APPROACH

1. Architecture:
   - Two convolutional layers were used to extract features from the input data.
   - Each convolutional layer was followed by a max-pooling layer to downsample the feature maps and introduce translation invariance.
   - The output of the last max-pooling layer was flattened into a 1D vector using a flatten layer.
   - Two fully connected (dense) layers were added to perform classification based on the learned features.
   - The ReLU activation function was applied to the outputs of the convolutional and dense layers to introduce non-linearity and improve convergence.
   - The output layer utilized the softmax activation function to convert the dense layer outputs into probabilities for multi-class classification.

2. Optimizer: The Adam optimizer was employed during training.
   - Adam optimizer dynamically adjusted the learning rate based on the first and second moments of the gradients.
   - This adaptive learning rate helped in efficient gradient-based optimization and faster convergence.

3. Loss function: Sparse categorical cross-entropy loss was utilized.
   - Sparse categorical cross-entropy is suitable for multi-class classification tasks with mutually exclusive classes.
   - It measured the dissimilarity between the predicted probabilities and the true labels, guiding the model towards better classification.

4. Dataset splitting: The Fashion MNIST dataset was split into training and testing sets.
   - 80% of the data was allocated for training the model.
   - The remaining 20% was reserved for testing the model's performance on unseen samples.

5. Training process:

- The model was trained for 10 epochs, meaning it went through the entire training dataset 10 times.
- After each epoch, the model's performance was evaluated on the validation set.
- This evaluation helped monitor the model's progress, detect overfitting or convergence issues, and make adjustments if necessary.