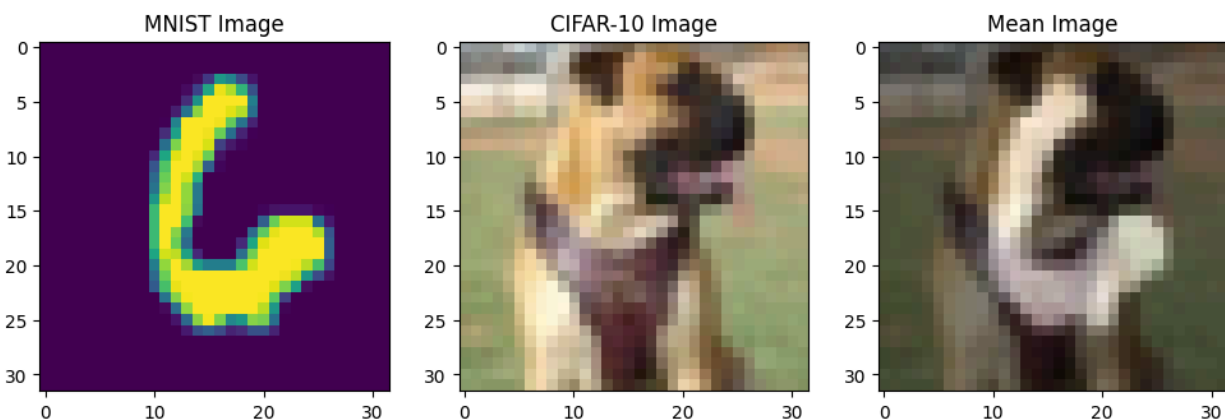


# دوره یادگیری عمیق

## تمرین ۷

### مجموعه داده‌های ادغام شده



### هدف

در این تمرین، شما با خودرمزگذارها (Autoencoders) و خودرمزگذارهای تغییریافته (Variational Autoencoders) کار خواهید کرد تا مدلی ایجاد کنید که بتواند دو تصویر اصلی را از یک تصویر ادغام شده بازسازی کند. تصویر ادغام شده، میانگین یک تصویر از مجموعه داده CIFAR-10 و یک تصویر از مجموعه داده MNIST است. شما از PyTorch برای این کار استفاده خواهید کرد.

### وظایف

1. ایجاد مجموعه داده  
مجموعه داده‌ای ایجاد کنید که هر تصویر آن میانگین یک تصویر از CIFAR-10 و یک تصویر از MNIST باشد.
2. طراحی مدل  
مدلی با استفاده از خودرمزگذارها یا خودرمزگذارهای تغییریافته طراحی کنید تا دو تصویر اصلی را از تصویر ادغام شده بازسازی کند.
3. آموزش و ارزیابی  
مدل خود را با استفاده از مجموعه داده ادغام شده آموزش دهید و عملکرد آن را در بازسازی تصاویر اصلی ارزیابی کنید.

## مراحل

### 1. آماده‌سازی مجموعه داده

- مجموعه داده‌های CIFAR-10 و MNIST را دانلود کنید.
- مجموعه داده‌ها را پیش‌پردازش کنید تا اطمینان حاصل شود که شکل‌های سازگاری دارند.
- مجموعه داده ادغام شده را با گرفتن میانگین تصاویر مربوطه از CIFAR-10 و MNIST ایجاد کنید.

### 2. طراحی مدل

- مدلی با استفاده از خودرمزگذارها یا خودرمزگذارهای تغییر یافته با PyTorch طراحی کنید.
- رمزگذار باید تصویر ادغام شده را به عنوان ورودی بگیرد و آن را به فضای نهان رمزگذاری کند.
- رمزگشا باید دو تصویر اصلی را از نمایش نهان بازسازی کند.

### 3. آموزش مدل

- مدل را با استفاده از مجموعه داده ادغام شده آموزش دهید.
- از توابع زیان مناسب استفاده کنید تا اطمینان حاصل شود که تصاویر بازسازی شده تا حد ممکن به تصاویر اصلی CIFAR-10 و MNIST نزدیک هستند.

### 4. ارزیابی

- عملکرد مدل خود را با استفاده از معیارهای مناسب ارزیابی کنید.
- تصاویر بازسازی شده را بصری سازی کنید و آن‌ها را با تصاویر اصلی مقایسه کنید.

## کد برای آماده‌سازی مجموعه داده

در زیر نمونه کدی برای ایجاد مجموعه داده ادغام شده از CIFAR-10 و MNIST با استفاده از PyTorch آمده است:

```
class MergedDataset(Dataset):
    def __init__(self, dataset1, dataset2):
        self.dataset1 = dataset1
        self.dataset2 = dataset2
        self.resize = transforms.Resize((32, 32))

    def __len__(self):
        return min(len(self.dataset1), len(self.dataset2))

    def __getitem__(self, idx):
        image1, _ = self.dataset1[idx]
        image2, _ = self.dataset2[idx]
```

```

# Resize image1 to match the size of image2
image1 = self.resize(image1)

# If image1 is grayscale, duplicate the channel dimension to match
image2
if image1.shape[0] == 1:
    image1 = torch.cat([image1] * 3, dim=0)

# Compute the mean of the two images
mean_image = (image1 + image2) / 2

return mean_image

```

## راهنمایی های اضافی

- اطمینان حاصل کنید که معماری مدل شما به خوبی تعریف و مستند شده است.
- از تکنیک های بهینه سازی و پارامترهای مناسب استفاده کنید.
- گزارش کاملی شامل طراحی مدل، فرآیند آموزش و نتایج ارزیابی ارائه دهید.
- تصویری سازی هایی از برخی تصاویر بازسازی شده و مقایسه آن ها با تصاویر اصلی ارائه کنید.

## نحوه ارسال

موارد زیر را ارسال کنید:

- کد شما برای ایجاد مجموعه داده ادغام شده.
- کد مدل PyTorch برای خودرمزگذار یا خودرمزگذار تغییر یافته.
- گزارشی که روش، نتایج و چالش های پیش روی شما را به تفصیل توضیح دهد.
- تصویری سازی هایی از تصاویر بازسازی شده و تصاویر اصلی.