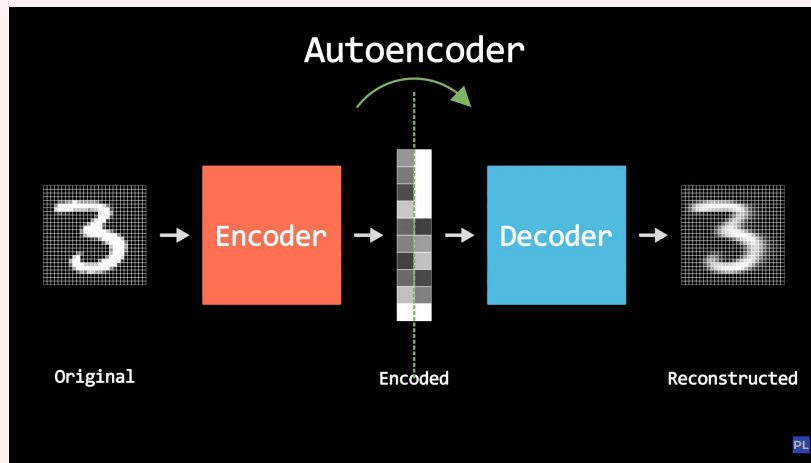


AUTO ENCODER

By: Marcus Blea

WHAT IS AN AUTOENCODER?

An autoencoder is a neural network that learns efficient data representations by encoding the input into a lower-dimensional latent space and then decoding it back to reconstruct the original input.



Linear Autoencoder

A type of autoencoder that uses only linear transformations in its encoder and decoder to learn a lower-dimensional linear representation of the input data

```
# Defines a linear autoencoder model with an encoder that reduces the input size through
class Autoencoder_linear(nn.Module):
    def __init__(self):
        super().__init__()
        self.encoder = nn.Sequential(
            nn.Linear(28 * 28, 128), # (N, 784) -> (N, 128)
            nn.ReLU(),
            nn.Linear(128, 64),
            nn.ReLU(),
            nn.Linear(64, 12),
            nn.ReLU(),
            nn.Linear(12, 3) # -> N, 3
        )

        self.decoder = nn.Sequential(
            nn.Linear(3, 12),
            nn.ReLU(),
            nn.Linear(12, 64),
            nn.ReLU(),
            nn.Linear(64, 128),
            nn.ReLU(),
            nn.Linear(128, 28 * 28),
            nn.Sigmoid()
        )

    def forward(self, x):
        encoded = self.encoder(x)
        decoded = self.decoder(encoded)
        return decoded

# Input [-1, +1] -> use nn.Tanh
✓ 0.0s
```

Convolutional Autoencoder

Employs convolutional layers in its encoder to learn spatial hierarchies of features and transpose convolutional layers in its decoder to reconstruct the input from a compressed representation.

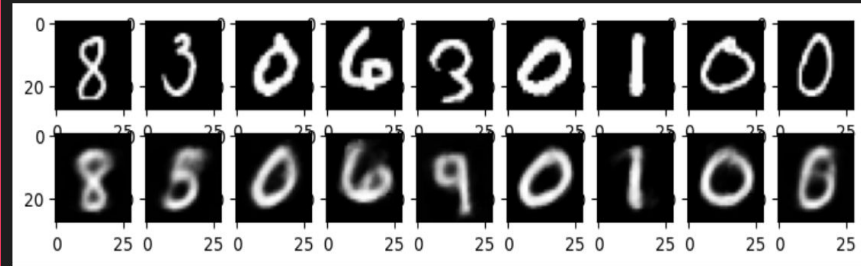
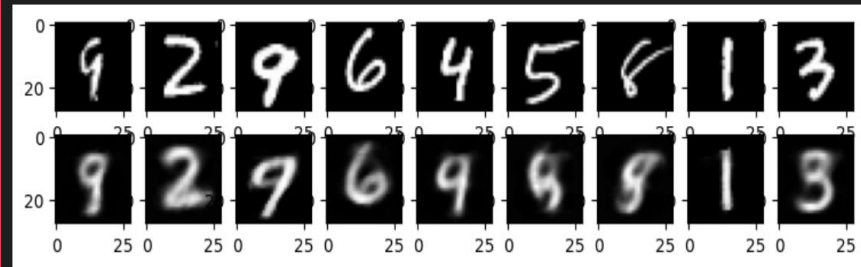
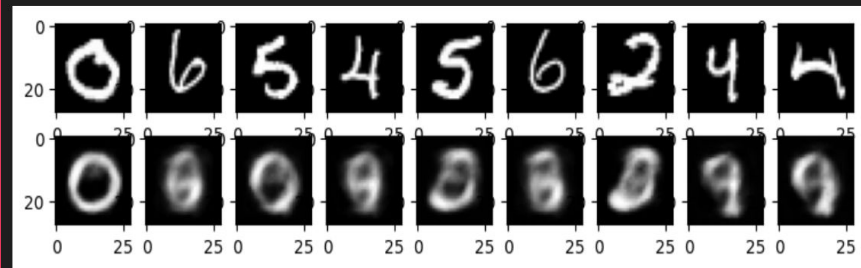
```
class Autoencoder(nn.Module):
    def __init__(self):
        super().__init__()
        # N, 1, 28, 28
        self.encoder = nn.Sequential(
            nn.Conv2d(1, 16, 3, stride=2, padding=1), # -> N, 16, 14, 14
            nn.ReLU(),
            nn.Conv2d(16, 32, 3, stride=2, padding=1), # -> N, 32, 7, 7
            nn.ReLU(),
            nn.Conv2d(32, 64, 7) # -> N, 64, 1, 1
        )

        # N, 64, 1, 1
        self.decoder = nn.Sequential(
            nn.ConvTranspose2d(64, 32, 7), # -> N, 32, 7, 7
            nn.ReLU(),
            nn.ConvTranspose2d(32, 16, 3, stride=2, padding=1, output_padding=1), # N, 16
            nn.ReLU(),
            nn.ConvTranspose2d(16, 1, 3, stride=2, padding=1, output_padding=1), # N, 1,
            nn.Sigmoid()
        )

    def forward(self, x):
        encoded = self.encoder(x)
        decoded = self.decoder(encoded)
        return decoded
```

WHAT IS THE POINT OF THESE EXAMPLES?

It tries to learn a compressed representation of an input so that it can be used to reconstruct the original input





**LETS LOOK
AT THE
CODE!**

**THANK
YOU!**