how can i modify this code by using for loop for each epoch rather giving epoch=max\_num\_epoch and make epoch=1 in .fit\_generator()

You can modify the code to use a for loop for each epoch by replacing the model.fit\_generator() call with a for loop that iterates for each epoch. Here's an example:

python

Copy code

for epoch in range(max\_num\_epochs):

print("Epoch ", epoch+1, "/", max\_num\_epochs)

train\_loss = 0.0

train\_jacc\_coef = 0.0

num\_batches\_train = 0

# Train the model for one epoch

for batch\_idx, (x\_train, y\_train) in enumerate(mybatch\_generator\_train(list(zip(train\_img\_split, train\_msk\_split)), in\_rows, in\_cols, batch\_sz, max\_bit)):

# Train on one batch

loss, jacc\_coef = model.train\_on\_batch(x\_train, y\_train)

train\_loss += loss

train\_jacc\_coef += jacc\_coef

num\_batches\_train += 1

# Compute the average train loss and jaccard coefficient for this epoch

train\_loss /= num\_batches\_train

train\_jacc\_coef /= num\_batches\_train

# Evaluate the model on the validation set

val\_loss, val\_jacc\_coef = model.evaluate\_generator(

mybatch\_generator\_validation(list(zip(val\_img\_split, val\_msk\_split)), in\_rows, in\_cols, batch\_sz, max\_bit),

steps=np.ceil(len(val\_img\_split) / batch\_sz))

# Print the train and validation metrics for this epoch

print("Train Loss: ", train\_loss, " - Train Jacc Coef: ", train\_jacc\_coef)

print("Val Loss: ", val\_loss, " - Val Jacc Coef: ", val\_jacc\_coef)

# Save the model weights if the validation loss has improved

if val\_loss < best\_val\_loss:

print("Validation loss improved. Saving weights.")

best\_val\_loss = val\_loss

model.save\_weights(weights\_path)

# Update the learning rate

lr\_reducer.on\_epoch\_end(epoch, logs={'val\_loss': val\_loss})

# Log the train and validation metrics to a CSV file

csv\_logger.on\_epoch\_end(epoch, logs={'epoch': epoch+1, 'train\_loss': train\_loss, 'train\_jacc\_coef': train\_jacc\_coef, 'val\_loss': val\_loss, 'val\_jacc\_coef': val\_jacc\_coef})

In this code, we iterate over max\_num\_epochs epochs using a for loop. Within each epoch, we train the model on the training set using model.train\_on\_batch() and evaluate the model on the validation set using model.evaluate\_generator(). We then update the learning rate using lr\_reducer.on\_epoch\_end() and log the train and validation metrics to a CSV file using csv\_logger.on\_epoch\_end(). Finally, we save the model weights if the validation loss has improved.