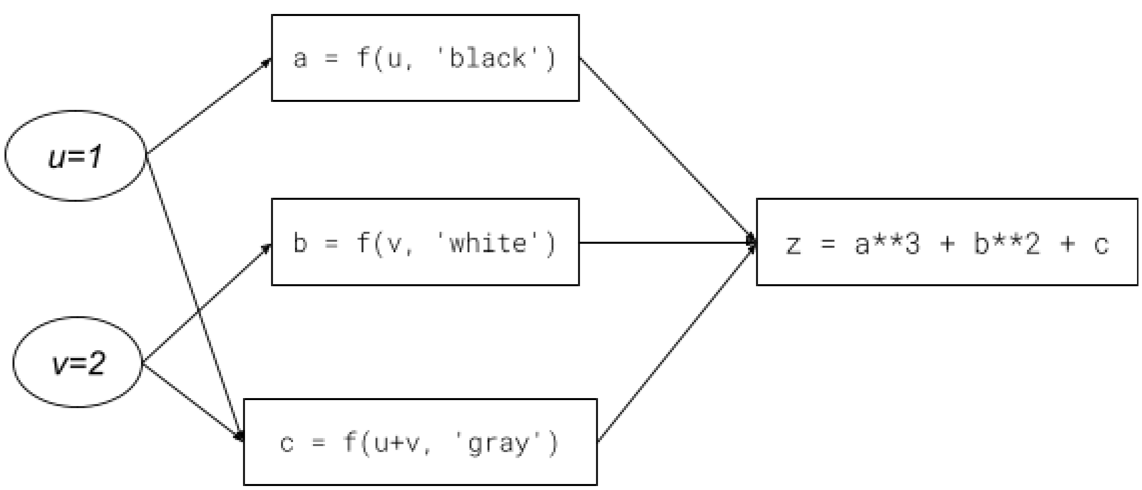
CSCI 4050U, Test 1

Feb 16, 2023

- Answer all questions.
- If **"I don't know"** is the only answer you provide, you will receive 20% of the total grade of that question.
- You are permitted to use any Web resources including the course notes. **However, you are forbidden to communicate with others, or access any online forums such as discords.**
- All academic dishonesty will be investigated to the fullest extent.
- If you witness any academic misconduct, you may report to https://forms.gle/ZcthT5ua8BeFQrSG6. All submissions are 100% anonymous.

Gradients and backpropagation

Consider the computational graph.



The definition of f(...) is given by:

```
import torch
```

```
def f(x:torch.Tensor, color:str) -> torch.Tensor:
    if color == 'black':
        return 1 * x
    elif color == 'white':
        return -x
    else:
        return 2*x
```

Q1. [6] Complete the following forward propagation calculations of the computational graph:

Node values	Edge values
u = 1	da/du = 1
v = 2	db/dv = -1
a = 1	dc/du = 2
b = -2	dc/dv = 2
c = 6	dz/da = 3a^2
z = 11	dz/db = 2b
	dz/dc = 1

====6

Q2. [6] Complete the following backpropagation of the computational graph:

dz/dz = 1
dz/du = 5
dz/dv = 6

====6

Linear algebra

Consider a potential function:

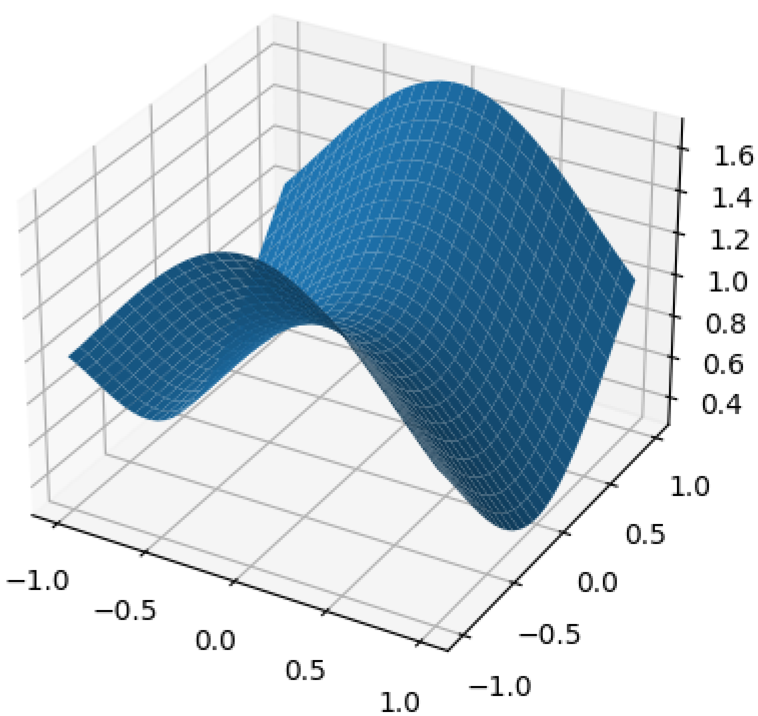
$f \colon R^2 \to R$

$f(x, y) \;=\; cos(x)^2 + sin(y)^2$

defined by:

```
def f(x:torch.Tensor, y:torch.Tensor) -> torch.Tensor:
    return torch.cos(x)**2 + torch.sin(y)**2
```

It is shown below as a surface plot, where f is the z-dimension of the 3D space. Thus, the surface is given by (x, y, f(x, y)).



Recall that the **projection** of a vector v=[v1, v2, v3] on the surface is the point on the surface (x, y, f(x, y)) that has the smallest distance to v.

Q3. [4] Define the function **dist** that measures the distance between a point on the surface and [0, 0, 0].

```python
def dist(x:torch.Tensor, y:torch.Tensor) -> torch.Tensor:
    # z coordinate is given by f(x, y)
    # d = sqrt((x-0)**2 + (y-0)**2 + (f(x,y)-0)**2)
    #   = sqrt(x**2 + y**2 + f(x, y)**2)
    z = f(x, y)
    return torch.sqrt(x**2 + y**2 + z**2)
```

====4

Q4. [15] Write a PyTorch training loop that will find the projection of [0, 0, 0] on the surface by performing gradient descent to minimize the distance.

```python
def train(x0:float, y0:float, epochs:int, learning_rate:float):
    # perform gradient descent to minimize dist to [0,0,0]
    x = torch.tensor(x0, requires_grad=True)
    y = torch.tensor(y0, requires_grad=True)
    history = [[x.item(), y.item(), f(x,y), dist(x,y)]]
    for i in range(epochs):
        d = dist(x, y)
        d.backward()
        with torch.no_grad():
            x -= learning_rate * x.grad
            y -= learning_rate * y.grad
        tmp = [x.item(), y.item(), f(x, y), dist(x,y)]
        history.append(tmp)
        x.grad.zero_()
        y.grad.zero_()
    return torch.round(torch.tensor(history), decimals=3)
```

====15

Q5. [4] Run your training loop with **10 epochs** with the initial starting point of **[1.0, 1.0, f(1.0, 1.0)]**, and learning rate of **0.3**.

Report the final result, with all your answers **rounded to the nearest two decimal places.**

x = 0.63
y = 0.00
z = 0.66
distance to [0, 0, 0] = 0.91

====4

Final grade=35