Neural Networks and Deep Learning - ICP5

Nalluri Prajwala

Student ID- 700766230

Github Link: https://github.com/Prajwalanalluri/Neural-Assignment-5.git

1. Add one more hidden layer to autoencoder

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

```
[ ] from keras.layers import Input, Dense
    from keras.models import Model

    # Define the size of our encoded representation
    encoding_dim = 32  # 32 floats -> compression factor of 24.5 assuming the input is 784 floats

    # Input placeholder
    input_img = Input(shape=(784,))
    # Encoded representation
    encoded = Dense(128, activation='relu')(input_img)  # Increased size of hidden layer
    # Additional hidden layer
    hidden_layer = Dense(64, activation='relu')(encoded)  # Adding a hidden layer with 64 neurons
    # Decoded representation
    decoded = Dense(784, activation='sigmoid')(hidden_layer)
    # Autoencoder model
    autoencoder = Model(input_img, decoded)

    # Compile the autoencoder (defaults to adam optimizer with default learning rate)
    autoencoder.compile(optimizer='adam', loss='binary_crossentropy')

    # Loading and preprocessing the Fashion MNIST data
    from keras.datasets import fashion_mnist
    import numpy as np

    (x_train, _), (x_test, _) = fashion_mnist.load_data()
    x_train = x_train.astype('float32') / 255.
    x_test = x_test.astype('float32') / 255.
    x_train = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
    x_test = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))

    # Training the autoencoder
    autoencoder.fit(x_train, x_train,
                    epochs=8,  # Increased epochs
                    batch_size=128,  # Smaller batch size
                    shuffle=True,
                    validation_data=(x_test, x_test))
```

✓ 2m 28s completed at 3:29 PM

```
Epoch 1/8
469/469 [==============================] - 5s 9ms/step - loss: 0.3446 - val_loss: 0.3068
Epoch 2/8
469/469 [==============================] - 4s 10ms/step - loss: 0.2946 - val_loss: 0.2914
Epoch 3/8
469/469 [==============================] - 6s 12ms/step - loss: 0.2863 - val_loss: 0.2858
Epoch 4/8
469/469 [==============================] - 5s 11ms/step - loss: 0.2819 - val_loss: 0.2826
Epoch 5/8
469/469 [==============================] - 4s 9ms/step - loss: 0.2792 - val_loss: 0.2803
Epoch 6/8
469/469 [==============================] - 6s 12ms/step - loss: 0.2772 - val_loss: 0.2786
Epoch 7/8
469/469 [==============================] - 4s 9ms/step - loss: 0.2757 - val_loss: 0.2774
Epoch 8/8
469/469 [==============================] - 4s 9ms/step - loss: 0.2746 - val_loss: 0.2766
<keras.src.callbacks.History at 0x7eacbfacf8e0>
```

2. Do the prediction on the test data and then visualize one of the reconstructed version of that test data. Also, visualize the same test data before reconstruction using Matplotlib

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

```
[ ] import matplotlib.pyplot as plt

    # Predict on test data
    decoded_imgs = autoencoder.predict(x_test)

    # Function to plot images
    def plot_images(original_images, decoded_images, num_images=10):
        plt.figure(figsize=(20, 4))
        for i in range(num_images):
            # Original images
            ax = plt.subplot(2, num_images, i + 1)
            plt.imshow(original_images[i].reshape(28, 28), cmap='gray')
            ax.get_xaxis().set_visible(False)
            ax.get_yaxis().set_visible(False)

            # Reconstructed images
            ax = plt.subplot(2, num_images, i + 1 + num_images)
            plt.imshow(decoded_images[i].reshape(28, 28), cmap='gray')
            ax.get_xaxis().set_visible(False)
            ax.get_yaxis().set_visible(False)
        plt.show()

    # Visualize original and reconstructed images
    plot_images(x_test, decoded_imgs)
```

```
313/313 [==============================] - 2s 5ms/step
```

✓ 2m 28s completed at 3:29 PM

3. Repeat the question 2 on the denoisening autoencoder

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

[2]

```python
import numpy as np
import matplotlib.pyplot as plt
from keras.layers import Input, Dense, GaussianNoise
from keras.models import Model

# Define the size of our encoded representation
encoding_dim = 32  # 32 floats -> compression factor of 24.5 assuming the input is 784 floats

# Input placeholder with added noise
input_img = Input(shape=(784,))
noisy_img = GaussianNoise(0.5)(input_img)  # Adding Gaussian noise with stddev=0.5
# Encoded representation
encoded = Dense(128, activation='relu')(noisy_img)  # Increased size of hidden layer
# Additional hidden layer
hidden_layer = Dense(64, activation='relu')(encoded)  # Adding a hidden layer with 64 neurons
# Decoded representation
decoded = Dense(784, activation='sigmoid')(hidden_layer)
# Denoising Autoencoder model
autoencoder = Model(input_img, decoded)

# Compile the autoencoder (defaults to adam optimizer with default learning rate)
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')

# Loading and preprocessing the Fashion MNIST data
from keras.datasets import fashion_mnist

(x_train, _), (x_test, _) = fashion_mnist.load_data()
x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
x_train = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
x_test = x_test.reshape((len(x_test), np.prod(x_test.shape[1:])))

# Add Gaussian noise to the training and test data
noise_factor = 0.5
x_train_noisy = x_train + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_train.shape)
x_test_noisy = x_test + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_test.shape)
```

2m 28s completed at 3:29 PM

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

```python
x_train_noisy = np.clip(x_train_noisy, 0., 1.)
x_test_noisy = np.clip(x_test_noisy, 0., 1.)

# Training the denoising autoencoder
autoencoder.fit(x_train_noisy, x_train,
                epochs=25,  # Increased epochs for better training
                batch_size=100,
                shuffle=True,
                validation_data=(x_test_noisy, x_test))

# Predict on test data
decoded_imgs = autoencoder.predict(x_test_noisy)

# Function to plot images
def plot_images(original_images, noisy_images, decoded_images, num_images=10):
    plt.figure(figsize=(20, 6))
    for i in range(num_images):
        # Original images
        ax = plt.subplot(3, num_images, i + 1)
        plt.imshow(original_images[i].reshape(28, 28), cmap='gray')
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)
        if i == 0:
            ax.set_title('Original Images')

        # Noisy images
        ax = plt.subplot(3, num_images, i + 1 + num_images)
        plt.imshow(noisy_images[i].reshape(28, 28), cmap='gray')
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)
        if i == 0:
            ax.set_title('Noisy Input')
```

2m 28s completed at 3:29 PM

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

```
        # Reconstructed images
        ax = plt.subplot(3, num_images, i + 1 + 2 * num_images)
        plt.imshow(decoded_images[i].reshape(28, 28), cmap='gray')
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)
        if i == 0:
            ax.set_title('Denoised Output')
    plt.tight_layout()
    plt.show()

# Visualize original, noisy, and denoised images
plot_images(x_test, x_test_noisy, decoded_imgs)
```

```
Epoch 1/25
600/600 [==============================] - 9s 14ms/step - loss: 0.3709 - val_loss: 0.3536
Epoch 2/25
600/600 [==============================] - 6s 10ms/step - loss: 0.3293 - val_loss: 0.3397
Epoch 3/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3235 - val_loss: 0.3260
Epoch 4/25
600/600 [==============================] - 7s 11ms/step - loss: 0.3203 - val_loss: 0.3200
Epoch 5/25
600/600 [==============================] - 7s 12ms/step - loss: 0.3179 - val_loss: 0.3135
Epoch 6/25
600/600 [==============================] - 6s 10ms/step - loss: 0.3160 - val_loss: 0.3173
Epoch 7/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3147 - val_loss: 0.3096
Epoch 8/25
600/600 [==============================] - 6s 10ms/step - loss: 0.3136 - val_loss: 0.3061
Epoch 9/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3127 - val_loss: 0.3067
Epoch 10/25
600/600 [==============================] - 6s 10ms/step - loss: 0.3122 - val_loss: 0.3033
```

✓ 2m 28s completed at 3:29 PM

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

```
Epoch 9/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3127 - val_loss: 0.3067
Epoch 10/25
600/600 [==============================] - 6s 10ms/step - loss: 0.3122 - val_loss: 0.3033
Epoch 11/25
600/600 [==============================] - 12s 20ms/step - loss: 0.3115 - val_loss: 0.3019
Epoch 12/25
600/600 [==============================] - 9s 15ms/step - loss: 0.3110 - val_loss: 0.3020
Epoch 13/25
600/600 [==============================] - 7s 12ms/step - loss: 0.3105 - val_loss: 0.3013
Epoch 14/25
600/600 [==============================] - 8s 14ms/step - loss: 0.3102 - val_loss: 0.3008
Epoch 15/25
600/600 [==============================] - 6s 10ms/step - loss: 0.3098 - val_loss: 0.2992
Epoch 16/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3096 - val_loss: 0.2993
Epoch 17/25
600/600 [==============================] - 6s 10ms/step - loss: 0.3093 - val_loss: 0.3001
Epoch 18/25
600/600 [==============================] - 8s 14ms/step - loss: 0.3090 - val_loss: 0.2994
Epoch 19/25
600/600 [==============================] - 7s 11ms/step - loss: 0.3089 - val_loss: 0.2991
Epoch 20/25
600/600 [==============================] - 8s 14ms/step - loss: 0.3087 - val_loss: 0.2985
Epoch 21/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3086 - val_loss: 0.2987
Epoch 22/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3083 - val_loss: 0.2984
Epoch 23/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3082 - val_loss: 0.2986
Epoch 24/25
600/600 [==============================] - 7s 11ms/step - loss: 0.3082 - val_loss: 0.2979
Epoch 25/25
600/600 [==============================] - 8s 13ms/step - loss: 0.3080 - val_loss: 0.2984
313/313 [==============================] - 1s 3ms/step
```

Original Images

✓ 2m 28s completed at 3:29 PM

4. plot loss and accuracy using the history object

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

```python
# Train the denoising autoencoder
history = autoencoder.fit(x_train_noisy, x_train,
                          epochs=20,
                          batch_size=128,
                          shuffle=True,
                          validation_data=(x_test_noisy, x_test))

# Plot training history (loss)
plt.figure(figsize=(10, 5))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
Epoch 1/20
469/469 [==============================] - 7s 12ms/step - loss: 0.3796 - val_loss: 0.3535
Epoch 2/20
469/469 [==============================] - 7s 15ms/step - loss: 0.3314 - val_loss: 0.3391
Epoch 3/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3248 - val_loss: 0.3373
Epoch 4/20
469/469 [==============================] - 7s 16ms/step - loss: 0.3214 - val_loss: 0.3272
Epoch 5/20
469/469 [==============================] - 6s 14ms/step - loss: 0.3188 - val_loss: 0.3169
Epoch 6/20
469/469 [==============================] - 7s 15ms/step - loss: 0.3172 - val_loss: 0.3146
Epoch 7/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3158 - val_loss: 0.3114
Epoch 8/20
469/469 [==============================] - 7s 15ms/step - loss: 0.3148 - val_loss: 0.3069
Epoch 9/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3139 - val_loss: 0.3059
Epoch 10/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3131 - val_loss: 0.3059
Epoch 11/20
```

✓ 2m 28s completed at 3:29 PM

Home - Google Drive | Untitled3.ipynb - Colab

colab.research.google.com/drive/1RcGv9dW0SD4HBgAud9YRdkv1r7pCVyCA#scrollTo=cqqeWhRQ_3VO

Untitled3.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Gemini

```
469/469 [==============================] - 7s 15ms/step - loss: 0.3124 - val_loss: 0.3056
Epoch 12/20
469/469 [==============================] - 7s 14ms/step - loss: 0.3118 - val_loss: 0.3020
Epoch 13/20
469/469 [==============================] - 7s 16ms/step - loss: 0.3114 - val_loss: 0.3019
Epoch 14/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3110 - val_loss: 0.3014
Epoch 15/20
469/469 [==============================] - 7s 15ms/step - loss: 0.3106 - val_loss: 0.3018
Epoch 16/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3103 - val_loss: 0.3003
Epoch 17/20
469/469 [==============================] - 7s 16ms/step - loss: 0.3101 - val_loss: 0.3004
Epoch 18/20
469/469 [==============================] - 6s 12ms/step - loss: 0.3098 - val_loss: 0.3002
Epoch 19/20
469/469 [==============================] - 7s 15ms/step - loss: 0.3095 - val_loss: 0.3005
Epoch 20/20
469/469 [==============================] - 6s 13ms/step - loss: 0.3093 - val_loss: 0.2997
```

✓ 2m 28s completed at 3:29 PM