



Small primer in tensor manipulation

Tensor manipulation

- In ML, we mostly operate with tensors
- These are matrices of numbers of multiple dimensions.
- Often (batch, ...others)
- In text: (batch, seq_pos) or (batch,seq_pos, d_resid)
- In images: (batch, hight, width, channel) or (batch channel, hight, width)
- Manipulating these Tensors is important

rearrange

- **Umordnen von Dimensionen:**

- Ändert die Reihenfolge der Dimensionen

```
output = einops.rearrange(x, "b c h w -> b h w c")
```

- **Splitting von Dimensionen**

- Teilt eine Dimension in mehrere auf

```
output = einops.rearrange(x, "b (h1 h2) w -> b h1 h2 w", h1=2)
```

- **Combining von Dimensionen**

- Fasst mehrere Dimensionen zu einer zusammen

```
output = einops.rearrange(x, "b h w c -> b (h w) c")
```

- **Pattern mit führenden Dimensionen**

- Behält beliebig viele führende Dimensionen bei

```
output = einops.rearrange(x, "... h w -> ... (h w)")
```

reduce

- **Mean Reduktion:**

- Berechnet den Durchschnitt entlang einer Dimension

```
output = einops.reduce(x, "b c h w -> b c", "mean")
```

- **Max Reduktion**

- Findet das Maximum entlang einer Dimension (das selbe existiert mit „min“, „min“ und „sum“)

```
output = einops.reduce(x, "b c h w -> b c", "max")
```

- **Custom Reduktion**

- Wendet eine benutzerdefinierte Reduktionsfunktion an

```
output = einops.repeat(x, "h w -> h (2 w)")
```

repeat

- **Einfache Wiederholung**

- Wiederholt eine Dimension mehrfach

```
output = einops.repeat(x, "h w -> h (2 w)")
```

- **Wiederholung mit neuer Dimension**

- Fügt eine neue Dimension durch Wiederholung hinzu

```
output = einops.repeat(x, "h w -> b h w", b=32)
```

Casting

- Tensors can be operated together even if the size does not match
- Non-fitting dimension must be length 1, then it gets copied along that dimension

```
a = torch.rand((2,3,4))
b = torch.rand((2,3,5))
print((a+b).shape)
```

⊗ 0.0s

RuntimeError Traceback (most recent call last)

Cell In[4], line 3

```
1 a = torch.rand((2,3,4))
2 b = torch.rand((2,3,5))
----> 3 print((a+b).shape)
```

RuntimeError: The size of tensor a (4) must match the size of tensor b (5) at non-singleton dimension 2

```
a = torch.rand((2,3,4))
b = torch.rand((2,3,1))
print((a+b).shape)
```

✓ 0.0s

torch.Size([2, 3, 4])