

# einsum

einsum

Free indices Summation indices

- $a_{ik}$   $b_{kj}$   $c_{ij}$
- $c_{ij} = \sum_k a_{ik} b_{kj}$

einsum

- $c_{ij} = \sum_k a_{ik} b_{kj}$  equation  $c_{ij}$   $a_{ik}$   $b_{kj}$
- $c_{ij} = \sum_k a_{ik} b_{kj}$  equation  $c_{ij}$   $a_{ik}$   $b_{kj}$
- $c_{ij} = \sum_k a_{ik} b_{kj}$  equation  $c_{ij}$   $a_{ik}$   $b_{kj}$

einsum

einsum

- $c_{ij} = \sum_k a_{ik} b_{kj}$  equation  $c_{ij}$   $a_{ik}$   $b_{kj}$
- $c_{ij} = \sum_k a_{ik} b_{kj}$  equation  $c_{ij}$   $a_{ik}$   $b_{kj}$

```
a = torch.randn(2,3,5,7,9)
# i = 7, j = 9
b = torch.einsum('...ij->...ji', [a])
```

bilinear transformation

```
np_a = a.numpy()
np_b = b.numpy()
```

```

np_c = c.numpy()
np_out = np.empty((2, 5), dtype=np.float32)

np_out = torch.einsum('ik,jkl,il->ij', [a, b, c]).numpy()
# ik broadcast  ikl
# il broadcast  ikl
# 'ik,jkl,il->ij'  'ikl,jkl,ikl->ij'

for i in range(0, 2):
    for j in range(0, 5):
        #  k  l
        sum_result = 0
        for k in range(0, 3):
            for l in range(0, 7):
                sum_result += np_a[i, k] * np_b[j, k, l] * np_c[i, l]
        np_out[i, j] = sum_result

```



```

a = torch.rand(2,3)
b = torch.rand(3,4)
c = torch.einsum("ik,kj->ij", [a, b])
#  torch.mm(a, b)

```

equation

c c[i, j] a[i, k] b[k, j] k

Revision #1

Created 11 January 2025 09:46:28 by Colin

Updated 12 January 2025 06:36:11 by Colin