1.

C/C++ 10

6 4

project —— ——

IR

Node IRVisitor SDD, SDT

project

pass

project

2.

2.1

Tensorflow, PyTorch

```
X is a tensor of shape [4, 3, 28, 28]
T is a label of shape [4, 8 * 28 * 28]
Y1 = Conv2d(X, kernel=(8, 3, 3, 3), padding=1, stride=1) # result shape is [4, 8, 28, 28]
Y2 = flatten(Y1)  # result shape is [4, 8 * 28 * 28]
loss = mse_loss(Y2, T)  # loss is scalar
```

X

loss loss 1 Y2 Y2 loss

mse_loss mse_loss grad_mse_loss Y2

Y1 Y2 flatten flatten

grad_flatten Y1 X Conv2d

X

/

project

/

2.2

$Output = expr(Input_1, Input_2, ..., Input_n)$ $Output, Input_i$
, $expr()$ loss $Output$
$dOutput = \frac{\partial loss}{\partial Output}$ loss $dInput_i = \frac{\partial loss}{\partial Input_i}$
Ọ ỗ K O ı ẩ **IR** ṕ ɗ ῶ O ı ẩ ɺ ʌ —"O**case** ª ḓ ˋ ˥ ’

2.3

```
C<M, N>[i, j] = A<M, K>[i, k] * B<K, N>[k, j]
```

project
$loss$ C dC C dC
dA

$$dA[i,k] = \frac{\partial loss}{\partial A[i,k]} = \sum_j \frac{\partial loss}{\partial C[i,j]} \times \frac{\partial C[i,j]}{\partial A[i,k]} = dC[i,j] \times B[k,j]$$

A

```
dA<M, K>[i, k] = dC<M, N>[i, j] * B<K, N>[k, j]
```

C

```
for (int i = 0; i < M; ++i) {
  for (int k = 0; k < K; ++k) {
    dA[i][k] = 0.0;
    for (int j = 0; j < N; ++j) {
      dA[i][k] += dC[i][j] * B[k][j];
    }
  }
}
```

B

```
dB<K, N>[k, j] = dC<M, N>[i, j] * A<M, K>[i, k]
```

3. Project

project 10 project case
json "grad_to" /

case1 json

```
{
  "name": "grad_case1",
  "ins": ["A", "B"],
  "outs": ["C"],
  "data_type": "float",
  "kernel": "C<4, 16>[i, j] = A<4, 16>[i, j] * B<4, 16>[i, j] + 1.0;",
  "grad_to": ["A"]
}
```

A

```
dA<4, 16>[i, j] = dC<4, 16>[i, j] * B<4, 16>[i, j]
```

dC C
d

json
C/C++ kernels/ cmake project
solution run2.cc run2.cc

3.1

10

- NP

10

- project trivial

-

10

1. element-wise
2.
3. dense MTTKRP
4.
5.
6. flatten
7. broadcast
8. blur

ground truth run2.cc

case1 test_case1

```
bool test_case1(std::mt19937 &gen, std::uniform_real_distribution<float> &dis) {
    // "C<4, 16>[i, j] = A<4, 16>[i, j] * B<4, 16>[i, j] + 1.0;"
    // "dA<4, 16>[i, j] = dC<4, 16>[i, j] * B<4, 16>[i, j];"
    float B[4][16] = {{0}};
    float dA[4][16] = {{0}};
    float dC[4][16] = {{0}};
    float golden[4][16] = {{0}};
    // initialize
    for (int i = 0; i < 4; ++i) {
        for (int j = 0; j < 16; ++j) {
            B[i][j] = dis(gen);
            dC[i][j] = dis(gen);
        }
    }
    // compute golden
    for (int i = 0; i < 4; ++i) {
        for (int j = 0; j < 16; ++j) {
            golden[i][j] = dC[i][j] * B[i][j];
        }
    }
    try {
        grad_case1(B, dC, dA);
    } catch (...) {
        std::cout << "Failed because of runtime error\n";
        return false;
    }

    // check
    for (int i = 0; i < 4; ++i) {
        for (int j = 0; j < 16; ++j) {
            if (std::abs(golden[i][j] - dA[i][j]) >= 1e-5) {
                std::cout << "Wrong answer\n";
                return false;
            }
        }
    }
    // correct
    return true;
}
```

golden

4.

4.1

project2	2020	5	16	12:00
project2	2020	6	21	12:00

4.2

4.2.1

project

- compiler2020spring@163.com
- github compiler2020spring@163.com

4.2.2

1. pdf project2
2. / run2.h, run2.cc, clean2.cc
3. stdin json
4. C++11 gcc 4.8.5

build

```
mkdir build
cd build
cmake ..
make -j 4
cd project2
./test2
```

C++11

project2

github public

public

```
git clone --recursive <  github  > CompilerProject
cd CompilerProject
mkdir build
cd build
cmake ..
make -j 4
cd project2
./test2
```

4.3

project 20% 20 5 pdf 15
pdf

pdf

- 2
- 1
- 1
- 1

- case ( )

case	0	1	2	3	4	5	6	7	8	9	10
	0	2.25	4.5	6.75	9	10.5	12	12.75	13.5	14.25	15

- ./test2

4.4

- run2.h/run2.cc/clean2.cc
-
-
-

1. 39 3 13
2. x
3. case json kernels/
4.
5. 12

4

9 60% 6 12 80%