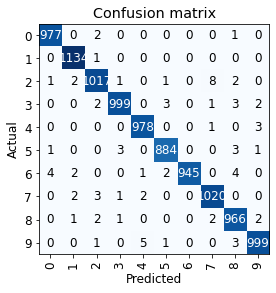
I applied the resnet18 network on the MNIST dataset in fast.ai and trained it for 3 epochs. The code was straightforward and finding the documentation to interpret the predictions was also easy to find. In my search for documentation, I also stumbled upon a Chinese version of MNIST, which I thought was pretty cool and shows the wide applications of neural networks and their ability to generalize. Similarly, there are even Japanese and other language versions of MNIST. I don’t know if it was because I was running the Google Collab file on my school laptop or not, but the actual training took nearly an hour, with each epoch running for 15-17 minutes. The improvement in accuracy and decrease in train\_loss and val\_loss was also incredible, especially since I only ran it for 3 epochs but was still able to create a model with over 99% accuracy. Meanwhile, it took me 20 or even more epochs on my own version of the neural network using backpropagation to get even 98% accuracy on the testing dataset. The difference in performance is staggering.

Confusion Matrix:



List of Most Confused Labels:

[('2', '7', 8),

('9', '4', 5),

('6', '0', 4),

('6', '8', 4),

('3', '5', 3),

('3', '8', 3),

('4', '9', 3),

('5', '3', 3),

('5', '8', 3),

('7', '2', 3),

('9', '8', 3),

('0', '2', 2),

('2', '1', 2),

('2', '8', 2),

('3', '2', 2),

('3', '9', 2),

('6', '1', 2),

('6', '5', 2),

('7', '1', 2),

('7', '4', 2),

('8', '2', 2),

('8', '7', 2),

('8', '9', 2),

('0', '8', 1),

('1', '2', 1),

('2', '0', 1),

('2', '3', 1),

('2', '5', 1),

('3', '7', 1),

('4', '7', 1),

('5', '0', 1),

('5', '9', 1),

('6', '4', 1),

('7', '3', 1),

('8', '1', 1),

('8', '3', 1),

('9', '2', 1),

('9', '5', 1)]

Plot of Top Losses:

