In [1]:
```
%matplotlib inline
import time
import torch
import torchvision
from torchvision import transforms
from d2l import torch as d2l

d2l.use_svg_display()
```

In [2]:
```
class FashionMNIST(d2l.DataModule):  #@save
    def __init__(self, batch_size=64, resize=(28, 28)):
        super().__init__()
        self.save_hyperparameters()

        trans = transforms.Compose([transforms.Resize(resize),
                                    transforms.ToTensor()])
        self.train = torchvision.datasets.FashionMNIST(
            root=self.root, train=True, transform=trans, download=True)
        self.val = torchvision.datasets.FashionMNIST(
            root=self.root, train=False, transform=trans, download=True)
```

In [3]:
```
data = FashionMNIST(resize=(32, 32))
len(data.train), len(data.val)
```

```
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-images-idx3-ubyte.gz to ../data/Fas
hionMNIST/raw/train-images-idx3-ubyte.gz
100%|████████████████████████████| 26421880/26421880 [00:05<00:00, 4731535.78it/s]
Extracting ../data/FashionMNIST/raw/train-images-idx3-ubyte.gz to ../data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/train-labels-idx1-ubyte.gz to ../data/Fas
hionMNIST/raw/train-labels-idx1-ubyte.gz
100%|████████████████████████████████| 29515/29515 [00:00<00:00, 99545.58it/s]
Extracting ../data/FashionMNIST/raw/train-labels-idx1-ubyte.gz to ../data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-images-idx3-ubyte.gz to ../data/Fash
ionMNIST/raw/t10k-images-idx3-ubyte.gz
100%|████████████████████████████| 4422102/4422102 [00:01<00:00, 3521825.30it/s]
Extracting ../data/FashionMNIST/raw/t10k-images-idx3-ubyte.gz to ../data/FashionMNIST/raw

Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz
Downloading http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz to ../data/Fash
ionMNIST/raw/t10k-labels-idx1-ubyte.gz
100%|████████████████████████████████| 5148/5148 [00:00<00:00, 3375375.49it/s]
Extracting ../data/FashionMNIST/raw/t10k-labels-idx1-ubyte.gz to ../data/FashionMNIST/raw
```

Out[3]:
```
(60000, 10000)
```

In [4]:
```
data.train[0][0].shape
```

Out[4]:
```
torch.Size([1, 32, 32])
```

In [5]:
```
@d2l.add_to_class(FashionMNIST)  #@save
def text_labels(self, indices):
    labels = ['t-shirt', 'trouser', 'pullover', 'dress', 'coat',
              'sandal', 'shirt', 'sneaker', 'bag', 'ankle boot']
    return [labels[int(i)] for i in indices]
```

In [6]:
```
@d2l.add_to_class(FashionMNIST)  #@save
def get_dataloader(self, train):
    data = self.train if train else self.val
    return torch.utils.data.DataLoader(data, self.batch_size, shuffle=train,
                                       num_workers=self.num_workers)
```

In [7]:
```
X, y = next(iter(data.train_dataloader()))
print(X.shape, X.dtype, y.shape, y.dtype)
```

```
torch.Size([64, 1, 32, 32]) torch.float32 torch.Size([64]) torch.int64
```

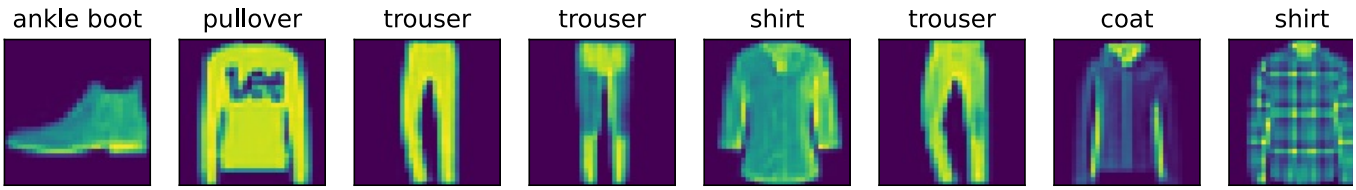
In [8]:
```
tic = time.time()
for X, y in data.train_dataloader():
    continue
f'{time.time() - tic:.2f} sec'
```

Out[8]:
```
'5.13 sec'
```

In [9]:
```
def show_images(imgs, num_rows, num_cols, titles=None, scale=1.5):  #@save
```

```
    raise NotImplementedError
```

In [10]:
```
@d2l.add_to_class(FashionMNIST)  #@save
def visualize(self, batch, nrows=1, ncols=8, labels=[]):
    X, y = batch
    if not labels:
        labels = self.text_labels(y)
    d2l.show_images(X.squeeze(1), nrows, ncols, titles=labels)
batch = next(iter(data.val_dataloader()))
data.visualize(batch)
```



Discussion: In this chapter, we worked on loading and visualizing images using an actual dataset. Usually, examples tend to use the handwritten digits dataset, so this was a bit different and interesting. The code for resizing the data to a specified size and loading it in batches using 'get_dataloader' felt somewhat unfamiliar but fascinating. The performance measurement showed a result of 5.13 seconds, which made me realize how important time is in these tasks.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js