1 Automatic Differentiation

1.1 Multiple choice (5 pts)

Which of the following statements is true? In all these questions, you can assume the functions are differentiable.

True	False	
☐	☐	Reverse-mode automatic differentiation computes the derivative of the input of a function with regards to its output.
☐	☐	To differentiate a function that has one input and 10 outputs, we should use reverse-mode automatic differentiation.
☐	☐	The Jacobian of a function is a symmetric matrix.
☐	☐	Reverse-mode differentiation is generally more expensive to use than forward-mode automatic differentiation because it requires recording all the operations.
☐	☐	Central finite differences always gives the same result as forward-mode automatic differentiation.

Solution: True: D, False: A, B, C, E

1.2 Jacobian (5 pts)

Consider a function $f : \mathbb{R}^n \to \mathbb{R}^m$. Find a way to use automatic differentiation in order to compute the Jacobian matrix $\mathbf{J}_f(\mathbf{x})$ of that function in the least number of differentiation passes possible. Describe the steps you would take and which AD mode you would use.

Solution: If $n < m$, we should use forward-mode AD. We can compute n passes of forward-mode AD to compute the products $\mathbf{J}_f(\mathbf{x})\mathbf{e}_i$, where $\mathbf{e}_i$ is the i-th canonical basis vector of $\mathbb{R}^n$. This will give us the rows of the Jacobian.
Conversely, if $n > m$, we should use reverse-mode AD. We can compute m passes of reverse-mode AD to compute the products $\mathbf{d}_i^T\mathbf{J}_f(\mathbf{x})/$, where $\mathbf{d}_i$ is the i-th canonical basis vector of $\mathbb{R}^m$. This will give us the columns of the Jacobian.

1.3 Differentiation of a function (5 pts)

Consider the following computation:

```
a = exp(-2*x)
b = 3 * a + 1
y = sqrt(b)
```

We now want to apply automatic-differentiation to this computation. Write the sequence of steps that forward-mode automatic differentiation would take to compute the derivative of y with regards to x. Do the same for reverse-mode automatic differentiation.

Solution: Forward-mode AD:

```
dx = 1
da = -2*dx*exp(-2*x) = -2 * exp(-2*x)
db = 3 * da = -6 * exp(-2*x)
dy = - db / (2*sqrt(b)) = -3 * exp(-2*x) / sqrt(b)
```

Reverse-mode AD:

```
dy = 1
db = - dy / (2*sqrt(b)) = -1 /(2*sqrt(b))
da = 3 * db = -3 / (2 * sqrt(b))
dx = -2*da*exp(-2*x) = -3 * exp(-2*x) / sqrt(b)
```

1.4 Solving a linear system (5 pts)

We provide you with a black-box function $linear_solve(b)$, that takes as input a vector b and returns the solution x of the linear system $Ax = b$ for a constant symmetric matrix A.

(1) Is solving a linear system a differentiable operation ? Why ? (2 pts)

Solution: Solving a (square) linear system is the same operation as computing a matrix-vector product: $x = A^{-1}b$, which is differentiable.

(2) similar to Homework 4, write the backward pass of the function: (3pts)

Solution:

```
@mytorch_bwd
def solve_system(b, out):
    return linear_solve(A, out.grad),
```