Part 2 (12 marks):

4

$$f(r) = \frac{(r-7)^3}{3}$$

$$\left(\frac{r-7}{3}\right)^3 = 0$$

$$(r-3)^3 = 0$$

$$r - 3 = 0$$

$$r = 3$$

$$f(s) = (4s-20)^2 + (2s-10)^2 = 0$$

$$20s - 200s + 50$$

$a = 20$ $c = 500$

$b = -200$

$$s = \frac{-5 \pm \sqrt{5^2 - 400}}{2a}$$

$$s = \frac{200 \pm \sqrt{(200)^2 - 4(20 \times 500)}}{2(20)}$$

$$s = \frac{200 \pm \sqrt{0}}{40}$$

$$s = 5.$$

5

$$3 + (t) = \left(\frac{4t^2}{t} - 12\right)^2$$

$$\left[\frac{4t^2}{t} - 12\right]^2 = 0$$

$$(4t - 12)^2 = 0$$

$$4t = 12$$

$$t = 3$$

$$y = \cancel{3(5^2 + 5) + 20^4}$$

Part 2b (2 marks):

To find the true values of r, s, and t such that $E = y^2 = 0$, we need to solve the equation $E(r, s, t) = 0$ analytically. This means finding the exact solutions where the error function is equal to zero.

Given the error function $E = y^2$, where $y = f(r, s, t) = (r - 7)^3 + (4s - 20)^2 + (2s - 10)^2 + \left(\frac{4t^2}{t} - 12\right)^2$, we need to solve $y = 0$ to find the true values of r, s, and t where $E = 0$.

Let's set up the equation $y = 0$ and solve it for r, s, and t separately:

1. For r: $(r - 7)^3 = 0 \; r - 7 = 0 \; r = 7$
2. For s: $(4s - 20)^2 + (2s - 10)^2 = 0$

Both terms in the expression are squares, so they are always non-negative. The only way for the sum of two squares to be zero is if both squares are zero individually:

$(4s - 20)^2 = 0 \; 4s - 20 = 0 \; s - 5 = 0 \; s = 5$

$(2s - 10)^2 = 0 \; 2s - 10 = 0 \; s - 5 = 0 \; s = 5$

So, $s = 5$.

3. For t: $\left(\frac{4t^2}{t} - 12\right)^2 = 0 \; (4t - 12)^2 = 0 \; 4t - 12 = 0 \; t - 3 = 0 \; t = 3$

Therefore, the true values of r, s, and t such that $E = 0$ are: $r = 7 \; s = 5 \; t = 3$

These are the exact solutions where the error function is equal to zero, verified mathematically.

Task 2 – Regression using PyTorch (35%)

Part 1 (13 marks):

1. $y = 3(t^2 + 2)^2$, where $t = 2x + c$

In [ ]:
```
import torch

# Initialize variables
x = torch.tensor(1.0, requires_grad=True)
c = torch.tensor(1.0)

# Define equation 1
t = 2*x + c
y = 3 * (t**2 + 2)**2

# Calculate gradient
y.backward()

# Print the gradient dy/dx
print("Gradient dy/dx for equation 1:", x.grad)
```

```
Gradient dy/dx for equation 1: tensor(792.)
```

2. $y = 3(s^3 + s) + 2c^4$, where $s = 2x$

In [ ]:
```
# Reset gradient
x.grad = None

# Define equation 2
s = 2*x
y = 3 * (s**3 + s) + 2*c**4

# Calculate gradient
y.backward()

# Print the gradient dy/dx
print("Gradient dy/dx for equation 2:", x.grad)
```

```
Gradient dy/dx for equation 2: tensor(78.)
```

3. $y = 2t + c$, where $t = (p^2 + 2p + 3)^2$, $p = 2r^3 + 3r$, $r = 2q + 3$, $q = 2x + c$

In [ ]:
```
# Reset gradient
x.grad = None

# Define equation 3
q = 2*x + c
r = 2*q + 3
p = 2*r**3 + 3*r
t = (p**2 + 2*p + 3)**2
y = 2*t + c

# Calculate gradient
y.backward()

# Print the gradient dy/dx
print("Gradient dy/dx for equation 3:", x.grad)
```

```
Gradient dy/dx for equation 3: tensor(5.1347e+13)
```

Q4) Draw (by-hand) two separate diagrams/ computational maps for the functions shown in Q1 & Q2 of this Task. That is, the diagram/ computational map should highlight the significant sub-components, demonstrating how inputs x & c are converted to the final function �

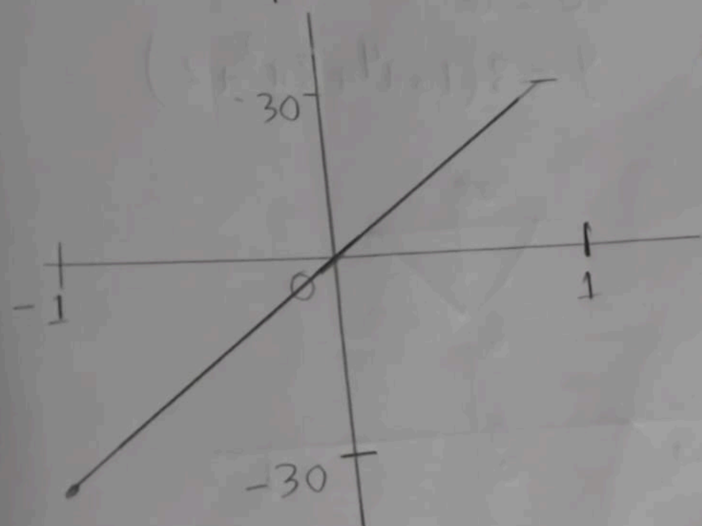
6

$y = 3(s^2 + s) + 2c^4 \quad s = 2x$

$y = 24x^3 + 6c + 2c^4$ Assume $c = 0$

when

x	-1	0	1
y	-30	0	30



7

$y = 3(-t^2+2)^2 \qquad t = 2x + c$

$y = 3(16c^4 + 8c^2x^2 + c^4 + 8x^2c + 4(x+2)$

Assume

$c = 0$

$y = 3(16x^4 + 8x^2 + 2)$