

REPORT

Code settings:

input nodes = 784

1 hidden nodes = 200

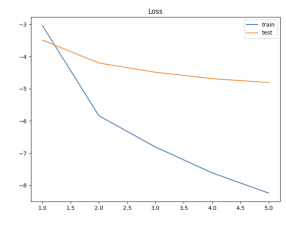
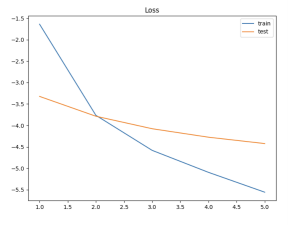
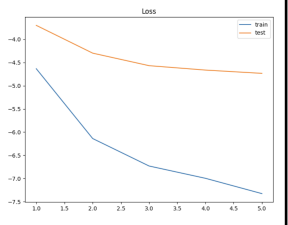
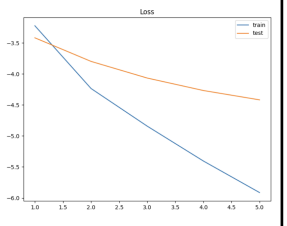
2 hidden nodes = 200

output nodes = 10

learning rate = 0.1

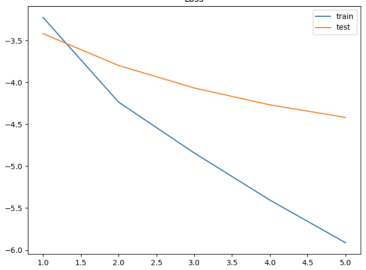
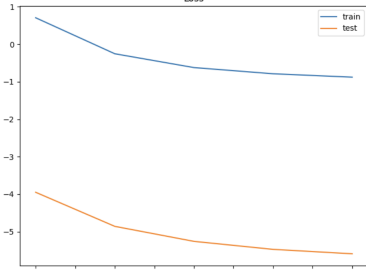
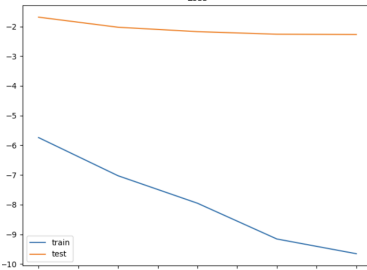
epochs = 5

Experiment presence / absence of dropout and batch normalization

dropout (p=0,5) ✓ batch normalization ✓	dropout (p=0,5) ✓ batch normalization ✗	dropout ✗ batch normalization ✓	dropout ✗ batch normalization ✗
			
Validation accuracy = 97% (9747/10000)	Validation accuracy = 96% (9630/10000)	Validation accuracy = 97% (9728/10000)	Validation accuracy = 96% (9637/10000)

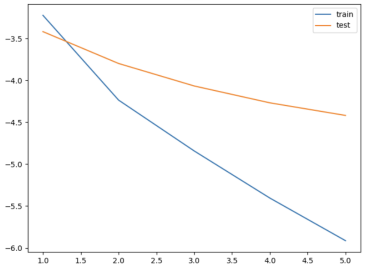
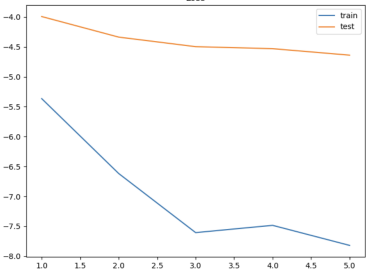
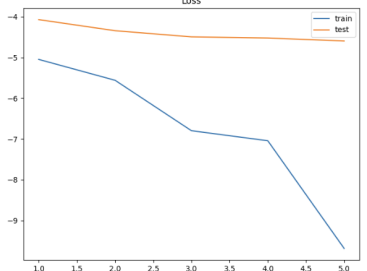
Results: в нашому випадку dropout не зміг підвищити точність прогнозування, на відміну від batch normalization, застосування якого призвело до підвищення точності на 1%.

Experiment with batch size

batch size = 10	batch size = 100	batch size = 1
		
Validation accuracy = 96% (9637/10000)	Validation accuracy = 89% (8912/10000)	Validation accuracy = 97% (9721/10000)

Results: зменшивши batch size до 1 ми змогли досягти більшої точності в 97%. А збільшивши batch size до 100 - точність зменшилася до 89%.

Experiment with activation functions

Activation functions: Sigmoid	Activation functions: Softsign	Activation functions: ReLU
		
Validation accuracy = 96% (9637/10000)	Validation accuracy = 97% (9714/10000)	Validation accuracy = 97% (9732/10000)

Results: змінивши функцію активації з Sigmoid на Softsign та ReLU ми змогли досягти більшої точності на 1%.