Tensor Gradients

Matrix Operations

$A = \mathbb{R}^{j \times i}, B = \mathbb{R}^{i \times k}$

operations

$$AB_{jk} = \sum_i a_{ji} b_{ik}$$

gradients

$$\frac{\partial AB}{\partial A_{ji}} = \sum_k b_{ik}$$

$$\frac{\partial AB}{\partial B_{ik}} = \sum_j a_{ji}$$

Tensor Operations

$A = \mathbb{R}^e, \;\; B = \mathbb{R}^e, \;\; e = \prod e_i, \;\; scalar^e \mid_e = scalar$

operations

$$A + B \mid_e = \quad a_e + b_e$$

$$A - B \mid_e = a_e - b_e$$

$$A \otimes B \mid_e = a_e b_e$$

$$\frac{A}{B} \mid_e = \frac{a_e}{b_e}$$

$$F(A) = F(a_e)$$

gradients

$$\frac{\partial A + B}{\partial A} = 1^e$$

$$\frac{\partial A + B}{\partial B} = 1^e$$

$$\frac{\partial A - B}{\partial A} = 1^e$$

$$\frac{\partial A - B}{\partial B} = -1^e$$

$$\frac{\partial A \otimes B}{\partial A} = B$$

$$\frac{\partial A \otimes B}{\partial B} = A$$

$$\frac{\partial A/B}{\partial A} = \frac{1^e}{B}$$
$$\frac{\partial A/B}{\partial B} = \frac{-A}{B^2}$$
$$\frac{\partial F(A)}{\partial A} = F'(A)$$

Tensor Rules

conversions

$A = \mathbb{R}^{e\times 1}$, $B = \mathbb{R}^{e\times e_{new}}$,
$def.\ e \times e_{new} = e_1 \times e_2 \ldots e_j \times e_{new} \times e_{j+1} \cdots \times e_m = e_1 \ldots e_m \times e_{new} = e_{new} \times e_1 \ldots e_m$

$$A \times scalar \rightarrow A \otimes scalar^{e1}$$

$$newA_{e\times e_{new}} = A_{e\times 1},\ \ e \times 1 = e \times e_b\ s.t.\ e_b = 1$$

$$A_{e\times 1} = \sum_{e_{new}} newA$$

all are irreversible

e.g.

1. [1] * [1,2,3,4]
   -> [1, 1, 1, 1] * [1, 2, 3, 4]
2. [[1], [2]] * [[1,2,3,4], [2,3,4,5]]
   -> [[1, 1, 1, 1], [2, 2, 2, 2]] * [[1,2,3,4], [2,3,4,5]]
3. [[[[2.0, 4.0, 6.0, 8.0]]]] * [[[2.0, 4.0, 6.0, 8.0], [3.0, 5.0, 7.0, 9.0]]]
   -> [[[[2.0, 4.0, 6.0, 8.0], [2.0, 4.0, 6.0, 8.0]]]] * ...

gradients

$$\frac{\partial A\ op\ B}{\partial A}\ |_e = \sum_{e_{new}} \frac{\partial newA\ op\ B}{\partial newA}|_{e\times e_{new}}$$

$$\frac{\partial scalar\ op\ B}{\partial scalar} = \sum_{e} \frac{\partial scalar^e\ op\ B}{\partial scalar^e}|_e$$

chain rule

tensor operations

$$\frac{\partial C}{\partial B} = \frac{\partial C}{\partial A} \otimes \frac{\partial A}{\partial B}$$

matrix multiply

$$C = AB$$
$$\frac{\partial E}{\partial A} = \frac{\partial E}{\partial C} B^T$$
$$\frac{\partial E}{\partial B} = A^T \frac{\partial E}{\partial C}$$

tensorflow

```
Matrix Multiply Gradients
AB
grad = dE/dAB
t_a = op.get_attr("transpose_a")
t_b = op.get_attr("transpose_b")

if not t_a and not t_b:
  return (grad * B.transpose, A.transpose * grad)
elif not t_a and t_b:
  return (grad * B, grad.transpose * A)
elif t_a and not t_b:
  return (B * grad.transpose, A * grad)
elif t_a and t_b:
  return (B.transpose * grad.transpose, grad.transpose * A.transpose)
```

transpose_a, b という attribute　は matmul(x, y, transpose_a=True, transepose_b=True) というtranspose_a, bにTrueを入れた場合のみ入る．　つまり，　普通に使うのであれば,

return (grad *B.transpose, A.transpose* grad)

が呼ばれる．

```
@ops.RegisterGradient("MatMul")
def _MatMulGrad(op, grad):
  t_a = op.get_attr("transpose_a")
  t_b = op.get_attr("transpose_b")
  if not t_a and not t_b:
    return (math_ops.matmul(
        grad, op.inputs[1], transpose_b=True), math_ops.matmul(
            op.inputs[0], grad, transpose_a=True))
  elif not t_a and t_b:
    return (math_ops.matmul(grad, op.inputs[1]), math_ops.matmul(
        grad, op.inputs[0], transpose_a=True))
  elif t_a and not t_b:
    return (math_ops.matmul(
        op.inputs[1], grad, transpose_b=True),
            math_ops.matmul(op.inputs[0], grad))
  elif t_a and t_b:
    return (math_ops.matmul(
        op.inputs[1], grad, transpose_a=True, transpose_b=True),
            math_ops.matmul(
                grad, op.inputs[0], transpose_a=True, transpose_b=True))
```