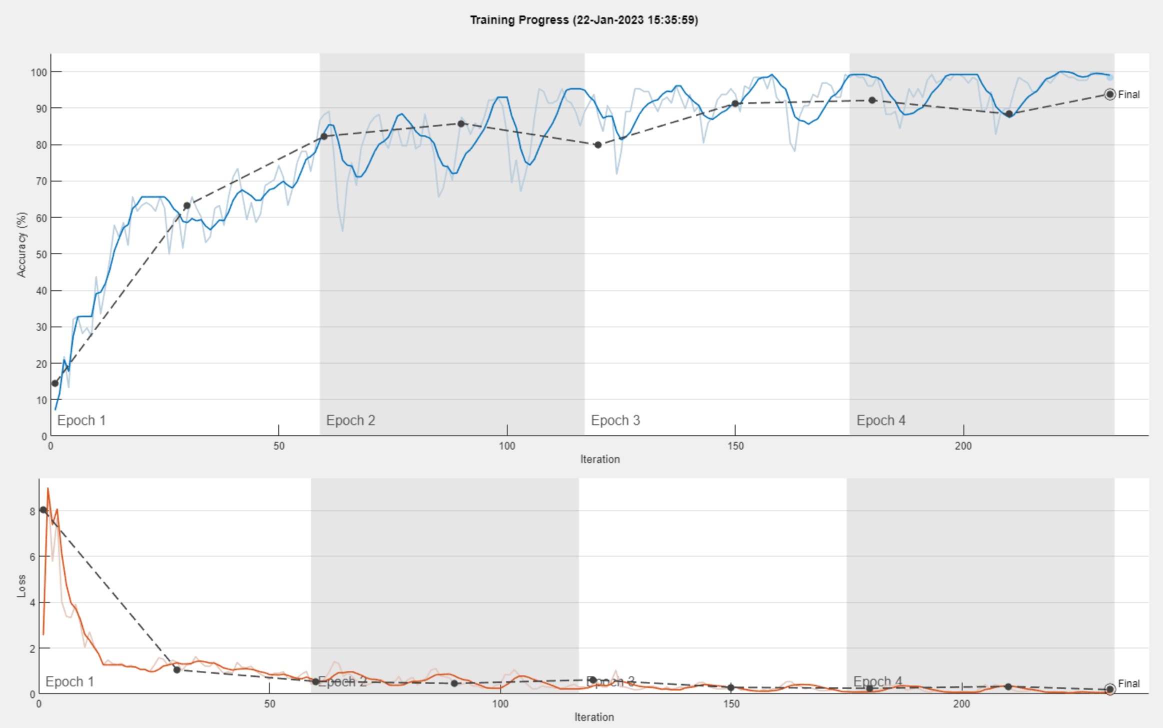


Figure 21: Accuracy and loss graph for SGDM



Figure

22: Accuracy and loss graph for RMSPROP

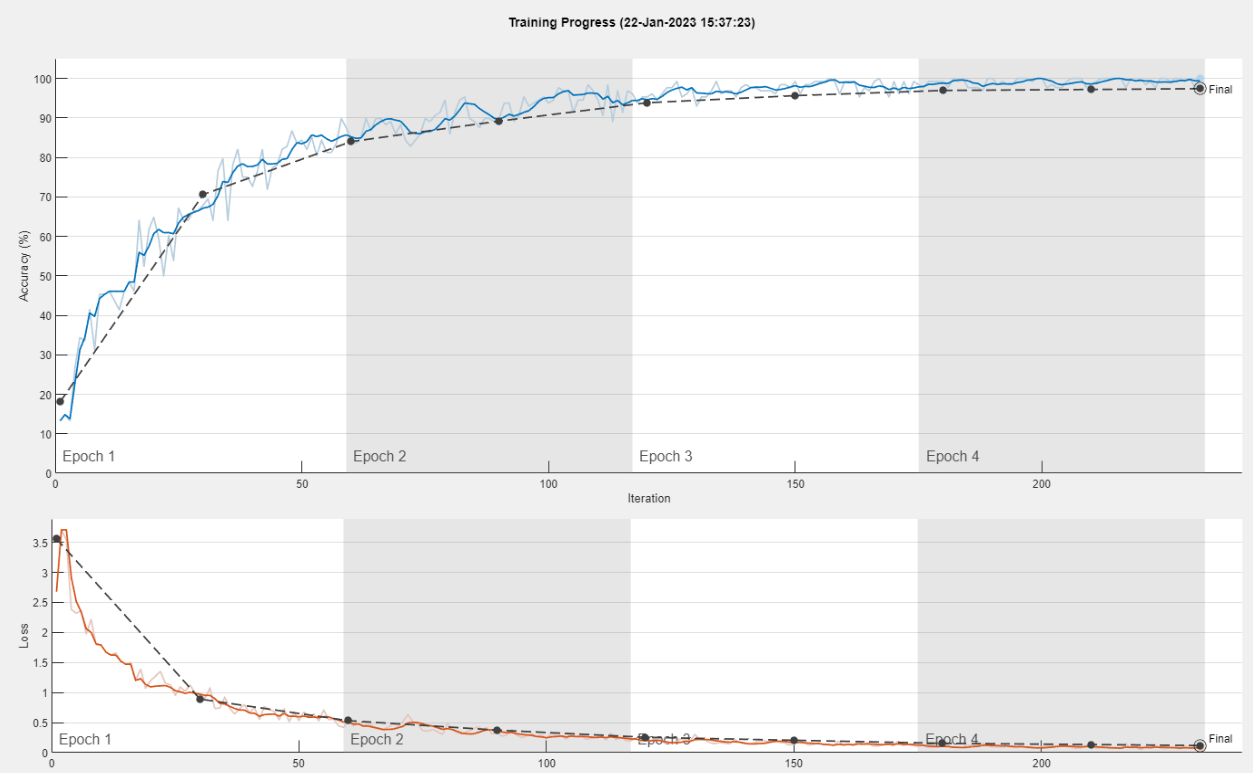


Figure 23: Accuracy and loss graph for ADAM

The final accuracy rates are:

- Gradient Descent with Momentum : 0.9852
- RMSPROP : 0.9552
- ADAM : 0.9748

The most accurate classification is achieved by Gradient Descent with Momentum and the least accurate result is achieved by RMSPROP. There is not a big difference between the results because the classficiation task is very simple at this case. Therefore, comparing the accuracy results are not fair to compare these three optimization algorithms.

The oscillations of the accuracy graph can convey some idea about the efficiecny of the algorithms. Most oscillations are observed at RMSPROP optimization because it does not utilize momentum mechanism.

Moreover, the least oscillations are observed at ADAM optimizer because it combines the mechanisms of both the Gradient Descent with Momentum and the RMSPROP algortihms.

It can be concluded that for a complicated image classification problem, ADAM optimizer can perform more efficiently compared to the other optimization algorithms.

CONCLUSION

In this project, three common optimization algorithms that are used for machine learning are introduced. First, the linear regression problem is defined to visualizing the implementation of the optimization algorithms. Then, the exponential weighted moving average operation is defined as a prerequisite topic to implement the algorithms. Then, each algorithm is formulized and defined step by step. Their application to solve linear regression problem is visualized step by step to observe how they are converging.

After the structure of the algorithms observed well, their applications on image processing/deep learning is observed. A simple image classification task is completed with each optimization algorithm to analyse their difference.

Historically, the gradient descent, RMSPROP and ADAM optimizers were designed respectively. The motivation behind the improvements is reducing the oscillations of the propagation when training our model. Therefore, the “momentum“ and the “exponential weighted moving average method to square of the gradient vector” methods are utilized. To design ADAM optimizer, these two methods are combined to get more robust propagation algorithm.