Assignment 7

Due Thursday, April 3rd before midnight (California time)

Single-cell RNA-Seq (scRNA-Seq) data is well known for its high level of noise and sparsity. Due to the challenges in amplifying the very low amounts of RNA present in individual cells, only a small portion of the transcribed RNAs is captured during sequencing. This results in situations where many genes may be active at once, and two scRNA-Seq measurements might originate from cells with identical expression profiles despite appearing different.

In this assignment, you will implement and analyze autoencoders to extract meaningful clusters from scRNA-Seq count data. These clusters could reflect diverse expression states within the same tissue, such as differences arising from various cell cycle phases or distinct cell types.

Dataset

You are provided with:

- **`counts.npy`**: A gene expression matrix (normalized log-transformed) of scRNA-Seq data with dimensions corresponding to 5000 cells by 1000 genes. The entry e_{ij} indicates the expression level of the j-th gene in the i-th cell.
- **`labels.txt`**: A text file containing cluster labels for the cells, which fall into a total of 3 clusters.

Questions

Q1: Autoencoder Training for Dimensionality Reduction [1 pts]

- Load the **`counts`** and **`labels`**.
- Train a fully connected autoencoder to learn a low-dimensional representation of the gene expression data.
- Use two different latent space sizes: **`10`** and **`50`** (i.e., build two separate models).
- Use Mean Squared Error (MSE) as the loss function.
- Train the model for approximately **`50`** epochs (adjust as needed for optimal results).
- Report the training and testing history plots.

Hints:

- Use a **symmetric** encoder-decoder architecture (e.g., `1000 -> 100 -> 10` for encoding and `10 -> 100 -> 1000` for decoding).
- Do not use activation functions in the input, latent, or output layers.
- Normalization of input data is **not required**.

Q2: Reconstruction Error Analysis [1 pts]

- Compare the training and testing history plots for the models with latent space sizes of 10 and 50.
- Report the MSE between the reconstructed data and the original data for each latent space size.

Discuss:

- Discuss the differences in convergence speed between the two models.
- Explain how the latent space size (10 vs. 50) affects the reconstruction error, and in particular, why one model might achieve a lower final MSE despite converging more slowly.

Q3: Visualization of Reconstructions [2 pts]

- Using the autoencoder with a latent space size of **10**, create the following visualizations:
  - Generate PCA plots comparing the original data to the reconstructed data.
  - Generate t-SNE plots comparing the original data to the reconstructed data.
- Use **labels.txt** as labels when plotting.
- Compare the PCA and t-SNE visualizations between the original and reconstructed data.

Discuss:

- Discuss any observable differences in clustering between the original and reconstructed data for PCA and t-SNE.
- Explain possible reasons for these differences based on the autoencoder's reconstruction process.

Q4: Latent Space Visualization with 2-Dimensional Latent Space [1 pts]

- Retrain the autoencoder with a latent space of 2 dimensions.
- Extract the 2-dimensional latent representations.
- Create a 2D scatter plot of the latent space, using *labels.txt* to color-code the clusters.

Discuss:

- Discuss the observed clustering in the 2D latent space, including how well the clusters are separated and what the results suggest about the autoencoder’s ability to capture the underlying structure of the data.

Q5: Alternative Loss Function Experiment [2 pts]

The new loss function, **`nonzero_mse_loss`**, computes the MSE only over the nonzero elements of the input data. This is particularly useful for sparse datasets (like scRNA-seq) where zeros are prevalent and may not carry meaningful signals. By ignoring zeros, the loss focuses on capturing errors where there is actual gene expression data, potentially leading to improved learning of meaningful features.

- Replace the MSE loss with the following custom loss function when training the autoencoder (using a latent space size of **10**):

```
import tensorflow as tf
def nonzero_mse_loss(y_true, y_pred):
        # Create a mask for non-zero elements
        mask = tf.cast(tf.math.not_equal(y_true, 0), tf.float32)
        nonzero_count = tf.reduce_sum(mask)  # Count the number of non-zero elements
        # Calculate the squared difference for non-zero elements
        nonzero_squared_diff = tf.square(y_true - y_pred * mask)
        # Compute the mean of the non-zero squared differences
        nonzero_mse = tf.reduce_sum(nonzero_squared_diff) / nonzero_count
        return nonzero_mse
```

- After training with this alternative loss, do the following:
- Compare the reconstruction MSE from this experiment with the MSE reported in Q2.
- Generate the following four plots:
    - PCA plot: Reconstructed data vs. Original data.
    - PCA plot: Latent embeddings vs. Original data.
    - t-SNE plot: Reconstructed data vs. Original data.
    - t-SNE plot: Latent embeddings vs. Original data.

Discuss:

- Discuss any observed differences, noting if the model with the new loss better captures the relevant structure in the data by focusing on nonzero values.