**GANs with AWS DeepComposer**

**Summary**

We’ll begin our journey of popular generative models in AWS DeepComposer with generative adversarial networks or GANs. Within an AWS DeepComposer GAN, models are used to solve a creative task: adding accompaniments that match the style of an input track you provide. Listen to the input melody and the output composition created by the AWS DeepComposer GAN model:

* [Input melody](https://video.udacity-data.com/topher/2021/April/607ddd51_input-twinkle-twinkle-input/input-twinkle-twinkle-input.mp3)
* [Output melody](https://video.udacity-data.com/topher/2021/April/607ddd92_output-twinkle-twinkle-rock/output-twinkle-twinkle-rock.mp3)

**What are GANs?**

A GAN is a type of generative machine learning model which pits two neural networks against each other to generate new content: a generator and a discriminator.

* A *generator* is a neural network that learns to create new data resembling the source data on which it was trained.
* A *discriminator* is another neural network trained to differentiate between real and synthetic data.

The generator and the discriminator are trained in *alternating cycles*. The generator learns to produce more and more realistic data while the discriminator iteratively gets better at learning to differentiate real data from the newly created data.

**Collaboration between an orchestra and its conductor**

A simple metaphor of an orchestra and its conductor can be used to understand a GAN. The orchestra trains, practices, and tries to generate polished music, and then the conductor works with them, as both judge and coach. The conductor judges the quality of the output and at the same time provides feedback to achieve a specific style. The more they work together, the better the orchestra can perform.

The GAN models that AWS DeepComposer uses work in a similar fashion. There are two competing networks working together to learn how to generate musical compositions in distinctive styles.

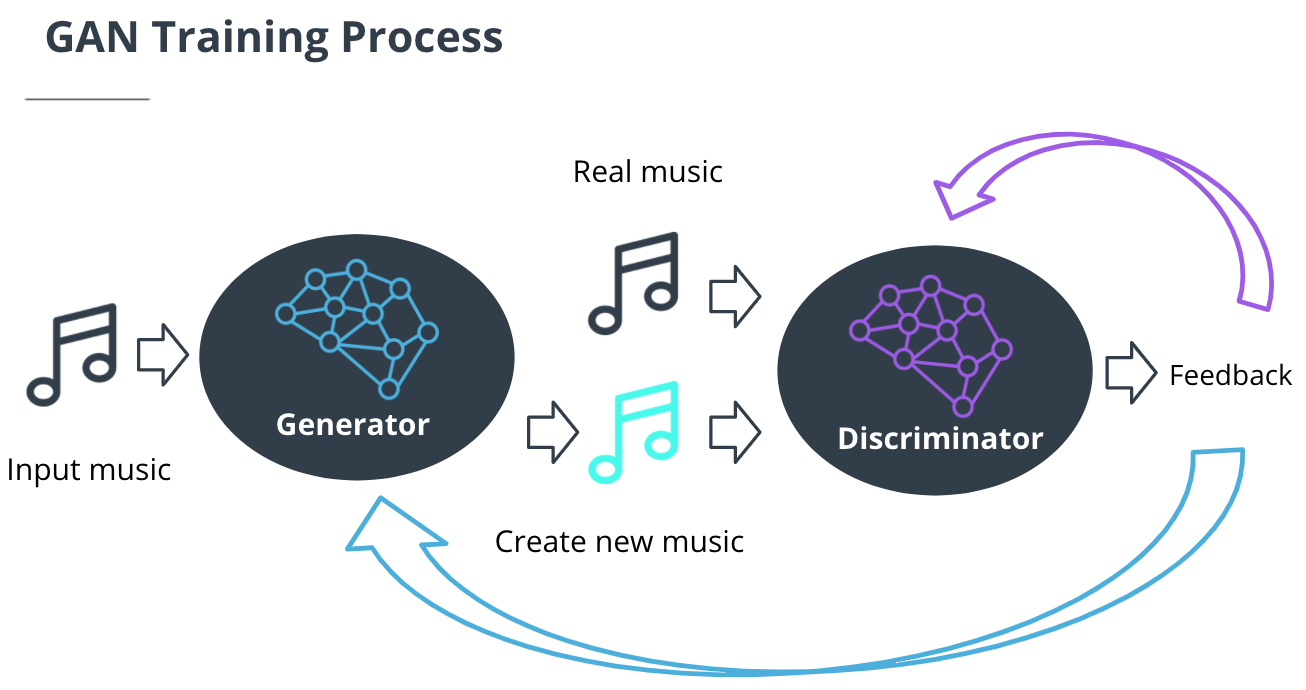
A GAN's generator produces new music as the orchestra does. And the discriminator judges whether the music generator creates is realistic and provides feedback on how to make its data more realistic, just as a conductor provides feedback to make an orchestra sound better.



An orchestra and its conductor

**Training Methodology**

Let’s dig one level deeper by looking at how GANs are trained and used within AWS DeepComposer. During training, the generator and discriminator work in a tight loop as depicted in the following image.



A schema representing a GAN model used within AWS DeepComposer

**Note:** While this figure shows the generator taking input on the left, GANs in general can also generate new data without any input.

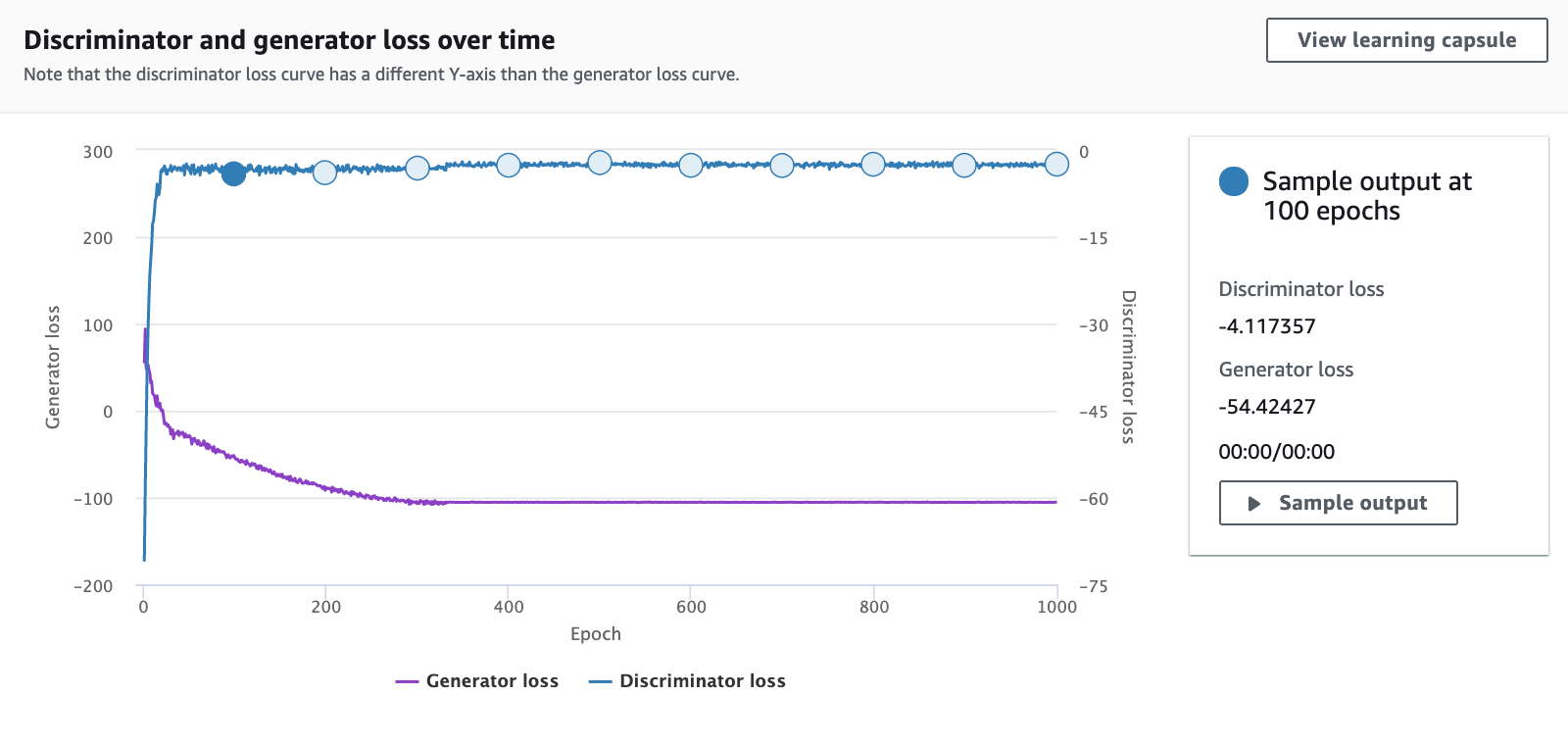
**Generator**

* 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 they deviate from the real data present in the training dataset. This deviation is called the *generator loss*. This feedback from the discriminator is used by the generator to incrementally get better at creating realistic output.

**Discriminator**

* As the generator gets better at creating music accompaniments, it begins fooling the discriminator. So, the discriminator needs to be retrained as well. The discriminator measures the *discriminator loss* to evaluate how well it is differentiating between real and fake data.

Beginning with the discriminator on the first iteration, we **alternate training these two networks** until we reach some stop condition; for example, the algorithm has seen the entire dataset a certain number of times or the generator and discriminator loss reach some plateau (as shown in the following image).



Discriminator loss and generator loss reach a plateau

**New Terms**

* **Generator:** A neural network that learns to create new data resembling the source data on which it was trained.
* **Discriminator**: A neural network trained to differentiate between real and synthetic data.
* **Generator loss:** Measures how far the output data deviates from the real data present in the training dataset.
* **Discriminator loss:** Evaluates how well the discriminator differentiates between real and fake data.

**Supporting Materials**

* [Input Twinkle Twinkle Input](https://video.udacity-data.com/topher/2021/April/607ddd51_input-twinkle-twinkle-input/input-twinkle-twinkle-input.mp3)
* [Output Twinkle Twinkle Rock](https://video.udacity-data.com/topher/2021/April/607ddd92_output-twinkle-twinkle-rock/output-twinkle-twinkle-rock.mp3)

;