Model1:
80% of the image from training, 20% of the images for validation.
We used last 40,000 images as our training data, and it matches the last 40,000 noisy labels. For the validation set, we use the first 10,000 images and first 10,00 clean labels to develop our first model.

Covert the train labels and validation labels to one-hot encoding vector to match the sizes.

```
test_y:  [1. 7. 3. ... 9. 1. 1.]
test_y_one_hot:  [[0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]]
```

Before we apply the neural network, we convert the datatype into float 32, and resacle the pixel by dividing 255 because the images pixel values are from 0 to 255.

We use Leaky ReLU activation function to help us to learn the non-linear boundaries. It can help us to separate the 10 classes that are not linearly separable.

We used three 2d convolutional layers with 3 by 3 kernel. The first one is 32, second one is 64, and three one is 128. Besides that, We use another subsample technique to reduce the sample size and overfitting, which is max pooling. we also use 3 max_pooling layers to get the maximum pixel values from the regions.

Finally, we add those features to dense layer and use softmax activation function to calculate the probabilities. We need the model with batch_size of 64, epochs of 20 classed of 10.

Model2:

Reduce the layers and less features to help us improve the model.