Gradient Descent Method [6 Points]

Problem 1. Consider learning a linear regression function (without bias term) as follows.

$$y = 3x_1 + 4x_2 + \epsilon \tag{1}$$

Generate 500 samples in uniformly random manner with $x_1, x_2 \in [-10, 10]$. Add a gaussian noise ϵ with mean = 0 and variance = 0.01 to each generated sample value y obtained from x_1, x_2.

- Consider learning a linear regression function (without bias term). Hence, there will be two learnable parameters only. Assuming squared error loss function, write the objective function J to be minimized. **[0.5 marks]**
- Plot the error curve with respect to the parameters for this problem. **[0.5 marks]**
- Consider minimizing J using gradient descent with constant step size η_{opt}. Calculate and explain the optimal learning rate η_{opt}. **[1 mark]**
- Try gradient descent with following step sizes $\eta = \frac{0.9\eta_{opt}}{2}, \frac{1.5\eta_{opt}}{2}, \eta, 1.5\eta_{opt}$. For definiteness, we consider convergence to be complete $J < 0.001$. For every case,
  - Plot the error vs epoch curve. **[2 marks (0.5 for each η)]**
  - Using contour plots show the convergence path. **[2 marks (0.5 for each η)]**

Gradient Descent and Variants - 1 [4 Points]

Problem 2. Consider the Rosenbrock function $f(x, y) = x^2 + 100(y - x^2)^2$, which is used to benchmark optimization algorithms and the following variant admits a global minimum at $(0, 0)^T$. Use random initialization of the parameters.

- Run gradient descent with constant step size to minimize $f(x, y)$. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]
- Use gradient descent with Polyak's momentum method to minimize $f(x, y)$. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]
- Minimize $f(x, y)$ using Nesterov accelerated gradient descent. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]
- Minimize $f(x, y)$ using Adam optimizer. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]

Gradient Descent and Variants - 2 [4 Points]

Problem 3. Consider the following function

$$f(x,y) = \frac{50}{9}(x^2+y^2)^3 - \frac{209}{18}(x^2+y^2)^2 + \frac{59}{9}(x^2+y^2).$$

This function has global minimum at $(0,0)^T$ and local minima at $x^2+y^2=1$. Consider minimizing $f(x,y)$ using the methods below. Use random initialization of the parameters.

- Use gradient descent with constant step size to minimize $f(x,y)$. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]
- Use Polyak's momentum method to minimize $f(x,y)$. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]
- Minimize $f(x,y)$ using Nesterov accelerated gradient descent. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]
- Minimize $f(x,y)$ using Adam optimizer. Show contour plot of the function. After every update, using arrow show the movement in the contour plots. Do it till convergence. [1 Point]

Preprocessing the data speeds up learning [5 Points]

Problem 6. Demonstrate that pre-processing data can lead to significant reduction in time of learning. Consider a single linear output unit for a two-category classification task, with teaching values ± 1 and squared error criterion.

1. Write a program to train three weights based on training samples.
2. Generate training set having 400 samples, 200 from each of the two categories $P(w_1) = P(w_2) = .5$ and $p(\mathbf{x}|w_i) \sim \mathcal{N}(\mu_i, \Sigma)$, where $\mathcal{N}$ is Gaussian distribution, $\mu_1 = (-3,4)^T$ and $\mu_2 = (4,-3)^T$ and

   $$\Sigma = \begin{bmatrix} 16 & 0 \\ 0 & 9 \end{bmatrix}$$

   Generate 200 test samples, 100 from each category.
3. Find the optimal learning rate empirically by trying a few values. Plot error versus epoch curve for both training and test sets. [2 Points]
4. Train to minimum error. Why is there no danger of overtraining in this case? [0.5 Point]
5. Why can we be sure that it is at least possible that this network can achieve the minimum (Bayes) error. [0.5 Point]
6. Now pre-process the data by subtracting off the mean and scaling with standard deviation in each dimension. Repeat the above, and find the optimal learning rate. [2 Points]