q1_code

Sunday, February 28, 2021 11:04 PM

Gradient Descent code for Q1 (gelu.py)

```python
import numpy as np
import math

# Gradient Descent, GeLU Activation
def sigmoid(x):
  return 1 / (1 + math.exp(-x))

def gelu(x):
  return x * sigmoid(1.702*x)

# gradient
def grad(x):
    return sigmoid(1.702*x) * (1+1.702*x*(1-sigmoid(1.702*x)))

# Set up variables and run GD
def main():
    lr = 0.1                                    # learning rate
    x = {0:-3,1:0,2:0,3:0}                      # store x[i]
    gelu_dict = {0:0, 1:0, 2:0, 3:0}            # store gelu(x[i])
    num_iters = 3

    for i in range(num_iters):
        x[i+1] = x[i] - lr * (grad(x[i]))       # update x[i+1]
        gelu_dict[i+1] = gelu(x[i+1])           # get gelu of new x[i+1]


if 'name == __main__':
    main()
```

GD with Momentum code (momentum.py)

```python
# Set up variables and run perceptron
def main():
    B = 0.9                                     # Set up variables
    lr = 1
    x = {0:-3,1:0,2:0,3:0}                      # Store x0, x1, x2, x3
    v = {0:grad(x[0]), 0:0, 0:0, 0:0}           # Store v0, v1, v2, v3
    gelu_dict = {0:0, 1:0, 2:0, 3:0}            # Store geLU of x0, x1, x2, x3
    num_iters = 3

    print(f"GD with Momentum")
    print(f"lr: {lr}, x_0: {x[0]}, num_iters: {num_iters}, B: {B}")

    for i in range(3):
        v[i+1] = (B*v[i]) + ((1-B)*grad(x[i]))  # update v[i+1]
        x[i+1] = x[i] - (lr * v[i+1])           # update x[i+1]
                                                # get gelu of new x[i+1]
        gelu_dict[i+1] = gelu(x[i+1])
        print(f"GeLU(x_{i+1}): ", gelu_dict[i+1])


if 'name == __main__':
    main()
```