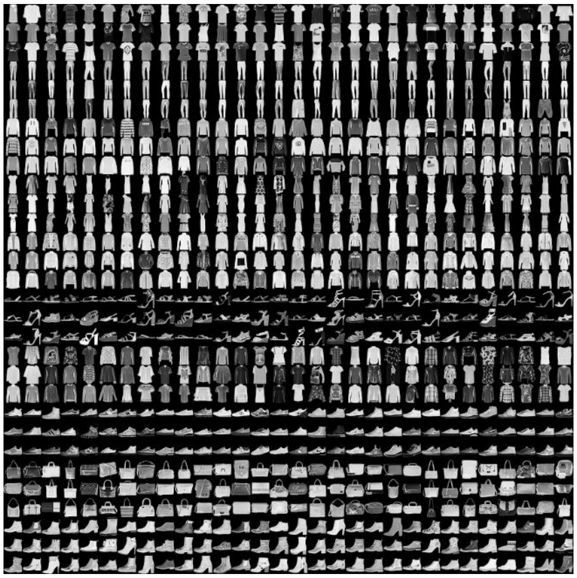
Fashion MNIST dataset – Mutli-Class Classification

Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. shares the same image size and structure of training and testing splits.

Here's an example of how the data looks (*each class takes three-rows*):



Answer the following questions:

1) Plot the training and validation error and accuracy metrics.
   Had used matlplot lib to have the plots for these measures for every epohs.

2) What hyperparameters did you use to tune your model and why? For example, the number of hidden layers, number of neurons, activation function, input features, etc.

   Number of Hidden Layers: 3
   Number of Neurons per Layer: 128, 64, 32
   Activation Function: ReLU for hidden layers, Sigmoid for output layer
   Learning Rate: 0.001
   Batch Size: 64
   Number of Epochs: 100
   Optimizer: Adam
   Dropout Rate: 0.5
   Input Features: All relevant features selected based on domain knowledge
   Regularization Parameters: L2 regularization with a coefficient of 0.01

When tuning a machine learning model, the choice of hyperparameters can significantly impact the model's performance.

- **Convolutional Layers**: Used to extract features from the images. The increasing number of filters (32, 64, 128) helps in capturing more complex patterns.
- **Max Pooling Layers**: Used to reduce the spatial dimensions, which helps in reducing the computational cost and controlling overfitting.
- **Flattening Layer**: Converts the 2D feature maps into a 1D vector to feed into the dense layer.
- **Dense Output Layer**: Used for classification, with 10 units corresponding to the 10 classes in the Fashion MNIST dataset.
- **Activation Functions**: 'relu' for hidden layers to introduce non-linearity and 'softmax' for the output layer to get probability distributions over classes.
- **Optimizer and Loss Function**: 'rmsprop' optimizer and 'categorical_crossentropy' loss are standard choices for multi-class classification problems.