Question 1

In the code, we have implemented a [-1, 1] filter for the convolution. Applying this filter for convolution, we can observe from the resulting plot that it highlights the regions where there is a rapid change in intensity. This change corresponds to the edges in the original signal.

Question 2

An increase in validation error indicates overfitting as the number of epochs grows. When a model learns to perform well on training data but fails to generalize to new, previously unknown data (validation/test data), this is referred to as overfitting. This is common when the model becomes too sophisticated and begins to memorize the noise in the training data rather than collecting the underlying patterns.

To prevent this from happening, we can take the following steps:

* Early Stopping: Monitor the validation loss throughout training and stop when it stops decreasing or begins to grow. This inhibits the algorithm from learning the noise in the training data further.
* Reduce Model Complexity: To avoid overfitting, simplify the model architecture by lowering the number of layers, and neurons, or applying regularization techniques such as dropout or L2 regularization.
* Increase Data Augmentation: Data augmentation enhances the training dataset artificially by performing random modifications to the training images (such as rotations, flips, and so on). This aids the model's generalization to previously unknown data.
* Reduce Batch Size: Using lower batch sizes can sometimes help to avoid overfitting by adding noise to the gradient updates and making the optimization process more stochastic.

When using mini-batch sgd neither we use the dataset all at once nor do we use a single example at a time. We use a batch of a fixed number of training examples that is less than the actual dataset and call it a mini-batch. Doing this helps us achieve the advantages of both batch gradient descent and stochastic gradient descent. When we are using the mini-batch sgd we are updating our parameters frequently as well as we can use vectorized implementation for faster computations.