AI VIET NAM - COURSE 2024

Optimization Algorithms in Deep Learning - Exercise

Quang-Vu Pham

November 26, 2024

1 Introduction:

This document is prepared to address a set of exercises assigned by AI Vietnam (aivietnam.edu.vn). The exercises focus on implementing and analyzing various optimization algorithms. The file is structured to provide detailed solutions to these tasks, using both theoretical explanations and practical implementations.

The optimization algorithms covered include Gradient Descent, Gradient Descent with Momentum, RMSProp, and Adam, each designed to minimize a given function.

2 Problem:

Given the following function: $f(w_1, w_2) = 0.1w_1^2 + 2w_2^2$ (1)

2.1 Problem 1:

Based on the Gradient Descent algorithm, finding the minimum point of function (1) with the following parameters initialized $w_1 = -5$, $w_2 = -2$, $\alpha = 0.4$:

(a) Provide detailed steps to find the minimum point using Gradient Descent for 2 epochs. (find w_1 and w_2 after 2 epochs).

(b) Implementing a Python solution to find the parameters w1, w2 after 30 epochs.

Solution for (a):

1. Partial Derivatives:

$$\frac{\partial f}{\partial w_1} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_1} = 0.2w_1, \quad \frac{\partial f}{\partial w_2} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_2} = 4w_2$$

$$\nabla f(w_t) = \left[\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}\right]$$

2. Gradient Descent Update Rules:

$$w_t = w_{t-1} - \alpha \cdot \nabla f(w_{t-1})$$

3. Initialization:

$$w_0 = [-5, -2],\ \alpha = 0.4$$

Epoch 1 (t = 1):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-5) = -1, \frac{\partial f}{\partial w_2} = 4 \cdot (-2) = -8$$

$$\nabla f(w_0) = [-1, -8]$$
$$w_1 = w_0 - \alpha \cdot \nabla f(w_0) = [-5, -2] - 0.4 \cdot [-1, -8] = [-4.6, 1.2]$$

Epoch 2 (t = 2):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-4.6) = -0.92, \frac{\partial f}{\partial w_2} = 4 \cdot 1.2 = 4.8$$
$$\nabla f(w_1) = [-0.92, 4.8]$$
$$w_2 = w_1 - \alpha \cdot \nabla f(w_1) = [-4.6, 1.2] - 0.4 \cdot [-0.92, 4.8] = [-4.232, -0.72]$$

(b) The code is implemented in gradient_descent.py.

2.2 Problem 2:

Based on the Gradient Descent + Momentum algorithm, finding the minimum point of function (1) with the following parameters initialized $w_1 = -5$, $w_2 = -2$, $v_1 = 0$, $v_2 = 0$, $\alpha = 0.6$, $\beta = 0.5$.

(a) Provide detailed steps to find the minimum point using Gradient Descent + Momentum algorithm for 2 epochs. (find w_1 and w_2 after 2 epochs).

(b) Implementing a Python solution to find the parameters w1, w2 after 30 epochs.

Solution for (a):

1. Partial Derivatives:

$$\frac{\partial f}{\partial w_1} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_1} = 0.2w_1, \quad \frac{\partial f}{\partial w_2} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_2} = 4w_2$$
$$\nabla f(w_t) = \left[\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}\right]$$

2. Gradient Descent + Momentum Update Rules:

$$v_t = \beta v_{t-1} + (1 - \beta)\nabla f(w_{t-1})$$
$$w_t = w_{t-1} - \alpha v_t$$

3. Initialization:

$$w_0 = [-5, -2],\ v_0 = [0, 0],\ \alpha = 0.6,\ \beta = 0.5$$

Epoch 1 (t = 1):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-5) = -1, \frac{\partial f}{\partial w_2} = 4 \cdot (-2) = -8$$
$$\nabla f(w_0) = [-1, -8]$$
$$v_1 = \beta \cdot v_0 + (1 - \beta) \cdot \nabla f(w_0) = 0.5 \cdot [0, 0] + (1 - 0.5) \cdot [-1, -8] = [-0.5, -4]$$
$$w_1 = w_0 - \alpha \cdot v_1 = [-5, -2] - 0.6 \cdot [-0.5, -4] = [-4.7, 0.4]$$

Epoch 2 (t = 2):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-4.7) = -0.94, \frac{\partial f}{\partial w_2} = 4 \cdot 0.4 = 1.6$$
$$\nabla f(w_1) = [-0.94, 1.6]$$
$$v_2 = \beta \cdot v_1 + (1 - \beta) \cdot \nabla f(w_1) = 0.5 \cdot [-0.5, -4] + (1 - 0.5) \cdot [-0.94, 1.6] = [-0.72, -1.2]$$
$$w_2 = w_1 - \alpha \cdot v_2 = [-4.7, 0.4] - 0.6 \cdot [-0.72, -1.2] = [-4.268, 1.12]$$

(b) The code is implemented in gradient_descent_momentum.py.

2.3 Problem 3:

Based on the RMSProp algorithm, finding the minimum point of function (1) with the following parameters initialized $w_1 = -5$, $w_2 = -2$, $s_1 = 0$, $s_2 = 0$, $\alpha = 0.3$, $\gamma = 0.9$, $\epsilon = 10^{-6}$.

(a) Provide detailed steps to find the minimum point using RMSProp for 2 epochs. (find w_1 and w_2 after 2 epochs).

(b) Implementing a Python solution to find the parameters w1, w2 after 30 epochs.

Solution for (a):

1. Partial Derivatives:

$$\frac{\partial f}{\partial w_1} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_1} = 0.2w_1, \quad \frac{\partial f}{\partial w_2} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_2} = 4w_2$$

$$\nabla f(w_t) = \left[\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}\right]$$

2. RMSProp Update Rules:

$$s_t = \gamma s_{t-1} + (1-\gamma) \cdot (\nabla f(w_{t-1}))^2$$

$$w_t = w_{t-1} - \frac{\alpha}{\sqrt{s_t + \epsilon}} \cdot \nabla f(w_{t-1})$$

3. Initialization:

$$w_0 = [-5, -2],\ s_0 = [0, 0],\ \alpha = 0.3,\ \gamma = 0.9,\ \epsilon = 10^{-6}$$

Epoch 1 (t = 1):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-5) = -1, \quad \frac{\partial f}{\partial w_2} = 4 \cdot (-2) = -8$$

$$\nabla f(w_0) = [-1, -8]$$

$$s_1 = \gamma s_0 + (1-\gamma) \cdot (\nabla f(w_0))^2 = 0.9 \cdot [0, 0] + (1 - 0.9) \cdot ([-1, -8])^2 = [0.1, 6.4]$$

$$w_1 = w_0 - \frac{\alpha}{\sqrt{s_1 + \epsilon}} \cdot \nabla f(w_0) = [-5, -2] - \frac{0.3}{\sqrt{[0.1, 6.4] + \epsilon}} \cdot [-1, -8] = [-4.0513, -1.0513]$$

Epoch 2 (t = 2):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-4.0513) = -0.8103, \quad \frac{\partial f}{\partial w_2} = 4 \cdot (-1.0513) = -4.2052$$

$$\nabla f(w_1) = [-0.8103, -4.2052]$$

$$s_2 = \gamma s_1 + (1-\gamma) \cdot (\nabla f(w_1))^2 = 0.9 \cdot [0.1, 6.4] + (1 - 0.9) \cdot ([-0.8103, -4.2052])^2 = [0.1557, 7.5284]$$

$$w_2 = w_1 - \frac{\alpha}{\sqrt{s_2 + \epsilon}} \cdot \nabla f(w_1)$$

$$= [-4.0513, -1.0513] - \frac{0.3}{\sqrt{[0.1557, 7.5284] + \epsilon}} \cdot [-0.8103, -4.2052] = [-3.4352, -0.5915]$$

(b) The code is implemented in rmsprop_algorithm.py.

2.4 Problem 4:

Based on the Adam Optimization Algorithm, find the minimum point of the function (1) with the following initial parameters: $w_1 = -5$, $w_2 = -2$, $v_1 = 0$, $v_2 = 0$, $s_1 = 0$, $s_2 = 0$, $\alpha = 0.2$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$.

(a) Provide detailed steps to find the minimum point using Adam for 2 epochs. (find w_1 and w_2 after 2 epochs).

(b) Implementing a Python solution to find the parameters w1, w2 after 30 epochs.

Solution for (a):

1. Partial Derivatives:

$$\frac{\partial f}{\partial w_1} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_1} = 0.2w_1, \quad \frac{\partial f}{\partial w_2} = \frac{\partial (0.1w_1^2 + 2w_2^2)}{\partial w_2} = 4w_2$$

$$\nabla f(w_t) = \left[\frac{\partial f}{\partial w_1}, \frac{\partial f}{\partial w_2}\right]$$

2. Adam Update Rules:

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1)\nabla f(w_{t-1})$$

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2)(\nabla f(w_{t-1}))^2$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_1^t}, \quad \hat{s}_t = \frac{s_t}{1 - \beta_2^t}$$

$$w_t = w_{t-1} - \alpha \frac{\hat{v}_t}{\sqrt{\hat{s}_t} + \epsilon}$$

3. Initialization:

$$w_0 = [-5, -2],\ s_0 = [0, 0],\ \alpha = 0.2,\ \beta_1 = 0.9,\ \beta_2 = 0.999,\ \epsilon = 10^{-6}$$

Epoch 1 (t = 1):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-5) = -1, \quad \frac{\partial f}{\partial w_2} = 4 \cdot (-2) = -8$$

$$\nabla f(w_0) = [-1, -8]$$

$$v_1 = \beta_1 v_0 + (1 - \beta_1) \cdot \nabla f(w_0) = 0.9 \cdot [0, 0] + (1 - 0.9) \cdot ([-1, -8]) = [-0.1, -0.8]$$

$$s_1 = \beta_2 s_0 + (1 - \beta_2) \cdot (\nabla f(w_0))^2 = 0.999 \cdot [0, 0] + (1 - 0.999) \cdot ([-1, -8])^2 = [0.001, 0.064]$$

$$\hat{v}_1 = \frac{v_1}{1 - \beta_1^1} = \frac{[-0.1, -0.8]}{1 - 0.9^1} = [-1, -8]$$

$$\hat{s}_1 = \frac{s_1}{1 - \beta_2^1} = \frac{[0.001, 0.064]}{1 - 0.999^1} = [1, 64]$$

$$w_1 = w_0 - \alpha \cdot \frac{\hat{v_1}}{\sqrt{s_1} + \epsilon} = [-5, -2] - 0.2 \cdot \frac{[-1, -8]}{\sqrt{[1, 64]} + \epsilon} = [-4.8, -1.8]$$

Epoch 2 (t = 2):

$$\frac{\partial f}{\partial w_1} = 0.2 \cdot (-4.8) = -0.96, \quad \frac{\partial f}{\partial w_2} = 4 \cdot (-1.8) = -7.2$$

$$\nabla f(w_1) = [-0.96, -7.2]$$

$$v_2 = \beta_1 v_1 + (1 - \beta_1) \cdot \nabla f(w_1) = 0.9 \cdot (-[0.1, 0.8]) + (1 - 0.9) \cdot ([-0.96, -7.2]) = [-0.186, -1.44]$$

$$s_2 = \beta_2 s_1 + (1-\beta_2)\cdot(\nabla f(w_1))^2 = 0.999\cdot[0.001, 0.064] + (1-0.999)\cdot([-0.96, -7.2])^2 = [0.00192, 0.11578]$$

$$\hat{v}_2 = \frac{v_2}{1-\beta_1^2} = \frac{[-0.186, -1.44]}{1-0.9^2} = [-0.97895, -7.57895]$$

$$\hat{s}_2 = \frac{s_2}{1-\beta_2^2} = \frac{[0.00192, 0.11578]}{1-0.999^2} = [0.96048, 57.91896]$$

$$w_2 = w_1 - \alpha\cdot\frac{\hat{v_2}}{\sqrt{s_2}+\epsilon} = (-[4.8, 1.8]) - 0.2\cdot\frac{[-0.97895, -7.57895]}{\sqrt{[0.96048, 57.91896]}+\epsilon} = [-4.6, -1.6]$$

(b) The code is implemented in adam_algorithm.py.

2.5 Problem 5:

This exercise involves modifying the optimization algorithms (**optimizers**) to observe how each algorithm addresses the **vanishing gradient problem**. A model is constructed with the following specifications:

- **Weights:** Initialized randomly using a **normal distribution** with $\mu = 0$, $\sigma = 0.05$.
- **Loss function:** Cross-Entropy Loss.
- **Optimizer:** Stochastic Gradient Descent (SGD).
- **Hidden layers:** 5.
- **Nodes per layer:** 128.
- **Activation function:** Sigmoid.

The following optimization algorithms will be applied to the model:

(a) Gradient Descent.

(b) Gradient Descent with Momentum.

(c) RMSProp.

(d) Adam.

(e) ADOPT (Adaptive Optimizer).

Solution: The code is implemented in `vanishing_problem.py`. The optimization algorithms can be changed by modifying the `optimizer` parameter passed into the following function:

```
model = FashionMNISTMLP(input_dims=784, hidden_dims=128, output_dims=10,
                        optimizer=torch.optim.Adam, model_name="FashionMNIST")
```

Code Listing 1: Code snippet for changing optimizer