



During training, the generator and discriminator work in a tight loop as following:

Generator

- The generator takes in a batch of single-track piano rolls (melody) as the input and generates a batch of multi-track piano rolls as the output by adding accompaniments to each of the input music tracks.
- The discriminator then takes these generated music tracks and predicts how far it deviates from the real data present in your training dataset.

Discriminator

- This feedback from the discriminator is used by the generator to update its weights. As the generator gets better at creating music accompaniments, it begins fooling the discriminator. So, the discriminator needs to be retrained as well.
- Beginning with the discriminator on the first iteration, we alternate between training these two networks until we reach some stop condition (ex: the algorithm has seen the entire dataset a certain number of times).

Finer control of AWS DeepComposer with hyperparameters

As you explore training your own custom model in the AWS DeepComposer console, you will notice you have access to several **hyperparameters** for finer control of this process. Here are a few details on each to help guide your exploration.

Number of epochs

When the training loop has passed through the entire training dataset once, we call that one epoch. Training for a higher number of epochs will mean your model will take longer to complete its training task, but it may produce better output if it has not yet converged. You will learn how to determine when a model has completed most of its training in the next section.

Training over more epochs will take longer but can lead to a better sounding musical output

Model training is a trade-off between the number of epochs (i.e. time) and the quality of sample output.

Sample training input

▶

0:02 / 0:19

🔊

⋮

Sample output at 100 epochs

▶

0:02 / 0:19

🔊

⋮

Sample output at 400 epochs

▶

0:02 / 0:19

🔊

⋮

Apply your learning in AWS DeepComposer

[Try training a custom model using less than 250 epochs](#)

Learning Rate

The learning rate controls how rapidly the weights and biases of each network are updated during training. A higher learning rate might allow the network to explore a wider set of model weights, but might pass over more optimal weights.

Update ratio

A ratio of the number of times the discriminator is updated per generator training epoch. Updating the discriminator multiple times per generator training epoch is useful because it can improve the discriminators accuracy. Changing this ratio might allow the generator to learn more quickly early-on, but will increase the overall training time.

While we provide sensible defaults for these hyperparameters in the AWS DeepComposer console, you are encouraged to explore other settings to see how each changes your model's performance and time required to finish training your model.

QUIZ QUESTION

What is a **hyperparameter**?

- ☐ A parameter whose value is set during the training process
- ☒ A parameter whose value is set before the training process begins