Image Classification with the MNIST Dataset – How Different Optimizers Affect Model

1. Data Collection and Pre-processing

MNIST data is image data, with a collection of 70,000 grayscale images of handwritten digits from 0 to 9. I loaded the MNIST data through Keras API. The MNIST data has 60,000 training data and 10,000 testing data. The images in the MNIST data are in 28x28 pixels, but they are already represented as a collection of unsigned 8-bit integer values between 0 and 255, the values corresponding with a pixel's grayscale value where 0 is black, 255 is white, and all other values are in between.

Before training the model, I flattened the image data(to simplify the image input into the model), and made the array into a single array of 784 continuous pixels. After this, I normalized the image data by dividing 255, since deep learning models are better at dealing with floating point numbers between 0 and 1. The final step of pre-processing was categorizing the labels – encoding the labels.

2. Create a neural network model and compile the model

I ran 4 models, and they all have 3 layers, with 512, 512 and 10 nodes. Apart from the original model, I created other 3 models with different optimizers- “Adam”, “SGD”, and “RMSprop” to see if the optimizer would affect model’s performance.

Table 1. indicates the minimum loss of validation data in each model. I also pointed out the epoch number when the model reaches the minimum loss in validation data.

Loss / Model	Epoch Number of Best Model (Min. Validation Loss)	Train Data	Validation Data
Model1 (Default)	2	0.1022	0.1007
Model2 (Adam)	7	0.0258	0.0790
Model3 (SGD)	10	0.0956	0.1013
Model4 (RMSprop)	2	0.0988	0.9988

Table1. The Minimum Loss in Different Model and Train and Validation Data

Figure 1 shows the performance of each model when epoch number increases. In this case, model3 – SGD performs better than other optimizers. Due to the loss of validation data and training data are both converged. After the number of epochs exceeds 7, the loss of validation data starting to increase, which means model start to overfitting.

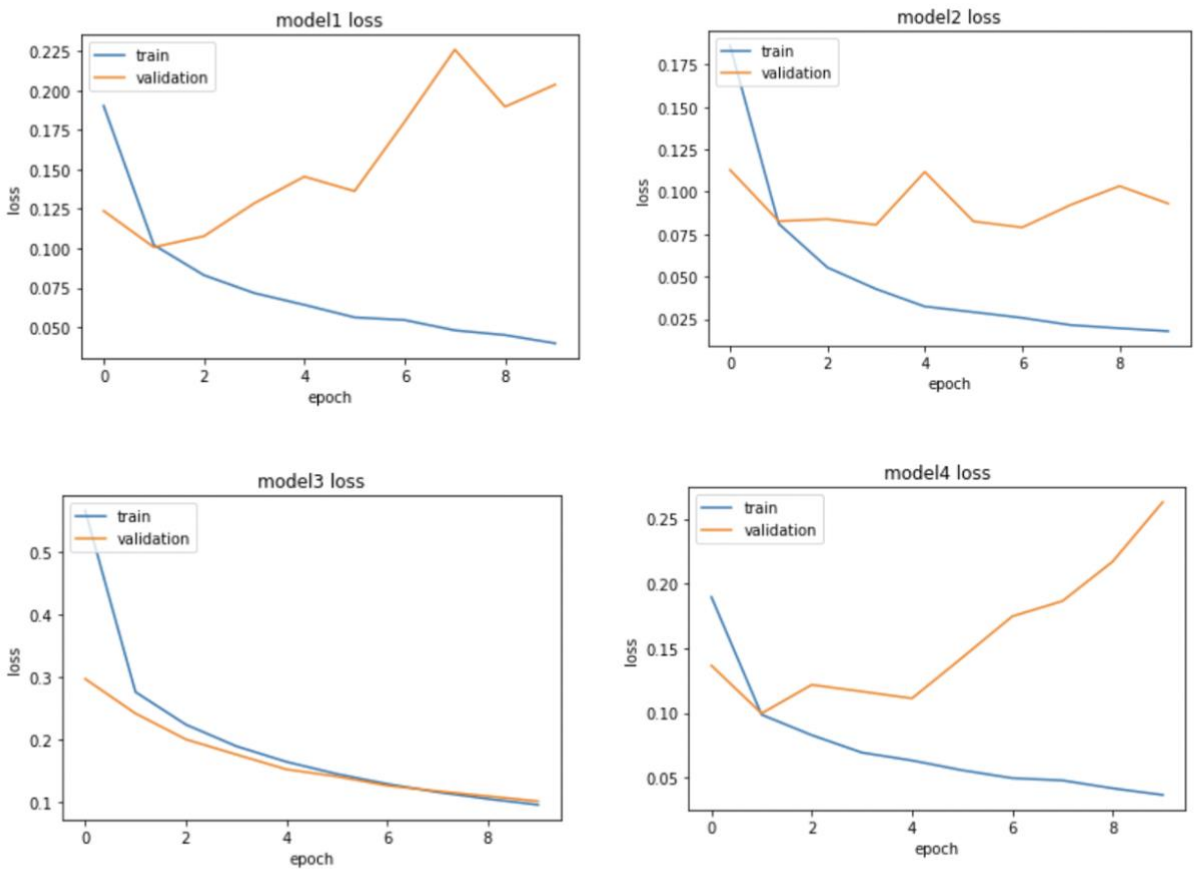

Figure 1. Performance of each Model when Epoch Number Increasing

3. **Managerial Conclusion:**
   The project shows that different optimizers will lead to a different result. If we want to deploy model, we want to choose a stable one. In this case, the ‘sgd’ optimizer with epoch = 7 is the best model.