
PROJECT REPORT

Image Denoising Using Autoencoders

NAME: RITESH KUMAR

ENROLLMENT NUMBER: 22112088

EMAIL: RITESH_K@CH.IITR.AC.IN

MOBILE NUMBER: 7988032771

Abstract:

This project explores the application of an autoencoder neural network for denoising images from the MNIST dataset, a collection of handwritten digits. The autoencoder, implemented using Keras and TensorFlow, is designed to encode noisy images into a latent representation and decode them back into clean, denoised versions.

Introduction:

In This report documents the implementation of an autoencoder neural network using the Keras framework for denoising MNIST dataset images. Autoencoders are neural networks designed to encode input data into a latent-space representation and then decode it back to reconstruct the input. They are effective for tasks like image denoising, where noisy images are passed through the network to obtain clean versions as output.

Dataset Overview:

The MNIST dataset consists of 28x28 grayscale images of handwritten digits (0-9). It is a popular benchmark dataset in machine learning and computer vision, widely used for training and testing models.

- **Training Set:** 60,000 images
- **Test Set:** 10,000 images

Data Preprocessing:

- **Loading:** The dataset is loaded using Keras's `mnist.load_data()` function, splitting it into training and test sets.

```
Load MNIST dataset
X_train, y_train), (X_test, y_test) = mnist.load_data()
```

- **Flattening:** Each 28x28 image is flattened into a 1D array of 784 pixels to be compatible with the input layer of the neural network.

```
# Flatten images
num_pixels = X_train.shape[1] * X_train.shape[2]
X_train = X_train.reshape(X_train.shape[0], num_pixels).astype('float32')
X_test = X_test.reshape(X_test.shape[0], num_pixels).astype('float32')
```

- **Normalization:** Pixel values are normalized to the range [0, 1] by dividing by 255.

```
X_train = X_train / 255
X_test = X_test / 255
```

- **Adding Noise:** Gaussian noise with mean 0 and standard deviation 1 is added to both training and test sets to create noisy versions of the images. The noise factor used was 0.2.

Autoencoder Architecture:

The autoencoder architecture used for this task consists of fully connected (Dense) layers:

- **Encoder:**
 - Input layer of 784 neurons.
 - Hidden layers with 500, 300, and 100 neurons respectively, using ReLU activation.
- **Decoder:**
 - Mirror image of the encoder architecture.
 - Reconstructs the flattened image with 784 neurons in the output layer, using sigmoid activation to output pixel values in the range [0, 1].

```
# Build the autoencoder model
model = Sequential()
model.add(Dense(500, input_dim=num_pixels, activation='relu'))
model.add(Dense(300, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(300, activation='relu'))
model.add(Dense(500, activation='relu'))
model.add(Dense(784, activation='sigmoid'))
```

Training:

- **Loss Function:** Mean Squared Error (MSE) was used as the loss function to measure the difference between the original and reconstructed images.
- **Optimizer:** Adam optimizer was employed for training the model.
- **Batch Size:** 200
- **Epochs:** The model was trained for 60 epochs on the noisy training data, validating on the noisy test data.

```
# Compile and train the model
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x_train_noisy, X_train, validation_data=(x_test_noisy, X_test), epochs=60, batch_size=200)
```

Evaluation:

- **Denoising Performance:** After training, the autoencoder was used to predict denoised images from the noisy test set.
- **Peak Signal-to-Noise Ratio (PSNR):** PSNR was calculated for each denoised image compared to its original counterpart. PSNR measures the quality of the reconstruction, with higher values indicating better quality.

```
psnr_scores = [peak_signal_noise_ratio(X_test[i], pred[i]) for i in range(len(X_test))]
import numpy as np
# Calculate the average PSNR score
average_psnr = np.mean(psnr_scores)

print(f"Average PSNR score: {average_psnr}")
```

- **Average PSNR:** The average PSNR score across all test images was computed to evaluate the overall denoising performance of the model.
-

Results:

- **Predicted Denoised Images:** The autoencoder successfully reduced noise from the input images, producing cleaner reconstructions.
- **Average PSNR Score:** The average PSNR score obtained is in the range of 23.2 to 23.6.

Average PSNR score: 23.252637187088666

Conclusion:

The autoencoder model effectively learned to denoise MNIST images corrupted with Gaussian noise. The approach demonstrates the capability of autoencoders in image reconstruction tasks and provides a basis for further exploration into more complex datasets and denoising techniques.
