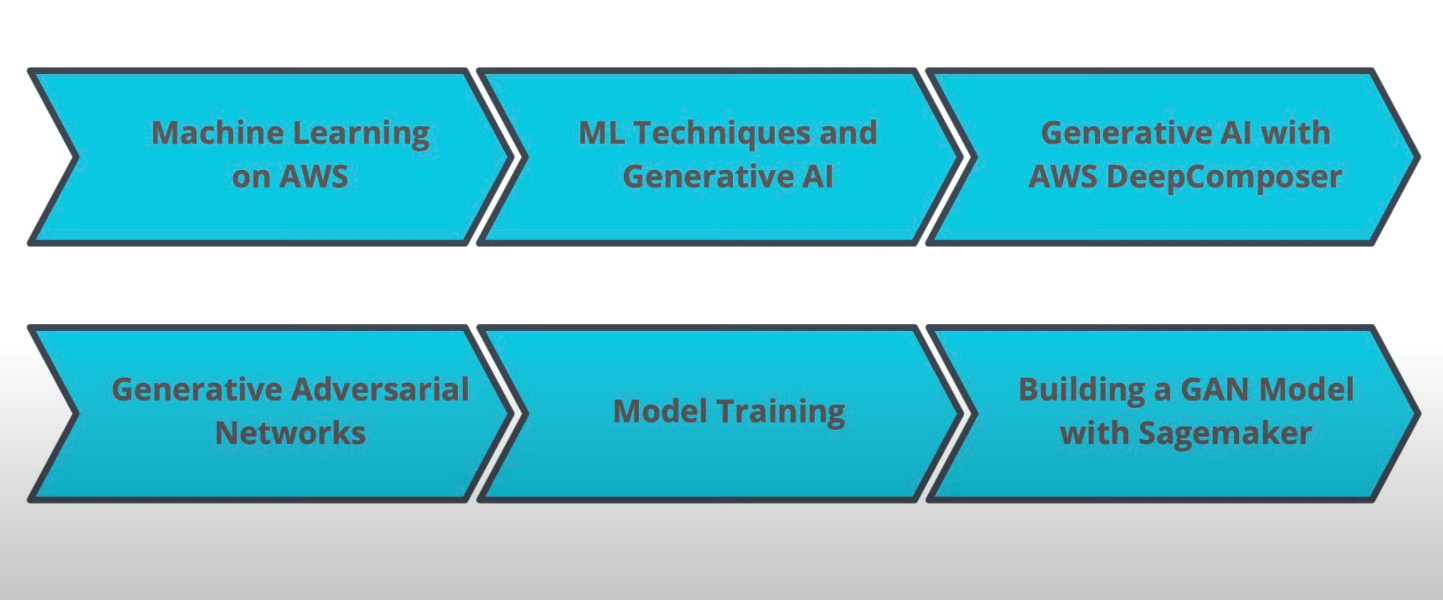
In this lesson you'll get an introduction to Machine Learning. You will learn about Generative AI and AWS DeepComposer. You'll also learn how to build a custom Generative Adversarial Network.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/dcd4f7b5-ea13-423b-a23c-9ef5672a1349)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/dcd4f7b5-ea13-423b-a23c-9ef5672a1349)**

**[What You Will Learn](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/dcd4f7b5-ea13-423b-a23c-9ef5672a1349)**

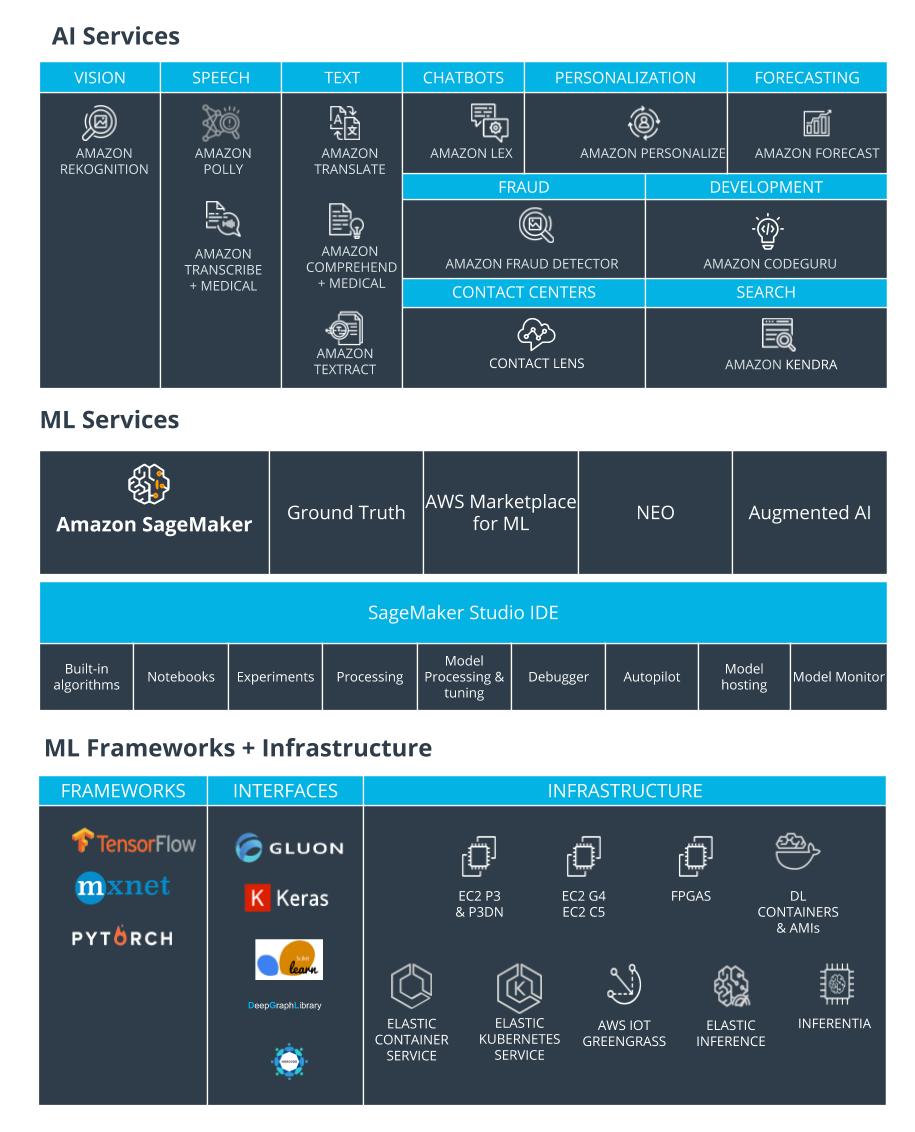
We hope this will be an interesting and exciting learning experience for you.

**AWS Mission**

Put machine learning in the hands of every developer.

**Why AWS?**

* AWS offers the broadest and deepest set of AI and ML services with unmatched flexibility.
* You can accelerate your adoption of machine learning with AWS SageMaker. Models that previously took months and required specialized expertise can now be built in weeks or even days.
* AWS offers the most comprehensive cloud offering optimized for machine learning.
* More machine learning happens at AWS than anywhere else.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)**

**[AWS Machine Learning Stack](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)**

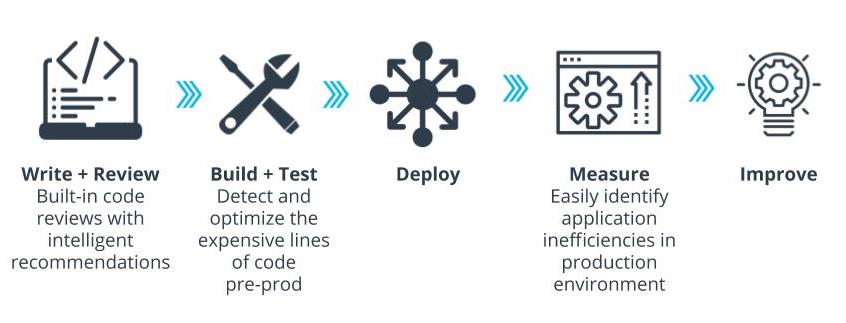
**More Relevant Enterprise Search With Amazon Kendra**

* Natural language search with contextual search results
* ML-optimized index to find more precise answers
* 20+ Native Connectors to simplify and accelerate integration
* Simple API to integrate search and easily develop search applications
* Incremental learning through feedback to deliver up-to-date relevant answers

**Online Fraud Detection with Amazon Fraud Detector**

* Pre-built fraud detection model templates
* Automatic creation of custom fraud detection models
* One interface to review past evaluations and detection logic
* Models learn from past attempts to defraud Amazon
* Amazon SageMaker integration

**Amazon Code Guru**

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)**

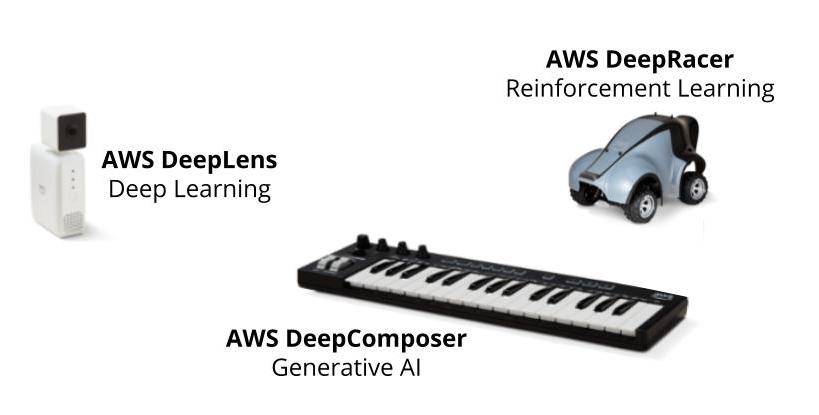
**[Amazon CodeGuru For High-performing Software](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)**

**Better Insights And Customer Service With Contact Lens**

* Identify common call types
* Identify recurring themes based on customer call feedback
* Alert supervisors when customers are having a poor experience
* Assist agents with a knowledge base to answer questions as they are being asked

**How to Get Started?**

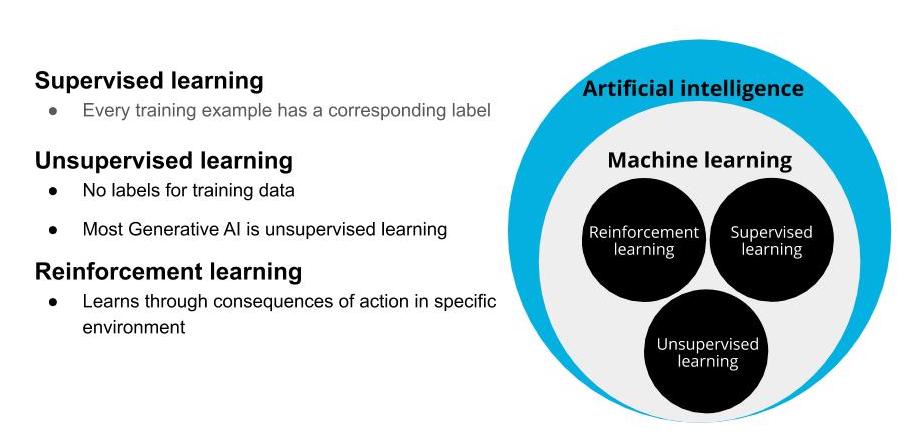
* [**AWS DeepLens**](https://aws.amazon.com/deeplens/): deep learning and computer vision
* [**AWS DeepRacer**](https://aws.amazon.com/deepracer/) and the [**AWS DeepRacer League**](https://aws.amazon.com/deepracer/league/): reinforcement learning
* [**AWS DeepComposer**](https://aws.amazon.com/deepcomposer/): Generative AI.
* [**AWS ML Training and Certification**](https://aws.amazon.com/training/learning-paths/machine-learning/): Curriculum used to train Amazon developers
* Partnerships with Online Learning Providers: Including this course and the Udacity [**AWS DeepRacer**](https://www.udacity.com/course/aws-deepracer--ud014) course!

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)**

**[AWS Educational Devices](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/f8cb43fe-de2a-4948-837c-0a2001566ca4)**

**Machine Learning Techniques**

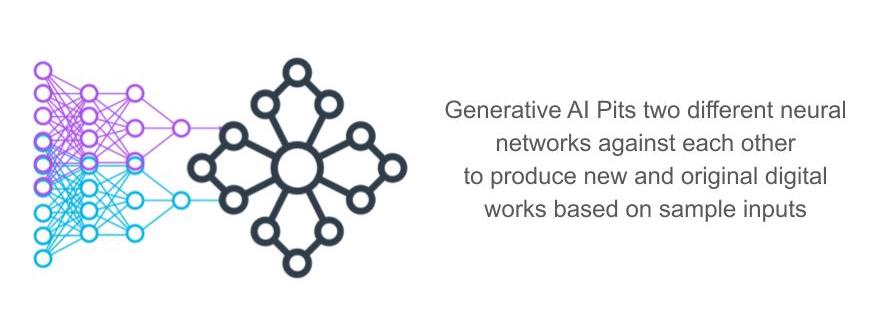
1. **Supervised Learning:** Models are presented wit input data and the desired results. The model will then attempt to learn rules that map the input data to the desired results.
2. **Unsupervised Learning:** Models are presented with datasets that have no labels or predefined patterns, and the model will attempt to infer the underlying structures from the dataset. Generative AI is a type of unsupervised learning.
3. **Reinforcement learning:** The model or agent will interact with a dynamic world to achieve a certain goal. The dynamic world will reward or punish the agent based on its actions. Overtime, the agent will learn to navigate the dynamic world and accomplish its goal(s) based on the rewards and punishments that it has received.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/9959d561-b593-4b8b-8095-153d35641365)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/9959d561-b593-4b8b-8095-153d35641365)**

**[Types of Machine Learning](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/9959d561-b593-4b8b-8095-153d35641365)**

**Generative AI**

Generative AI is one of the biggest recent advancements in artificial intelligence technology because of its ability to create something new. It opens the door to an entire world of possibilities for human and computer creativity, with practical applications emerging across industries, from turning sketches into images for accelerated product development, to improving computer-aided design of complex objects. It takes two neural networks against each other to produce new and original digital works based on sample inputs.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/9959d561-b593-4b8b-8095-153d35641365)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/9959d561-b593-4b8b-8095-153d35641365)**

**[Generative AI Opens the Door to Possibilities](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/9959d561-b593-4b8b-8095-153d35641365)**

**Additional Resources**

* [**AWS DeepLens**](https://aws.amazon.com/deeplens/)

## Generative AI

Generative AI has been described as one of the most promising advances in AI in the past decade by the MIT Technology Review.

Generative AI opens the door to an entire world of creative possibilities with practical applications emerging across industries, from turning sketches into images for accelerated product development, to improving computer-aided design of complex objects.

For example, Glidewell Dental is training a generative adversarial network adept at constructing detailed 3D models from images. One network generates images and the second inspects those images. This results in an image that has even more anatomical detail than the original teeth they are replacing.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)**

**[Glidewell Dental is training GPU powered GANs to create dental crown models](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)**

Generative AI enables computers to learn the underlying pattern associated with a provided input (image, music, or text), and then they can use that input to generate new content. Examples of Generative AI techniques include Generative Adversarial Networks (GANs), Variational Autoencoders, and Transformers.

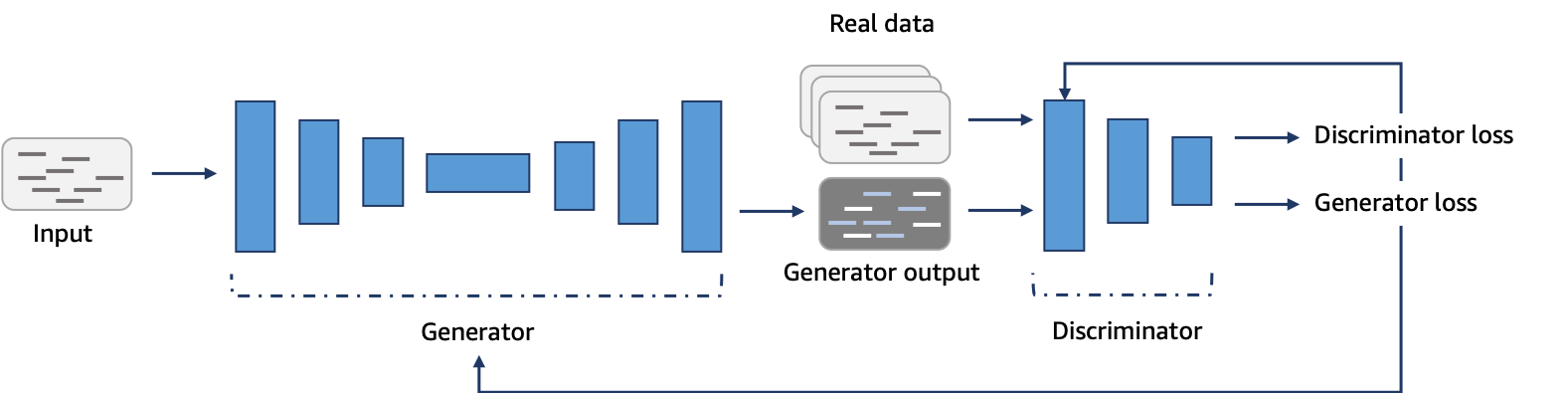
## What are GANs?

GANs, a generative AI technique, pit 2 networks against each other to generate new content. The algorithm consists of two competing networks: a **generator** and a **discriminator**.

A **generator** is a convolutional neural network (CNN) that learns to create new data resembling the source data it was trained on.

The **discriminator** is another convolutional neural network (CNN) that is trained to differentiate between real and synthetic data.

The generator and the discriminator are trained in alternating cycles such that 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.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)**

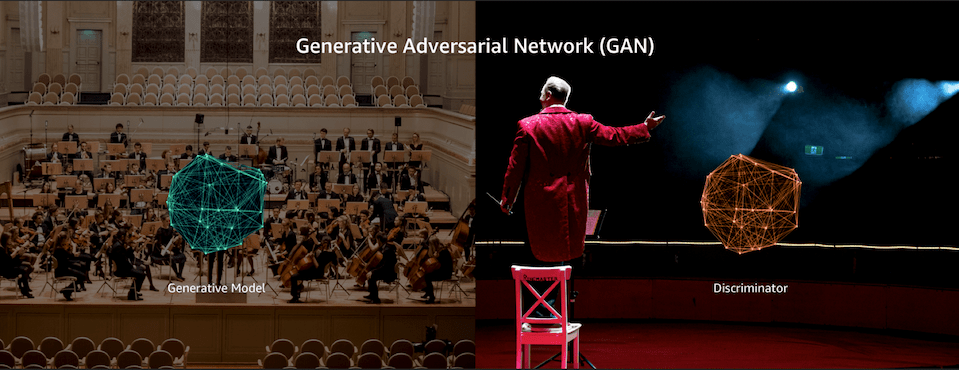
**[A schema representing a AWS DeepComposer GAN](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)**

### Like the collaboration between an orchestra and its conductor

The best way we’ve found to explain this is to use the metaphor of an orchestra and conductor. An orchestra doesn’t create amazing music the first time they get together. They have a conductor who both judges their output, and coaches them to improve. So an orchestra, trains, practices, and tries to generate polished music, and then the conductor works with them, as both judge and coach.

The conductor is both judging the quality of the output (were the right notes played with the right tempo) and at the same time providing feedback and coaching to the orchestra (“strings, more volume! Horns, softer in this part! Everyone, with feeling!”). Specifically to achieve a style that the conductor knows about. So, the more they work together the better the orchestra can perform.

The Generative AI that AWS DeepComposer teaches developers about uses a similar concept. We have two machine learning models that work together in order to learn how to generate musical compositions in distinctive styles.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)**

**[As a conductor provides feedback to make an orchestra sound better, a GAN's discriminator gives the generator feedback on how to make its data more realistic](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/64f219ba-1f44-4ccc-aff4-276abae890ba)**

In this demonstration we’re going to synchronize what you’ve learned about software development practices and machine learning, using AWS DeepComposer to explore those best practices against a real life use case.

## Coding Along With The Instructor (Optional)

To create the custom GAN, you will need to use an instance type that is not covered in the Amazon SageMaker free tier. **If you want to code along with the demo and build a custom GAN, you may incur a cost**.

You can learn more about SageMaker costs in the [**Amazon SageMaker pricing documentation**](https://aws.amazon.com/sagemaker/pricing/)

## Getting Started

**Setting Up the DeepComposer Notebook**

1. To get to the main Amazon SageMaker service screen, navigate to the AWS SageMaker console. You can also get there from within the AWS Management Console by searching for *Amazon SageMaker*.
2. Once inside the SageMaker console, look to the left hand menu and select **Notebook Instances**.
3. Next, click on **Create notebook instance**.
4. In the Notebook instance setting section, give the notebook a name, for example, DeepComposerUdacity.
5. Based on the kind of CPU, GPU and memory you need the next step is to select an instance type. For our purposes, we’ll configure a ml.c5.4xlarge
6. Leave the **Elastic Inference** defaulted to none.
7. In the Permissions and encryption section, create a new IAM role using all of the defaults.
8. When you see that the role was created successfully, navigate down a little ways to the Git repositories section
9. Select **Clone a public Git repository to this notebook instance only**
10. Copy and paste the public URL into the Git repository URL section: https://github.com/aws-samples/aws-deepcomposer-samples
11. Select **Create notebook instance**
12. Give SageMaker a few minutes to provision the instance and clone the Git repository

**Exploring the Notebook**

Now that it’s configured and ready to use, let’s take a moment to investigate what’s inside the notebook.

### Open the Notebook

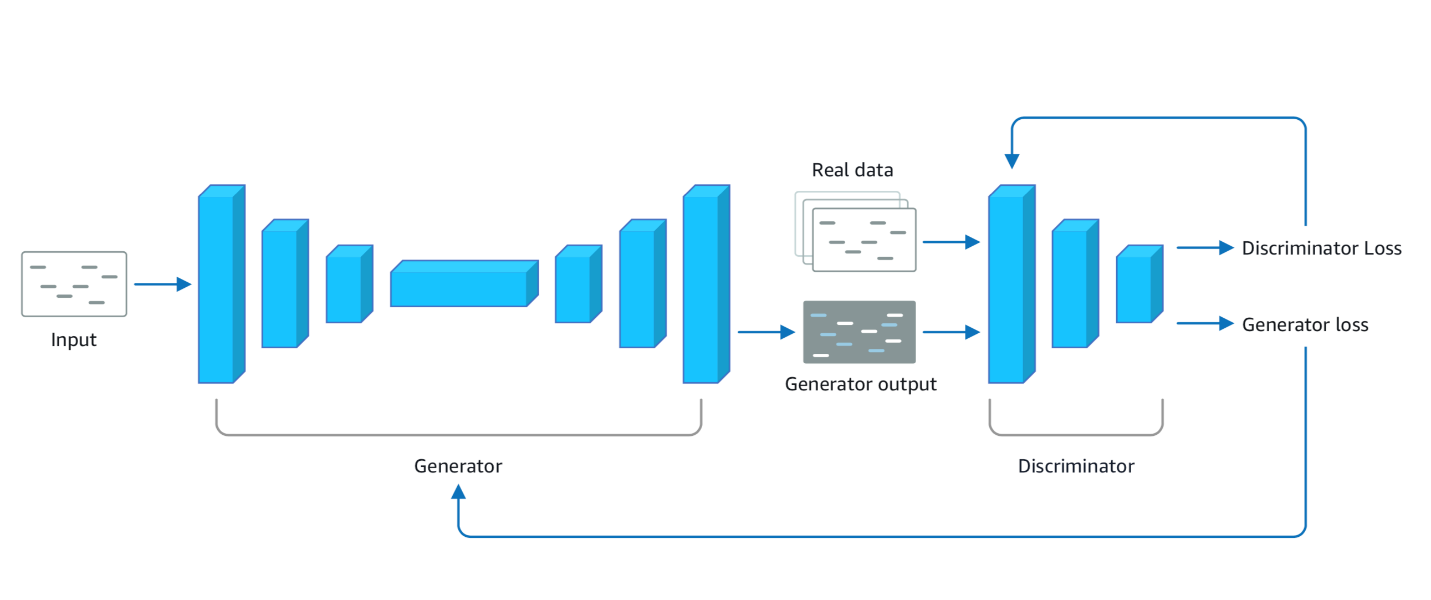
1. Click **Open Jupyter**.
2. When the notebook opens, click on **Lab 2**.
3. When the lab opens click on **GAN.ipynb**.

#### Review: Generative Adversarial Networks (GANs).

GANs consist of two networks constantly competing with each other:

* **Generator network** that tries to generate data based on the data it was trained on.
* **Discriminator network** that is trained to differentiate between real data and data which is created by the generator.

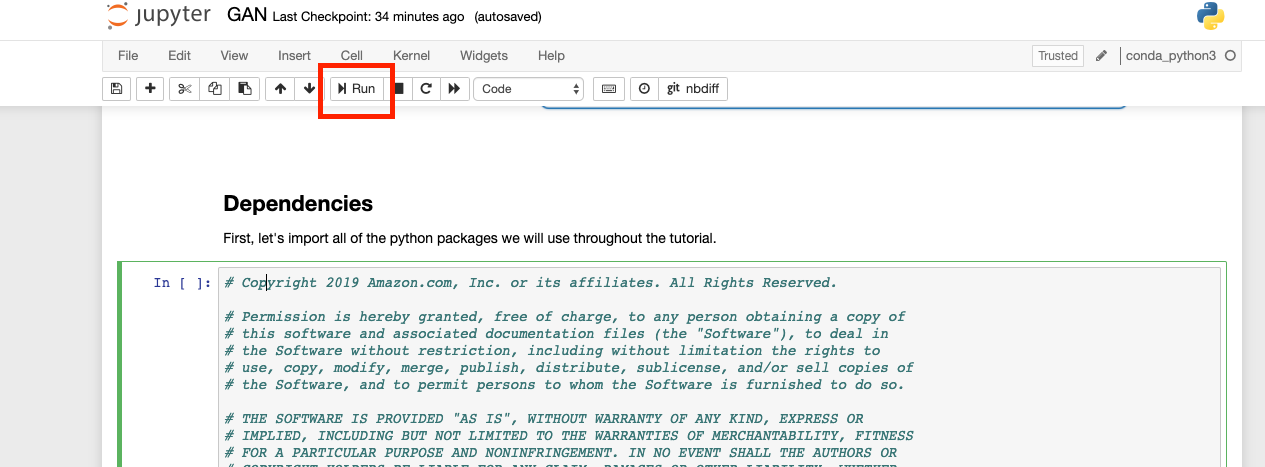
**Note**: The demo often refers to the **discriminator** as the **critic**. The two terms can be used interchangeably.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)**

### Set Up the Project

1. Run the first **Dependencies** cell to install the required packages
2. Run the second **Dependencies** cell to import the dependencies
3. Run the **Configuration** cell to define the configuration variables

**Note**: While executing the cell that installs dependency packages, you may see warning messages indicating that later versions of conda are available for certain packages. It is completely OK to ignore this message. It should not affect the execution of this notebook.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)**

**[Click Run or Shift-Enter in the cell](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)**

#### Good Coding Practices

* Do not hard-code configuration variables
* Move configuration variables to a separate config file
* Use code comments to allow for easy code collaboration

## Data Preparation

The next section of the notebook is where we’ll prepare the data so it can train the generator network.

**Why Do We Need to Prepare Data?**

Data often comes from many places (like a website, IoT sensors, a hard drive, or physical paper) and it’s usually not clean or in the same format. Before you can better understand your data, you need to make sure it’s in the right format to be analyzed. Thankfully, there are library packages that can help! One such library is called [**NumPy**](https://numpy.org/), which was imported into our notebook.

**Piano Roll Format**

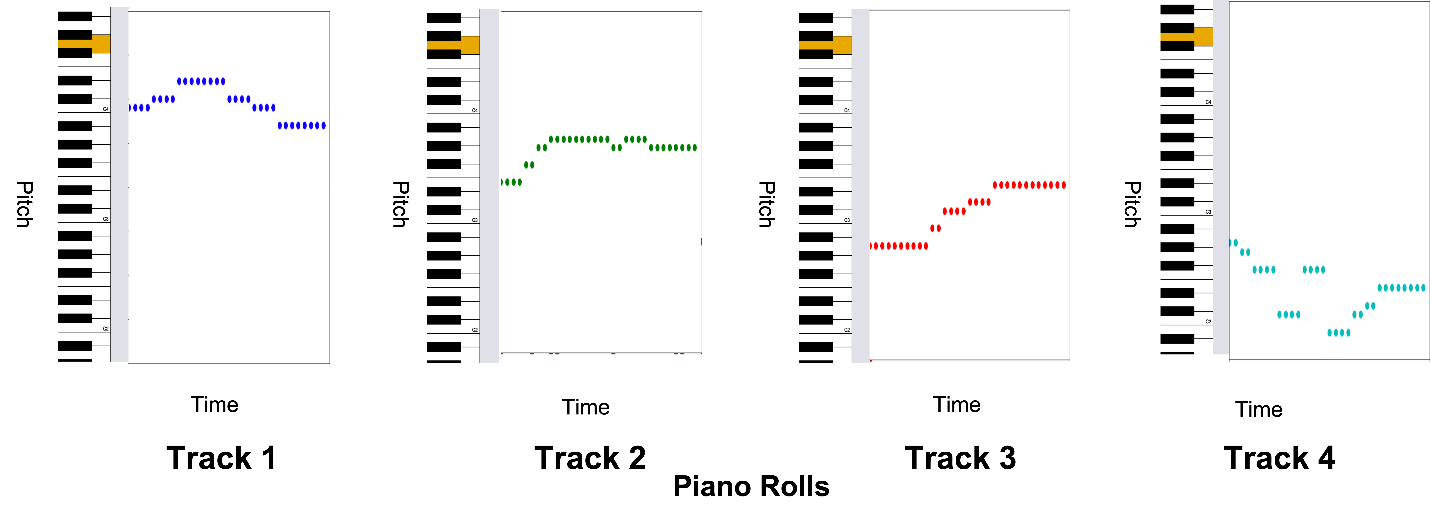
The data we are preparing today is music and it comes formatted in what’s called a “piano roll”. Think of a piano roll as a 2D image where the X-axis represents time and the Y-axis represents the pitch value. Using music as images allows us to leverage existing techniques within the computer vision domain.

Our data is stored as a NumPy Array, or grid of values. Our dataset comprises 229 samples of 4 tracks (all tracks are piano). Each sample is a 32 time-step snippet of a song, so our dataset has a shape of:

(num\_samples, time\_steps, pitch\_range, tracks)

or

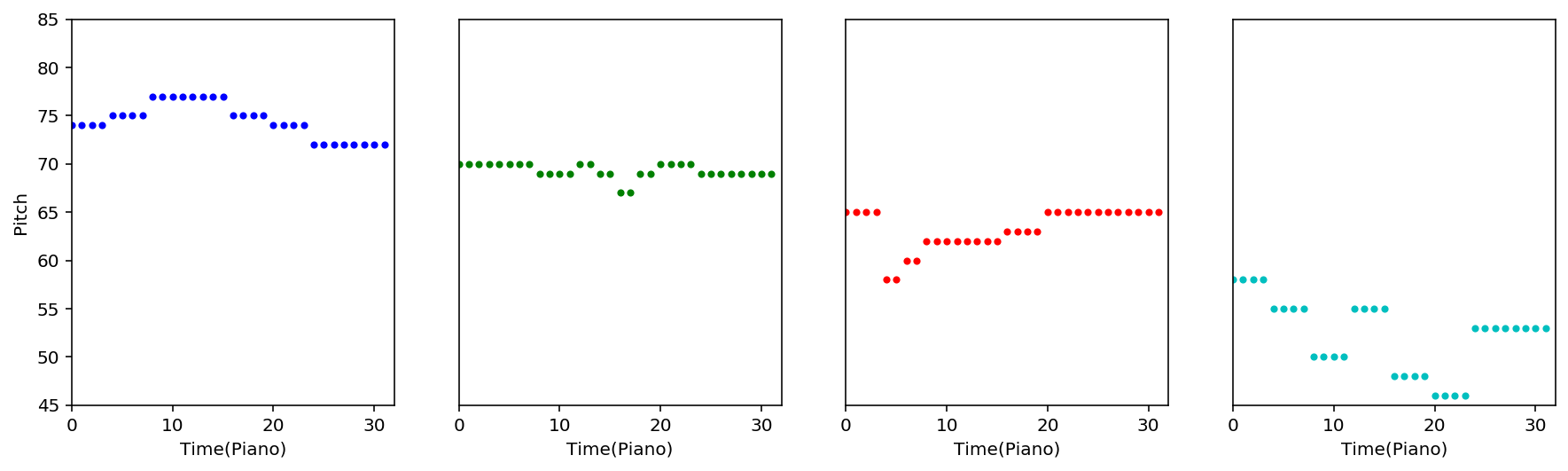
(229, 32, 128, 4)

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)**

**[Each Piano Roll Represents A Separate Piano Track in the Song](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)**

**Load and View the Dataset**

1. Run the next cell to play a song from the dataset.
2. Run the next cell to load the dataset as a nympy array and output the shape of the data to confirm that it matches the (229, 32, 128, 4) shape we are expecting
3. Run the next cell to see a graphical representation of the data.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)**

**[Graphical Representation of Model Data](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/81af7166-2a3b-4977-a17f-27bd5e23466e)**

**Create a Tensorflow Dataset**

Much like there are different libraries to help with cleaning and formatting data, there are also different frameworks. Some frameworks are better suited for particular kinds of machine learning workloads and for this deep learning use case, we’re going to use a **[Tensorflow](https://www.tensorflow.org/" \t "_blank)** framework with a **[Keras](https://keras.io/" \t "_blank)** library.

We'll use the dataset object to feed batches of data into our model.

1. Run the first Load Data cell to set parameters.
2. Run the second Load Data cell to prepare the data.

### Model Architecture

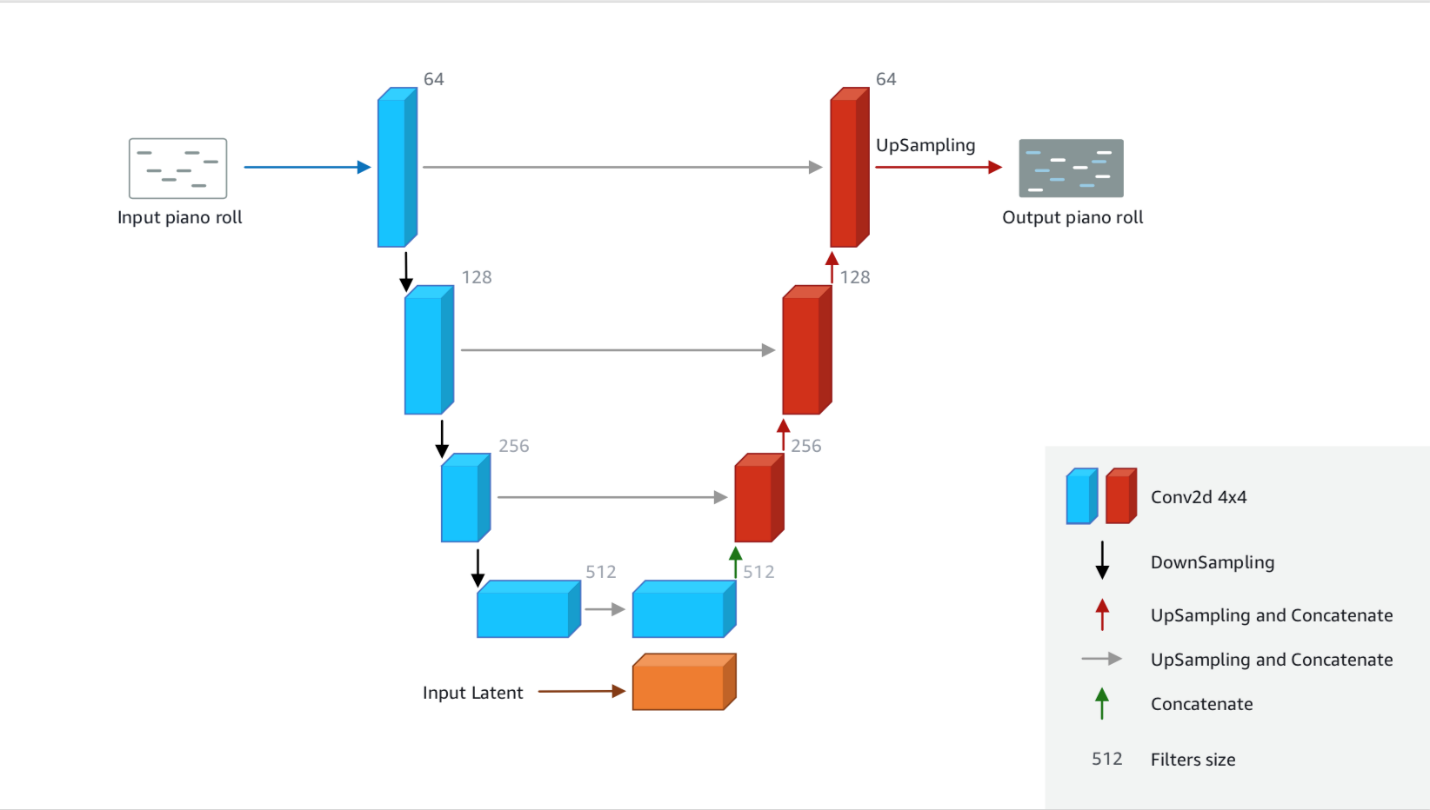
Before we can train our model, let’s take a closer look at model architecture including how GAN networks interact with the batches of data we feed it, and how they communicate with each other.

**How the Model Works**

The model consists of two networks, a generator and a critic. These two networks work in a tight loop:

* 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 evaluates the generated music tracks and predicts how far they deviate from the real data in the training dataset.
* The feedback from the discriminator is used by the generator to help it produce more realistic music the next time.
* As the generator gets better at creating better music and fooling the discriminator, the discriminator needs to be retrained by using music tracks just generated by the generator as fake inputs and an equivalent number of songs from the original dataset as the real input.
* We alternate between training these two networks until the model converges and produces realistic music.

The discriminator is a **binary classifier** which means that it classifies inputs into two groups, e.g. “real” or “fake” data.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

**Defining and Building Our Model**

1. Run the cell that defines the generator
2. Run the cell that builds the generator
3. Run the cell that defines the discriminator
4. Run the cell that builds the discriminator

**Model Training and Loss Functions**

As the model tries to identify data as “real” or “fake”, it’s going to make errors. Any prediction different than the ground truth is referred to as an error.

The measure of the error in the prediction, given a set of weights, is called a **loss function**. Weights represent how important an associated feature is to determining the accuracy of a prediction.

Loss functions are an important element of training a machine learning model because they are used to update the weights after every iteration of your model. Updating weights after iterations optimizes the model making the errors smaller and smaller.

**Setting Up and Running the Model Training**

1. Run the cell that defines the loss functions
2. Run the cell to set up the optimizer
3. Run the cell to define the generator step function
4. Run the cell to define the discriminator step function
5. Run the cell to load the melody samples
6. Run the cell to set the parameters for the training
7. Run the cell to train the model!!!!

Training and tuning models can take a very long time – weeks or even months sometimes. Our model will take around an hour to train.

**Model Evaluation**

Now that the model has finished training it’s time to evaluate its results.

There are several evaluation metrics you can calculate for classification problems and typically these are decided in the beginning phases as you organize your workflow.

In our example we:

* Checked to see if the losses for the networks are converging
* Looked at commonly used musical metrics of the generated sample and compared them to the training dataset.

**Evaluating Our Training Results**

1. Run the cell to restore the saved checkpoint. If you don't want to wait to complete the training you can use data from a pre-trained model by setting TRAIN = False in the cell.
2. Run the cell to plot the losses.
3. Run the cell to plot the metrics.

**Results and Inference**

Finally, we are ready to hear what the model produced and visualize the piano roll output!

There are several evaluation metrics you can calculate for classification problems and typically these are decided in the beginning phases as you organize your workflow.

In our example we:

* Checked to see if the losses for the networks are converging
* Looked at commonly used musical metrics of the generated sample and compared them to the training dataset.

**Evaluating Our Training Results**

1. Run the cell to restore the saved checkpoint. If you don't want to wait to complete the training you can use data from a pre-trained model by setting TRAIN = False in the cell.
2. Run the cell to plot the losses.
3. Run the cell to plot the metrics.

**Results and Inference**

Finally, we are ready to hear what the model produced and visualize the piano roll output!

Once the model is trained and producing acceptable quality, it’s time to see how it does on data it hasn’t seen. We can test the model on these unknown inputs, using the results as a proxy for performance on future data.

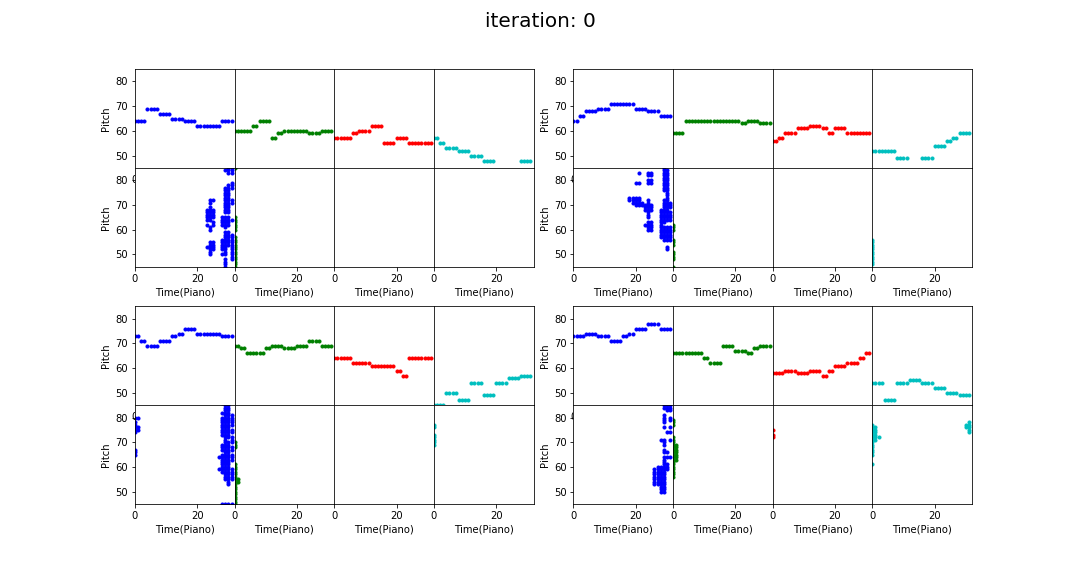
**Evaluate the Generated Music**

1. In the first cell, enter 0 as the iteration number:
2. iteration = 0

run the cell and play the music snippet.  
Or listen to this example snippet from iteration 0:

1. In the second cell, enter 0 as the iteration number:
2. iteration = 0

run the cell and display the piano roll.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

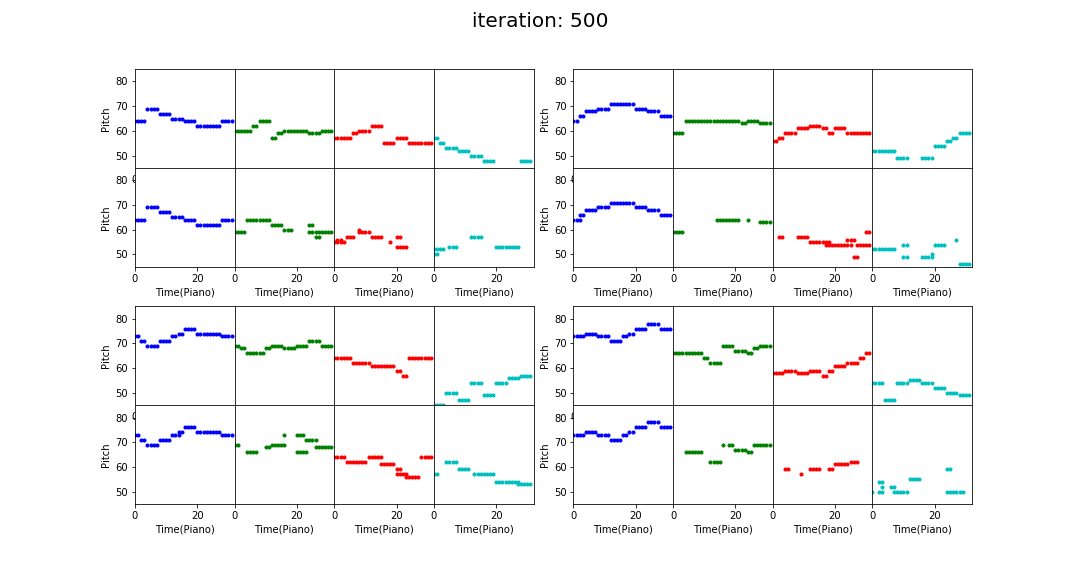
**[Example Piano Roll at Iteration 0](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

1. In the first cell, enter 500 as the iteration number:
2. iteration = 500

run the cell and play the music snippet.  
Or listen to the example snippet at iteration 500.

1. In the second cell, enter 500 as the iteration number:
2. iteration = 500

run the cell and display the piano roll.

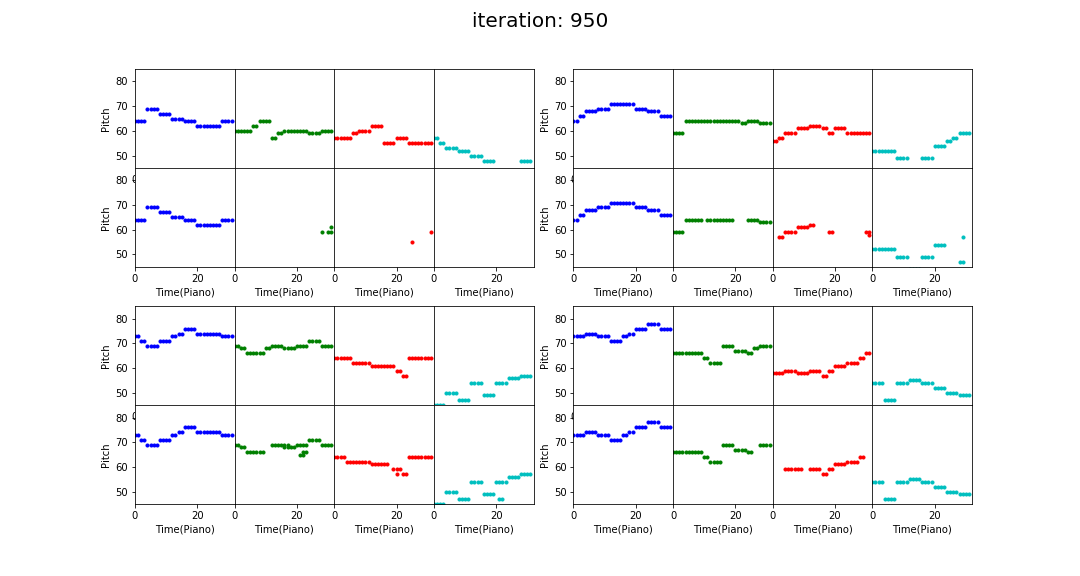
**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

**[Example Piano Roll at Iteration 500](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

Play around with the iteration number and see how the output changes over time!

Here is an example snippet at iteration 950

And here is the piano roll:

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

**[Example Piano Roll at Iteration 950](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

Do you see or hear a quality difference between iteration 500 and iteration 950?

**Watch the Evolution of the Model!**

1. Run the next cell to create a video to see how the generated piano rolls change over time.  
   Or watch the example video here:

**Inference**

Now that the GAN has been trained we can run it on a custom input to generate music.

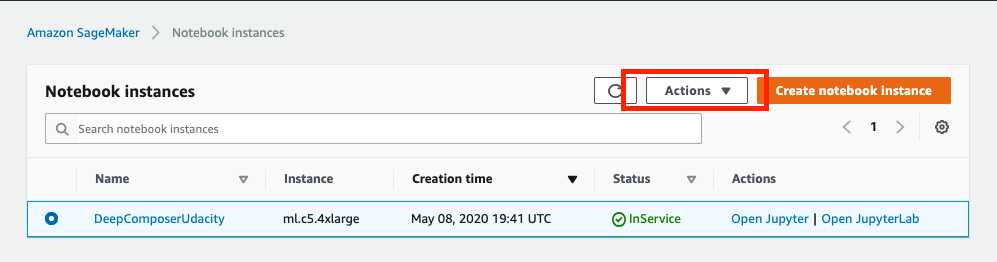
1. Run the cell to generate a new song based on "Twinkle Twinkle Little Star".  
   Or listen to the example of the generated music here:
2. Run the next cell and play the generated music.  
   Or listen to the example of the generated music here:

**Stop and Delete the Jupyter Notebook When You Are Finished!**

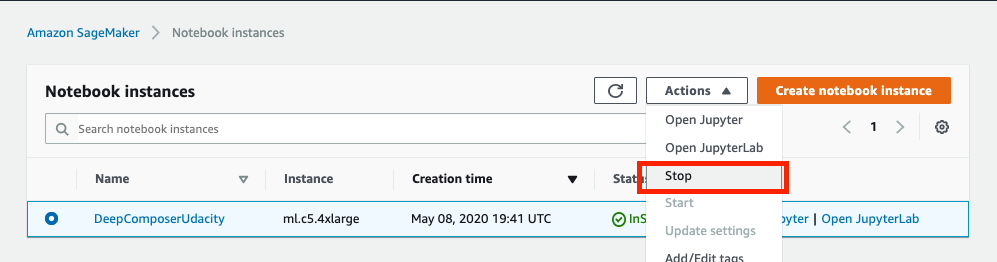
This project is not covered by the AWS Free Tier so ***your project will continue to accrue costs as long as it is running.***

Follow these steps to stop and delete the notebook.

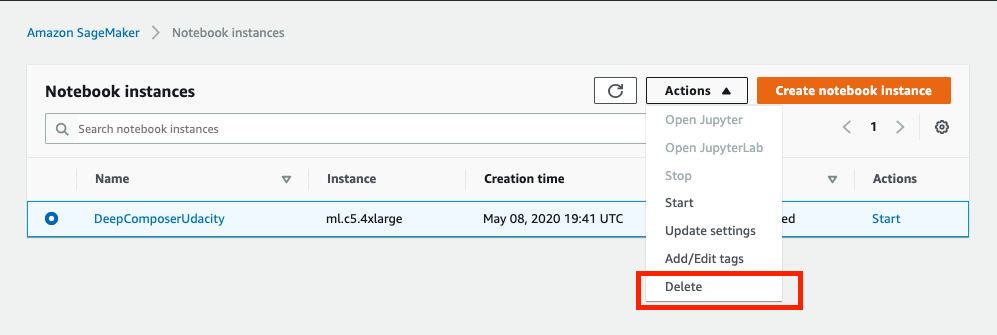
1. Go back to the [**Amazon SageMaker console**](https://console.aws.amazon.com/sagemaker/home?region=us-east-1#/notebook-instances).
2. Select the notebook and click **Actions**.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

1. Select **Stop** and wait for the instance to stop.

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

1. Select **Delete**

**[[](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)](https://classroom.udacity.com/courses/ud090/lessons/099925a2-4f01-41c7-a4d4-8ce246f7b801/concepts/45be01fd-ee02-4b5c-8826-b4e2e4082255)**

**Recap**

In this demo we learned how to setup a Jupyter notebook in Amazon SageMaker, about machine learning code in the real world, and what data preparation, model training, and model evaluation can look in a notebook instance. While this was a fun use case for us to explore, the concepts and techniques can be applied to other machine learning projects like an object detector or a sentiment analysis on text.