Report of the Fashion MNIST SOM Code

1. Imports and Data Loading

- Libraries like Numpy and Matplotlib are imported for array manipulation and plotting.
- The fashion_mnist dataset from TensorFlow is loaded, which contains grayscale images of clothing items, each of size 28x28 pixels.

2. Data Preprocessing

- The training and test images are reshaped from 28x28 pixels to a flat 784-length vector (28 * 28), making each image a one-dimensional array.
- The pixel values are normalized to a [0, 1] range by dividing by 255.

3. SOM Initialization and Training

- A 20x20 SOM grid is initialized, with each node having an input size of 784, matching the flattened image dimensions.
- The SOM is trained on the preprocessed Fashion MNIST data using random weight initialization and 10,000 iterations. This allows the SOM to learn the patterns within the dataset and organize similar patterns close together on the grid.

4. Distance Map Visualization

- The SOM's distance map, or U-matrix, is plotted to visualize node distances. This map indicates the similarity between nodes, where closer nodes in color or distance are more similar.

5. Label Overlay

- Each training data point is mapped to its winning node on the SOM grid (the node closest to the data point in feature space).
- Class labels (corresponding to each fashion item) are overlaid onto the winning nodes to visualize the clustering of labels on the SOM grid. This step helps to see which types of images (labels) are grouped together.

6. Cluster Label Frequency Display

- The code creates a dictionary that maps each SOM node to the labels of images that map to it. This displays how often each label appears within each cluster, giving insight into label distribution across nodes.

7. Node Prototype Visualization

- For each SOM node, the learned weight vector (which represents a prototype image for that node) is reshaped to a 28x28 matrix and visualized as a grayscale image. This displays what each SOM node has "learned" as a typical pattern or image, showing a representative item from the Fashion MNIST dataset for each node.

Overall, this code uses a Self-Organizing Map to cluster and visualize the Fashion MNIST dataset. It highlights the internal structure of the data by organizing similar items together and providing both a distance-based U-matrix view and a view of label distributions. The final prototype display offers a summary of the features that each node has learned, giving insight into the general visual features associated with each cluster.