一、变分自编码器生成MNIST 手写数字（结合代码描述实现步骤以及提交下面要求提交的结果）

推荐使用高斯分布随机初始化模型参数，可以避免一部分模式坍塌问题。 1、模型架构： ① 编码器（全连接层）： 输入图片维度：784 (28 × 28) 输出层维度（ReLU）： 400 ② 生成均值（全连接层）： 输入层维度：400 输出层维度：20 ③ 生成标准差（全连接层）： 输入层维度：400 输出层维度：20 ④ 使用均值和标准差生成隐变量z ⑤ 解码器（全连接层）： 输入维度：20 隐藏层维度（ReLU）： 400输 出层维度（Sigmoid）： 784 训练完网络，需要提交重构损失和KL散度的随迭代次数的变化图，以及10 张 生成的手写数字图片。

导入库

In [28]:
```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")
```

```
Using device: cuda
```

用一个类去封装并且定义VAE模型（实现上述模型架构）

采用了重参数化技巧：z=μ+σ·ϵ，σ=exp(0.5·logvar)，ϵ~N(0,I)。主要目的为：保留可微性：使得模型在采样隐变量 z 时仍然可以进行反向传播。引入随机性：通过标准正态分布的噪声 ϵ 来保持采样过程的随机性，使得 VAE 的隐变量具有生成分布的能力。

In [29]:
```
class VAE(nn.Module):
    def __init__(self):
        super(VAE, self).__init__()

        # 编码器
        self.fc1 = nn.Linear(784, 400)
        self.fc_mu = nn.Linear(400, 20)
        self.fc_logvar = nn.Linear(400, 20)

        # 解码器
        self.fc3 = nn.Linear(20, 400)
        self.fc4 = nn.Linear(400, 784)

    #将输入数据压缩为隐变量的均值和对数方差
    def encode(self, x):
        h1 = torch.relu(self.fc1(x))
        return self.fc_mu(h1), self.fc_logvar(h1)

    #采用了重参数化技巧
    def reparameterize(self, mu, logvar):
        std = torch.exp(0.5 * logvar)
        eps = torch.randn_like(std)
        return mu + eps * std

    def decode(self, z):
        h3 = torch.relu(self.fc3(z))
        return torch.sigmoid(self.fc4(h3))
```

```
    def forward(self, x):
        mu, logvar = self.encode(x)
        z = self.reparameterize(mu, logvar)
        return self.decode(z), mu, logvar
```

定义损失函数

损失函数由重构误差和KL散度构成 BCE=-sum(i=1->N)(XilogXi'+(1-Xi)log(1-Xi')) Xi':重构生成的数据 ,N=28x28 KL 散度可以强制隐变量分布接近标准正态分布，形成一个有规律的潜在空间结构。

$$\mathrm{KLD} = -\frac{1}{2}\sum_{j=1}^{M}\left(1+\log(\sigma_j^2)-\mu_j^2-\sigma_j^2\right)$$

μ：编码器生成的隐变量分布的均值向量。

$\sigma^2 = \exp(\mathrm{logvar})$：隐变量分布的方差，其中 logvar 是对数方差。

M：隐变量的维度（在这个模型中为 20）。

$$\mathrm{BCE} = -\sum_{i=1}^{N}\left[x_i\log(\hat{x}_i)+(1-x_i)\log(1-\hat{x}_i)\right]$$

In [30]:
```
def loss_function(recon_x, x, mu, logvar):
    BCE = nn.functional.binary_cross_entropy(recon_x, x, reduction='sum')#sum表示返回
    KLD = -0.5 * torch.sum(1 + logvar - mu.pow(2) - logvar.exp())
    return BCE + KLD, BCE, KLD
```

加载数据集，并使用数据加载器，transform：transforms.ToTensor() 将输入图像转换为 PyTorch 张量，并将像素值从 0-255（整型）范围缩放到 0-1（浮点型）范围。对于灰度图像（单通道），它将生成一个形状为 (1, H, W) 的张量；对于 RGB 图像，它将生成 (3, H, W) 的张量，其中 H 和 W 为图像的高度和宽度。transforms.Lambda(lambda x: x.view(-1)) 是一个自定义转换操作。它对输入的张量 x 应用 view(-1) 操作，将其展平成一维向量。view(-1) 会自动计算并调整张量的大小，使得所有元素都在一个一维张量中。比如说：对于 28x28 的灰度图像（例如 MNIST 数据集），在应用 ToTensor() 后张量形状为 (1, 28, 28)。经过 view(-1) 处理后，张量形状变成 (784,)。

In [31]:
```
transform = transforms.Compose([transforms.ToTensor(), transforms.Lambda(lambda x:
train_dataset = datasets.MNIST(root='./data', train=True, transform=transform, down
train_loader = DataLoader(train_dataset, batch_size=128, shuffle=True)
```

初始化模型，优化器，定义模型超参数

In [32]:
```
# 初始化模型、优化器
vae = VAE().to(device)
optimizer = optim.Adam(vae.parameters(), lr=1e-3)
# 训练模型
num_epochs = 100
reconstruction_losses = []
kl_losses = []
```

开始训练

In [33]:
```
for epoch in range(num_epochs):
    vae.train()#nn.Module内置
    train_loss = 0
    reconstruction_loss = 0
    kl_loss = 0

    for batch_idx, (data, _) in enumerate(train_loader):#（data,_）因为是无监督学习用
        data = data.to(device)#将数据迁移到GPU
        optimizer.zero_grad()
        recon_batch, mu, logvar = vae(data)#调用forward()函数
        loss, bce, kld = loss_function(recon_batch, data, mu, logvar)
        loss.backward()
        train_loss += loss.item()
        reconstruction_loss += bce.item()
        kl_loss += kld.item()
        optimizer.step()

    reconstruction_losses.append(reconstruction_loss / len(train_loader.dataset))
    kl_losses.append(kl_loss / len(train_loader.dataset))

    print(f"Epoch {epoch+1}, Loss: {train_loss / len(train_loader.dataset)}, BCE: {r
```

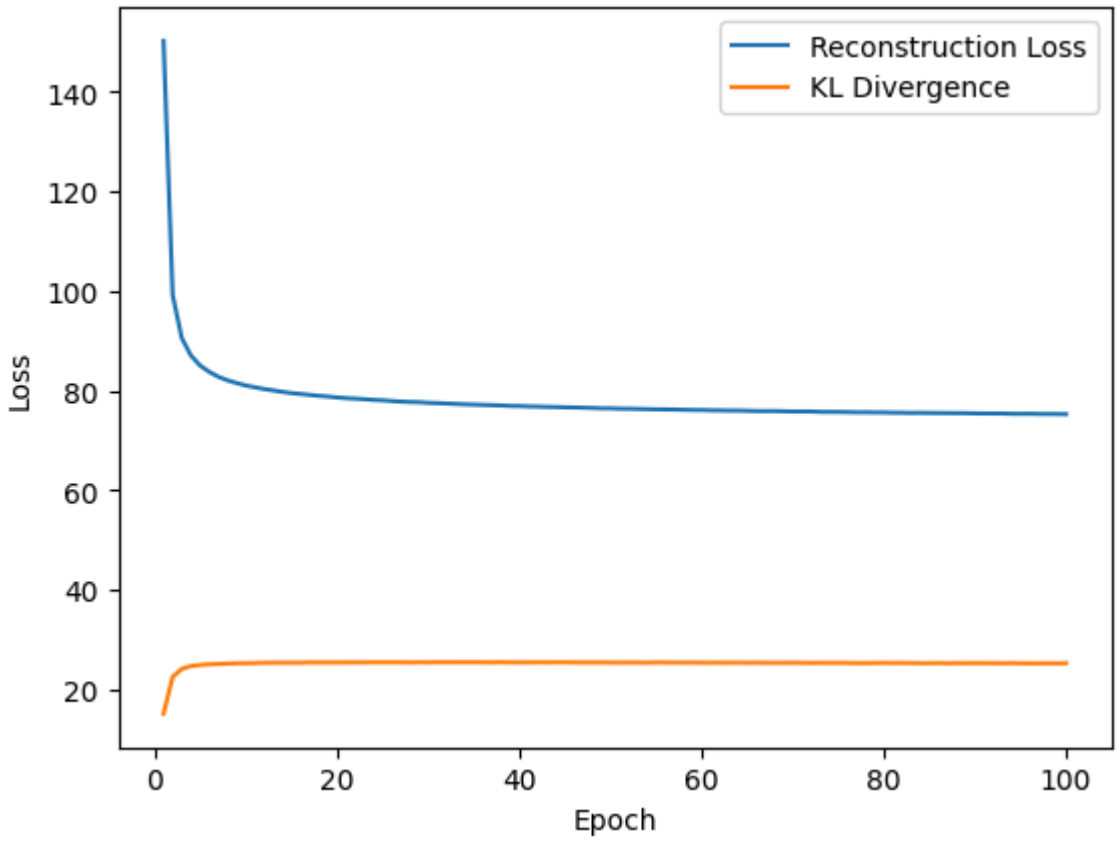
Epoch 1, Loss: 165.3020716796875, BCE: 150.12601051432293, KLD: 15.176061004479726
Epoch 2, Loss: 121.69891692708333, BCE: 99.14765349934896, KLD: 22.551263517252604
Epoch 3, Loss: 114.66297008463542, BCE: 90.51297451171875, KLD: 24.14999569091797
Epoch 4, Loss: 111.74801271158854, BCE: 87.03184103190104, KLD: 24.716171752929686
Epoch 5, Loss: 109.97101922200521, BCE: 85.02163464355469, KLD: 24.949384366861977
Epoch 6, Loss: 108.8215818359375, BCE: 83.74659449869792, KLD: 25.074987268066405
Epoch 7, Loss: 107.9133006998698, BCE: 82.75955424804687, KLD: 25.153746435546875
Epoch 8, Loss: 107.27075862630208, BCE: 82.05127303059896, KLD: 25.219485298665365
Epoch 9, Loss: 106.79034085286459, BCE: 81.5048848795573, KLD: 25.285456022135417
Epoch 10, Loss: 106.3172982421875, BCE: 81.01563967285156, KLD: 25.301658557128906
Epoch 11, Loss: 105.98319166666667, BCE: 80.65520839029948, KLD: 25.327983093261718
Epoch 12, Loss: 105.67151702473959, BCE: 80.29949213867188, KLD: 25.372024719238283
Epoch 13, Loss: 105.42885452473958, BCE: 80.0298123860677, KLD: 25.39904217529297
Epoch 14, Loss: 105.1284846516927, BCE: 79.73580721028645, KLD: 25.392677486165365
Epoch 15, Loss: 104.88894462890624, BCE: 79.49249405924479, KLD: 25.39645051676432
Epoch 16, Loss: 104.69861910807292, BCE: 79.30852049967447, KLD: 25.390098315429686
Epoch 17, Loss: 104.57295413411458, BCE: 79.11826509602865, KLD: 25.45468889567057
Epoch 18, Loss: 104.38634796549479, BCE: 78.94035893554687, KLD: 25.44598895670573
Epoch 19, Loss: 104.2473922688802, BCE: 78.80339705403645, KLD: 25.443995340983072
Epoch 20, Loss: 104.06151555989584, BCE: 78.61666559244792, KLD: 25.44485008951823
Epoch 21, Loss: 103.91801266276042, BCE: 78.4623317545573, KLD: 25.455681026204427
Epoch 22, Loss: 103.81981560872396, BCE: 78.36809670410156, KLD: 25.451718977864584
Epoch 23, Loss: 103.71546238606771, BCE: 78.25808345540365, KLD: 25.457379016113283
Epoch 24, Loss: 103.58360180664063, BCE: 78.11487314453124, KLD: 25.468728568522135
Epoch 25, Loss: 103.51142293294271, BCE: 78.04062459309895, KLD: 25.470798482259113
Epoch 26, Loss: 103.36306186523437, BCE: 77.89810404459635, KLD: 25.464957800292968
Epoch 27, Loss: 103.2553115234375, BCE: 77.77566677246094, KLD: 25.47964491373698
Epoch 28, Loss: 103.14201655273438, BCE: 77.70055619303385, KLD: 25.441460591634115
Epoch 29, Loss: 103.1394596516927, BCE: 77.66292403971354, KLD: 25.476535579427082
Epoch 30, Loss: 102.98273901367187, BCE: 77.53151927083333, KLD: 25.451219665527343
Epoch 31, Loss: 102.96655494791666, BCE: 77.46937898763021, KLD: 25.49717589518229
Epoch 32, Loss: 102.86257613932291, BCE: 77.39061297200521, KLD: 25.47196317952474
Epoch 33, Loss: 102.79892146809895, BCE: 77.3116362467448, KLD: 25.487285013834637
Epoch 34, Loss: 102.7309748046875, BCE: 77.23455688476562, KLD: 25.496418082682293
Epoch 35, Loss: 102.65872532552083, BCE: 77.1744927734375, KLD: 25.484232495117187
Epoch 36, Loss: 102.5991544921875, BCE: 77.12035055338542, KLD: 25.478803934733072
Epoch 37, Loss: 102.50546961263021, BCE: 77.05495727539062, KLD: 25.450512166341145
Epoch 38, Loss: 102.49072548828126, BCE: 77.0042119547526, KLD: 25.48651337890625
Epoch 39, Loss: 102.37018362630208, BCE: 76.90685082194011, KLD: 25.46333280843099
Epoch 40, Loss: 102.33613090820313, BCE: 76.88100688476563, KLD: 25.455124271647136
Epoch 41, Loss: 102.2744924967448, BCE: 76.81612768554687, KLD: 25.458364896647137
Epoch 42, Loss: 102.2507782148438, BCE: 76.7783169921875, KLD: 25.472461352539064
Epoch 43, Loss: 102.17886629231771, BCE: 76.72043367513021, KLD: 25.45843251953125
Epoch 44, Loss: 102.15780144856771, BCE: 76.68776215820313, KLD: 25.470039371744793
Epoch 45, Loss: 102.09620372721353, BCE: 76.6236814453125, KLD: 25.47252235107422
Epoch 46, Loss: 102.04954313151042, BCE: 76.5920006673177, KLD: 25.457542317708334
Epoch 47, Loss: 101.99102124023437, BCE: 76.55090003255208, KLD: 25.440121346028647
Epoch 48, Loss: 101.93675030924479, BCE: 76.4959839029948, KLD: 25.440766394042967
Epoch 49, Loss: 101.86213102213542, BCE: 76.42196911621093, KLD: 25.44016189778646
Epoch 50, Loss: 101.85459018554687, BCE: 76.42223973795573, KLD: 25.43235028483073
Epoch 51, Loss: 101.77279811197917, BCE: 76.35917451171875, KLD: 25.413623563639323
Epoch 52, Loss: 101.77775169270834, BCE: 76.34161223958333, KLD: 25.43613944905599
Epoch 53, Loss: 101.72746157226563, BCE: 76.30550322265626, KLD: 25.421958426920572
Epoch 54, Loss: 101.67367127278646, BCE: 76.26517255859375, KLD: 25.408498677571615
Epoch 55, Loss: 101.68085454101562, BCE: 76.23068173828125, KLD: 25.450172912597655
Epoch 56, Loss: 101.63019399414063, BCE: 76.20414072265625, KLD: 25.4260534383138
Epoch 57, Loss: 101.6027321126302, BCE: 76.18182985026041, KLD: 25.420902022298176
Epoch 58, Loss: 101.55730266927084, BCE: 76.14151384277343, KLD: 25.41578866780599
Epoch 59, Loss: 101.52625154622396, BCE: 76.09334806315104, KLD: 25.432903442382813
Epoch 60, Loss: 101.49106741536458, BCE: 76.0851674235026l, KLD: 25.40589989827474
Epoch 61, Loss: 101.42700760091145, BCE: 76.02876142578125, KLD: 25.39824619140625
Epoch 62, Loss: 101.42682644856771, BCE: 76.01342391764322, KLD: 25.41340251464844
Epoch 63, Loss: 101.37493636067708, BCE: 75.98386228841146, KLD: 25.39107380371094
Epoch 64, Loss: 101.40258865559896, BCE: 75.98351500651042, KLD: 25.419073575846355

```
Epoch 65, Loss: 101.31357252604167, BCE: 75.93440032552083, KLD: 25.379172131347655
Epoch 66, Loss: 101.29229225260417, BCE: 75.88466834309895, KLD: 25.40762383219401
Epoch 67, Loss: 101.24362216796875, BCE: 75.8686208984375, KLD: 25.375001501464844
Epoch 68, Loss: 101.25496819661458, BCE: 75.87869169921875, KLD: 25.37627654622396
Epoch 69, Loss: 101.20962503255208, BCE: 75.83664149576823, KLD: 25.372983418782553
Epoch 70, Loss: 101.17468430989584, BCE: 75.78252439778646, KLD: 25.39215997721354
Epoch 71, Loss: 101.15481263020834, BCE: 75.78536640625, KLD: 25.3694464070638
Epoch 72, Loss: 101.12917506510416, BCE: 75.79124047851562, KLD: 25.337934517415366
Epoch 73, Loss: 101.08565955403645, BCE: 75.71200999348959, KLD: 25.373649568684897
Epoch 74, Loss: 101.04934454752605, BCE: 75.70001940104167, KLD: 25.34932517903646
Epoch 75, Loss: 101.06612133789062, BCE: 75.70004326171875, KLD: 25.36607793375651
Epoch 76, Loss: 101.01729059244792, BCE: 75.66923006184896, KLD: 25.348060546875
Epoch 77, Loss: 100.94150563151042, BCE: 75.6314338704427, KLD: 25.310071960449218
Epoch 78, Loss: 100.95313409830729, BCE: 75.63678050130208, KLD: 25.31635382080078
Epoch 79, Loss: 100.95497153320312, BCE: 75.60318357747396, KLD: 25.351788037109376
Epoch 80, Loss: 100.95943273111979, BCE: 75.60351092936197, KLD: 25.355921785481772
Epoch 81, Loss: 100.88253776041667, BCE: 75.5486728515625, KLD: 25.33386494547526
Epoch 82, Loss: 100.90041305338542, BCE: 75.54360162760416, KLD: 25.356811466471353
Epoch 83, Loss: 100.83991458333334, BCE: 75.50567453613282, KLD: 25.334239929199217
Epoch 84, Loss: 100.82437449544271, BCE: 75.52050317382813, KLD: 25.303871341959635
Epoch 85, Loss: 100.80790169270833, BCE: 75.50105, KLD: 25.306851599121092
Epoch 86, Loss: 100.82026486002604, BCE: 75.48356854654948, KLD: 25.33669630126953
Epoch 87, Loss: 100.76355319010416, BCE: 75.47514554036458, KLD: 25.288407796223957
Epoch 88, Loss: 100.78811090494791, BCE: 75.46123269042968, KLD: 25.326878084309897
Epoch 89, Loss: 100.75740629882813, BCE: 75.45078807779947, KLD: 25.30661809895833
Epoch 90, Loss: 100.73359970703125, BCE: 75.4135776611328, KLD: 25.320022054036457
Epoch 91, Loss: 100.7042879720052, BCE: 75.39174456380208, KLD: 25.312543518066406
Epoch 92, Loss: 100.69693510742188, BCE: 75.38461258138021, KLD: 25.312322615559896
Epoch 93, Loss: 100.68137998046875, BCE: 75.38811157226563, KLD: 25.29326844075521
Epoch 94, Loss: 100.65877887369791, BCE: 75.33423385416667, KLD: 25.324545068359374
Epoch 95, Loss: 100.60273727213541, BCE: 75.31942287597656, KLD: 25.283314400227866
Epoch 96, Loss: 100.60981557617187, BCE: 75.33444883626302, KLD: 25.275366849772137
Epoch 97, Loss: 100.5593776204427, BCE: 75.29360310872396, KLD: 25.265774556477865
Epoch 98, Loss: 100.58418279622396, BCE: 75.29192788085938, KLD: 25.29225499267578
Epoch 99, Loss: 100.54277967122395, BCE: 75.27773606770833, KLD: 25.265043530273438
Epoch 100, Loss: 100.56256595052083, BCE: 75.24932430826823, KLD: 25.313241418457032
```

训练完成后，输出重构损失和KL散度的随迭代次数的变化图，以及10 张生成的手写数字图片。

In [34]:

```
# 绘制损失变化图
plt.plot(range(1, num_epochs + 1), reconstruction_losses, label="Reconstruction Loss
plt.plot(range(1, num_epochs + 1), kl_losses, label="KL Divergence")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



输出样本图像

In [35]:
```
# 生成样本图像
vae.eval()
with torch.no_grad():
    z = torch.randn(10, 20).to(device)  # 从标准正态分布采样
    sample = vae.decode(z).view(-1, 28, 28).cpu()#从GPU转移到CPU
    fig, ax = plt.subplots(1, 10, figsize=(15, 2)) #fig是图像的容器，ax是包含10个子图
    for i in range(10):
        ax[i].imshow(sample[i].cpu().numpy(), cmap='gray')#cmap='gray' 表示使用灰度颜
        ax[i].axis('off') #这一行隐藏了每个子图的坐标轴，简化图像展示。
    plt.show()
```