In this file, I test how parameters contribute/influence to the model.

Test results for each model									
	No. of epochs	Batch size	Number of neurons in a layer	Number of layers	Learning rate	Activation functions	Dropout rates	Test Loss	Test accuracy
Standard Model	5	32	512	3	0.0001	Relu	0.25	1.0913	0.618
Model2	7	32	512	3	0.0001	Relu	0.25	0.9967	0.648
Model3	5	64	512	3	0.0001	Relu	0.25	1.1122	0.6028
Model4	5	32	512	4	0.0001	Relu	0.25	1.3383	0.5321
Model5	5	32	512	3	0.0002	Relu	0.25	1.0001	0.6517
Model6	5	32	512	3	0.0001	elu	0.25	1.1427	0.5988
Model7	5	32	512	3	0.0001	Relu	0.1	1.0283	0.6377
Best Model	100	32	512	3	0.0002	Relu	0.1	0.7646	0.7648

Because the tests by changing parameters and compare models to standard model above, I'd like to build a model with 100 epochs, batch size 32, 512 neurons in every layer, 3 layers, 0.0002 learning rate, activation function 'relu' and dropout rate 0.1.

My best model fits data well but it can be still improved. Because the testing accuracy is about 0.76 and the loss is 0.76. These two parameters are very close to training results. My model underfits data because the loss of model is still a high value and can be decreased.
I think I can improve my model by doing follows:

1. Add more features. If the under-fitting is due to insufficient features, I can add more feature parameters.
2. Increase model complexity. If the model is too simple to handle complex tasks. More complexity can be used to reduce the regularization coefficients. For example, I can train model with more epochs, or more layers.