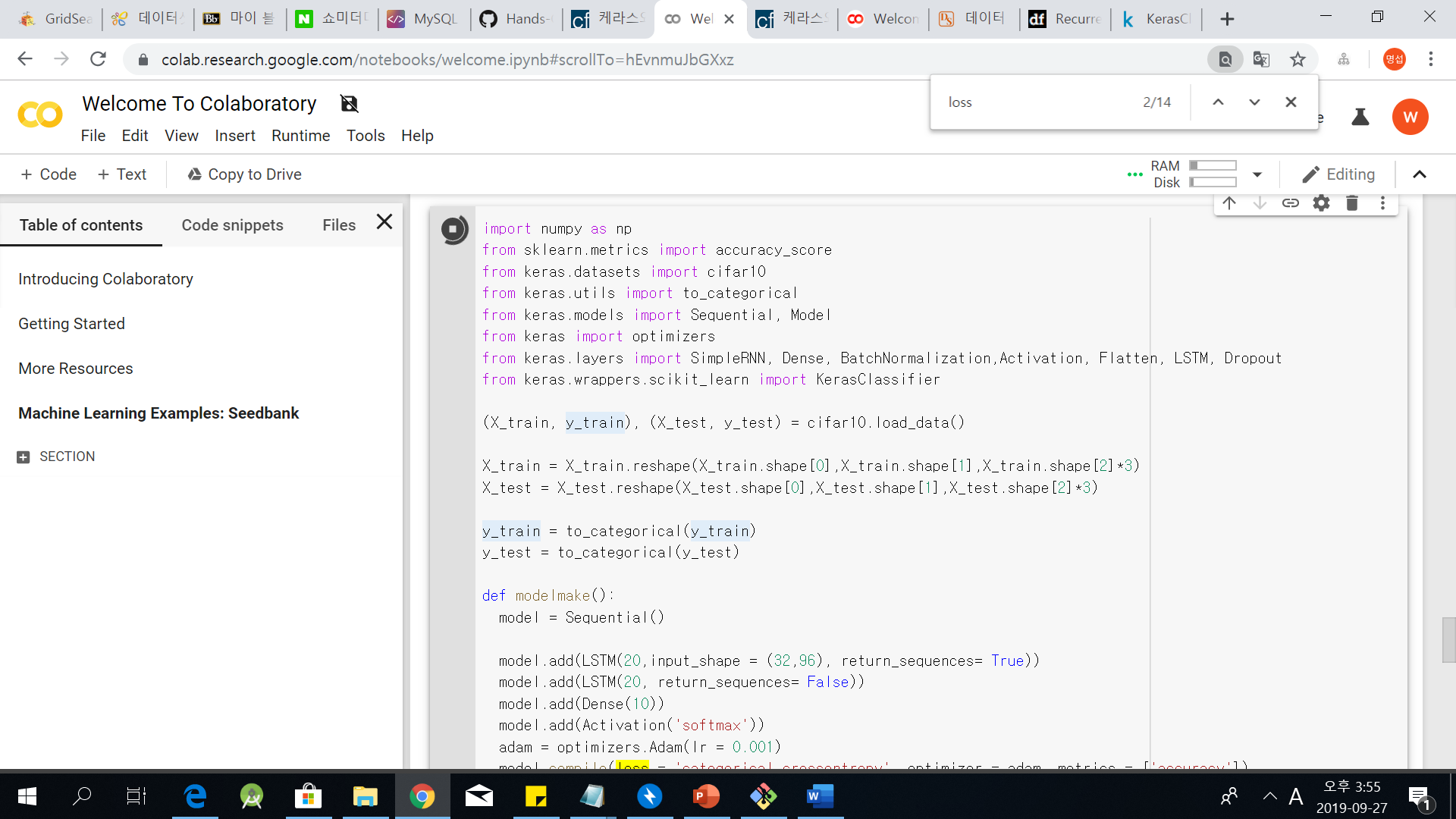
Cifar-10 RNN과 LSTM으로 구현하기

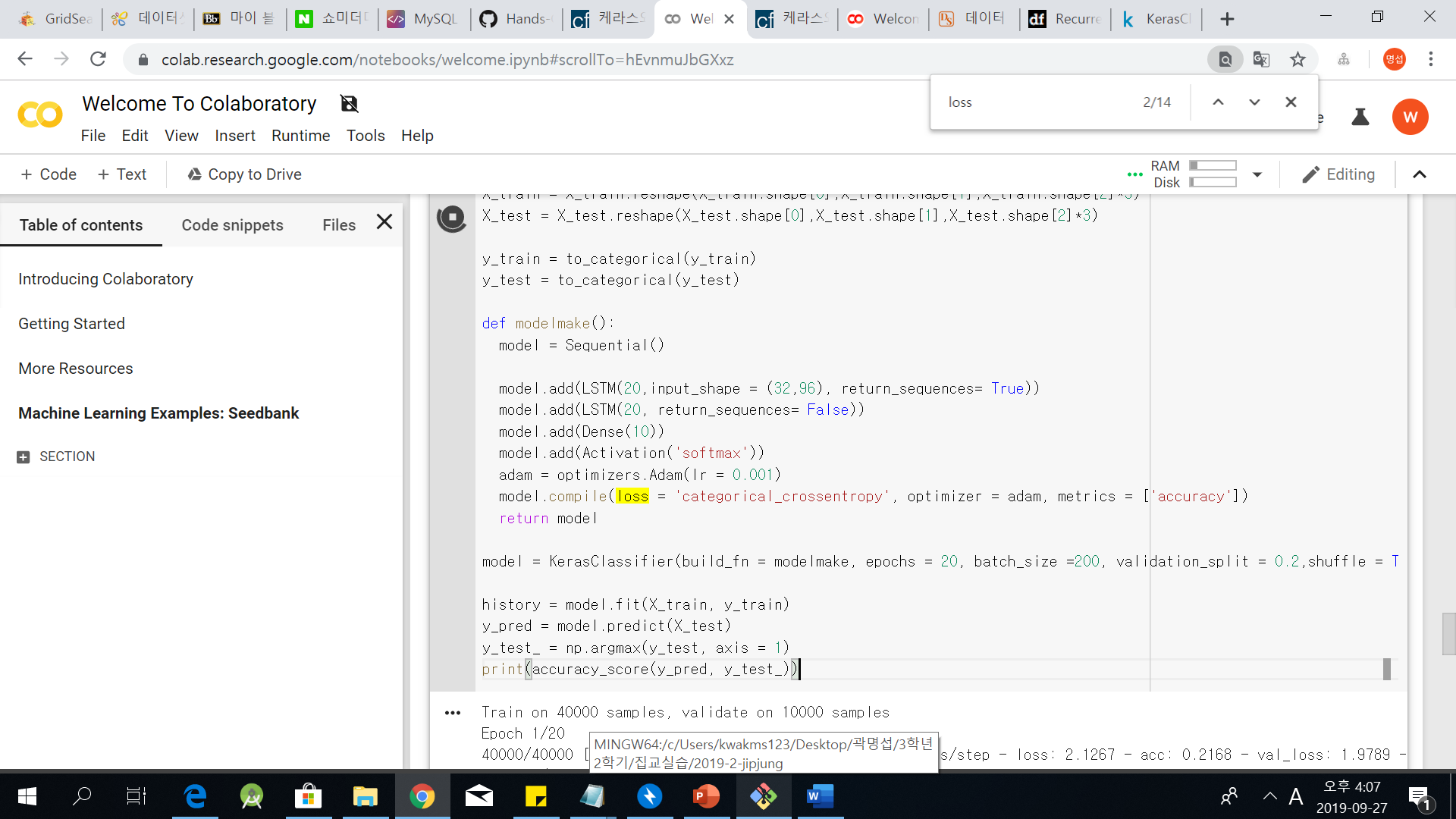
201520991

곽명섭

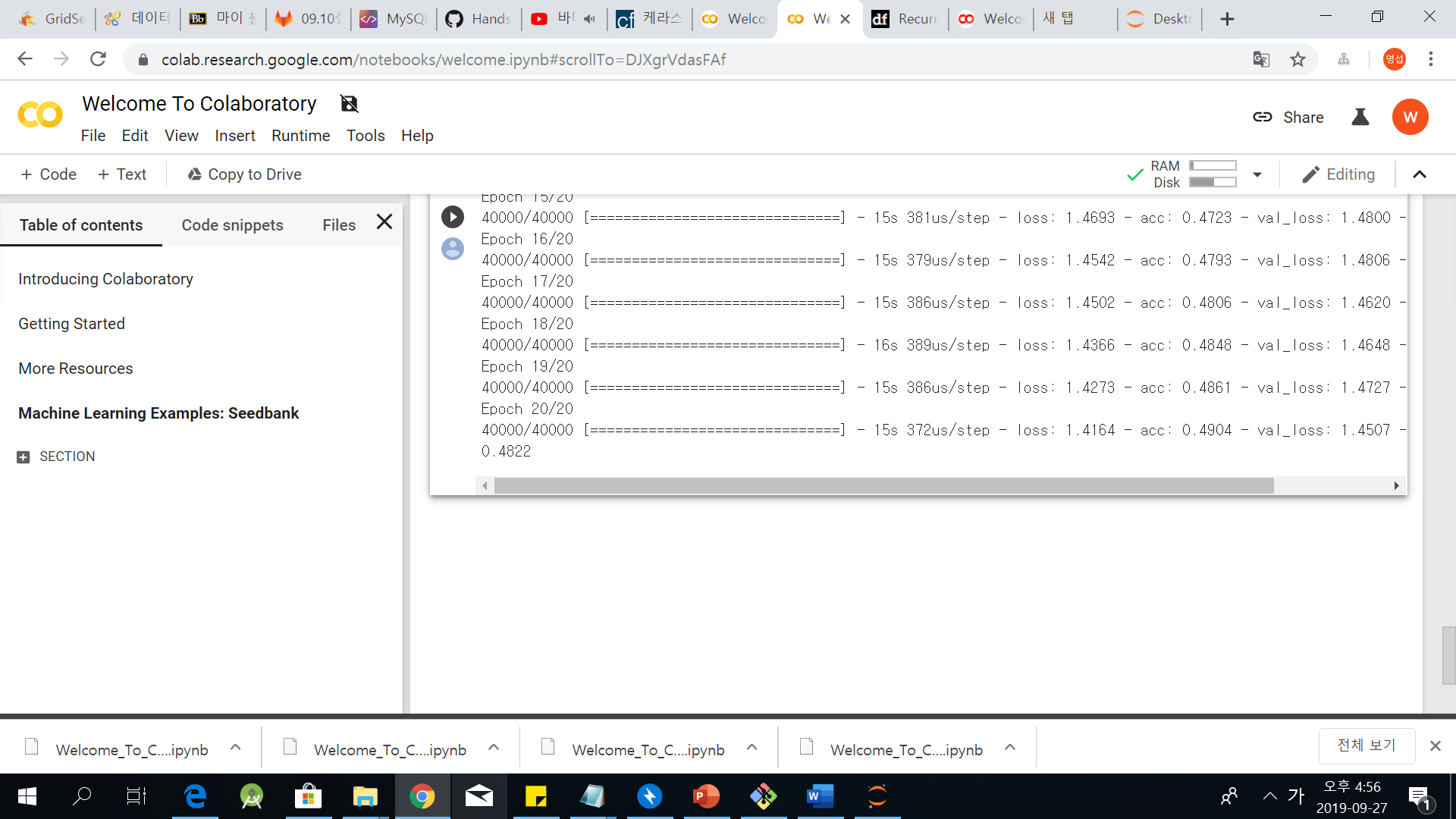
오늘은 저번에 CNN과 ANN을 이용하여 실습으로 하였던 Cifar10 데이터셋을 RNN과 LSTM을 이용하여 구현하는 실습을 해보았습니다. 먼저 일단 Cifar10 데이터셋을 불러오는 것부터 해야 했습니다.



그런데 데이터 전처리를 해주기위해서 RNN과 LSTM은 3차원 배열을 2차원 배열로 바꾸어 주는 처리부터 먼저 해주어야 했습니다. 왜냐하면 원래 cifar 10데이터는 training set같은경우 50000개의 32 \* 32 \* 3의 3차원 데이터로 되어있는데 RNN과 LSTM을 돌려 주기위해선 이것을 2차원 데이터로 바꿔주는 과정을 해주어야 했는데 그것을 한 게 저기 reshape함수를 사용한 부분입니다. 여기서 데이터를 어떤 식의 2차원 데이터로 바꿔 주어야 할지 모르겠 어서 그냥 32 \* 96 2차원데이터로 바꿔 버렸습니다. Test set도 똑같이 10000개의 32 \* 32 \* 3의 3차원 데이터로 되어 있는 걸 똑같이 32 \* 96의 2차원 데이터로 바꿔 주었습니다. 그리고 데이터를 255로나누는 정규화 과정도 있는데 사진에는 빠졌습니다. 모든 모델에서 epoch값은 20, optimizer는 adam, learning rate는 0.001로 맞춰 주었습니다

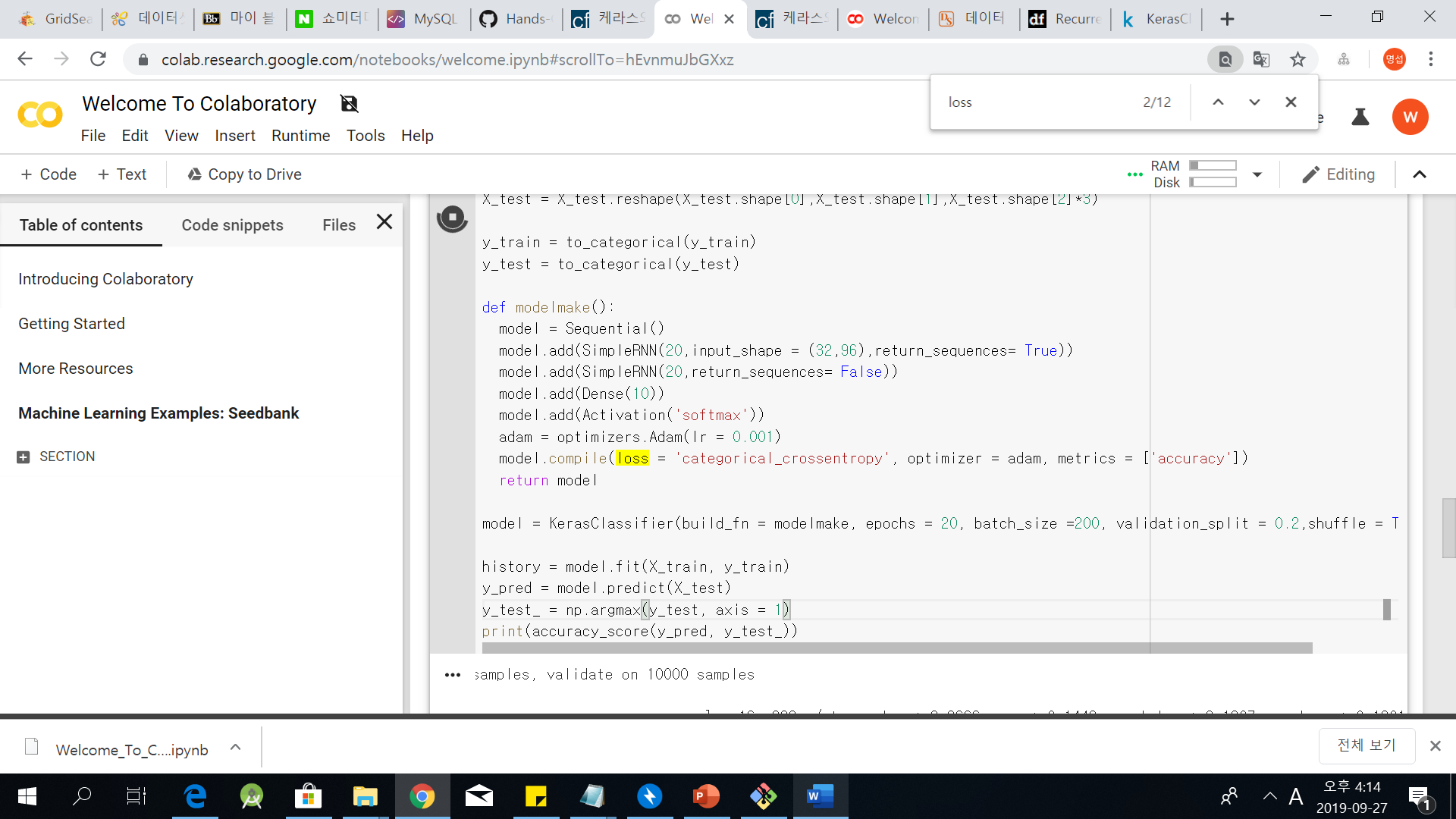


그리고 LSTM층을 2개를 쌓아주었습니다. 맨 첫 input layer의 LSTM을 쌓아줄때는 input\_shape라는 제가 넣어줄 데이터의 모양을 parameter로 넣어 주었습니다. 여기서 return\_sequences라는 parameter는 만약 true이면 강의노트에 있는 many to many이고 false이면 many to one 이기 때문에 무조건 마지막 layer에는 many to one으로 return\_sequences라는 parameter를 false로 설정을 해주 어야 한다. LSTM Input layer 1개와 hidden layer 1개를 쌓고 dense layer랑 activation function도 softmax 설정해주었고 optimizer는 Adam으로 설정해주었고 learning rate도 0.001로 설정해주었습니다.

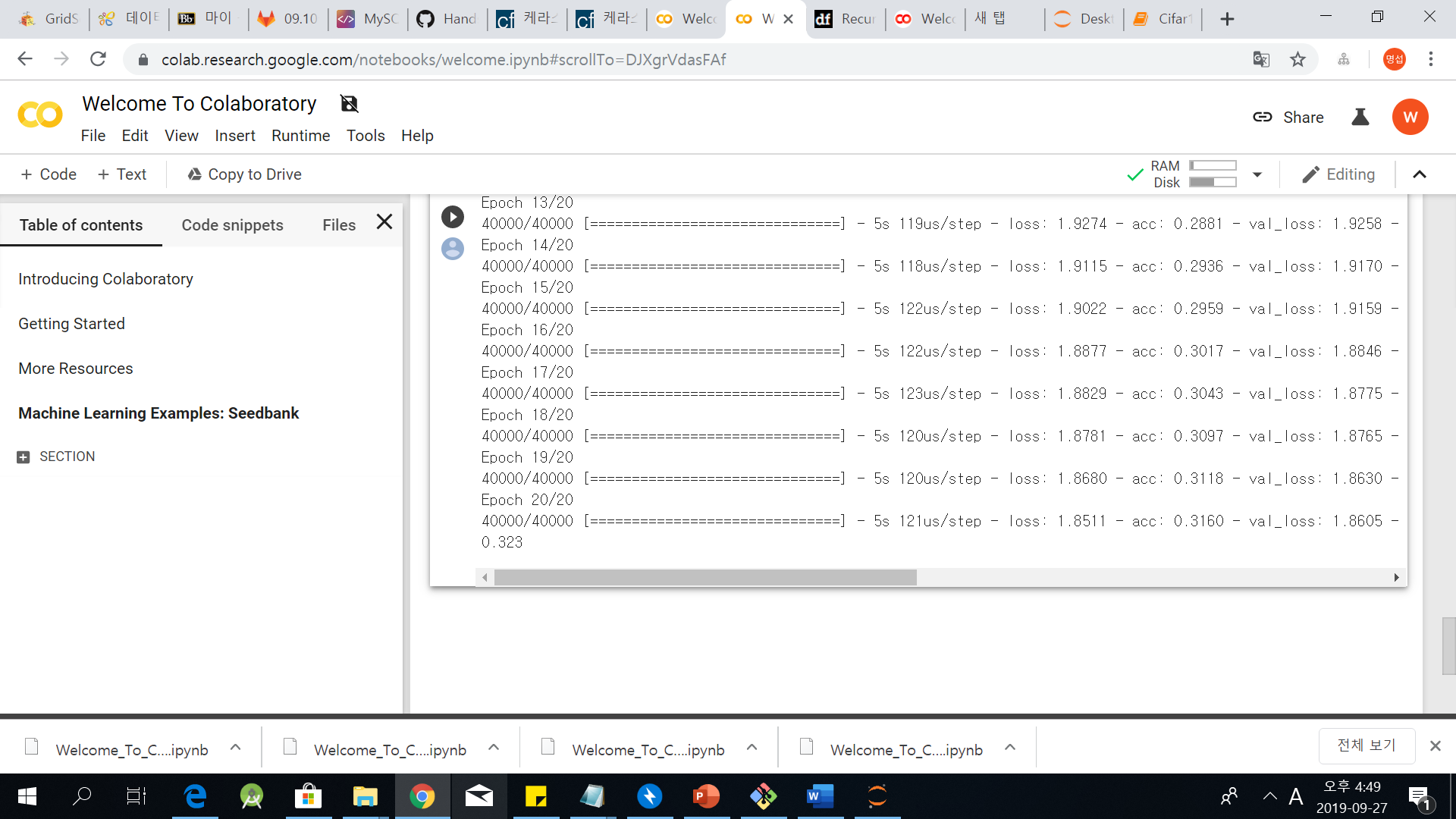


Accuracy는 0.4822가 나왔다.

다음은 RNN을 사용하여 cifar10에 적용을 해보았다. 데이터 전처리를 할때는 LSTM과 같이 3차원 데이터를 2차원 데이터로 바꿔 주는 그런 과정에는 별로 차이가 없고 모델을 만들 때만 약간 다르게 LSTM대신 SimpleRNN으로 바꿔주는 과정 만을 하였습니다

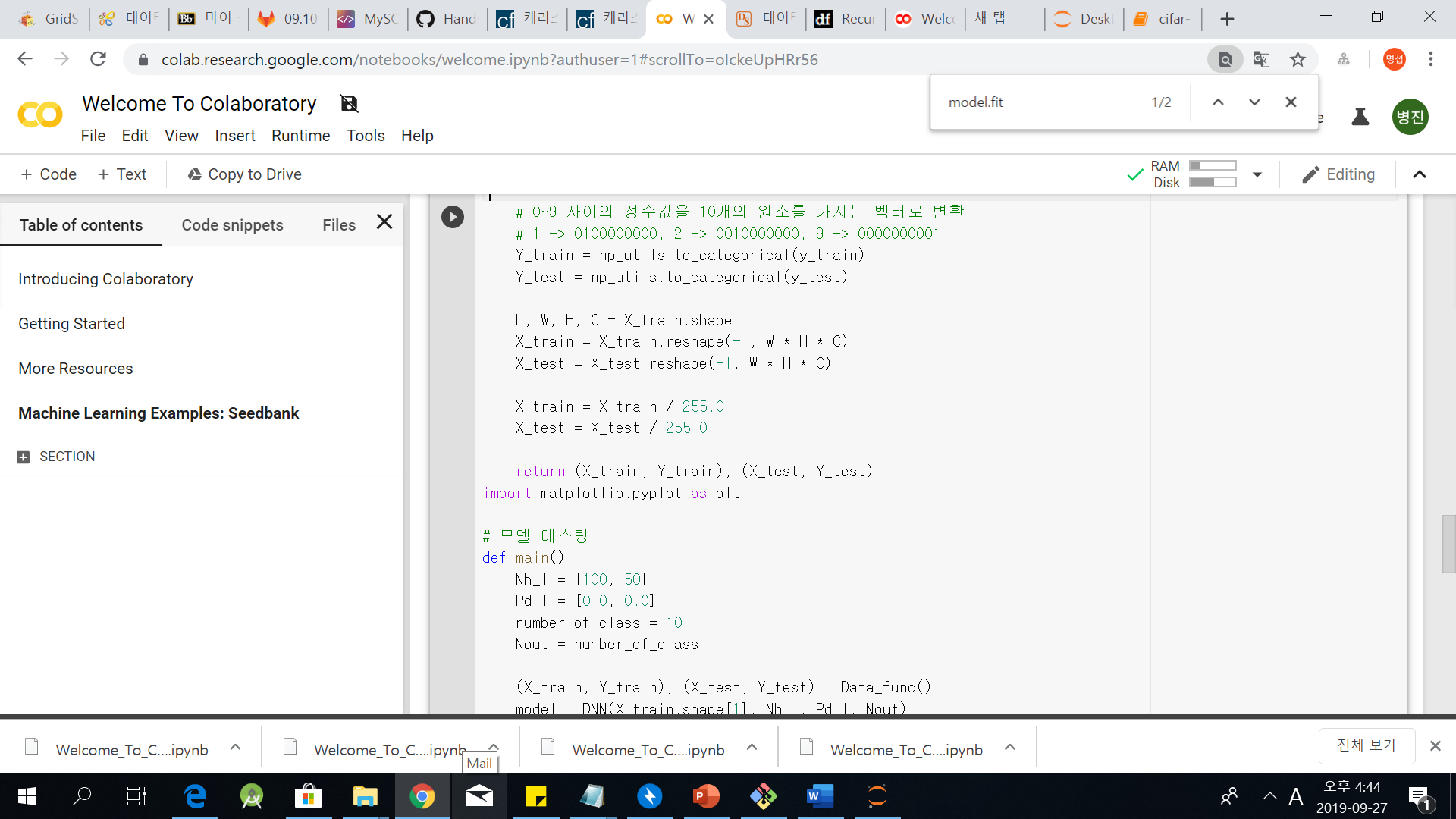


여기서도 input\_shape parameter같은 경우엔 32,96으로 바꿔 주었고 return\_sequences도 input layer에선 True, hidden layer에서는 False로 설정을 해주었고 나머지도 똑같이 해주었습니다.

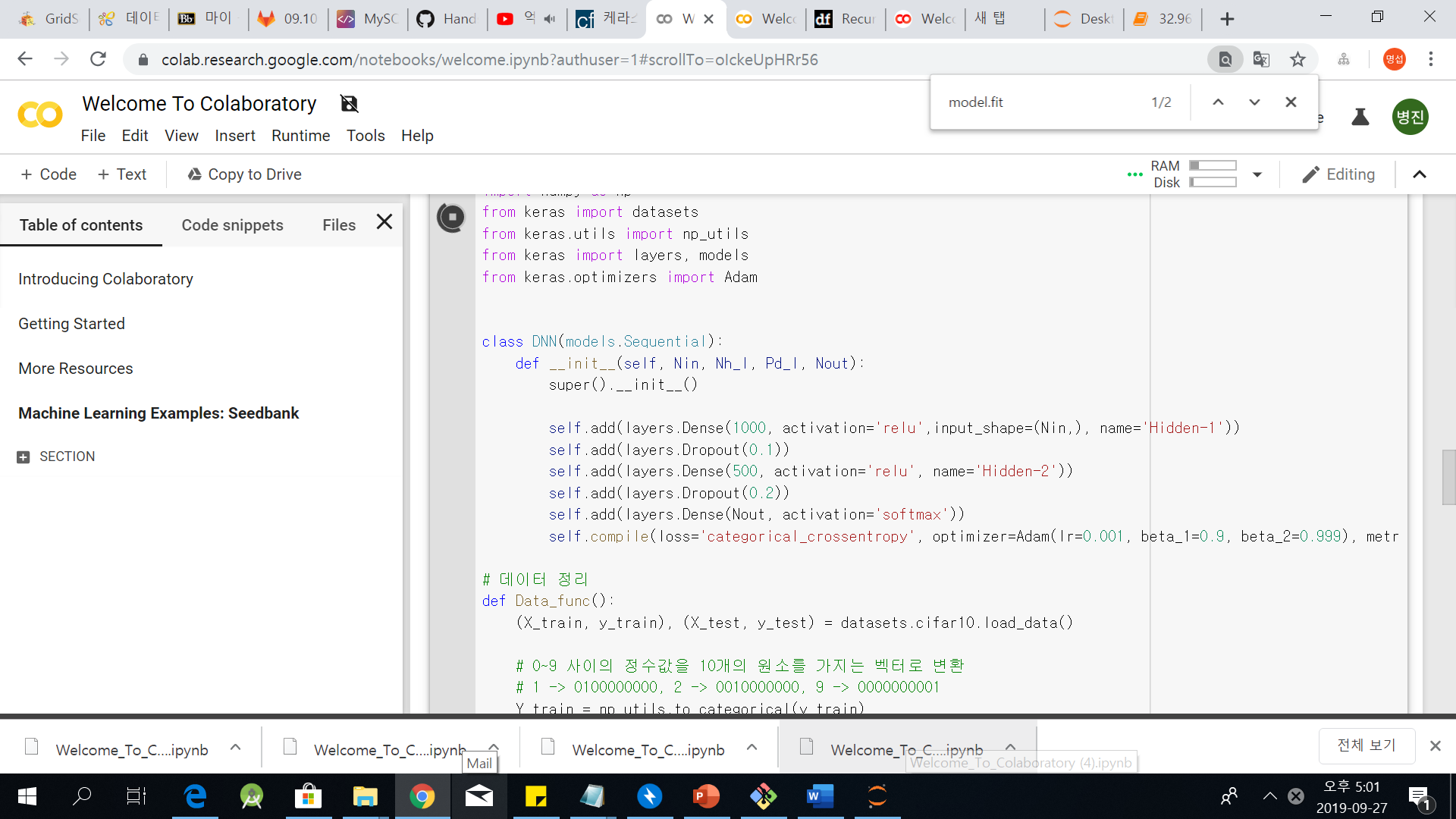


Accuracy 같은 경우엔 0.323정도가 나왔습니다.

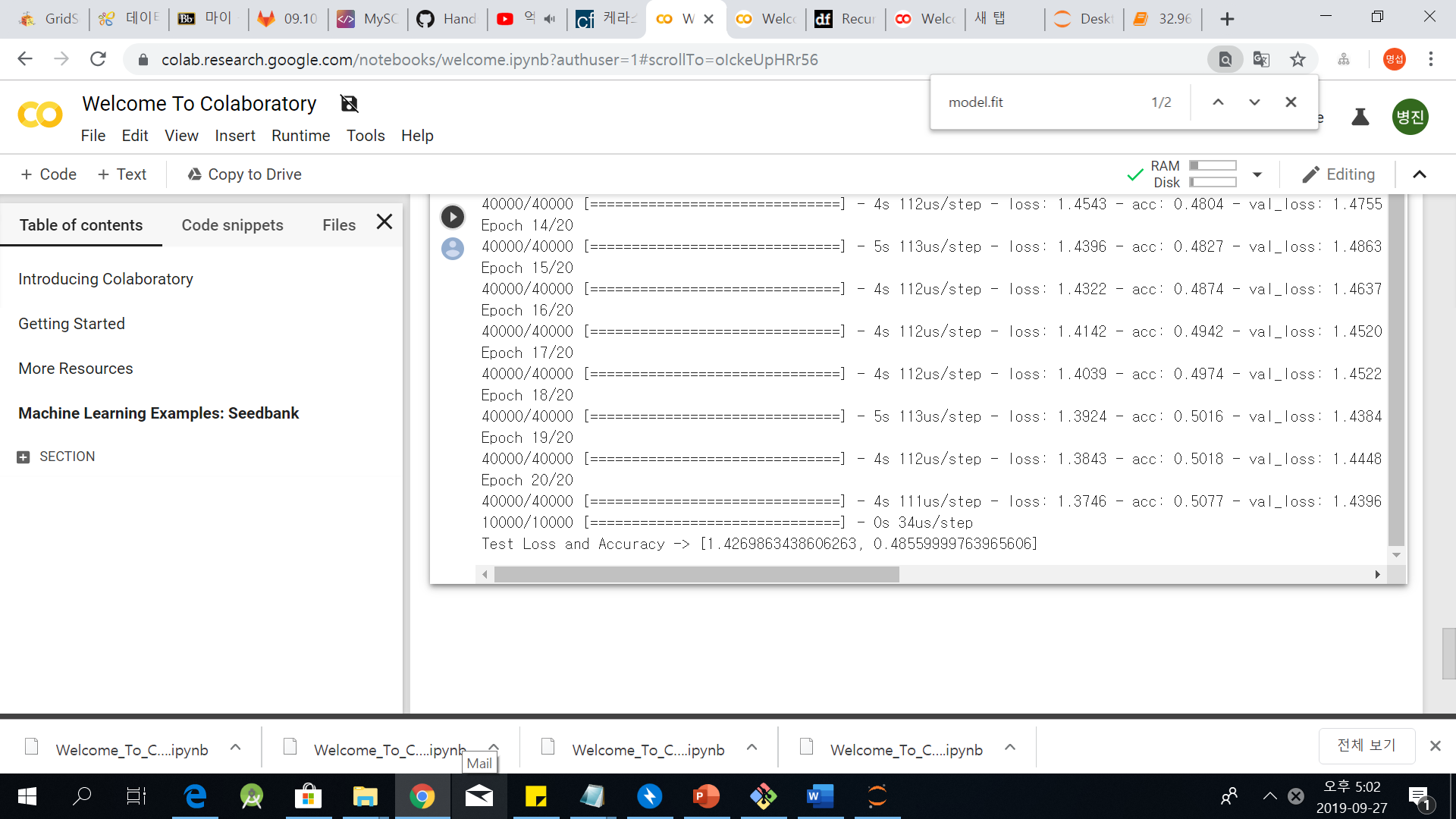
그리고 이번에는 저번에 했던 DNN과 CNN을 다시 돌려 보았습니다. 이번엔 DNN을 돌려 보았는데 DNN같은 경우엔 training set같은 경우엔 50000개 데이터의 32 \* 32 \* 3 의 3차원데이터를 그냥 1차원으로 쭉 펴버리는 과정으로 reshape를 해주었습니다.



다음은 DNN 모델을 쌓아주는 과정입니다.

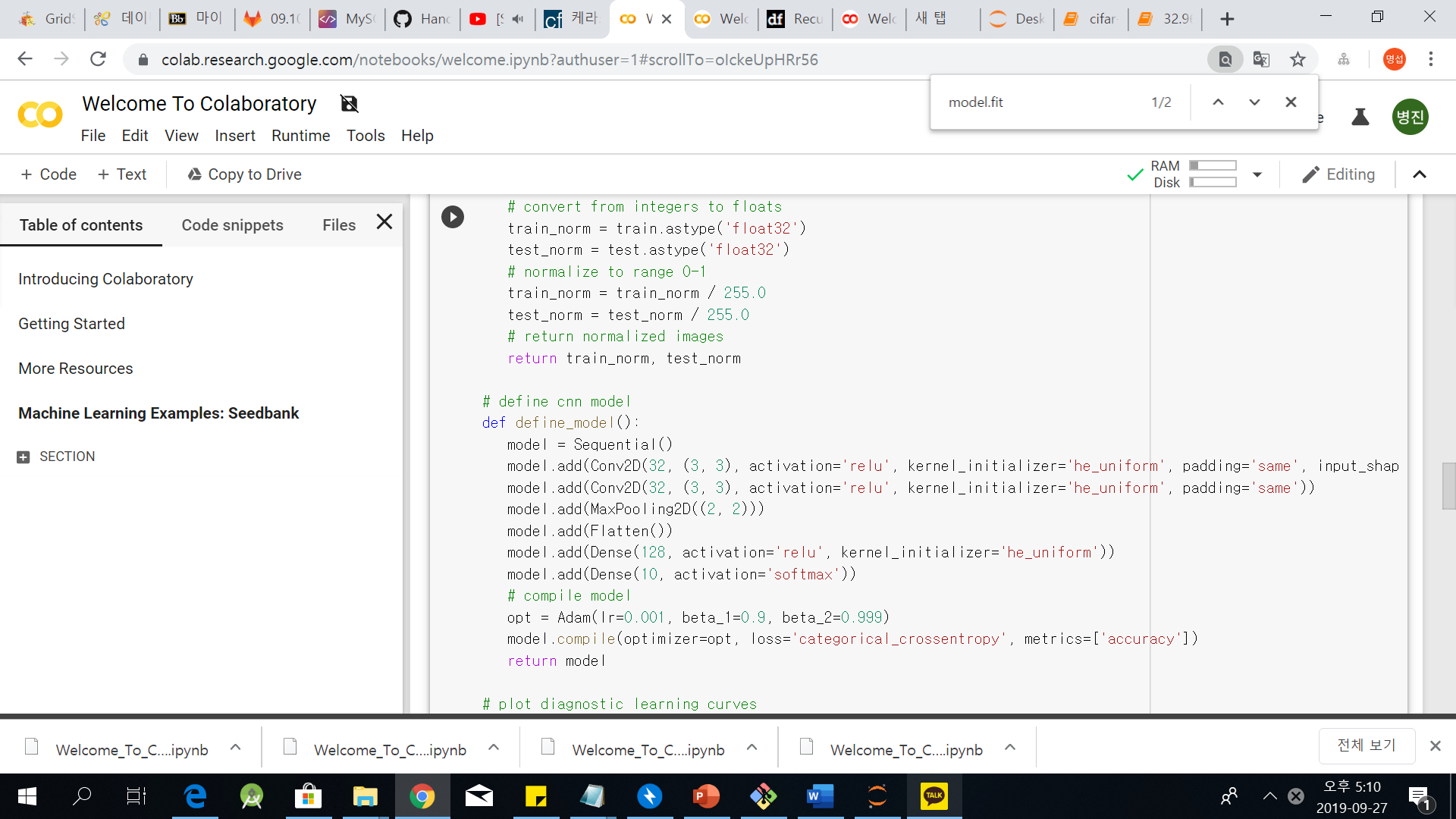


2층을 쌓아주었으며 노드의 수는 1000,500으로 맞춰 주었고 drop out도 0.2 한번 0.1한번 넣어 주었습니다. 최대한 각 방법에 대한 비교를 위해 optimizer도 앞에서 말한데로 adam, learning rate는 0.001로 통일해주고 돌려주었습니다.

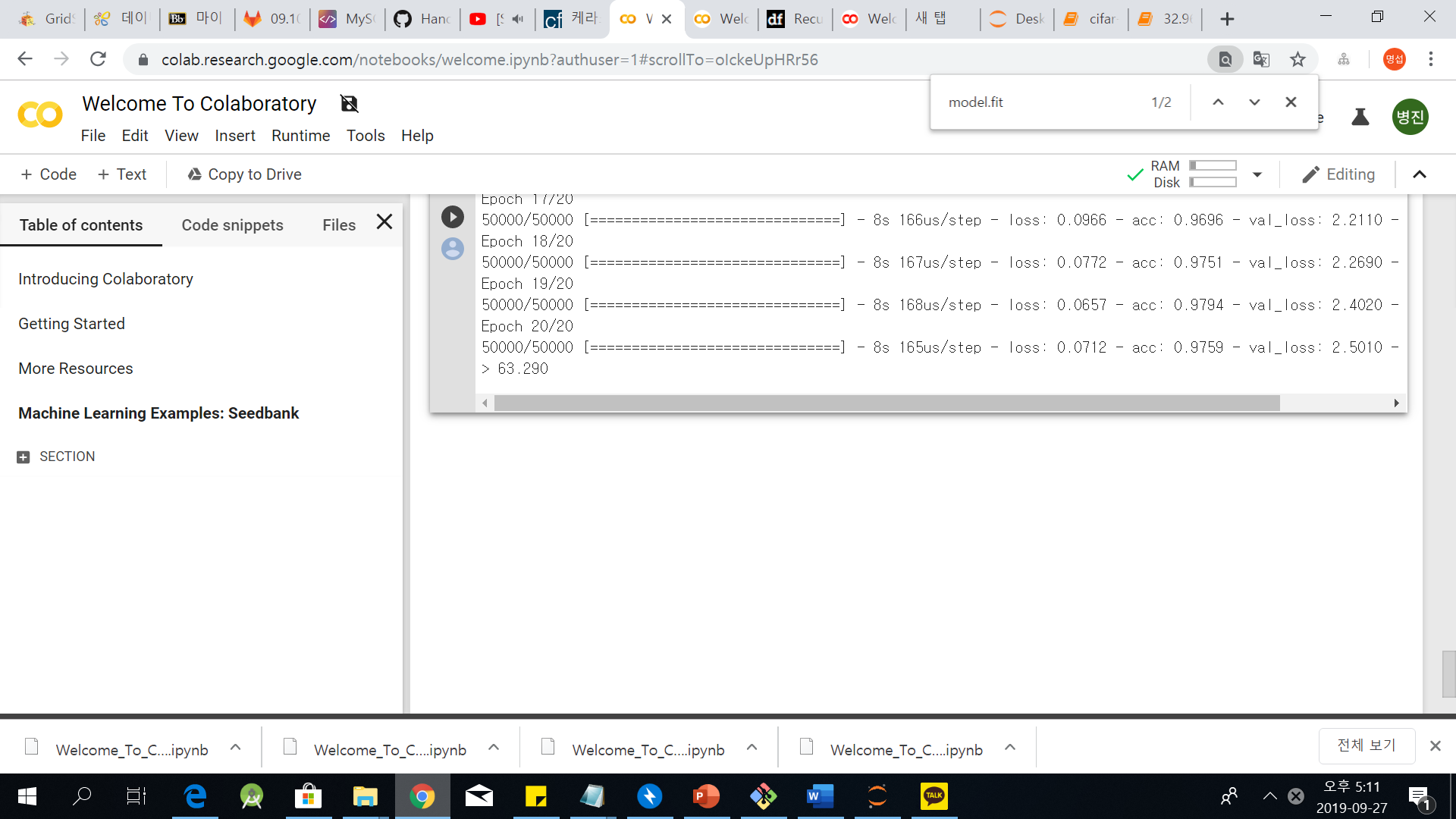


Accuracy는 0.4855가 나왔다.

CNN 같은 경우엔 딱히 데이터를 가공할 필요없이 넣어주면 되었고



Layer는 이렇게 쌓아주고 통일을 위해 optimizer는 adam, learning rate는 0.001로 설정을 해주었다. 그리고 epoch도 20으로 동일하게 설정해주었다.



Accuracy는 0.6329가 나왔습니다.

이번 실습은 제가 각 방법의 accuracy를 높이는데 초점을 두기보다는 그냥 같은 조건에서 어떤 방법을 썼을 때가 더 좋고 accuracy가 잘 나오는지 비교를 하는데에 초점을 두고 실습을 하였습니다. 아마 각각의 accuracy를 높이는데 초점을 두었다면 아마 epoch값도 더 길게 두고 learningrate도 더 낮추고 layer도 최적의 개수만큼 쌓아서 했을 것이지만 epoch = 20, lr = 0.001, layer는 2개로 통일을 해서 비교에 좀더 초점을 두었습니다 결과는 생각대로 CNN이 가장 높게 나왔고 DNN과 RNN은 비슷한 accuracy가 나왔으며 LSTM이 가장 낮은 accuracy가 나오게 되었습니다