AWS Machine Learning Blog

How to run an AI powered musical challenge: “AWS DeepComposer Got Talent”

by Maryam Rezapoor, David Bounds, David Coon, Jana Gnanachandran, Josh Schairbaum, Prem Ranga, Durga Sury, and Henry Wang | on 10 JUN 2021 | in Artificial Intelligence, AWS DeepComposer | Permalink | Comments | Share

To help you fast track your company’s adoption of machine learning (ML), AWS offers educational solutions for developers to get hands-on experience. We like to think of these programs as a fun way for developers to build their skills using ML technologies in real world scenarios. In this post, we walk you through how to prepare for and run an AI music competition using AWS DeepComposer. Through AWS DeepComposer, you can experience Generative AI in action and learn how to harness the latest in ML and AI. We provide an end-to-end kit that contains tools, techniques, processes, and best practices to run the event.

Since AWS re:Invent 2017, our team has launched three thought-leading products aimed to get developers hands-on with the latest machine learning technologies in a fun way:

- AWS DeepLens – The world’s first deep learning-enabled camera.
- AWS DeepRacer – A fully autonomous 1/18th scale race car driven by reinforcement learning.
- AWS DeepComposer – A product designed to teach Generative AI, which can take in a user provided melody and transform it into a completely original song in seconds.

Designed specifically to educate developers on generative AI, AWS DeepComposer includes tutorials, sample code, and training data in an immersive platform that can be used to build ML models with music as the medium of instruction. Developers, regardless of their background in ML or music, can get started with applying AI techniques including Generative Adversarial Networks (GANs), Autoregressive Convolutional Neural Networks (AR-CNN) and Transformers to generate new musical notes and accompaniments. With AWS DeepComposer, even non-musicians can use one of the pre-built models we provided to embellish a simple eight-bar tune to create unique compositions while more advanced users train and optimize ML models to create original music.

We have been running AI music events using AWS DeepComposer with customers, our partners, and internally within AWS, and these have received a lot of interest and engagement among both technical and non-technical users. Participants enjoy the opportunity to learn the latest in ML, compose music without the need for any knowledge of music theory, and win prizes in a talent show styled competition. And so, as a logical next step, we have packaged the resources required for you to run your own AI music challenge for employees, partners, and customers at corporate or meet-up style events. This not only scales out the educational experience but also helps accelerate ML adoption in your organization.

We walk you through the following topics in this post:

- Competition Overview
- Pre-requisites
- Overview of AWS DeepComposer
- Competition: Getting Started
- Competition: Composing and submitting tracks
- Competition: Scoring and Winning

To run this event, we suggest you as the organizer start planning and preparing 3 weeks in advance to ensure you have everything in place. For participation, all that is needed is access to AWS Management Console and specifically AWS DeepComposer.

Competition Overview

What happens during the competition?

The purpose of the competition is to make learning fun and engaging by presenting a real use case of generative AI. You can organize the event multiple ways, for example:

- Relaxed schedule with asynchronous submissions – extend the competition out for a week, with workshop on day 1, and participants having the next 3 days to submit their compositions, with the last day for scoring and announcing the winners
- Two day event with synchronous submissions – run the workshop and challenge on day 1 (total duration of 5 hours) with scoring and awards on day 2 (approximately 2–3 hours)
- Two day event with asynchronous submissions – run workshop on day 1 (3 hours), with offline submission of compositions (we typically set an ETA of end of the day), with scoring and awards on day 2 (2–3 hours)
- One day event with synchronous submissions – run the workshop and competition during the first half of the day, use an AI judge to select the winners, and present the awards at the end of the day

You can choose a format that fits your participant schedule. The more time you give participants, the more time they have to experiment with different options to compose some really cool music. In this section, we provide a sample agenda encompassing the options we discussed; feel free to modify them based on how you organize your event. We go through some of these items in more detail in the rest of this post.

Note on Judging process: As we will see from the following agenda and the rest of this post, we have two options for judging the musical tracks to decide the winners. We will show you how to organize a team of human judges, and we also provide you a ML model (i.e. an AI Judge) that can identify similarity with Bach music. In our events we select winners using the human judges for the main event and the AI judge as a special category.

Considering the following agenda example runs for several hours, please make sure you add a 15-minute break every 2 hours or so. We have included buffer in the session durations to accommodate the breaks throughout the day.

Sample agenda: Workshop and competition

Timing	**Topic**	**Type**	**Description**
9 AM – to 9:15 AM	Agenda Walkthrough	Presentation	Walk participants through the agenda and logistics for the day.
9:15 AM – 9:45 AM	Introduction to ML at AWS	Presentation	Introduce the three levels of the ML stack at AWS. This helps to set context for participants for the event. For more information, see Machine Learning on AWS.

9:45 AM – 10 AM	AWS DeepComposer Kahoot	Game	The competition is all about making learning fun. So we have a quick icebreaker to get everyone’s game face on. Kahoot is an interactive and engaging way to take a fun quiz on AWS DeepComposer. You can collect questions from the learning capsules in the Music Studio or you can make this an organization-specific exercise as well. We typically present the winner of the Kahoot game Amazon swag, such as stickers. We announce this upfront to encourage participation.
10 AM – 10:15 AM	AWS DeepComposer Music Studio Walkthrough	Demo	For many of the participants, this is the first time they have been exposed to the AWS DeepComposer Music Studio, so we spend 10–15 minutes walking through the steps necessary for creating their own compositions. Facilitators demonstrate how to navigate the Music Studio to generate compositions. For a video demonstrating the Music Studio walkthrough, see “AWS DeepComposer – Music Studio Walkthrough”.
10:15 AM – 10:45 AM	ML Algorithms in AWS DeepComposer	Presentation	We introduce participants to the ML algorithms that power AWS DeepComposer, such as GAN, AR-CNN, and Transformers.
10:45 AM – 11:30 AM	Composing Music in AWS DeepComposer	Lab	In this interactive lab, participants try out AWS DeepComposer features, play with the ML algorithms, and generate AI music, all under prescriptive guidance of the organizers. This session serves as a precursor to the competition.
11:30 AM – 12:30 PM	AWS DeepComposer Got Talent Kickoff	Presentation	Organizers walk participants through competition guidelines, rules, best practices, SoundCloud setup, scoring process, leaderboards, and introduce the judges.
12:30 – 1 PM	Lunch		
1 PM – 3 PM	AWS DeepComposer Got Talent Competition	Activity	Game on! The latest submission time for compositions is 3 PM.

3 PM – 4 PM	Create Playlists for Judges	Activity	Organizers create separate SoundCloud playlists for each of the judges. Depending on the number of participants and size of the event, you can have each of the judges listen to all of the tracks and take a weighted average score.
End of DAY 1			

Sample agenda: Deciding the winners and awards ceremony

Timing	**Task**	**Type**	**Description**
9 AM – 10:30 AM	Judge Music Compositions: Qualifiers	Activity	Judges listen to all the submissions and score them. A leaderboard keeps track of the scores and the top composers.
10:30 AM – 12	Judge Music Compositions: Finals	Activity	Judges listen to the top 15 (or more) tracks from the leaderboard and score these submissions again. From the revised leaderboard, the top three submissions are selected as winning submissions.
11:30 AM – 12	Run AI Judge	Activity	The ML model trained to listen to Bach music is run against the compositions to determine the top three submissions that are closest in similarity.
12 – 1 PM	Lunch		
1 PM – 2 PM	Awards Ceremony	Presentation	The top three composers for the main challenge and the top three composers selected by the AI Judge receive their awards.
End of DAY 2			

Rules and guidelines

The AWS DeepComposer Got Talent competition is intended to provide a fun backdrop for spending a day learning more about generative AI using AWS DeepComposer through the AWS DeepComposer console. Prizes may be at stake, so in the interest of fairness, we have created a short list of suggested rules for contestants to follow. You can come up with your own list or customize this list as needed.

1. Contestants must use an AWS DeepComposer Got Talent Event Team-provided input track, or they may record an input track of their own during the competition to use as the input track for their composition.
2. Participants must use the AWS DeepComposer Music Studio solely for the creation of their composition, except for recording their own input track on a hardware keyboard, if desired.
3. Contestants may only submit one composition for judging. Additional submissions to the AI judge are allowed.
4. For a track to be considered for judging, a contestant must submit the composition following the instructions set for the event by the AWS DeepComposer Got Talent Event Team.
5. Each judge scores each contestant’s submission. For events with more submissions than could reasonably be adjudicated by a team of judges, split the competition into two rounds by evenly and randomly assigning contestants to judges. The three highest-scoring submissions evaluated by each judge are moved to the final round, where each judge adjudicates the remaining entries from the other judges, but not their own.

Evolve the competition rules and format to scale with the number of judges and contestants expected. Each submission takes several minutes to judge, and you can scale the competition by increasing the number of judges or by allocating contestants to judges to limit the number of submissions required for each judge to listen to in preliminary rounds.

Pre-requisites

Event participants should either have or be provided the following before the start of the event:

- Desktop or laptop computer with access to the internet and the AWS Management Console in the Region selected for use. As of this writing, AWS DeepComposer is only supported in the `us-east-1` Region. Make sure to check if it’s supported in the Region you want to use. Mobile devices (smartphones or tablets) are not recommended for this event. The computer should have the following:
  - A pointer and keyboard.
  - A Chrome browser (this browser implements the full AWS DeepComposer console functionality).
  - An AWS DeepComposer physical device isn’t necessary, but can be used if desired. Custom audio elements can be played via the console or on the keyboard. You can also use the sample inputs provided in the AWS DeepComposer console.
- An account created on soundcloud.com, used to load and share the composed sound tracks.
- Minimal knowledge of computer technologies and website use.
- A shared video conferencing service that can support your total expected participants, such as Amazon Chime.
- A shared email distribution list or messaging platform where participants can send questions, requests, or feedback.

Overview of AWS DeepComposer

Introducing the AWS DeepComposer Music Studio

Many of your participants may be unfamiliar with the AWS DeepComposer Music Studio, so spend 10–15 minutes walking through the steps necessary to create compositions. The Music Studio isn’t difficult to operate, but providing an overview of the functionality gives participants an accelerated fast path to creating compositions of their own.

The basic steps for creating a composition in the Music Studio that requires a demonstration are:

1. Record, upload, or select an input melody.
2. Apply the AWS DeepComposer ML algorithms one or more times, in combination until satisfied with the results.

Read more about getting started with the AWS DeepComposer Music Studio.

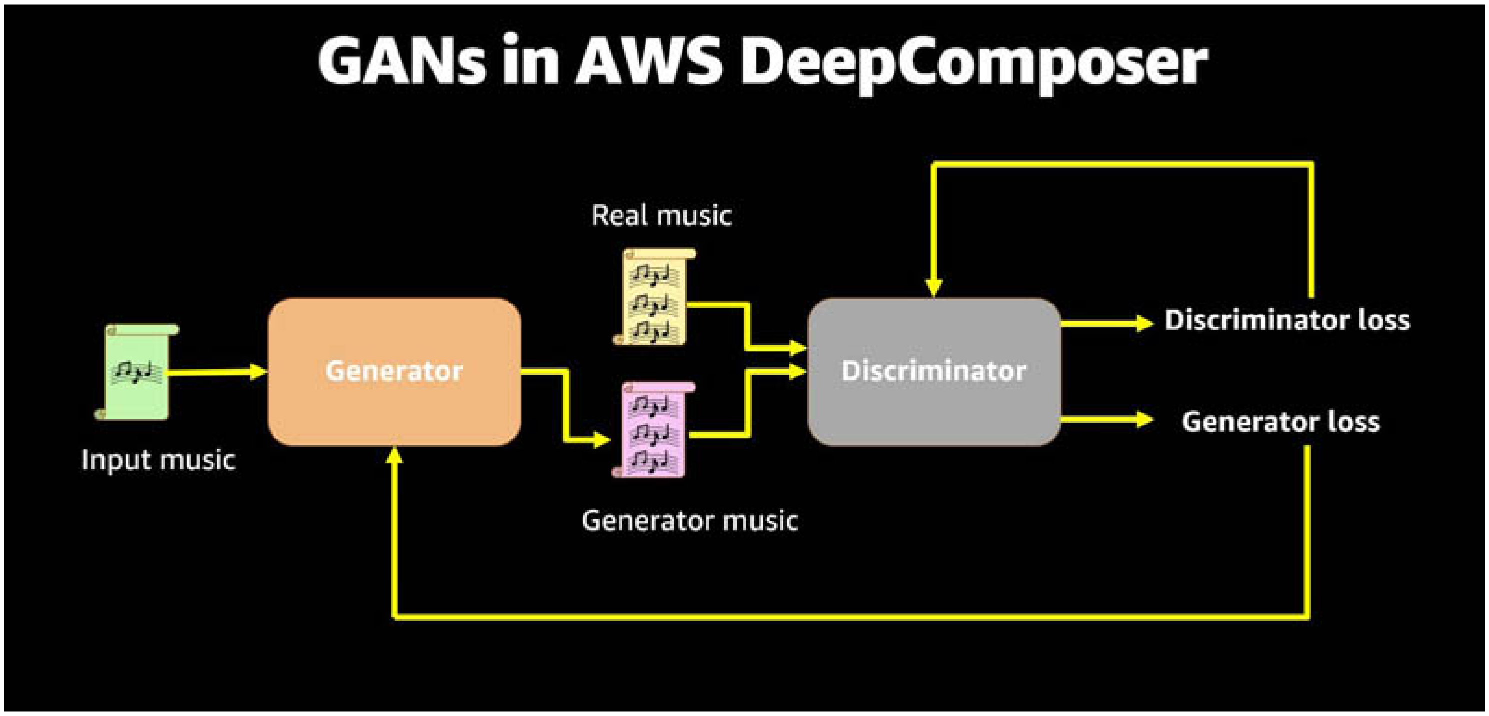
Review the AWS DeepComposer algorithms

We recommend providing a 100–200 level overview of each of the three algorithms utilized within AWS DeepComposer. With a basic understanding of the use case and output for GANs, AR-CNNs, and Transformers, your participants can more effectively utilize AWS DeepComposer to create an AWSome composition.

We recommend delivering this section as a 15-minute brief presentation that walks the participants through a conceptual and visual implementation of each algorithm. In our event, we condensed each of the included learning modules from the AWS DeepComposer console into a short slide deck that also included example musical samples as input and the generated composition from the model. In preparing this material, we recommend watching the re:Invent 2020 tech talk Getting Started with Generative AI with AWS DeepComposer

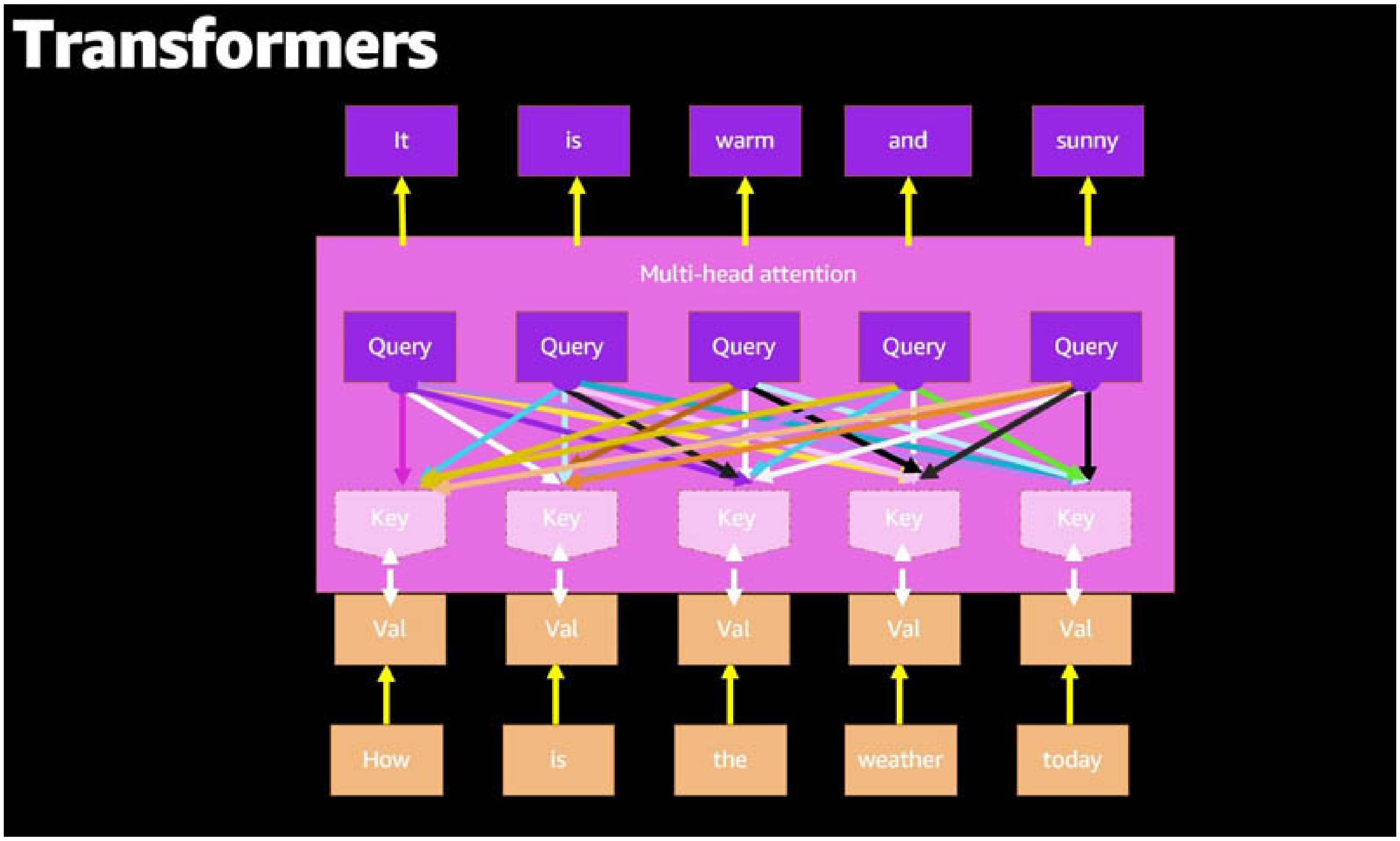
Generative Adversarial Networks (GANs) are one of the most popular generative AI techniques available today. Its core concept is to pit two neural networks against each other to generate new creative content such that one neural network generates new music, and the other discriminates and critiques the generate music to quantify how realistic the generated music is.

In AWS DeepComposer, we use GANs to generate accompanying tracks to an existing melody. The generator produces a new arrangement and sends it to the discriminator to receive feedback to create better music in the next iteration. Over time, the generator gets better at creating realistic music and eventually fools the discriminator. At this point, we retrain the discriminator using the new data from the generator so that it becomes even better at directing the generator.



Autoregressive Convolutional Neural Networks (AR-CNNs) work to sequentially and iteratively modify a melody to match a specific style more closely, or to make it more harmonious by removing wrong notes. At every step of an AR-CNN, we add or remove a single note from the composition, and each note that is added or removed depends on the notes around that step. The key idea is that each edit made depends on the other notes that have been added or removed. As part of the training process, the piano roll is modified and compared against the complete piano roll to more closely analyze the dataset it’s trying to replicate. Over iterations of training, the model learns to add or remove notes from a song so it can convert an out-of-tune song into a more correct song, or add more chords and melodies of a different style into a composition.

Transformers are state-of-the-art models for generating a complete set of sequential data such as text, music, and other time series data in a single operation. In contrast to sequential models like AR-CNNs, transformers take an entire sequential dataset such as a series of notes as a single input and operate on it all at once. Because of their ability to process an entire dataset as a single input, transformers stand out in their ability to capture near-term and long-term dependencies in a dataset. This enables the transformers within AWS DeepComposer to generate a longer version of an existing composition that is thematically and musically coherent with the original piece.



The ML algorithms used in AWS DeepComposer have already been applied to innovative and compelling use cases. Within the medical field, GANs are used to optimize the design process of new pharmaceutical compounds , and to improve cancer detection screening rates. Transformers are at the core of natural language processing, and power conversational AI, language translation, and text summarization.

Competition: Getting Started

In this section, we discuss what participants need to get started.

AWS accounts

We recommend that each participant is provided a separate AWS account in order to compose their scores. For smaller events, manual account creation via the console and distribution via existing team communication channels or email is acceptable, but for large-scale events, more automated measures are preferred. For very small-scale events with fewer than 10 participants, you can also set up individual users in a single AWS account. This allows users to log in and compose within a single account without colliding with each other’s compositions.

If your account is already set up with AWS Organizations, a service that helps you centrally manage and govern your environment as you grow and scale your AWS resources, you can create an Organizational Unit dedicated to the AWS DeepComposer event, which allows you to restrict usage of those accounts to the Regions where AWS DeepComposer is available and the associated resources permitted to be created within those accounts. If your enterprise uses AWS Control Tower, a service that provides the easiest way to set up and govern a secure, multi-account AWS environment, you can also use Account Factory to provision the accounts. For more details about large-scale account provisioning and governance, see Automate account creation, and resource provisioning using AWS Service Catalog, AWS Organizations, and AWS Lambda.

If your enterprise allocates these accounts as part of your organization, you can use AWS Single Sign-On, which centrally manages access to multiple AWS accounts and business applications, or use your federation provider to allow users to access these accounts with a specific role created for the event. If you’re not using AWS Organizations, you need a list of account IDs, user names, and passwords.

AWS account vending

For distribution, you can set up an application to allow users to self-obtain accounts, user names, and passwords for the event. AWS Amplify provides a set of tools and services that can be used together or on their own, to help front-end web and mobile developers build scalable full stack applications. Amplify also provides a simple method of creating a website that is accessible to specific users either via access controls like IP range, or by authentication like Amazon Cognito, an AWS service that provides simple and secure user signup, sign-in, and access control. When the application is ready, the user can log in to retrieve their AWS account credentials, access the console, and go to the AWS DeepComposer console to start composing.

For an example of how to use React JavaScript library, GraphQL, AWS AppSync, the Amplify Framework, and other AWS services to create a web application, see Building progressive web apps with the Amplify Framework and AWS AppSync

Competition: Composing and Submitting tracks

Composing music in AWS DeepComposer Music Studio

This section is an interactive hands-on lab where participants explore the AWS DeepComposer Music Studio and create their first compositions on their own. Prior to this lab, ensure that all participants have lab accounts or AWS accounts to access the AWS DeepComposer console. We recommend this section to have at least two moderators, in addition to the presenter, to monitor the chats and guide the participants. The lab is split into the following sections on a high level:

- **Access the Music Studio and understand the ways to provide input to AWS DeepComposer –** During the first section, participants sign in to the console and access the AWS DeepComposer service. Walk them through the menu, including the Music Studio, learning capsules, and more. The presenters should encourage the participants to record their first input using the console keyboard or digital keyboard if they own one, and try out the input settings. If providing input tracks for the competition, walk through the process to import a track as well.
- **Create an original composition through the AR-CNN model –** In this section, the participants create their first original composition using the AR-CNN model. The presenters can provide a refresher of the differences in using each of the three models. Encourage the participants to try different metrics and listen to their compositions to understand how each setting affects their resulting composition.
- **Feed the output of the AR-CNN model into a GAN or Transformers model –** After the composition is generated through AR-CNN, it can be used as the input to a GAN model to create accompaniments, or the Transformers model to extend the length of the composition. During this section, participants try out both options. Additionally, they can download a Transformers model output and import it as a melody for GANs.
- **Callouts to composition next steps and downloading files –** Each model shows a different set of next steps after composition is created. The presenters can walk through each of the options and ways to train custom models through the AWS DeepComposer console, or through Amazon SageMaker for ML practitioners. The final page provides options to download the melodies, and they can also be downloaded from the **Compositions** page on the AWS DeepComposer console.

For detailed instructions on the workshop, see AWS DeepComposer Lab. Throughout the lab, encourage participants to try out the various settings on both input and model inference pages to make their composition unique.

Best practices for composing

Share the following best practices with the participants:

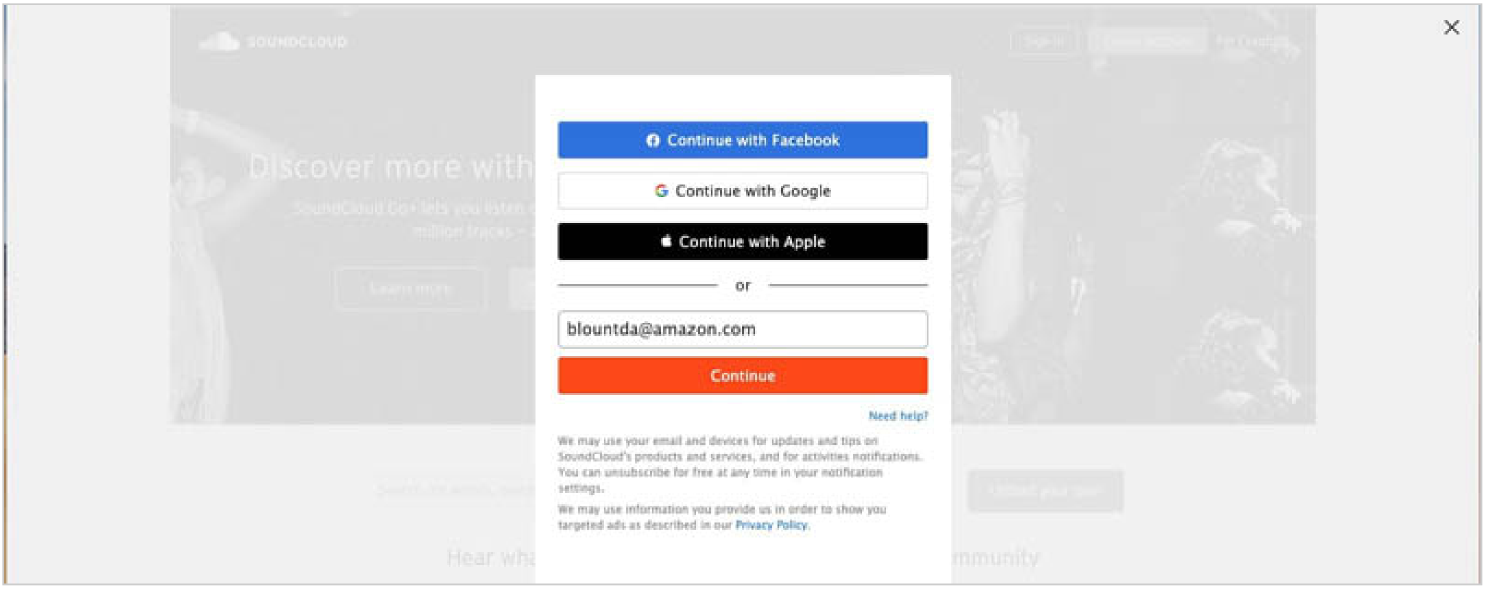
- The input melody shouldn’t exceed eight bars. Longer inputs are snipped to eight-bar length.
- Use the **Edit melody** feature to add or remove notes, change the pitch, and note length to avoid rerecording.
- Change accompaniments to understand how a composition sounds after using the GAN model. However, if they change the accompaniment instruments, this isn’t reflected when submitting the track to SoundCloud.
- Follow a proper naming convention to manage the participants’ output tracks in SoundCloud.
- Understand the order of using the algorithms: AR-CNN output can be fed into a GAN or Transformers model, but a GAN model output can’t be modified. Transformers output can be downloaded and re-uploaded to the console as a new melody.
- Explore the learning capsules to learn more about different generative AI algorithms, training parameters, and model evaluation tools.
- Custom models can’t be trained during the lab session because they take about 8 hours to train. However, encourage participants to train a custom model after the event. The AWS Free Tier includes training and storage of up to four custom models.

Submit compositions to the competition

AWS DeepComposer provides built-in integration with SoundCloud, a cloud-based music playlist app for submitting compositions for the event.

Before the competition begins, your participants should be directed to SoundCloud.com to create an account. If your participants already have SoundCloud accounts, they can use those instead.

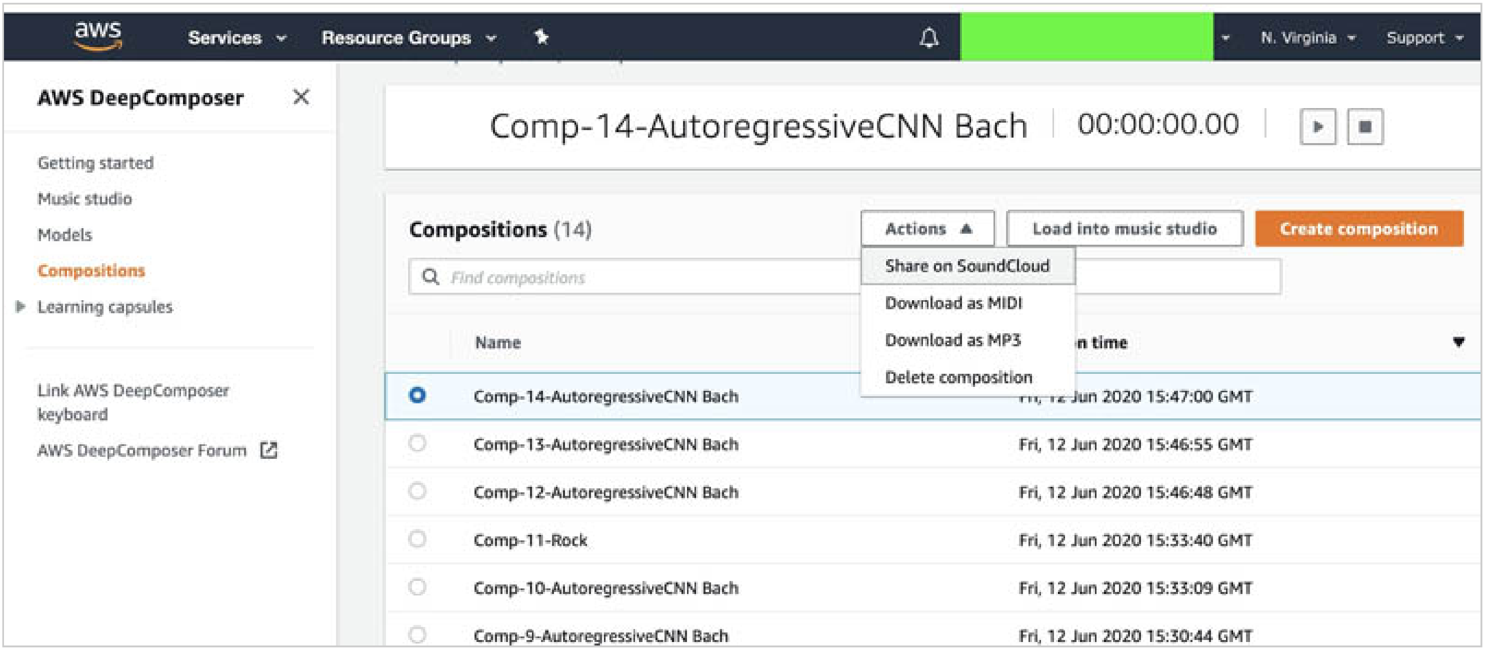
1. Sign up for a SoundCloud account by choosing **Create account**.
2. Continue by using a Facebook, Google, Apple, or email account.



We may use your email and devices for updates and tips on SoundCloud's products and services, and for activities notifications. You can unsubscribe for free at any time in your notification settings.

We may use information you provide us in order to show you targeted ads as described in our Privacy Policy.

3. On the AWS DeepComposer console, choose **Compositions** in the navigation pane.
4. Select a composition to submit.
5. On the **Actions** menu, choose **Share on SoundCloud**.



Name	Creation time
Comp-14-AutoregressiveCNN Bach	Fri, 12 Jun 2020 15:47:00 GMT
Comp-13-AutoregressiveCNN Bach	Fri, 12 Jun 2020 15:46:55 GMT
Comp-12-AutoregressiveCNN Bach	Fri, 12 Jun 2020 15:46:48 GMT
Comp-11-Rock	Fri, 12 Jun 2020 15:33:40 GMT
Comp-10-AutoregressiveCNN Bach	Fri, 12 Jun 2020 15:33:09 GMT
Comp-9-AutoregressiveCNN Bach	Fri, 12 Jun 2020 15:30:44 GMT

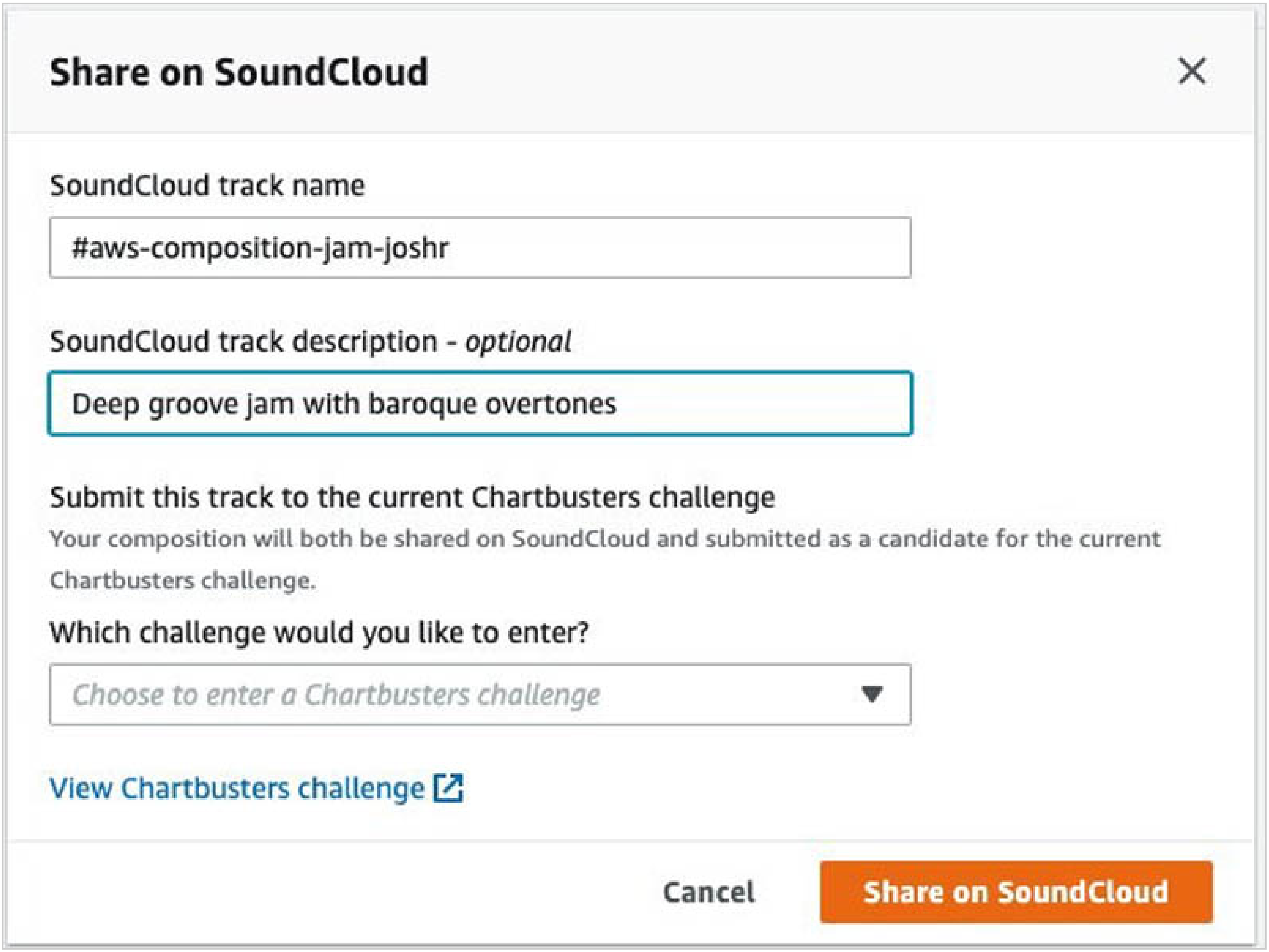
6. Choose **Connect and Continue**.

Prior to the start of your event, decide upon and communicate a naming convention for your participant submissions. This allows you to easily search for and manage the submissions for judgment. The suggested naming convention is `#eventName-userName-compositionName`. For example, for an AWS Event called *AWS Composition Jam* and a user with the internal user name of `joshr`, the submitted track name would be `#aws-composition-jam-joshr-BaroqueBeatz05`.

7. Enter a track name and description.
8. For **Which challenge would you like to enter?**, choose **None**.

Make sure participants don’t inadvertently submit their composition to the Chartbusters challenge.

9. Choose **Share on SoundCloud**.


Share on SoundCloud

SoundCloud track name

#aws-composition-jam-joshr

SoundCloud track description - *optional*

Deep groove jam with baroque overtones

Submit this track to the current Chartbusters challenge

Your composition will both be shared on SoundCloud and submitted as a candidate for the current Chartbusters challenge.

Which challenge would you like to enter?

Choose to enter a Chartbusters challenge

View Chartbusters challenge

Cancel Share on SoundCloud

You can decide if your participants are permitted to submit more than one composition for judgment, but we suggest having a defined upper limit of submissions. All users and staff can then use SoundCloud’s built-in search tool to locate submissions. For example, using the preceding naming convention, we can search based on the name prefix.

Competition: Scoring and Winning

Building the scoring tool and leaderboard

We wanted an easy, repeatable, and accessible mechanism for our judges to rate scores for compositions. To that end, we built an application within Amazon Honeycode, which gives you the power to build powerful mobile and web applications without writing any code. As our judges submit scores, the powerful spreadsheet and filtering functionality in Honeycode allows for real-time aggregation and ranking of compositions.

1. To start the build, first sign in to Amazon Honeycode and choose **Create Workbook**.
2. Import our table definitions into the workbook.

We have included sample composers and scores in your table to help validate your Honeycode tables and application. Please follow the instructions provided in README for Honeycode tables in our GitHub repository to complete the build.

As scores come in, the participants will be eager to see how their compositions measure up to their peers, and we recommend building a leaderboard website separate from your scoring tool to provide live updates as your judges submit their scores. To service this, we hosted a simple static HTML page out of an Amazon Simple Storage Service (Amazon S3) bucket that used JavaScript and DataTables to query the scoring tool and display the scores in a live and searchable scoreboard. To query the submitted scores, we wrote a simple AWS Lambda function to pull the data directly out of Honeycode using its QueryTableRows API function, and exposed it as an API endpoint via Amazon API Gateway. For instruction on how to build the leaderboard, please refer to the GitHub repo.

Select the winners

The compositions submitted by the participants are queued for judging. There are two types of Judges – Human and AI. The human judges are a team of leaders from your organization that you select and train to participate in the competition. The AI Judge is a ML model that we provide you to run inference on musical tracks submitted by participants.

Judging tenets

Human judges who listen to the tracks and use the scoring tool to rate their feedback. When selecting candidates to be part of the human judges team, make sure they have a musical disposition. Judges don’t need to have musical talent, but should be open for new experiences, enjoy what they’re doing, and carry a sense of objectivity. In short, select folks that subscribe to the following tenets:

- I believe in the power of AI to create great music
- I derive joy from nuances that achieve unforeseen creative embellishments
- While I appreciate the emotional depth of a piece, I will dispassionately consider all criteria before my decision

Evaluation criteria

The judges use the scoring tool that lists the evaluation criteria and an option to select one of several scores for each. The music is assessed based on the following criteria:

- **Originality** – The judges will be familiar with the input melodies, and will be interested in how much of a creative variance composers are able to introduce in their track compared to the input. The more variance, the

better.
- **Balance –** The music must be as enjoyable as its unique. The judges believe the right combination of balance between harmony, rhythm, and accompaniments will create a well-rounded composition. Cleanliness of sound will elevate the listening experience.
- **Emotional resonance –** The music that moves the judges is that which resonates with them. Compositions that are instantly likable, groovy, and create an emotional bond with the listener will be ranked higher.

Scoring scale

For the judging process, we use a five-point scale. For each of criteria, the judges pick a score of 1–5 based on their listening experience. The cumulative score drives the composer’s position in the leaderboard. The scores are titled as follows:

1. Very unimpressed
2. Slightly unimpressed
3. Neither impressed or unimpressed
4. Slightly impressed
5. Very impressed

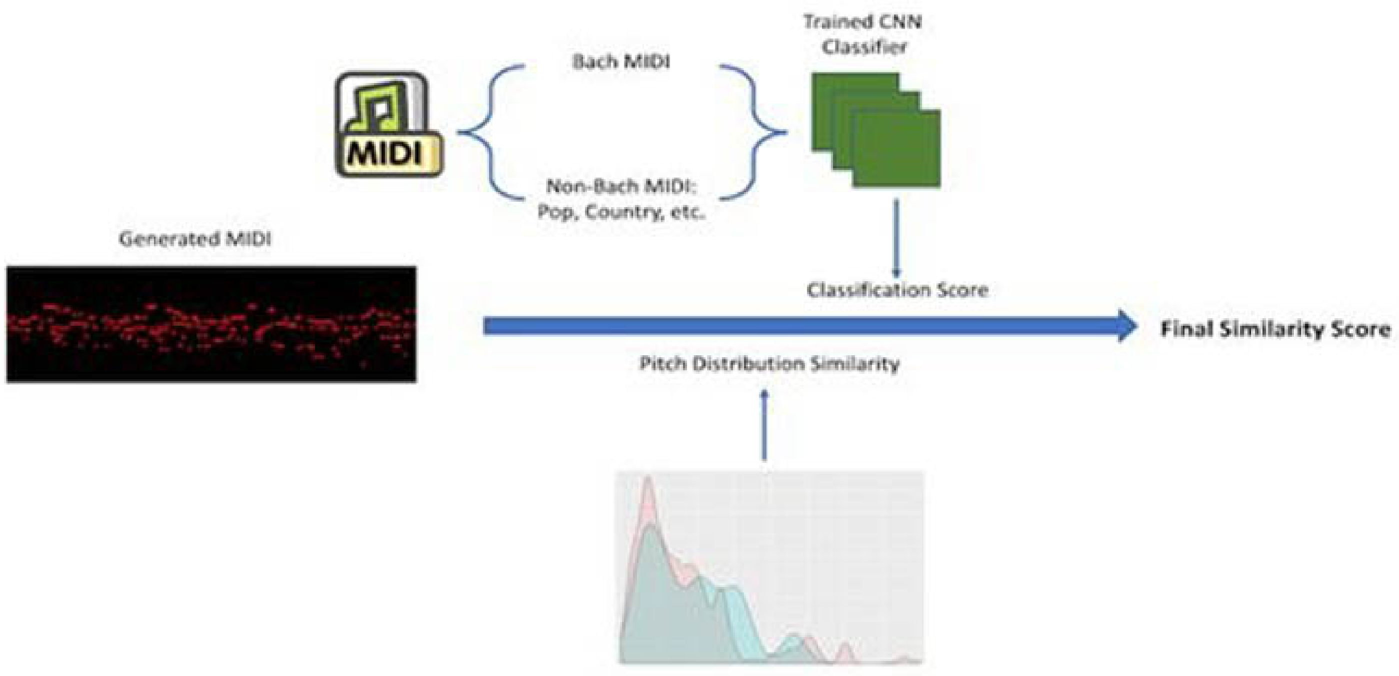
Scoring process

The scoring process involves the following steps:

1. Judges have access to their own SoundCloud playlists, which are updated with the submissions.
2. All judges should judge each submission.
3. When the judges are ready, they choose a submission in SoundCloud to listen to the piece. As they’re listening, they use the scoring tool to determine a score for each criteria and record that. We recommend that judges listen to a piece multiple times to formulate their decision. Also, they should make notes as to why they selected a particular score for a particular piece.
4. After they submit the score for the piece, the tool should first validate if submissions have been received from all the judges for a piece. If not, this composition isn’t be sent to the leaderboard.
5. After all judges have scored a piece, the tool calculates the final score and sends this to the leaderboard.
6. The top three composers from the leaderboard are selected for the awards.
7. During the awards ceremony, the judges should share their observations on the submissions from the top three composers.

AI judge: An ML model trained to recognize Bach-like music

AWS DeepComposer creates a completely original melody in seconds, powered by AI. It seems only appropriate that the composition is also judged by an AI model. Data scientists used samples of Bach and non-Bach input melodies to train an ML model that can provide a score of how Bach-like a composition is. MIDI (Musical Instrument Digital Interface) is a file format that is popularly used to store music, and the input melodies are provided to the model in the MIDI format. A Python library called PyPianoroll is used to read the MIDI files into a multitrack object, which allows us to easily wrangle and manipulate the music files.



The dataset for training is prepared with both positive examples (original Bach music MIDI files) and negative samples (non-Bach MIDI files). A CNN-based classifier is then trained on these samples to classify whether an input melody is in Bach style or not. To get the final score for each composition, the confidence score of the CNN model’s output is combined with the distribution similarity on pitch frequency of the input melody to original Bach music.

Use the AI judge to score melodies

To run the AI judge, the input files need to be in the MIDI format, and melodies can’t be more than eight bars long.

Before you run the judge, deploy a web application to collect input melodies. In this first step, we use the AWS Serverless Application Model (AWS SAM) to deploy an application that lets users upload files to an S3 bucket using Amazon S3 presigned URLs.

1. Clone the GitHub repository to your local development machine using `git clone`.

Make sure that the AWS Identity and Access Management (IAM) role you use to access the AWS account has sufficient permissions to deploy an AWS CloudFormation template, Lambda functions, and Amazon S3.

2. Update the `frontend/index.html` file with the `index.html` provided in this repository.

The new file lets users upload only MIDI files.

3. Update the Lambda function code in `getSignedURL/app.js` with the `app.js` provided.

We update this function to receive a file name along with the file as input, and upload the incoming file under a specific name.

4. From the repository’s parent directory, deploy the application using `sam deploy --guided`.

Refer to the AWS Samples repository for more detailed instructions on deploying the template. This step might take several minutes.

5. At the end of the deployment, note the output values, specifically the `APIendpoint` value.

It should look similar to `https://abcd123345.execute-api.us-west-2.amazonaws.com/`.

6. Update the `index.html` file with the API endpoint from the previous step.

The upload URL is your endpoint with the `/uploads` route added, and looks similar to `https://abcd123345.execute-api.us-west-2.amazonaws.com/uploads`.

7. Finally, upload the `index.html` file to Amazon S3 to be set up for public hosting.

You can use the same bucket that hosts the leaderboard.

We recommend that the unique name for each composition be the participant’s work alias, email, or a `firstname_lastname` format to ensure that the composition can be traced back to the participant. If they fail to provide a user name, the application stores the melody with a random file name and they lose their chance of winning.

You’re now ready to run the AI judge. The inference code to run the AI Judge is provided in the repository. A Jupyter notebook is provided with the repository to run the inference through an interactive session. This can be run either on your local machine, if you have Jupyter installed, or on a SageMaker notebook instance. If you use SageMaker, we recommend a p3 instance for faster processing of the input melodies.

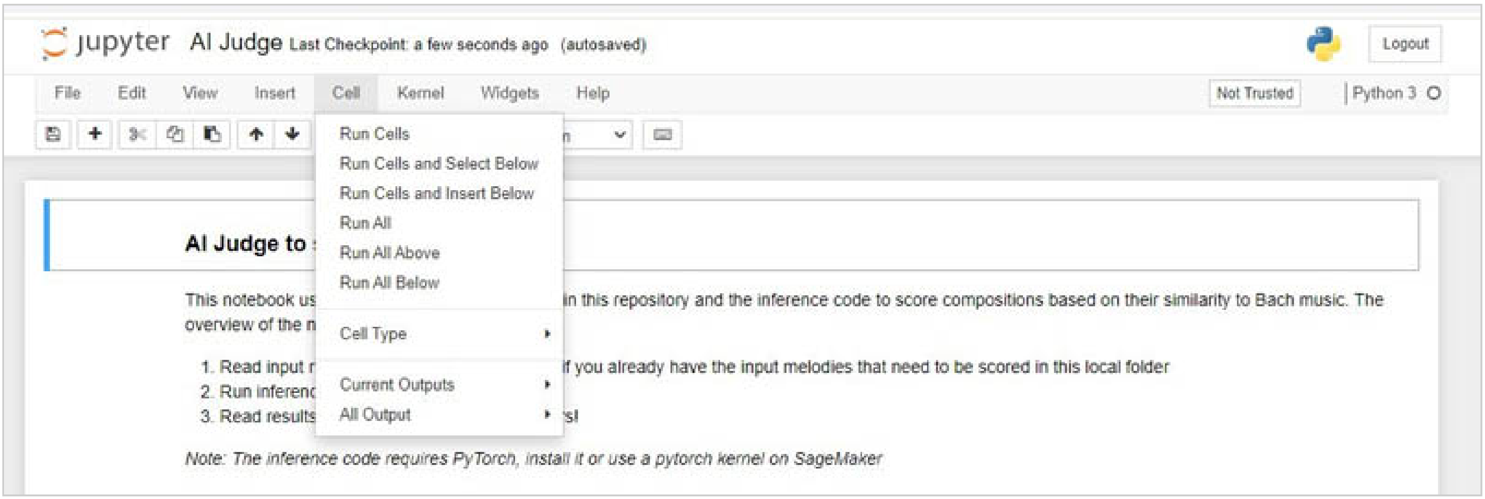
8. Download the AI judge folder from the repository to your Jupyter environment.

The folder contains three files: a pre-trained model, inference code, and the IPython notebook file that you can use to run the AI judge.

9. Make sure that all three files are within a single folder, and open the notebook file.
10. Update the notebook (cell 3) with the name of the S3 bucket created through the AWS SAM template.

The notebook downloads all input files from the S3 bucket, runs the inference script to generate scores based on the similarity to Bach music, and prints the results.

11. Run all cells by choosing the **Cell** menu and **Run All**.


jupyter AI Judge Last Checkpoint: a few seconds ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run Cells
Run Cells and Select Below
Run Cells and Insert Below
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output

AI Judge to

This notebook us in this repository and the inference code to score compositions based on their similarity to Bach music. The overview of the n

1. Read input r if you already have the input melodies that need to be scored in this local folder
2. Run inferenc
3. Read results s!

Note: The inference code requires PyTorch, install it or use a pytorch kernel on SageMaker

The last cell of the notebook outputs the scores for each composition in descending order.

Run the awards ceremony

A good way to drive engagement for the event is to select a senior leader (or multiple senior leaders) to present the awards and facilitate the ceremony. We provide the following recommendations based on our experience.

We have three 3 different awards categories:

- The top three composers from the leaderboard as the winners of AWS DeepComposer Got Talent
- The top three composer submissions for the AI judge selection as the winners of the special category
- One winner from the Kahoot competition played on day 1

We recommend the following order when the senior leaders present the awards:

1. Introduce themselves and share their excitement in recognizing the winners.
2. Talk about the importance of ML in their organization.
3. Provide a recap of the AWS DeepComposer Got Talent competition with some key statistics such as number of registrants, number of participants, number of submissions, feedback so far, and so on.
4. Announce the number of winners and recap the prizes for the competition in various categories.
5. Share the judges’ commentary on the winning tracks and announce the winners.
6. Conclude with the potential of ML that’s yet unexplored, and encourage the audience to continue innovating.

Conclusion

In this post, we presented a detailed walkthrough on how to prepare for and run your own AI-powered musical challenge for your employees, customers, and partners: AWS DeepComposer Got Talent.

To get you started with AWS DeepComposer we provide a Free Tier. Please refer to https://aws.amazon.com/deepcomposer/pricing/ for more details.

Music is always fun, and when combined with the potential to learn the latest in ML, and also win prizes, you are sure to drive a lot of engagement. We look forward to hearing your feedback. Have fun with your event!

About the Authors

Maryam Rezapoor is a Senior Product Manager with AWS DeepLabs team based in Santa Clara, CA. She works on developing products to put Machine Learning in the hands of everyone. She loves hiking through the US national parks and is currently training for 1-day Grand Canyon Rim to Rim hike. She is a fan of Metallica and Evanescence. The drummer, Lars Ulrich, has inspired her to pick up those sticks and play drum while singing “nothing else matters.”

Jana Gnanachandran is an Enterprise Solutions Architect at AWS based in Houston, Texas, focusing on Data Analytics, AI/ML and Serverless platforms. In his spare time, he enjoys playing tennis, learning new songs in his keyboard, and photography.

Durga Sury is a Data Scientist with AWS Professional Services, based in Houston, TX. Her current interests are Natural Language Processing, AutoML and MLOps. Outside of work, she loves binging on crime shows, motorcycle rides, and hiking with her husky. She is trained in Indian classical music and hopes to play Metallica on the guitar one day.

Josh Schairbaum is an Enterprise Solutions Architect based in Austin, TX. When he’s not assisting customers derive and implement the necessary technical strategy to meet their business goals, Josh spends a lot of time with his kids, exercising, and diving deeper into Hip Hop, both old and new.

David Coon is a Senior Technical Account Manager based out of Houston TX. He loves to dive deep into his customer’s technical and operational issues, and has a passion for music especially by Snarky Puppy, machine learning, and automation. When he’s not on the clock, David is usually walking his dogs, tinkering with a DIY project at home, or re-watching The Expanse.

David Bounds is an Enterprise Solutions Architect based in Atlanta, GA. When he isn’t obsessing over his customers, David is endlessly tinkering with his 3D Printers, running, cooking, or discovering a new, expensive, and timeconsuming hobby. David is usually accompanied by a very worried Boxer who quite likes anything by the Wu-Tang Clan.

Henry Wang is a Data Scientist at Amazon Machine Learning Solutions Lab where he helps AWS customers across industries to adopt Cloud and AI to tackle various challenges. In his spare time, he enjoys playing tennis and golf, watching StarCraft II tournaments, reading while listening to New Age music.

Prem Ranga is an Enterprise Solutions Architect based out of Houston, Texas. He is part of the Machine Learning Technical Field Community and loves working with customers on their ML and AI journey. Prem can moonwalk just like MJ, and is recently enamored by the deep sounds of the Liquid Mind collection, Max Richter, Nils Frahm and considers the soundtrack of Ad Astra profound.