Topics
My posts
More
CATEGORIES
General
Site Feedback
All categories

2025 USA-NA-AIO Round 2, Problem 1, Part 9

USAAIO May 2025

Part 9 (10 points, coding task)

The purpose of this part is to guide you to learn using `torch.autograd.grad`.

For each given input (t, x), we not only compute $U(t, x \mid \theta)$ (output from the model), but also its 1st and 2nd order partial derivatives.

In PyTorch, these can be done by using `torch.autograd.grad`.

Part 9.1

Consider the following function

$$f(p, q) = p^2 + q^3 + pq^2.$$

Do the following tasks at $(p, q) = (1, 2)$:

1. Define tensors `p` and `q` that have values 1.0 and 2.0 (float data type), respectively, an identical shape `()` (that is, 0-dim), and `requires_grad = True`.
2. Compute tensor `f` according to the formula above.
3. Compute $\frac{\partial f(p,q)}{\partial p}$ by using

```
f_p = autograd.grad(f, p, create_graph=True)[0]
```

Print `f_p`.

Skip to main content

4. Compute $\frac{\partial f(p,q)}{\partial q}$ by using

```
f_q = autograd.grad(f, q, create_graph=True)[0]
```

Print `f_q`.

5. Compute $\frac{\partial^2 f(p,q)}{\partial p^2}$ by using

```
f_pp = autograd.grad(f_p, p, create_graph=True)[0]
```

Print `f_pp`.

6. Compute $\frac{\partial^2 f(p,q)}{\partial q^2}$ by using

```
f_qq = autograd.grad(f_q, q, create_graph=True)[0]
```

Print out `f_qq`.

7. Compute $\frac{\partial^2 f(p,q)}{\partial p \partial q}$ by using

```
f_pq = autograd.grad(f_p, q, create_graph=True)[0]
```

Print `f_pq`.

8. Compute $\frac{\partial^3 f(p,q)}{\partial q^3}$ by using

```
f_qqq = autograd.grad(f_qq, q, create_graph=True)[0]
```

Print `f_qqq`.

Skip to main content

Part 9.2

Consider the following function

$$g(x) = x^2.$$

Let x be a vector with values $0, 0.1, \cdots, 0.9, 1$.

Do the following tasks.

1. Generate `x` as a 1-dim tensor and set `x.requires_grad = True`.
2. Generate `g = x**2`. Thus, `g` has the same shape as `x`.
3. Define `g_x` to be an element-wise 1st-order derivative of function g with respect to x. Thus, `g_x` has the same shape as `x`.

   Write code to compute `g_x`. Print `g_x` and `g_x.shape`.

   Hint: by using `autograd.grad(f, x, create_graph=True)[0]`, tensor `x` can be with any dimension, but tensor `f` must be with dimension 0. In this problem, tensor `g` is not with dimension 0. So you need to think about how to address this issue.

4. Define `g_xx` to be an element-wise 2st-order derivative of function g with respect to x. Thus, `g_xx` has the same shape as `x`. Write code to compute `g_xx`. Print it out `g_xx` and `g_xx.shape`.

USAAIO May 2025

Skip to main content

```
### WRITE YOUR SOLUTION HERE ###

# Part 9.1
p = torch.tensor(1.0, requires_grad=True)
```

```
q = torch.tensor(2.0, requires_grad=True)

f = p**2 + q**3 + p*q**2

f_p = autograd.grad(f, p, create_graph=True)[0]
print(f_p)

f_q = autograd.grad(f, q, create_graph=True)[0]
print(f_q)

f_pp = autograd.grad(f_p, p, create_graph=True)[0]
print(f_pp)

f_qq = autograd.grad(f_q, q, create_graph=True)[0]
print(f_qq)

f_pq = autograd.grad(f_p, q, create_graph=True)[0]
print(f_pq)

f_qqq = autograd.grad(f_qq, q, create_graph=True)[0]
print(f_qqq)

# Part 9.2

x = torch.linspace(0, 1, 10)
x.requires_grad_(True)

g = x**2
g_x = autograd.grad(torch.sum(g), x, create_graph=True)[0]
print(g_x)
print(g_x.shape)

g_xx = autograd.grad(torch.sum(g_x), x, create_graph=True)[0]
print(g_xx)
print(g_xx.shape)


""" END OF THIS PART """
```

Skip to main content

Related topics

Topic	Replies	Activity
2025 USA-NA-AIO Round 1, Problem 2, Part 2	2	Dec 2025
2026 USAAIO Round 1 Sample problems, Problem 6	1	28d
2025 USA-NA-AIO Round 2, Problem 1, Part 6	1	May 2025
2025 USA-NA-AIO Round 2, Problem 1, Part 7	1	May 2025
2025 USA-NA-AIO Round 1, Problem 3, Part 13	3	12d