

PAT 463/563 (Fall 2026)

Music & AI

Lecture 12: Music Classification

Instructor: Hao-Wen (Herman) Dong

Music Classification

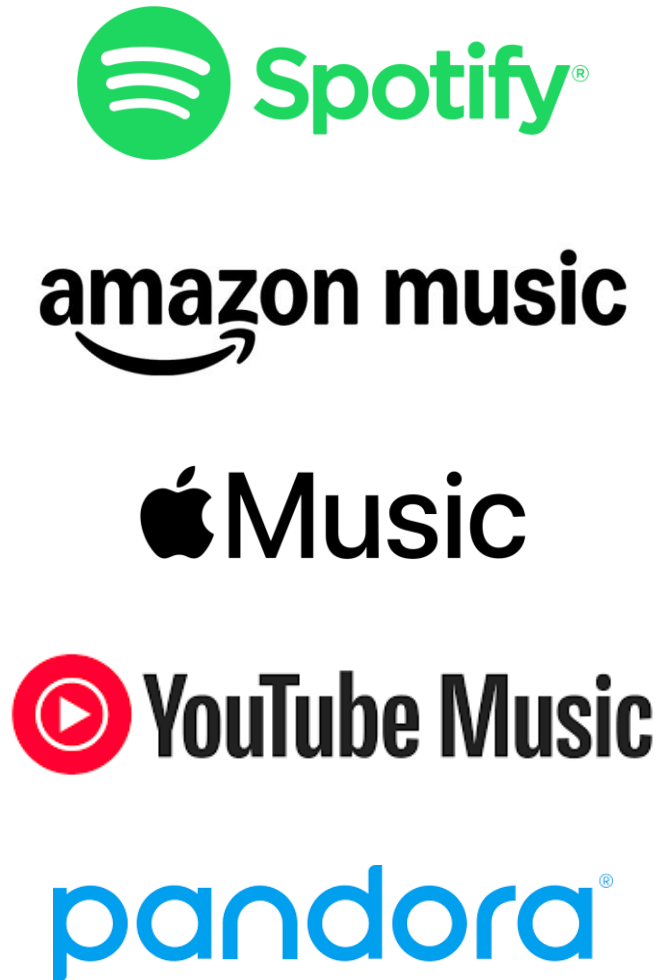
Music Classification Tasks

- **Genre** classification (pop, rock, r&b, jazz, hip-hop, classical, etc.)
- **Mood** classification (happy, sad, calm, aggressive, cheerful, etc.)
- **Instrument** recognition
- **Composer** identification
- **Key** detection
- **Chord** estimation
- **Music tagging** → Can cover everything above!

| Applications of Music Classification Models

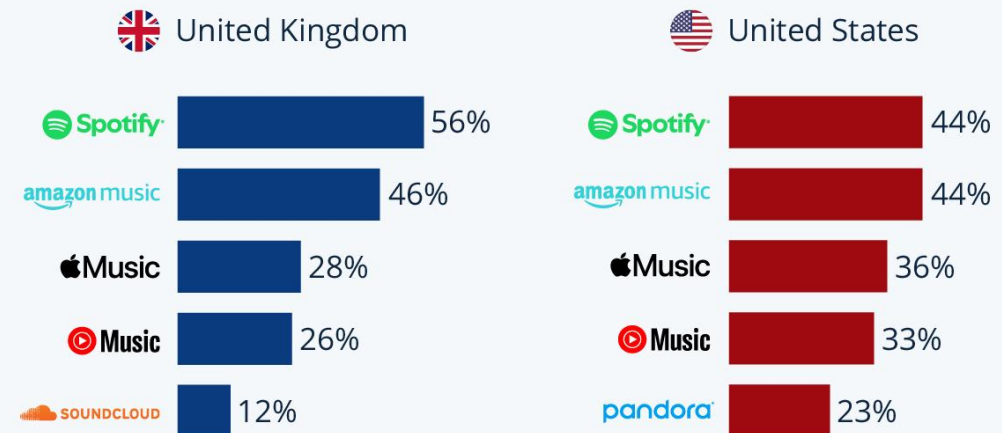
- Recommendation
- Curation
- Playlist generation
- Listening behavior analysis
- Musicology research

Music Classification for Recommendation



The Most Loved Digital Audio Streaming Platforms

Share of respondents who have paid for audio downloads or streaming services from the following platforms*



* in the 12 months prior to the survey
2,362 (UK)/4,944 (USA) respondents (18-64 y/o) surveyed Jul. 2023-Jun. 2024
Source: Statista Consumer Insights



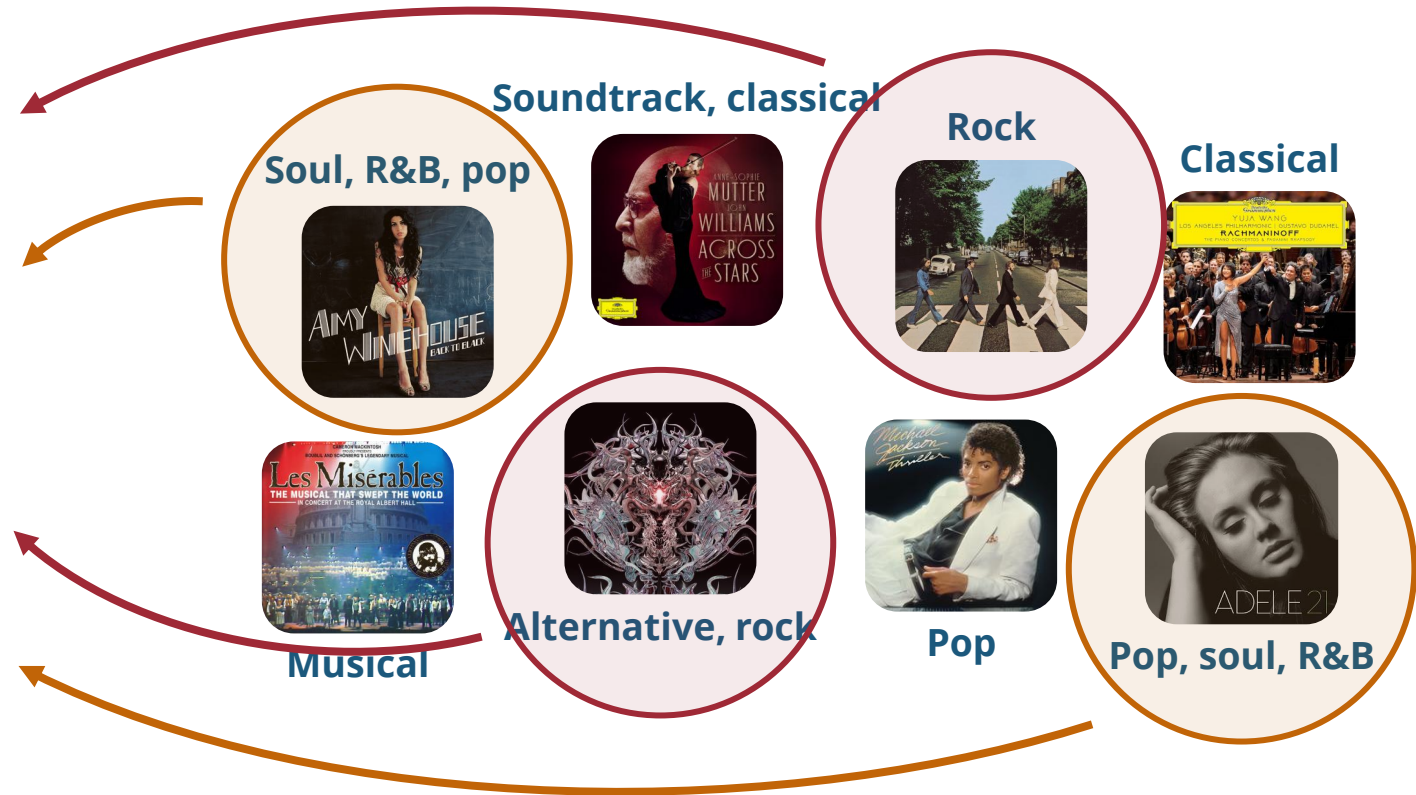
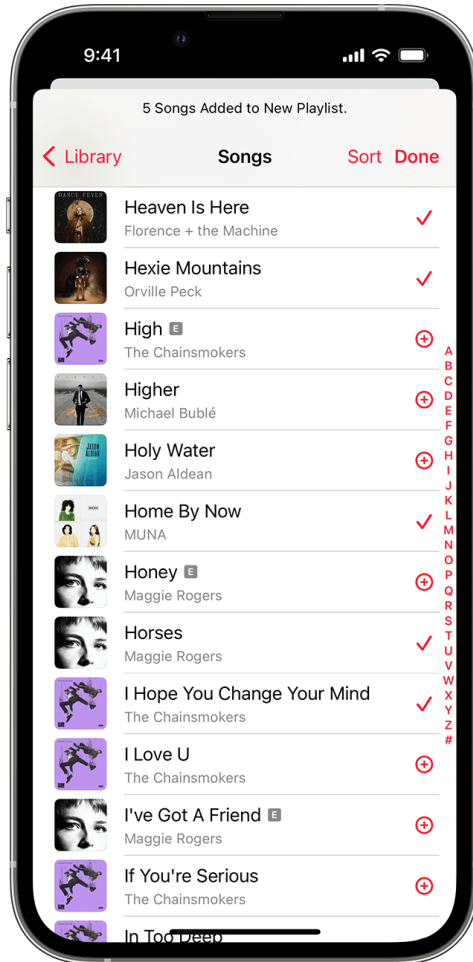
statista

Music Classification for Recommendation

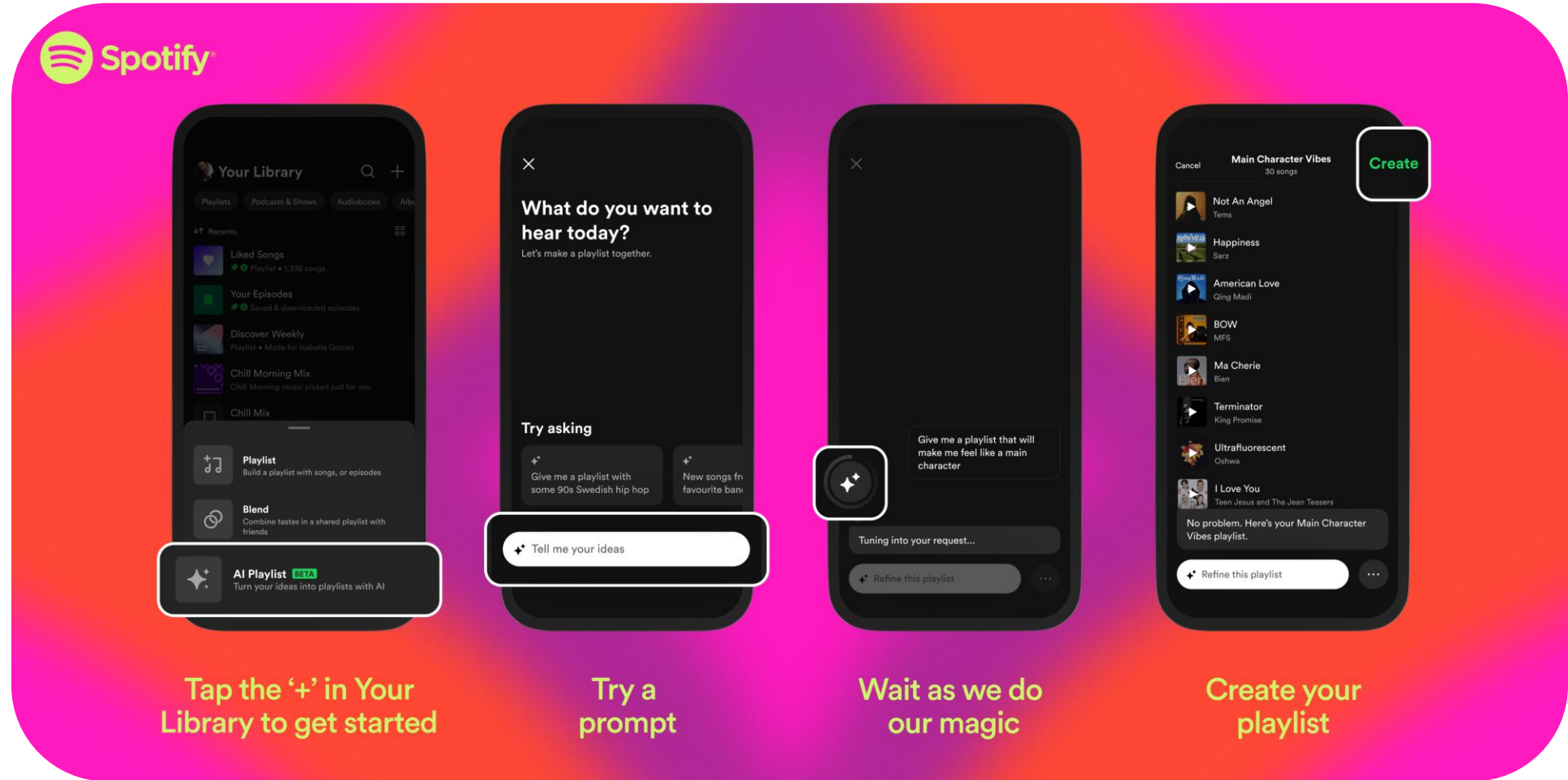
What to play next?



Music Classification for Playlist Generation

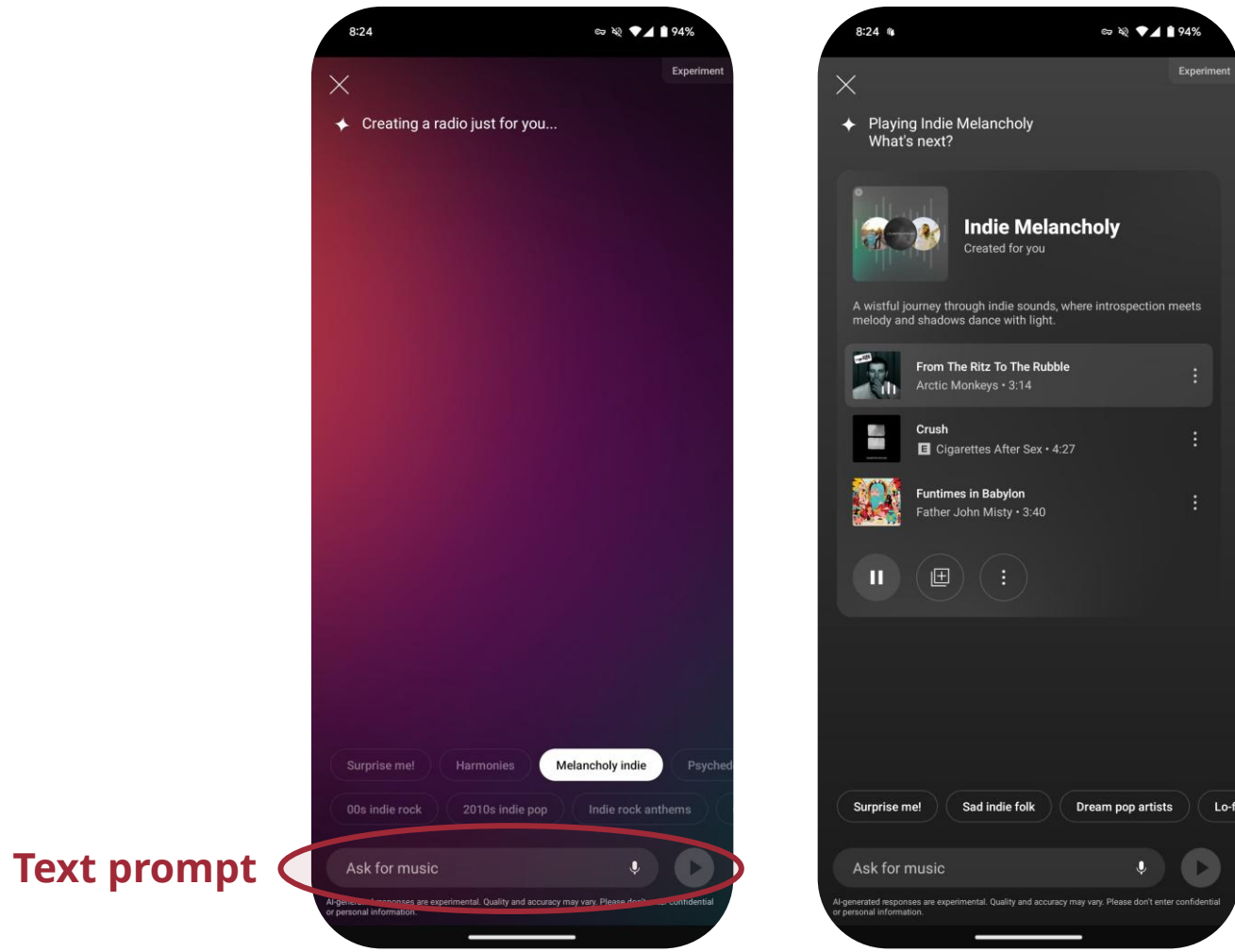


Spotify's AI Playlist (2024)



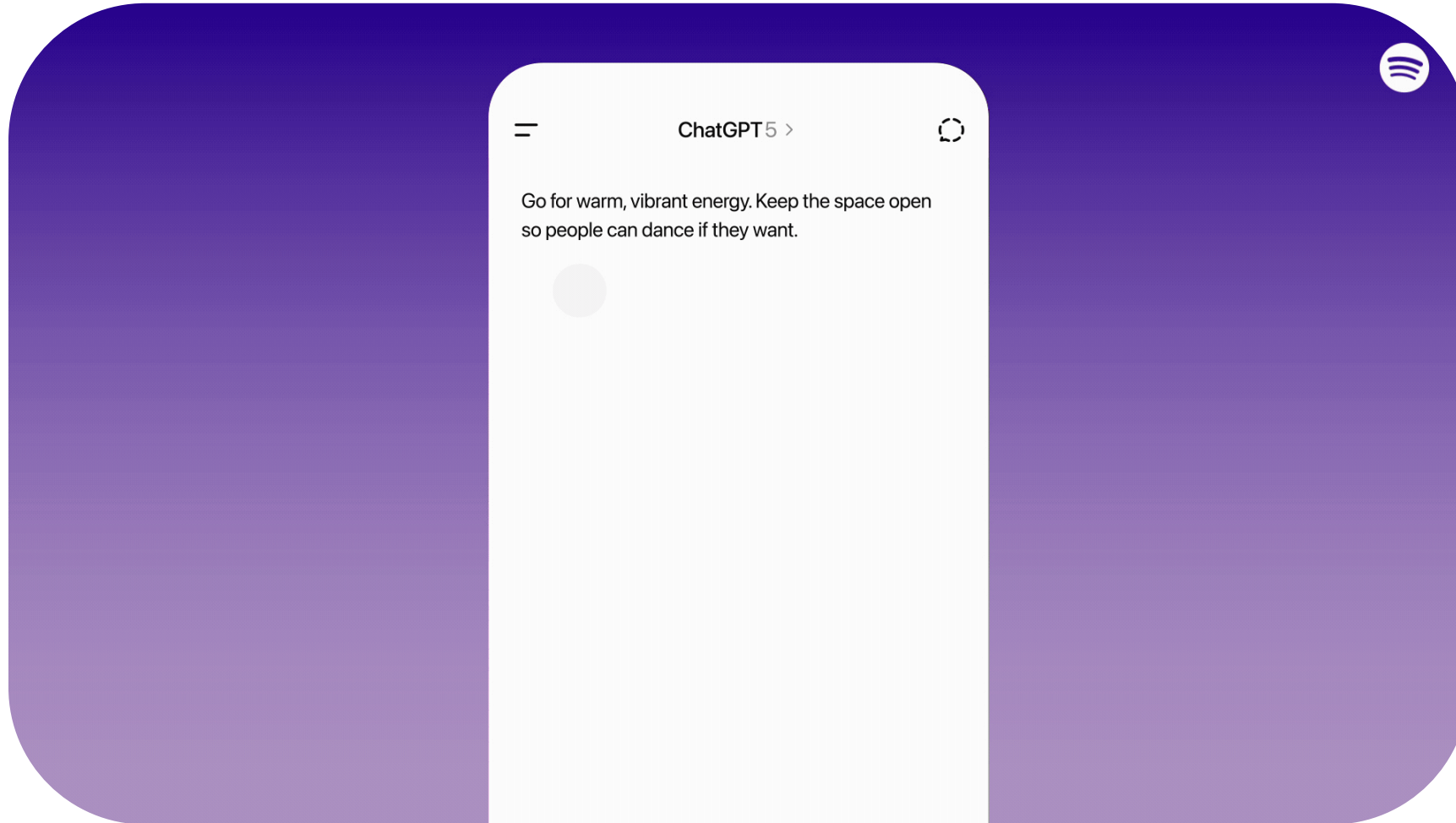
(Source: Spotify)

YouTube Music's Ask Music (2024)



(Source: Android Police)

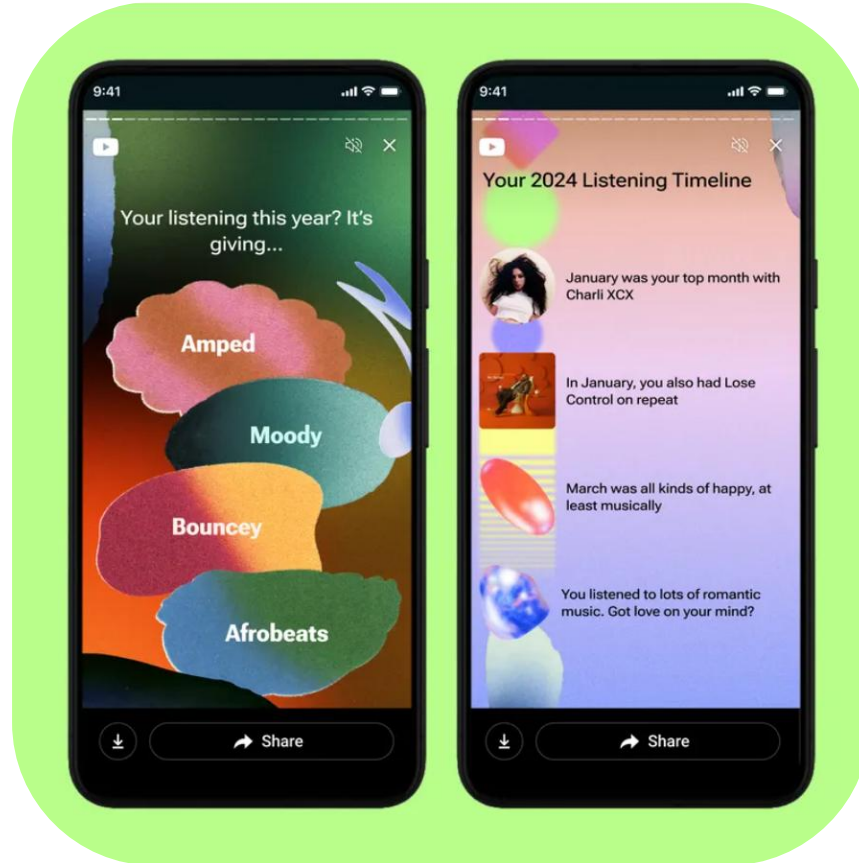
| Spotify's Personalized Picks in ChatGPT (2025)



(Source: Spotify)

Music Classification for Listening Behavior Analysis

YouTube's Music Recap



(Source: YouTube)

Spotify's Listening Personality



(Source: Spotify)

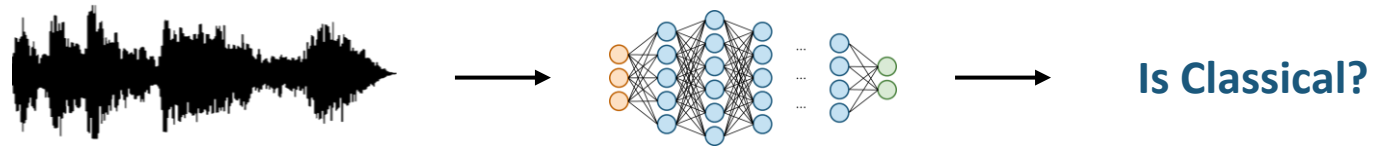
blog.youtube/news-and-events/2024-music-recap-youtube-engineering.atspotify.com/2023/01/whats-a-listening-personality/

Types of Classification Tasks

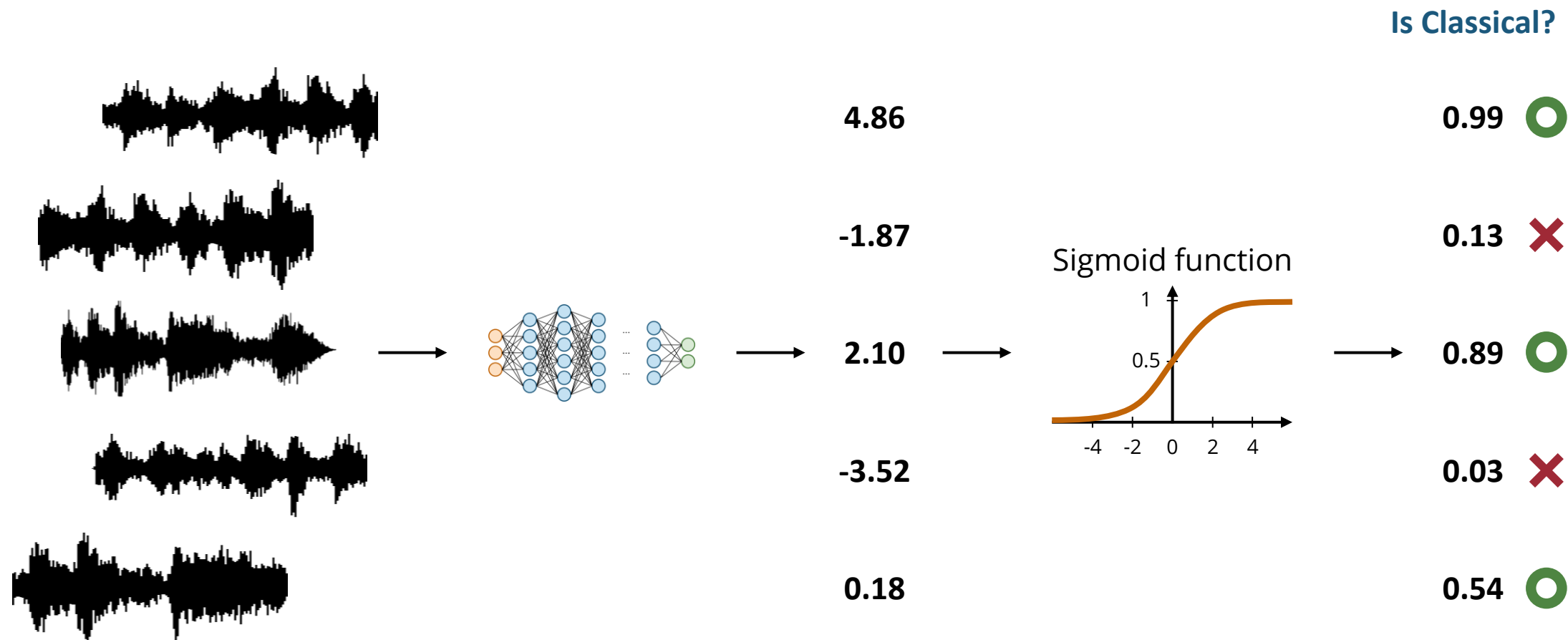
| Types of Classification Tasks

- **Binary** classification
- **Multiclass** classification
- **Multi-label** classification

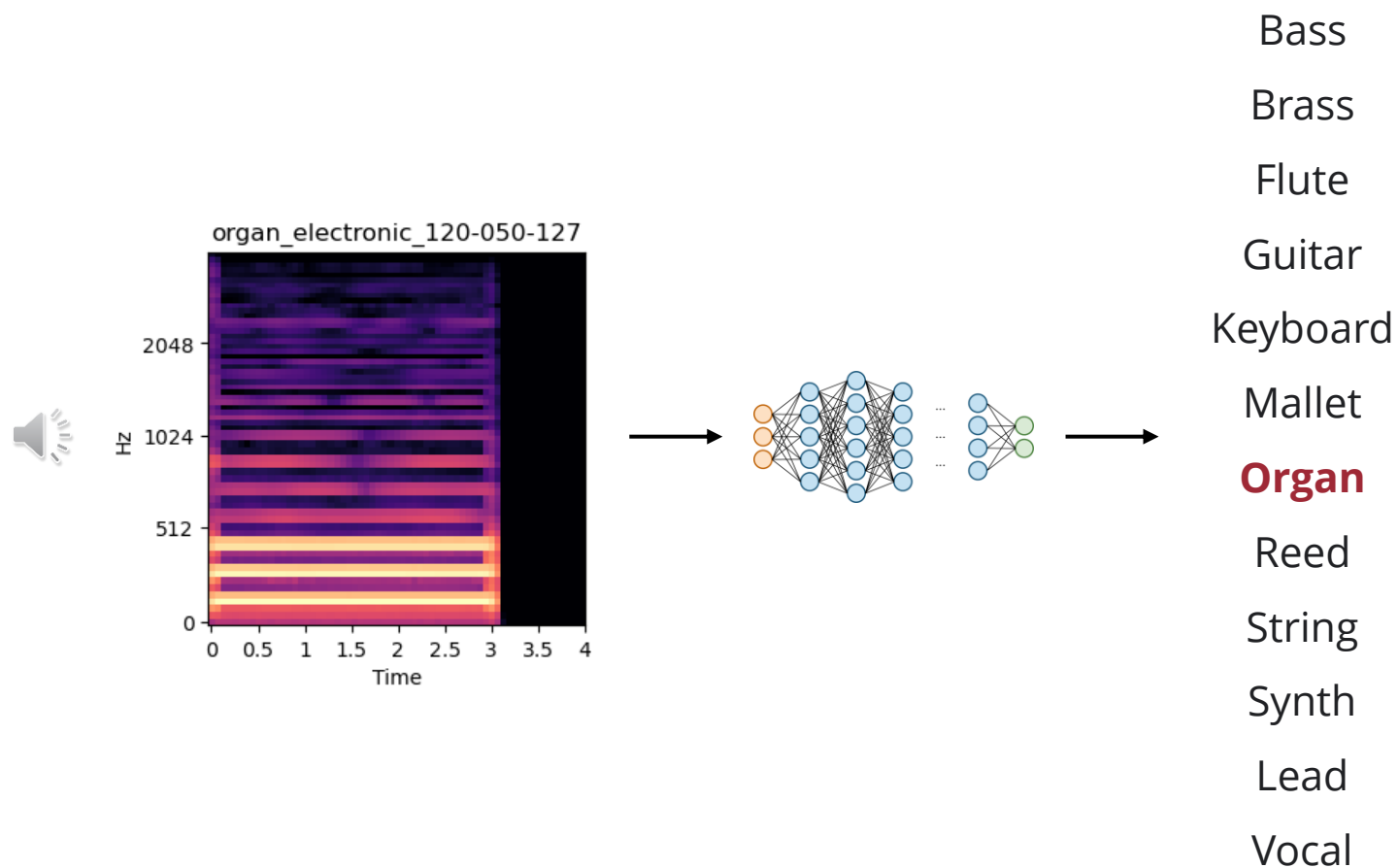
| Binary Classification



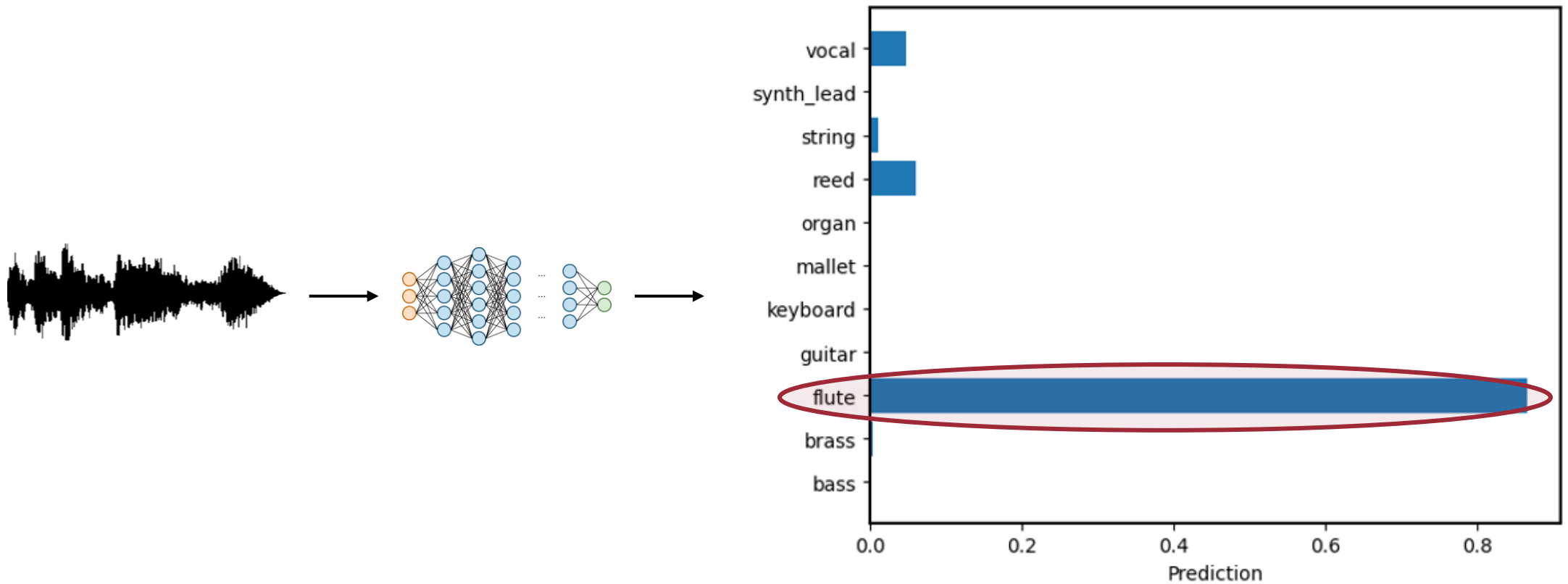
Binary Classification



Multiclass Classification



Multiclass Classification



Multi-label Classification



Soul, R&B, pop



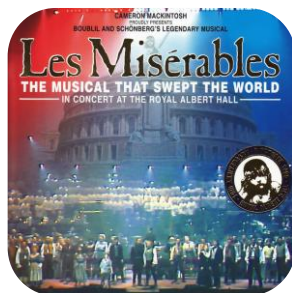
Soundtrack, classical



Rock



Classical



Musical



Alternative, rock

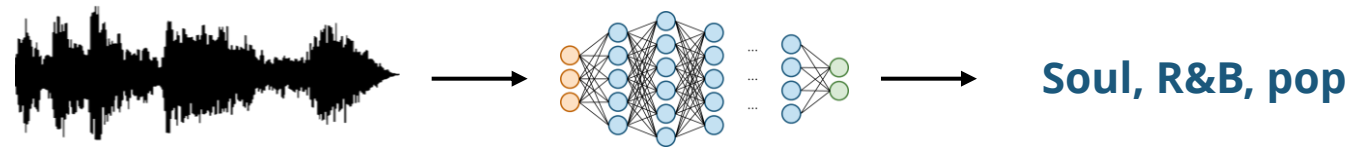


Pop

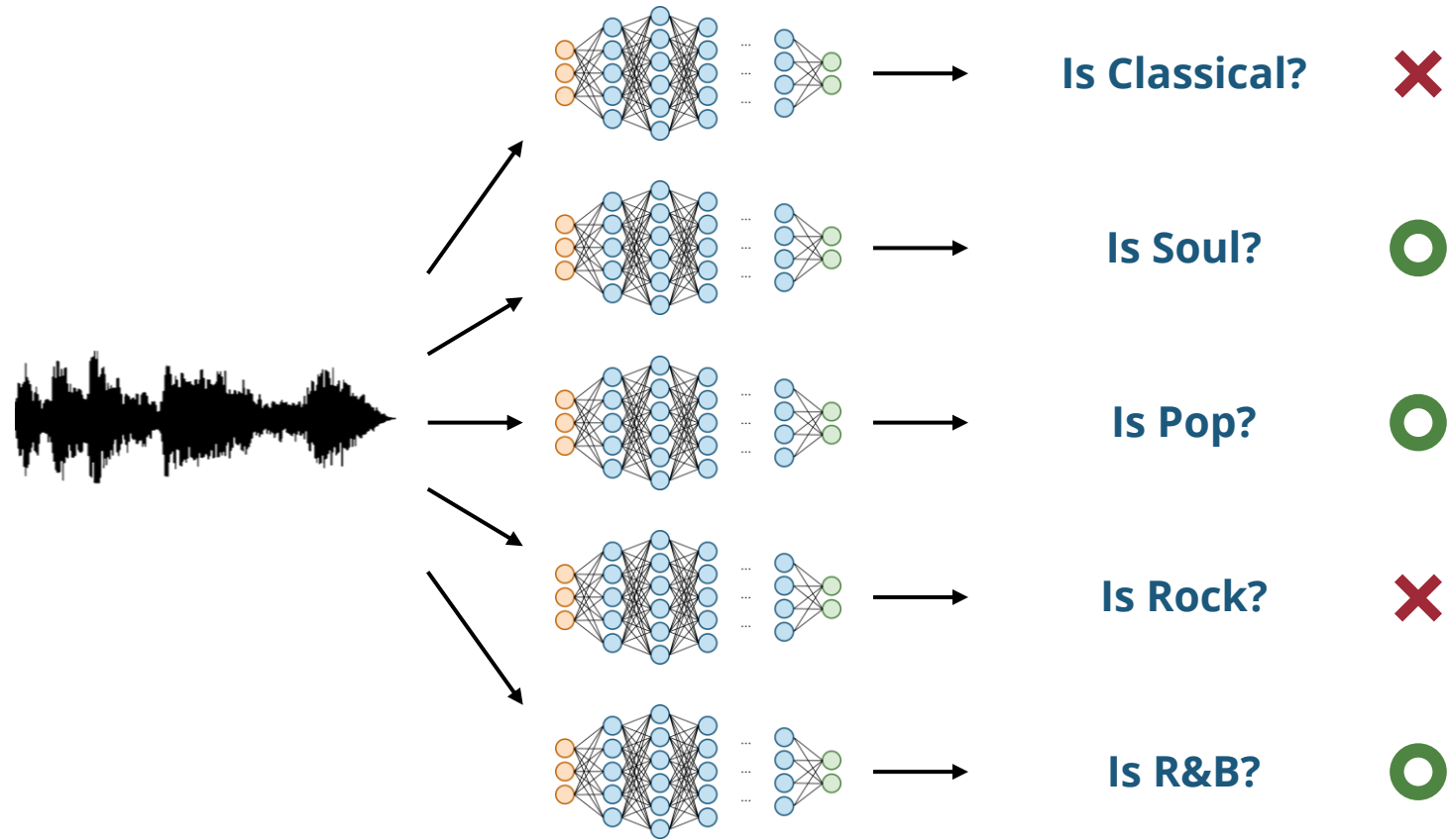


Pop, soul, R&B

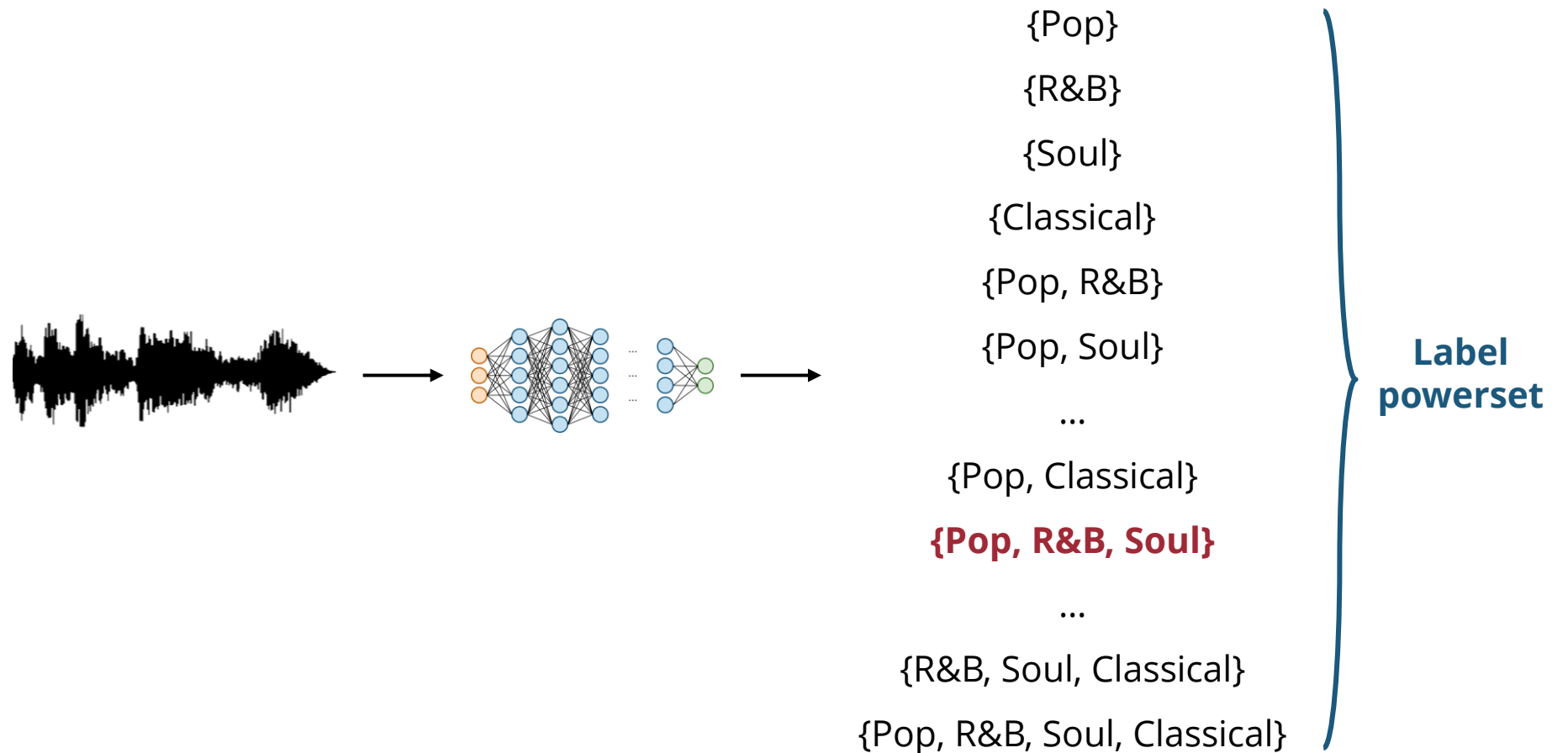
| Multi-label Classification



Multi-label Classification as Binary Classification

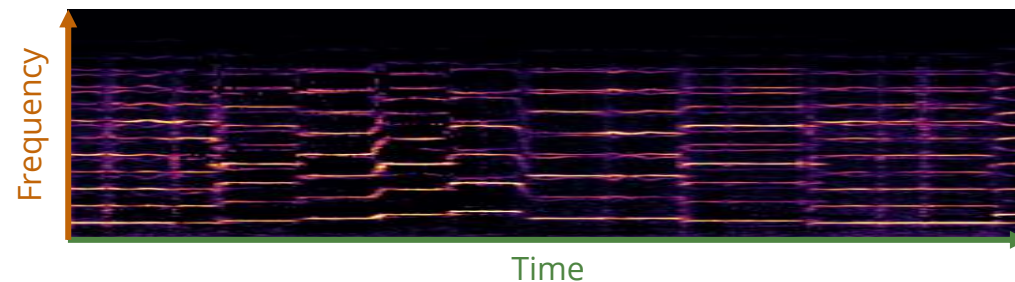


Multi-label Classification as Multi-class Classification



| Input Features

- **Waveforms**
- Time-frequency representation (**spectrograms**)
- **Hand-crafted features** or **features provided in metadata**
 - **Acoustic:** loudness, pitch, timbre
 - **Rhythmic:** beat, tempo, time signature
 - **Tonal:** key, scale, chords
 - **Instrumentation, expressions, structures,** etc.



Common Datasets

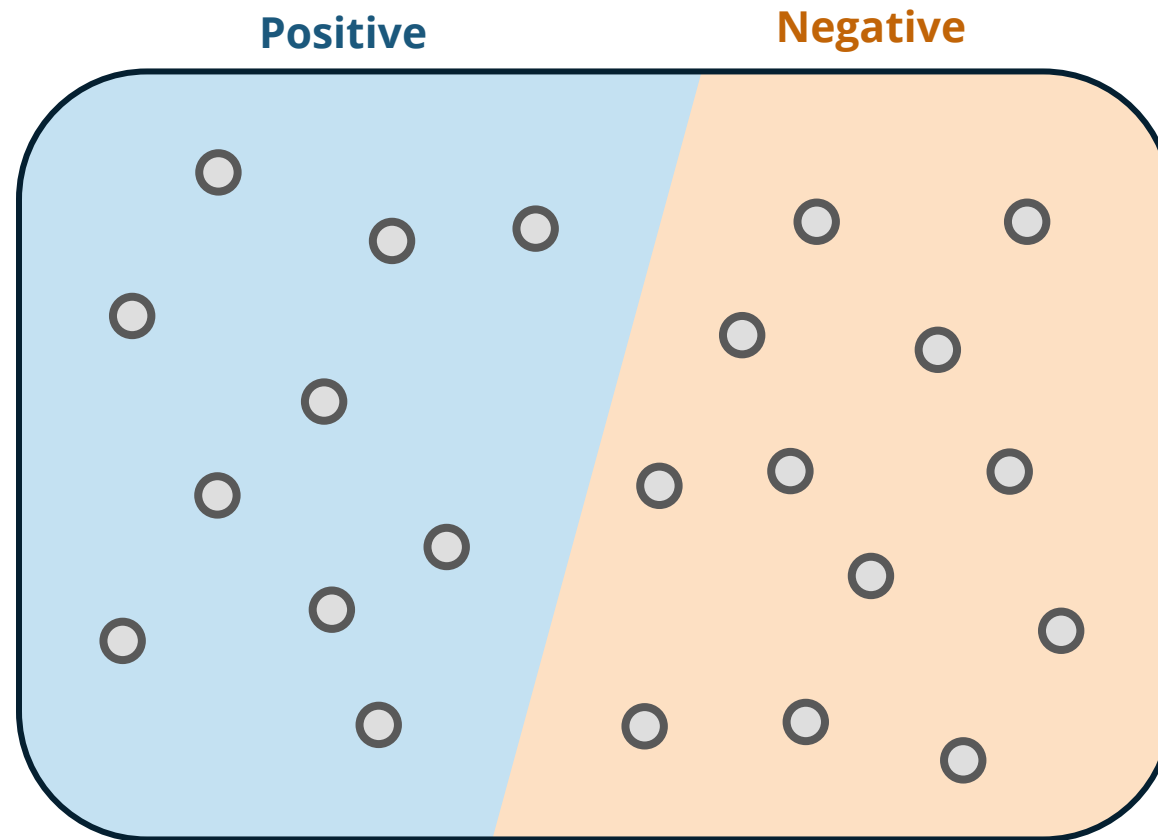
- **GTZAN**: 1,000 30-sec songs, 10 genres
- **MagnaTagATune**: 5,405 29-sec songs, 188 tags, 230 artists
- **Million Song Dataset (MSD)**: 1M 30-sec songs, >500K tags, tricky to access
- **Free Music Archive (FMA)**: >10K full songs, 163 genres
- **MTG-Jamendo**: 55K full songs, 195 tags
- **AudioSet**: 1M songs, YouTube URLs, low-quality audio
- **NSynth**: ~306K 4-sec one-shot instrument sounds

Evaluation Metrics

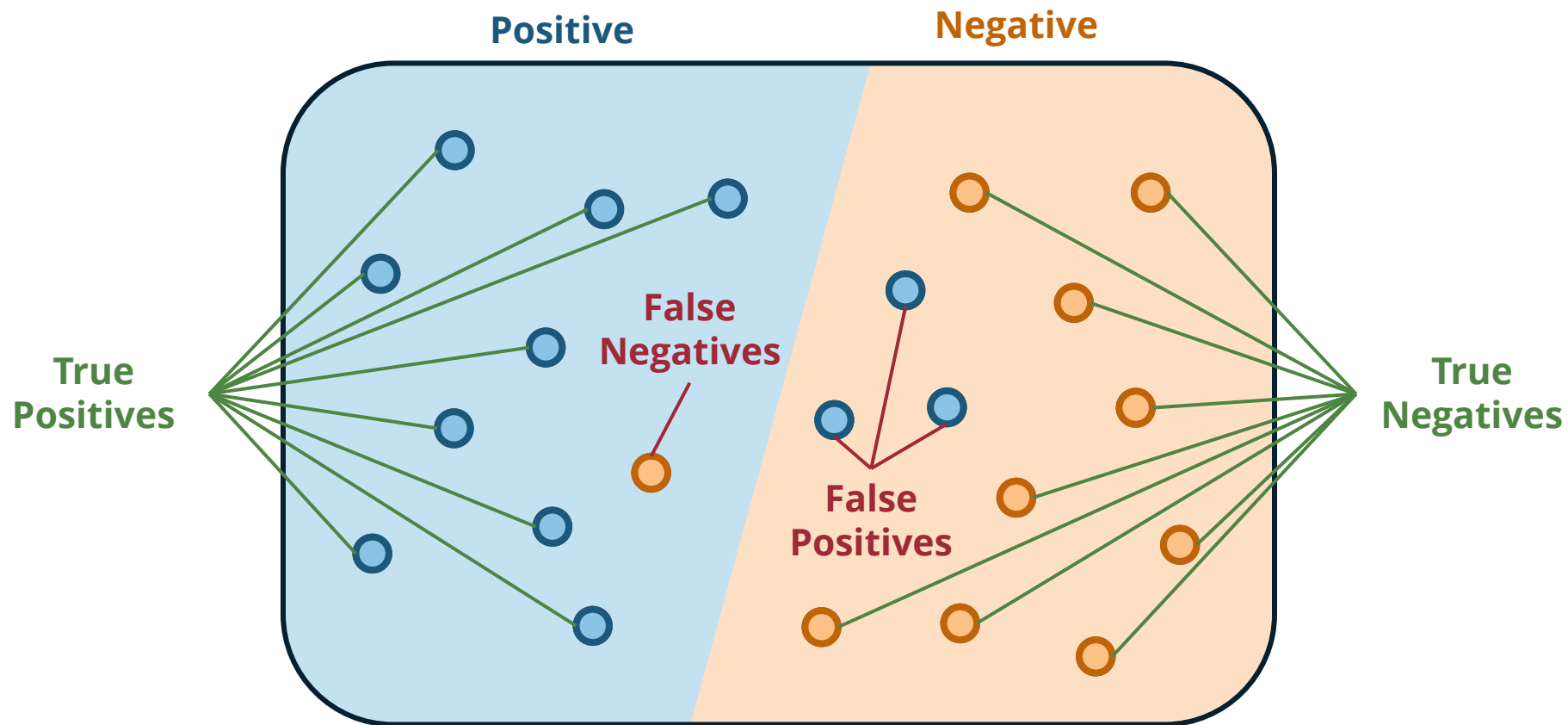
Evaluation Metrics

- **Key:** Capture what you care the most!
- The best evaluation metric depends on the actual use case
- Best to use several evaluation metrics to obtain a holistic view of your model's performance

| Toy Example: Binary Classification

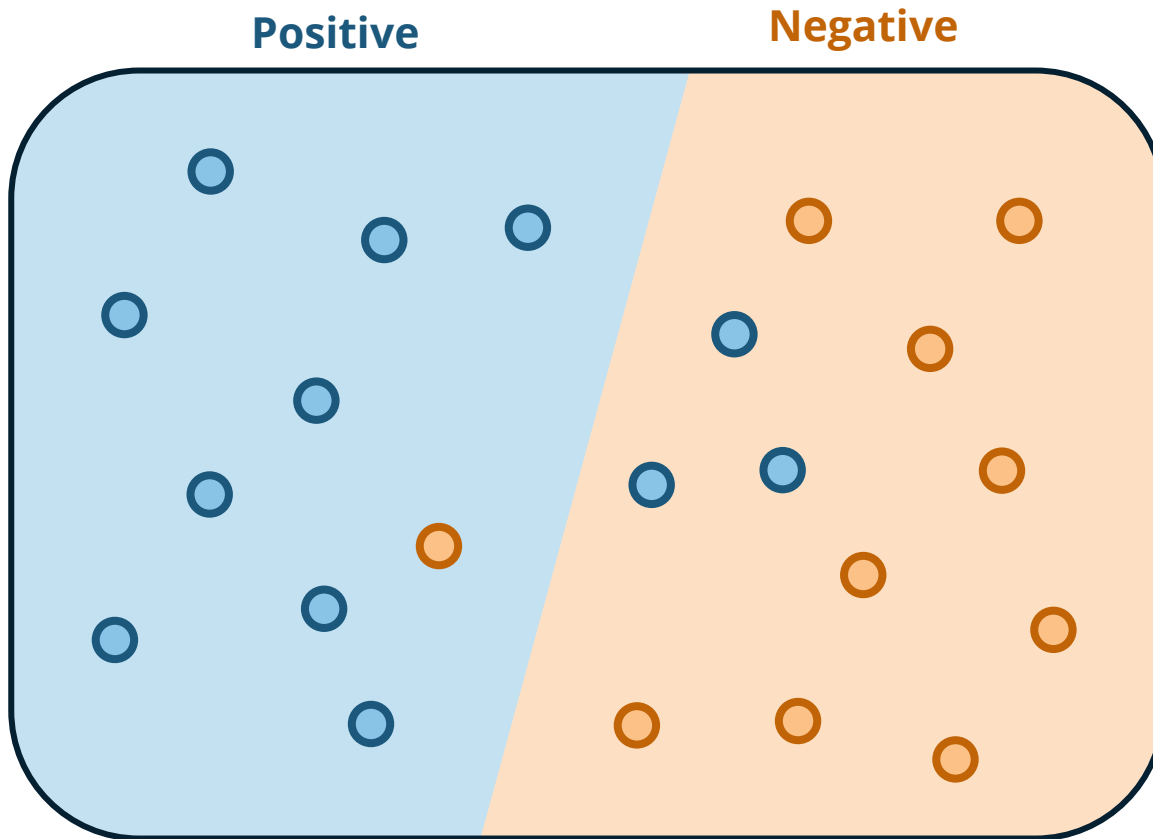


Toy Example: Binary Classification



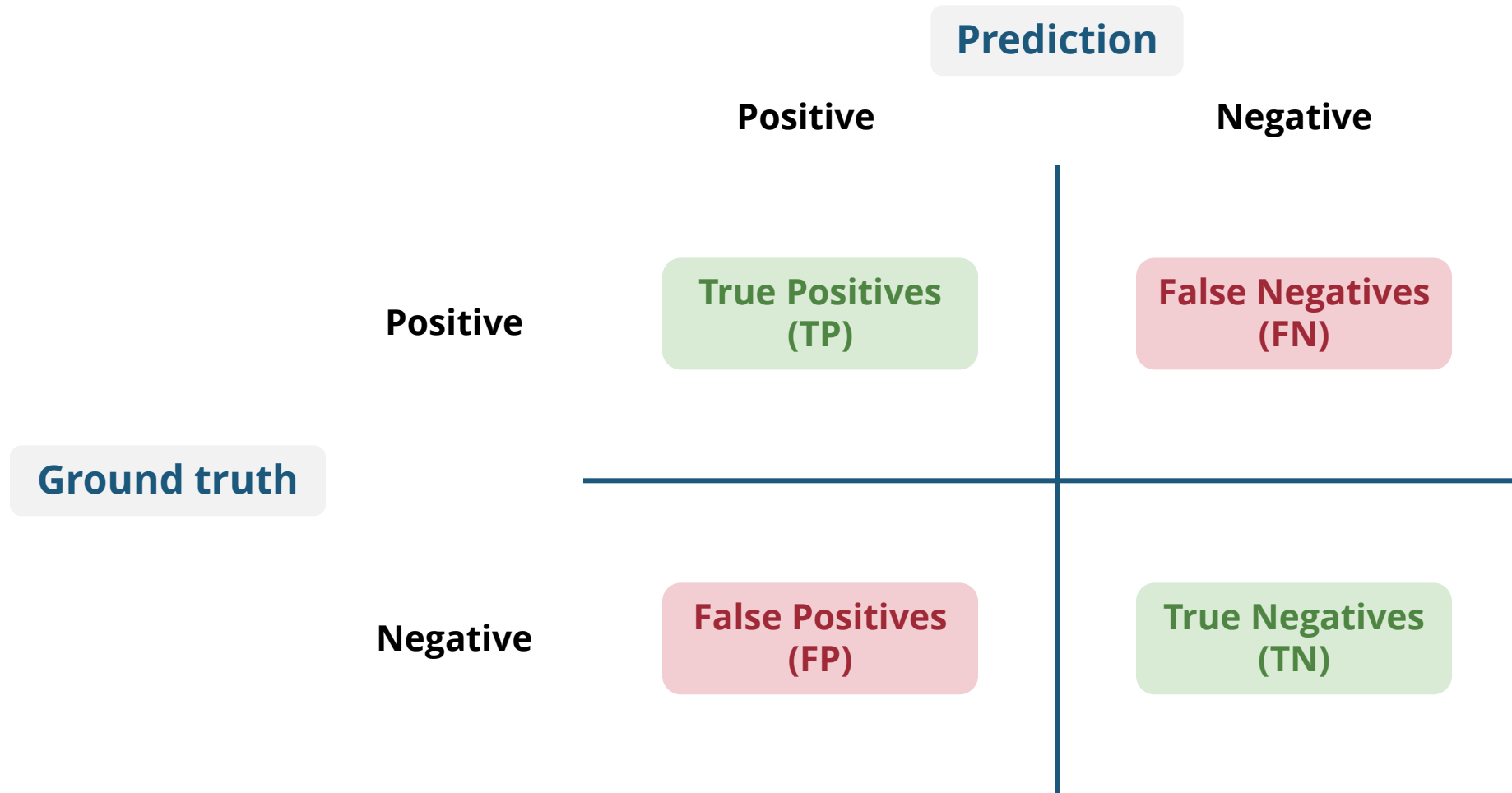
Accuracy

- **Definition:** Percentage of correct predictions across all classes

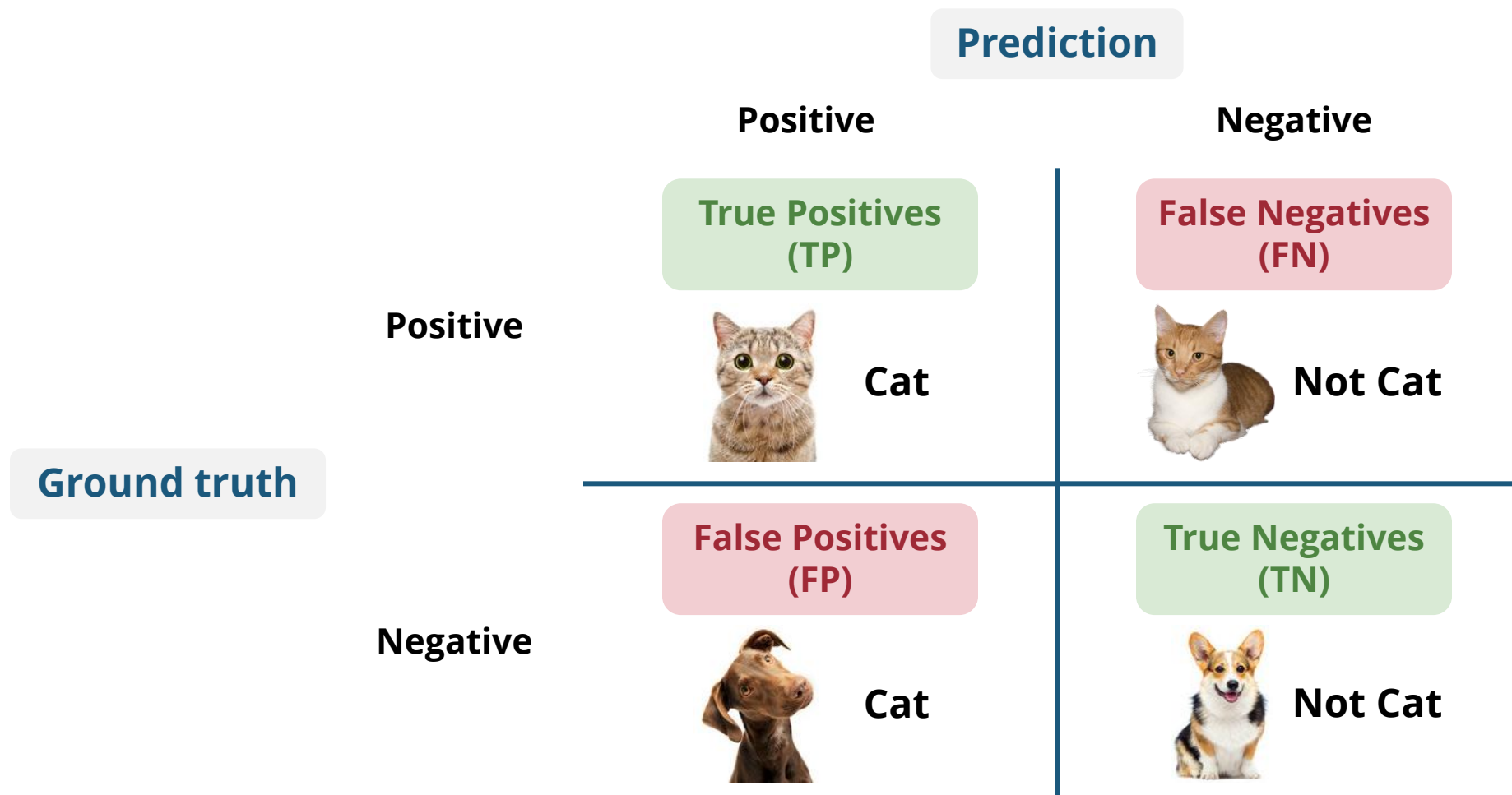


$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{12 + 12}{14 + 1 + 2 + 13} = 0.82$$

Confusion Matrix for Binary Classification



Confusion Matrix for Binary Classification

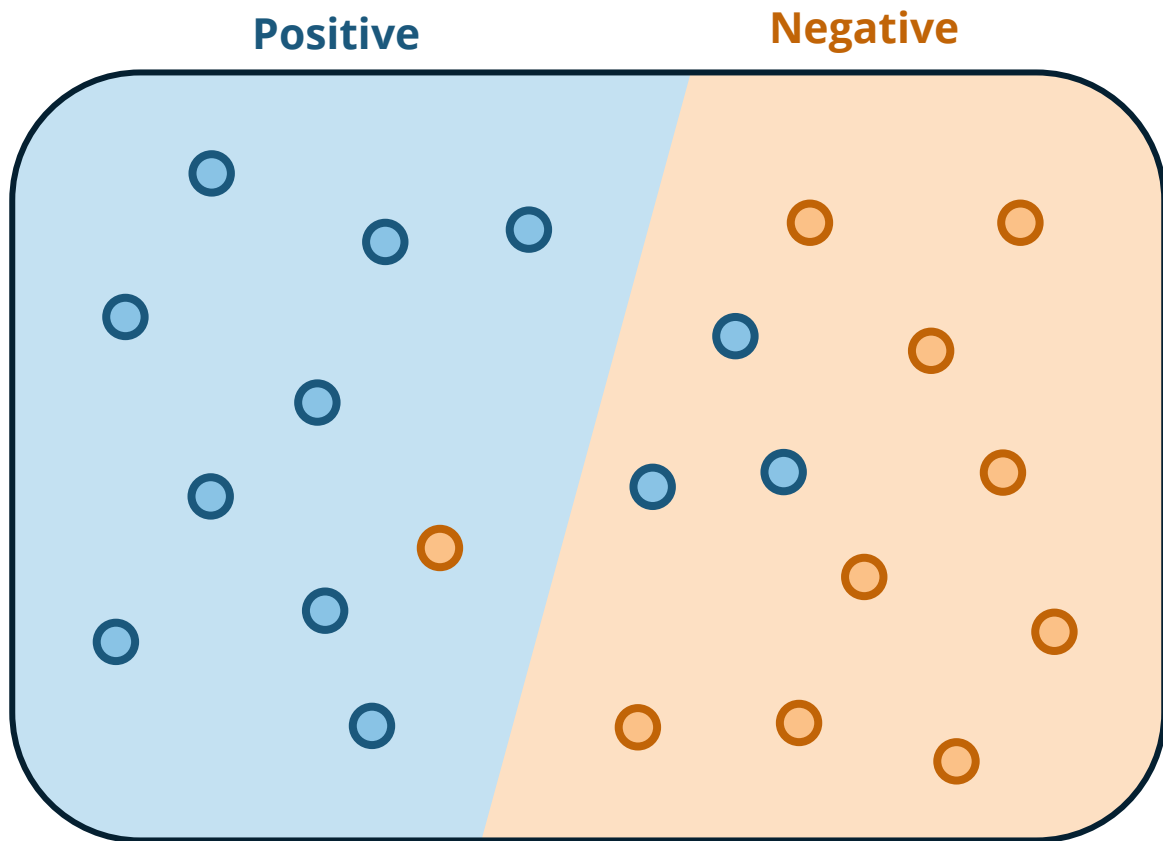


Accuracy on Imbalanced Datasets

- Accuracy does **not** work well on **imbalanced dataset**
- Take **a disease with a 1% prevalence** for example:
 - What if we simply say **negative to all diagnoses**?

		Prediction		
		Positive	Negative	
Ground truth	Positive	True Positives (TP) 0	False Negatives (FN) 1	Accuracy = 0.99 🤔
	Negative	False Positives (FP) 0	True Negatives (TN) 99	

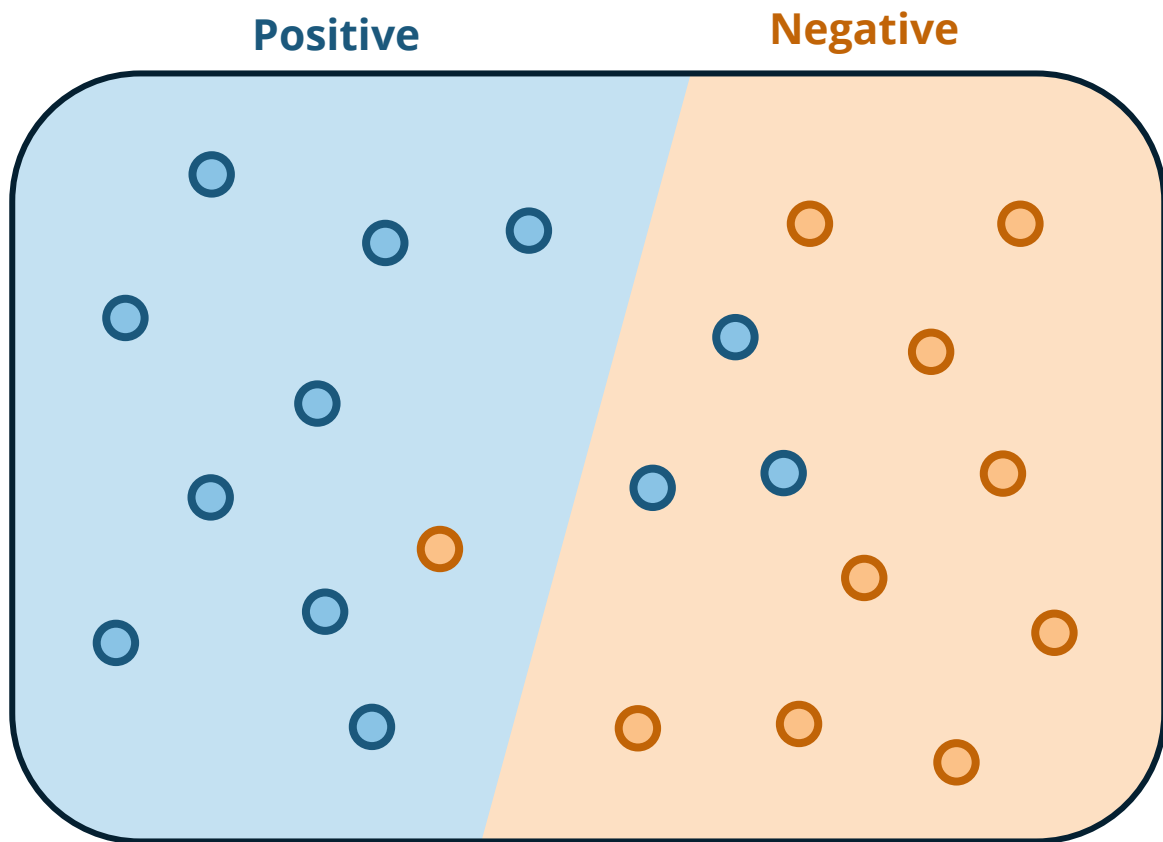
Precision



$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{10}{10 + 3} = 0.75$$

How often predictions for the positive are correct

Recall

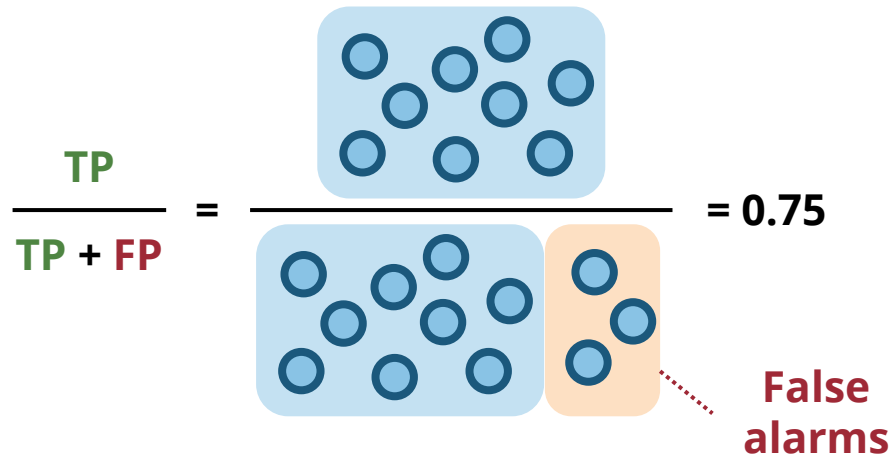


$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{9 (True Positives)}}{\text{9 (True Positives)} + \text{1 (False Negative)}} = 0.9$$

How well the model finds all positive instances in the dataset

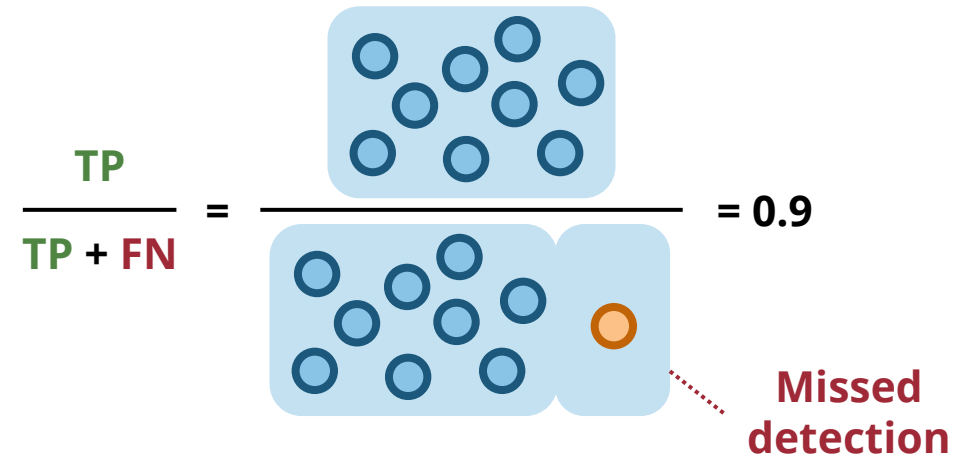
Precision vs. Recall

Precision



How often predictions for the positive are correct

Recall



How well the model finds all positive instances in the dataset

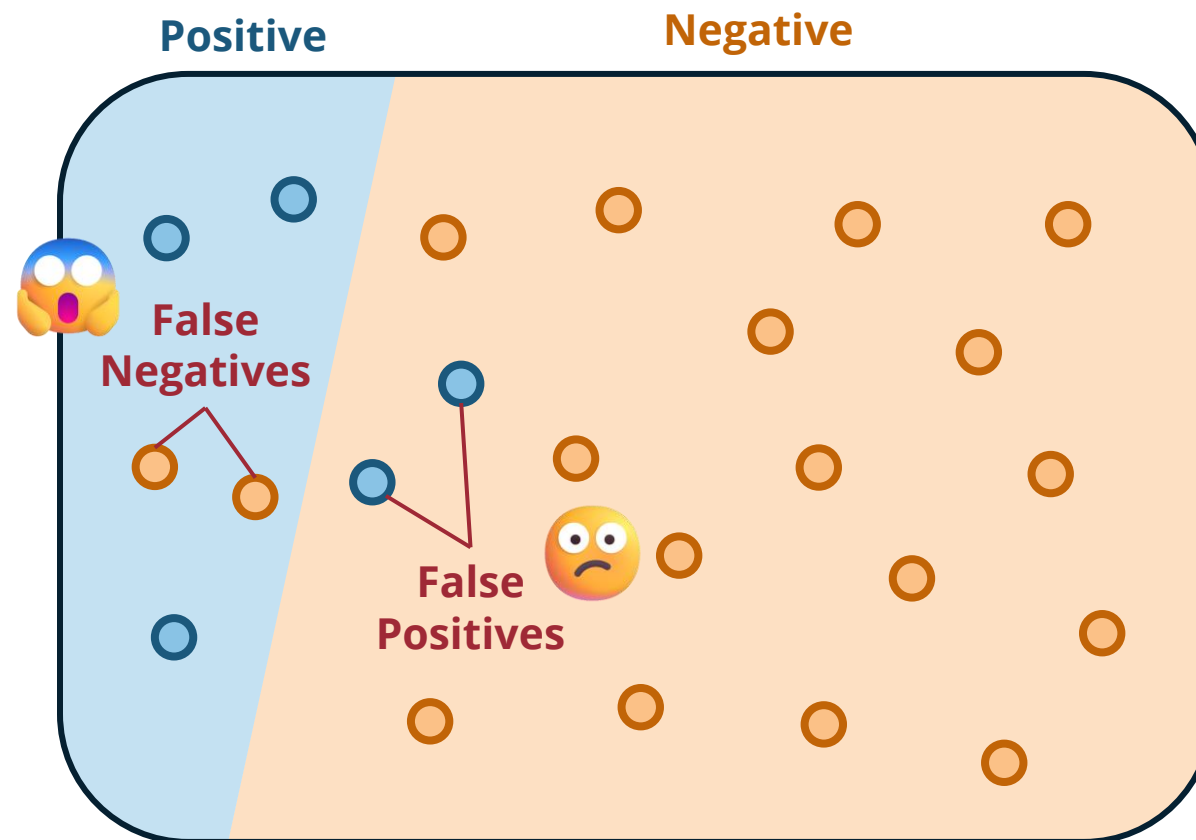
When should we care about Precision & Recall?

Rare cancer detection



Aim for high precision or high recall?

High recall ensures most cancer cases are identified.



False alarms vs. Missed detections

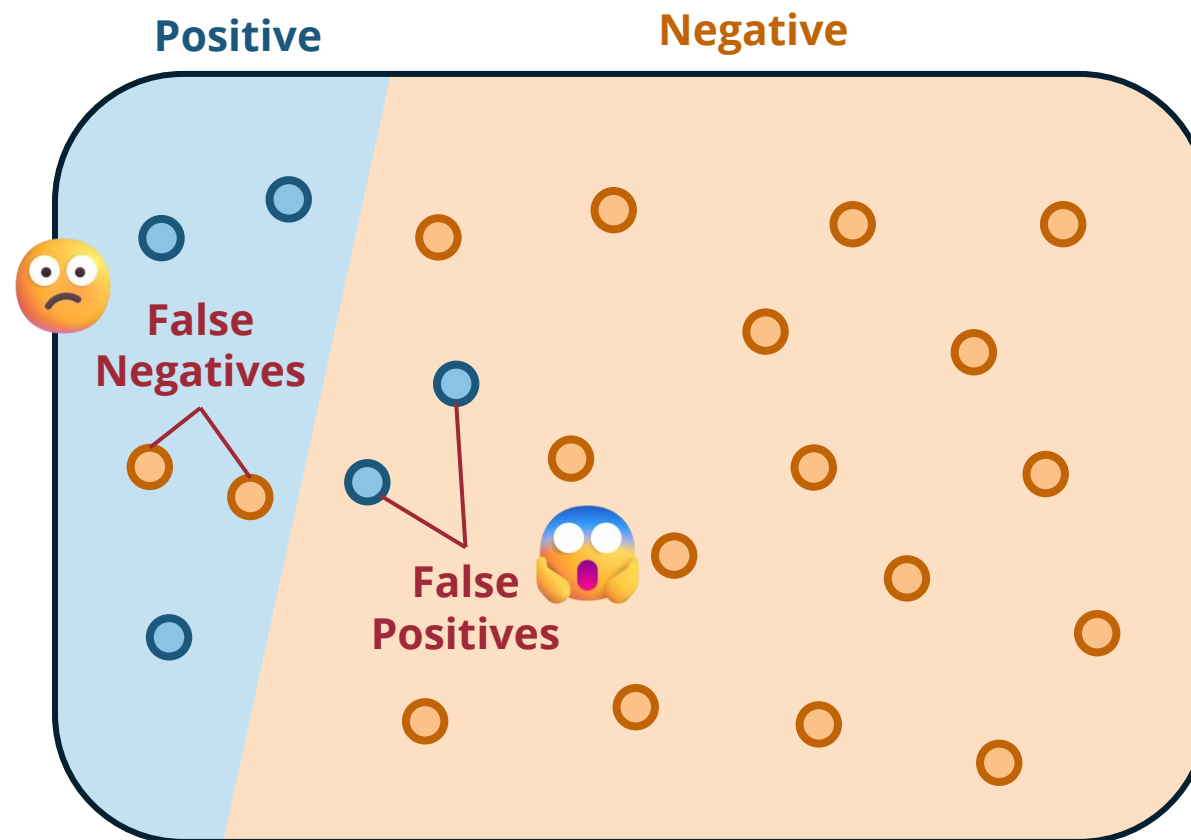
When should we care about Precision & Recall?

Music recommendation



Aim for high precision or high recall?

High precision ensures that the model won't recommend irrelevant items.



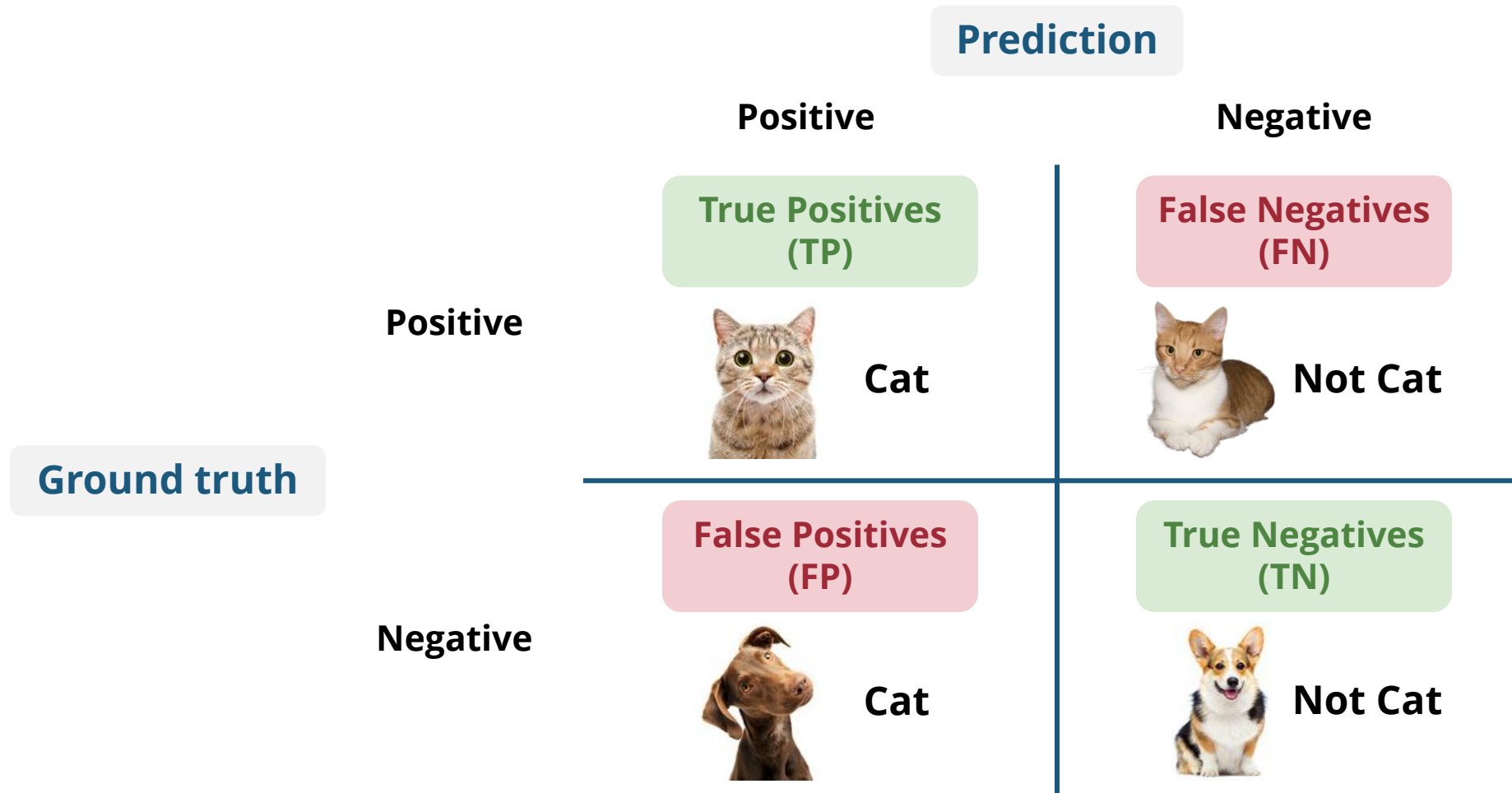
False alarms vs. Missed recommendations

F1 Score: Considering both Precision & Recall

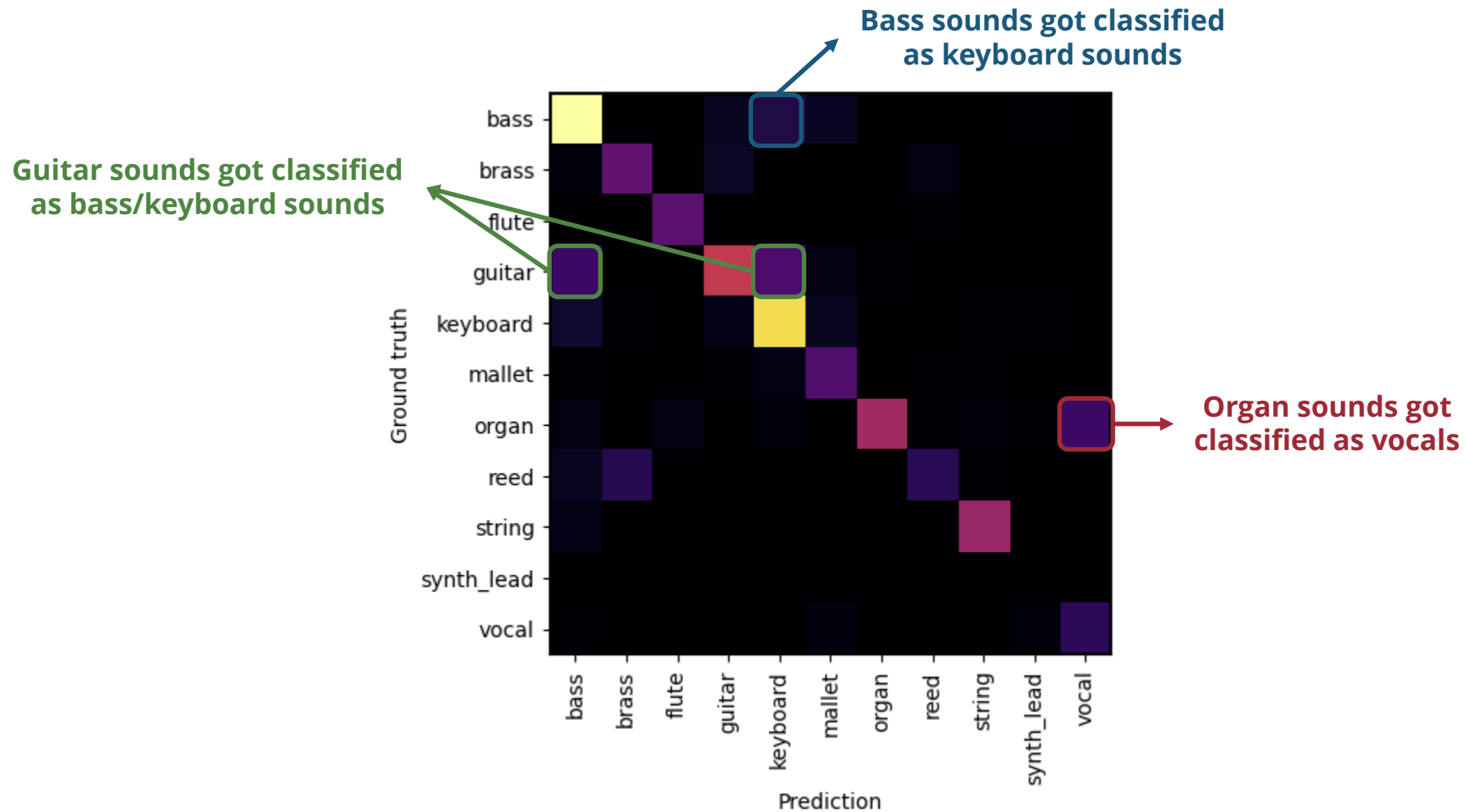
- Particularly useful for **imbalanced datasets**
 - Work better than accuracy when the dataset is imbalanced
 - For example, **music search, retrieval**, and **recommendation**

$$\begin{aligned} F_1 &= \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \\ &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \end{aligned}$$

Confusion Matrix for Binary Classification



Confusion Matrix for Multiclass Classification

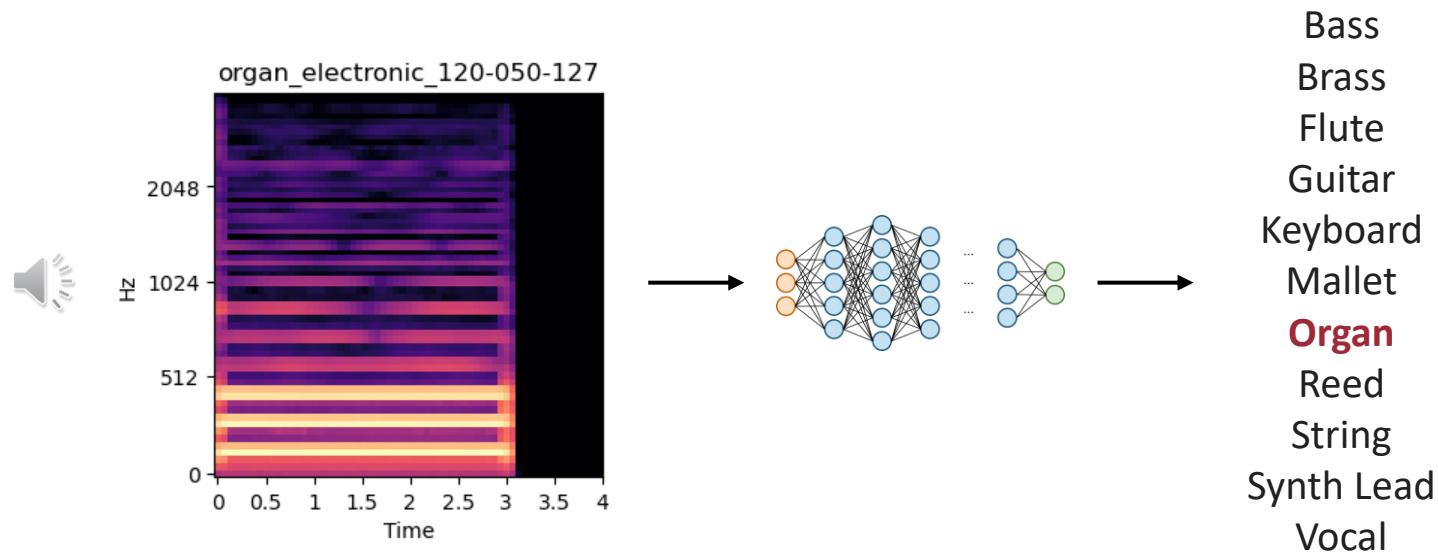


Resources on Music Classification

- Minz Won, Janne Spijkervet, and Keunwoo Choi, "[Music Classification: Beyond Supervised Learning, Towards Real-world Applications](#)," *Tutorials of ISMIR*, 2021.
- Open source music classification models
 - github.com/minzwon/sota-music-tagging-models
 - github.com/jordipons/musicnn

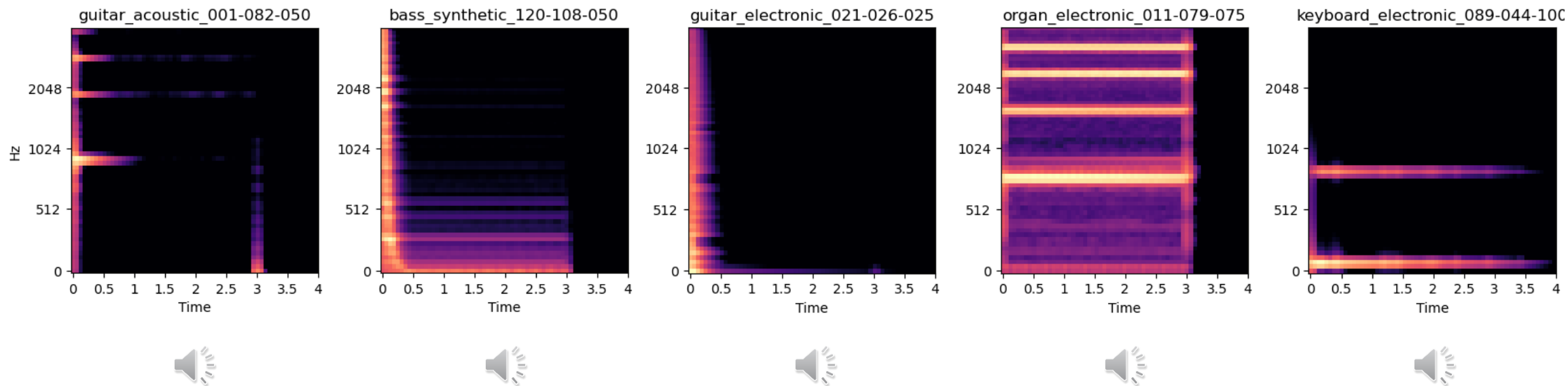
🔥 HW 3: Musical Note Classification using CNNs

- Instructions will be sent by **emails** and released on the **course website**
- Train a CNN that can classify audio files into their **instrument families**
 - **Input:** 64x64 mel spectrogram
 - **Output:** 11 instrument classes



NSynth Dataset

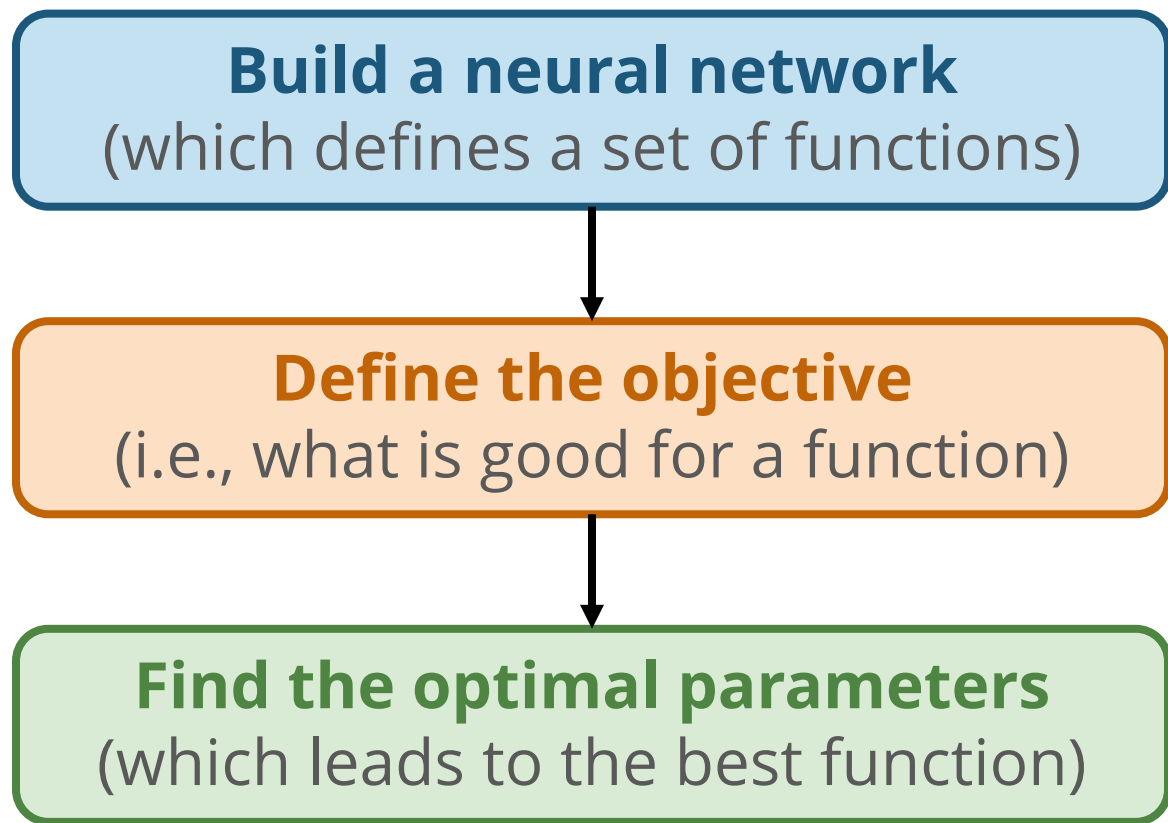
- A collection of 305,979 **one-shot musical notes** (Engel et al., 2017)
 - Produced from 1,006 **commercial sample libraries**
 - With different **MIDI pitches** (21–108) and **velocities** (25, 50, 75, 100, 127)



Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi, “[Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders](#),” *ICML*, 2017.

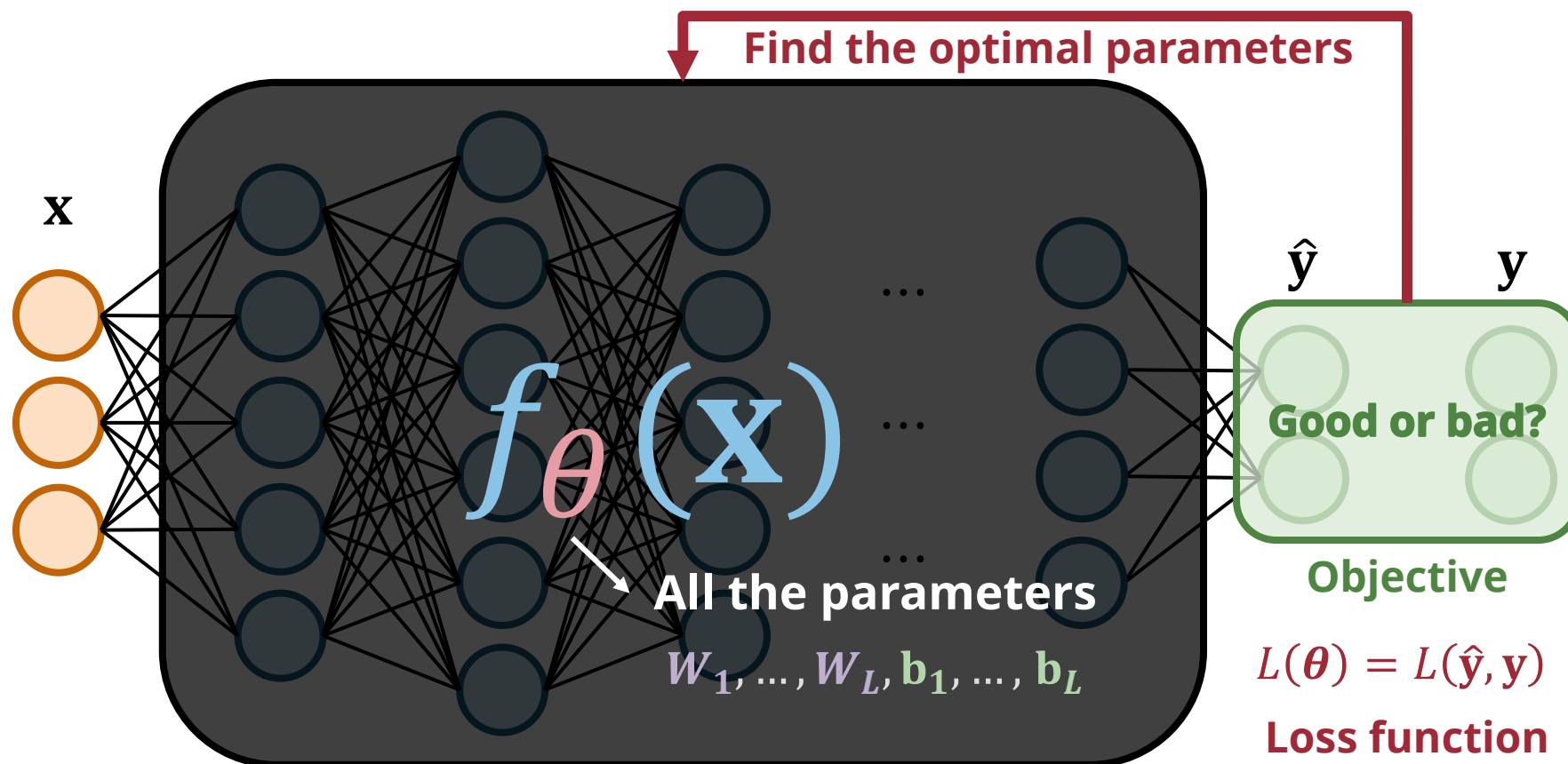
More on Optimization

| Training a Neural Network

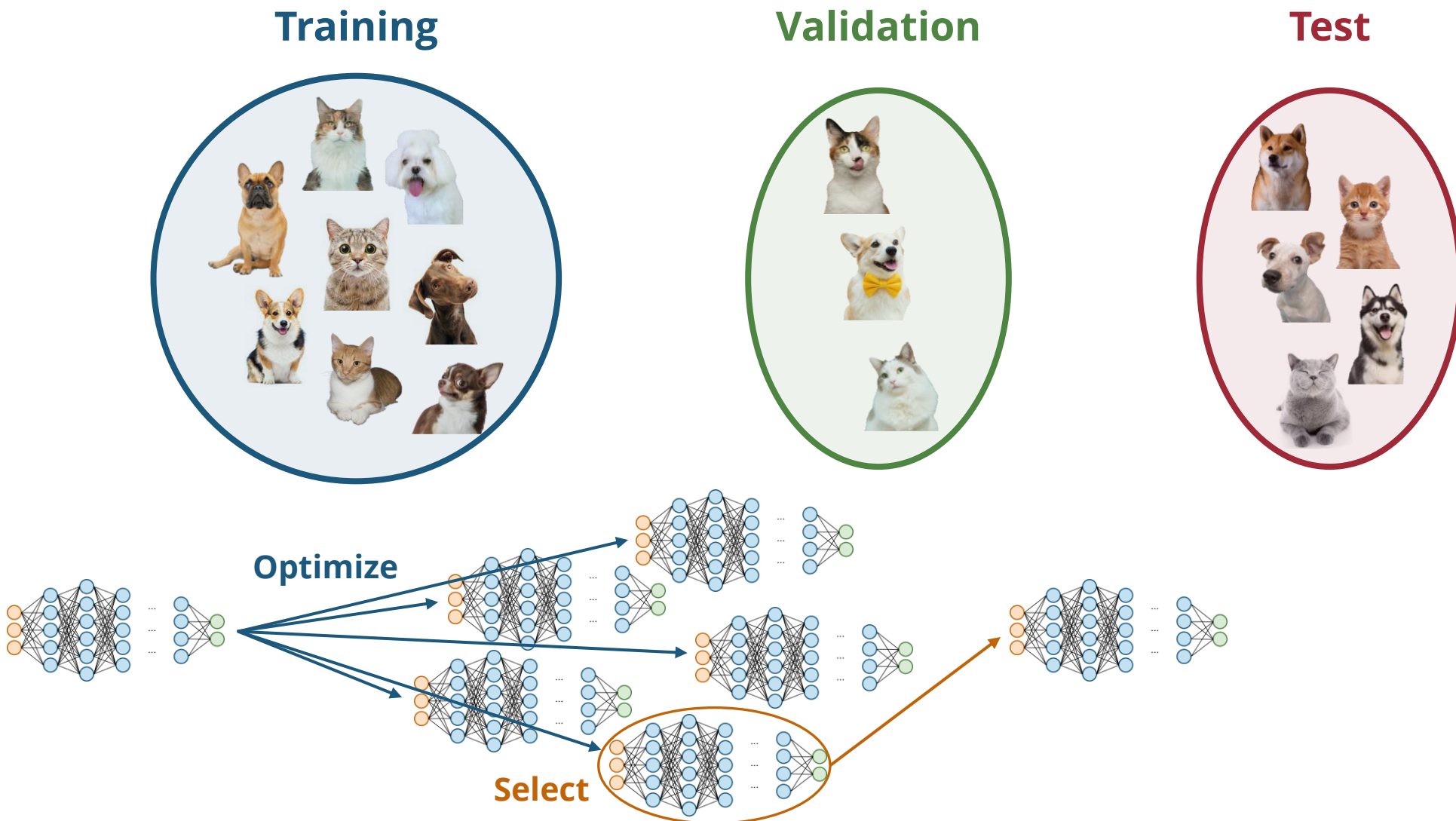


Neural Networks are Parameterized Functions

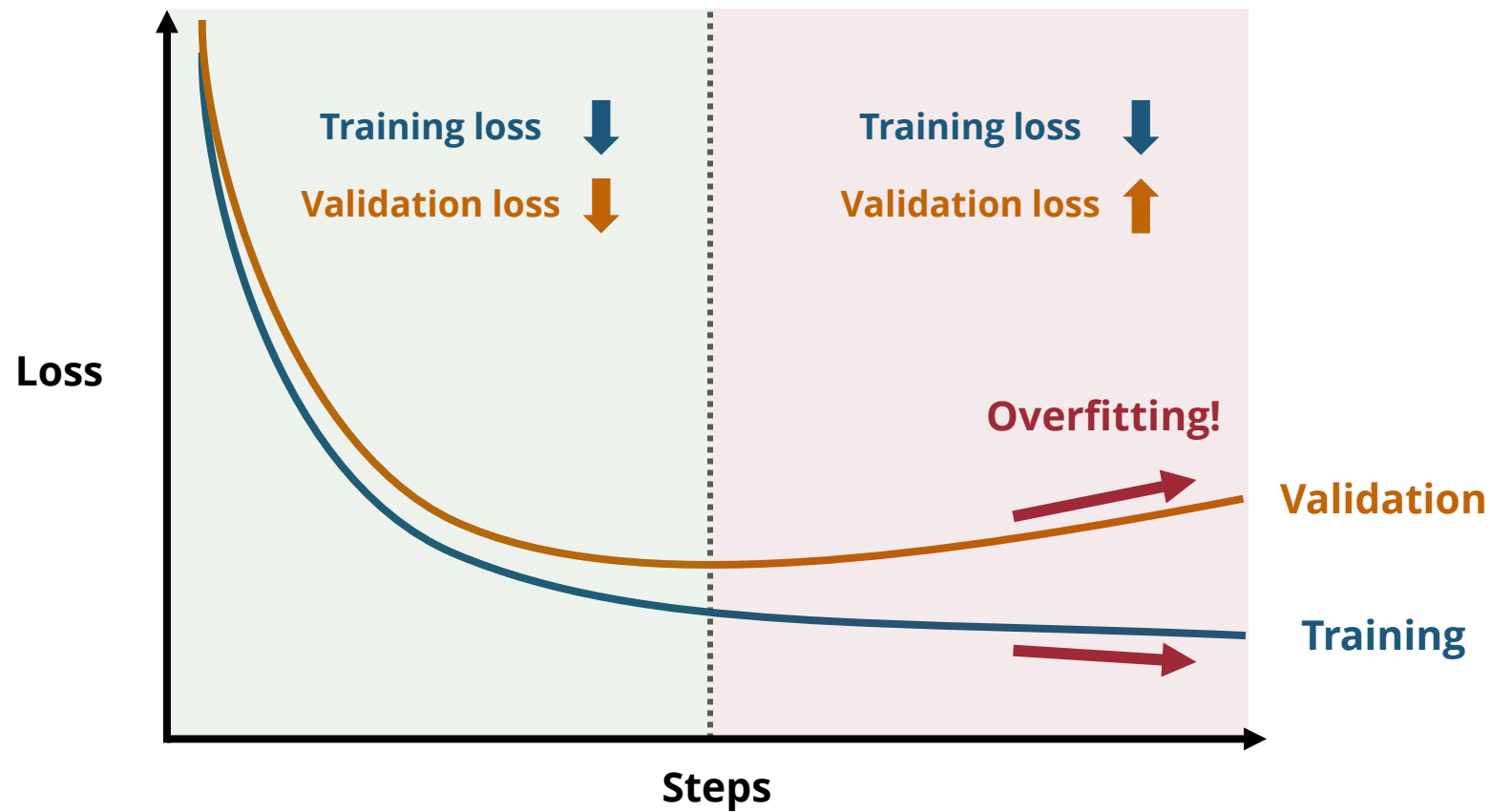
- A neural network represents **a set of functions**



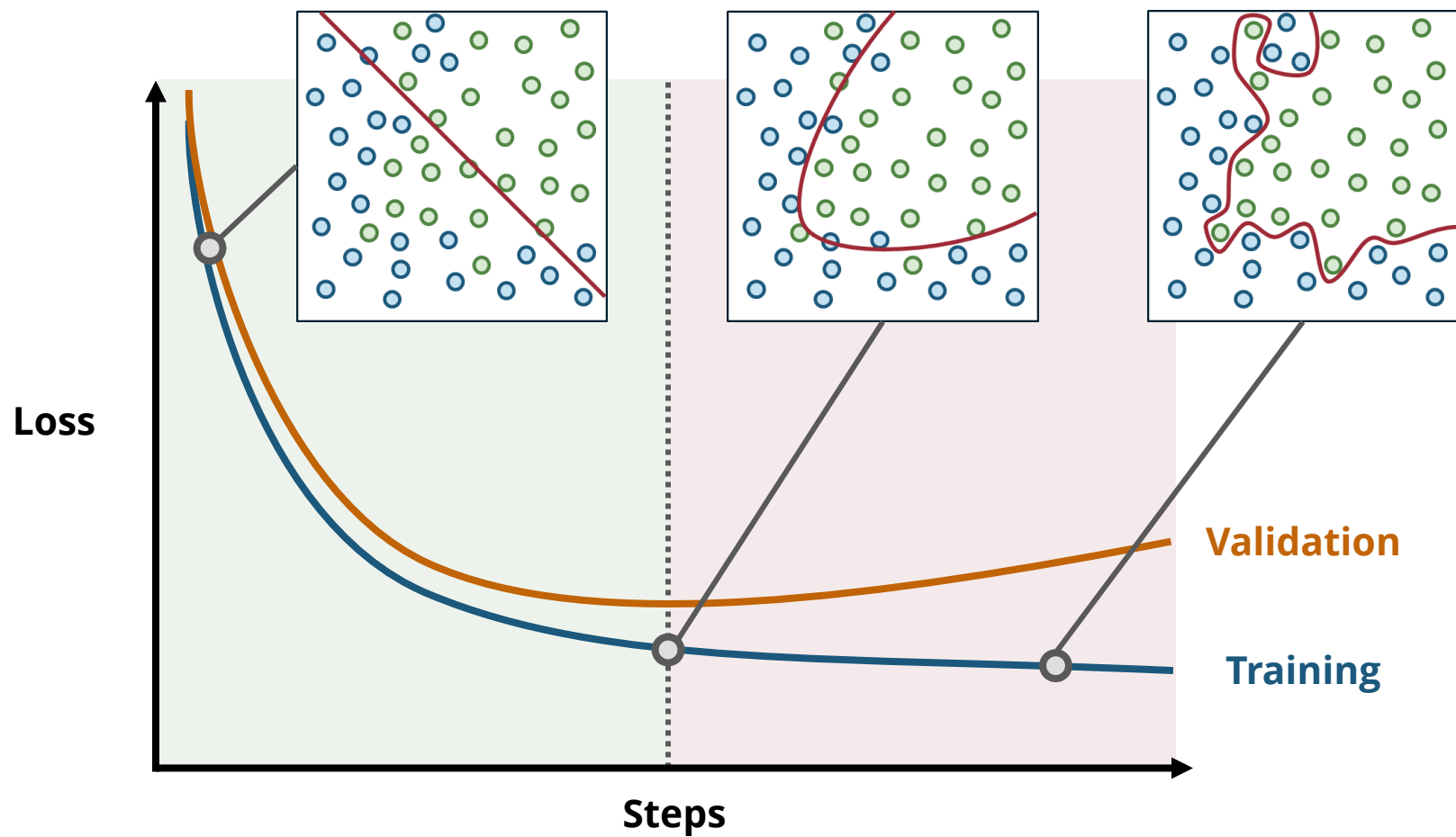
Training-Validation-Test Pipeline



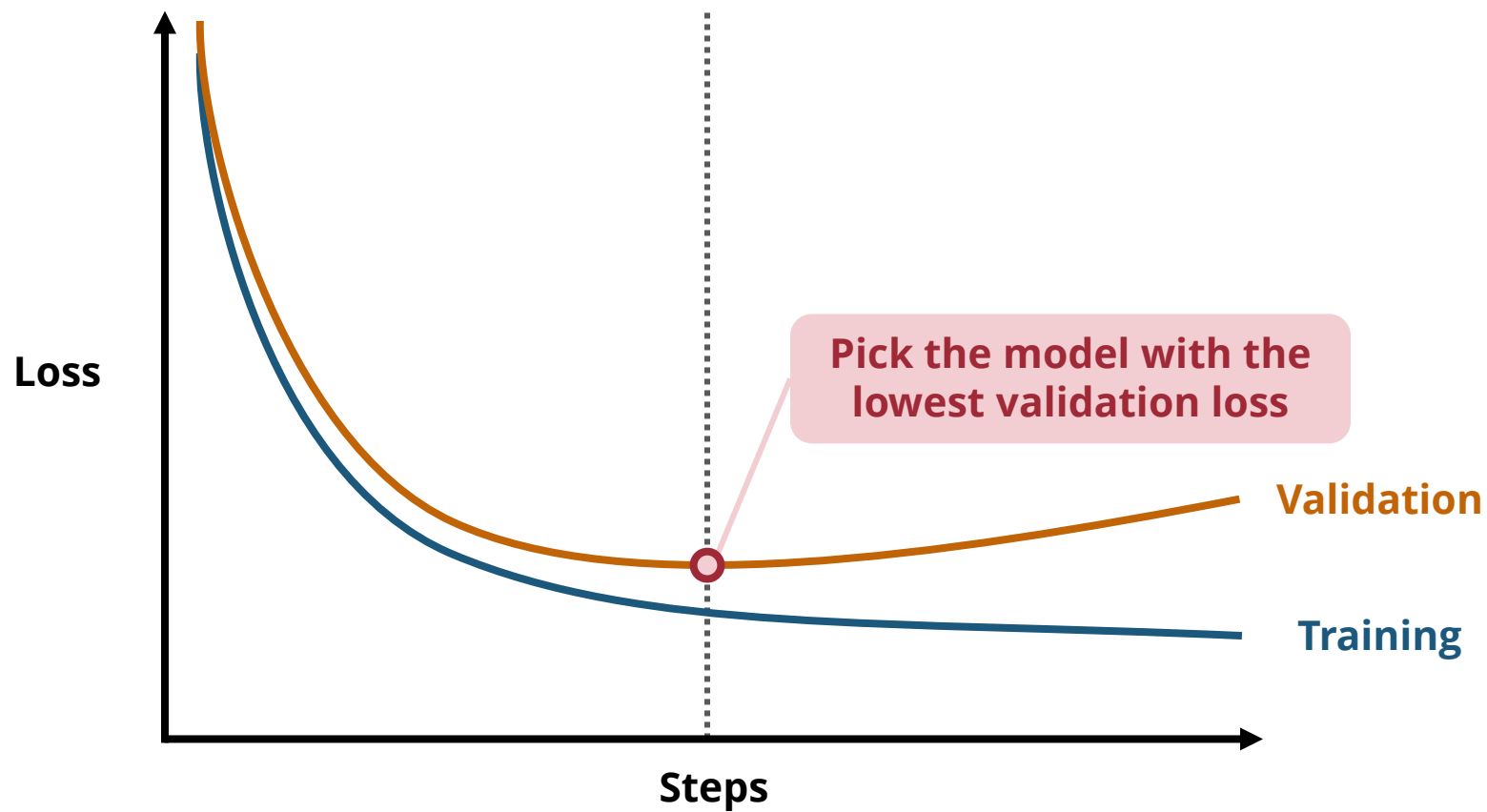
| Training vs. Validation Losses



Training vs. Validation Losses



| Training vs. Validation Losses



| Train–Validation–Test Split

- **Keys**

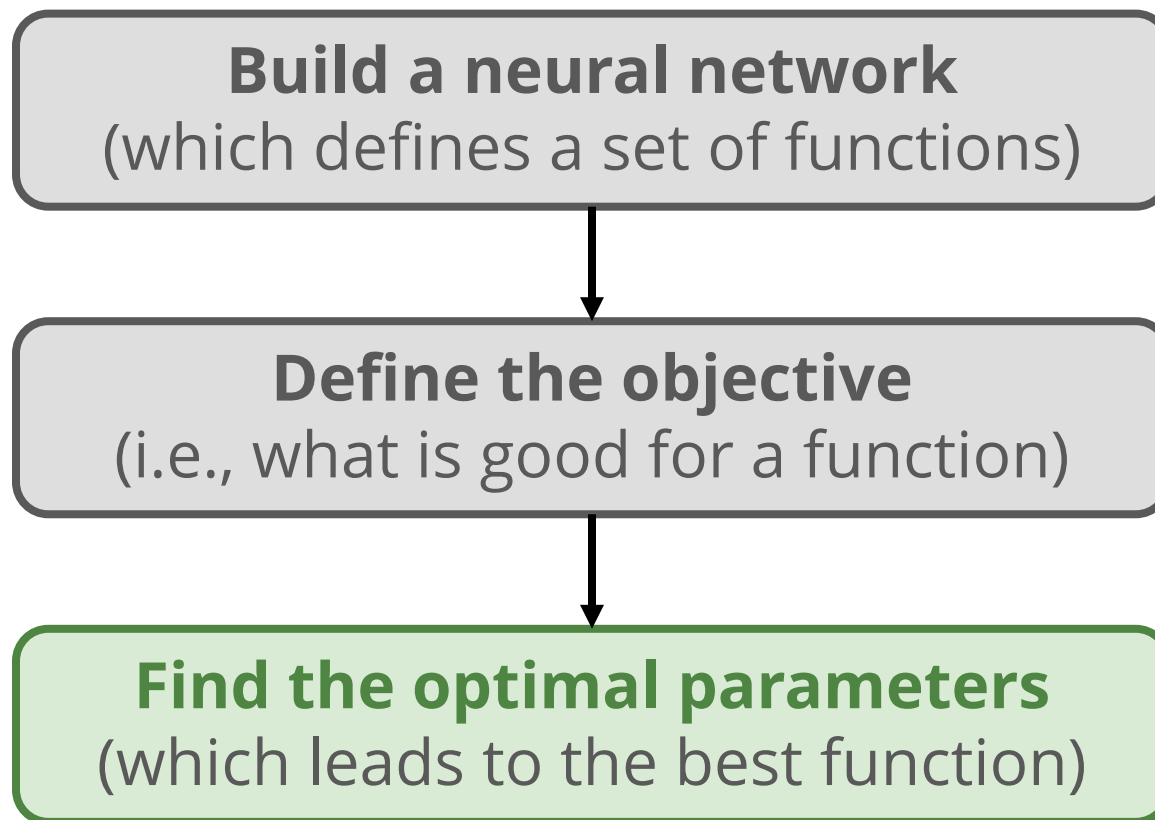
- **Never train or select your model on test samples!**
- Don't over-select your model on the validation set

- What's the **best ratio**?

- Most common: **8:1:1** or 9:0.5:0.5
- For smaller dataset, you might even want 6:2:2

Adaptive Optimizers

| Training a Neural Network

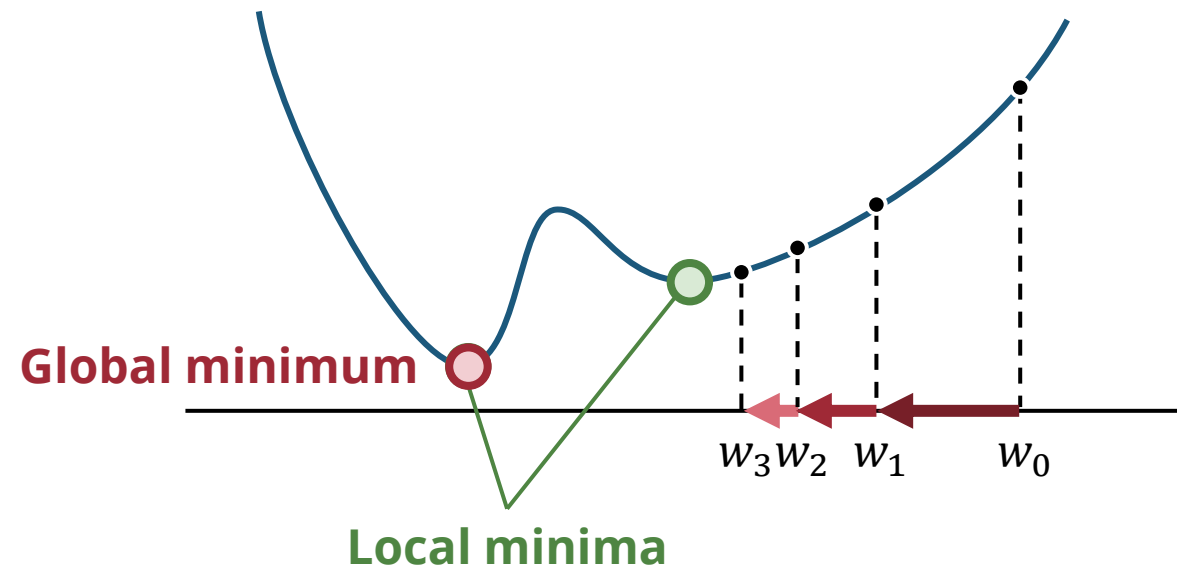


$$\hat{\mathbf{y}} = f_{\boldsymbol{\theta}}(\mathbf{x})$$

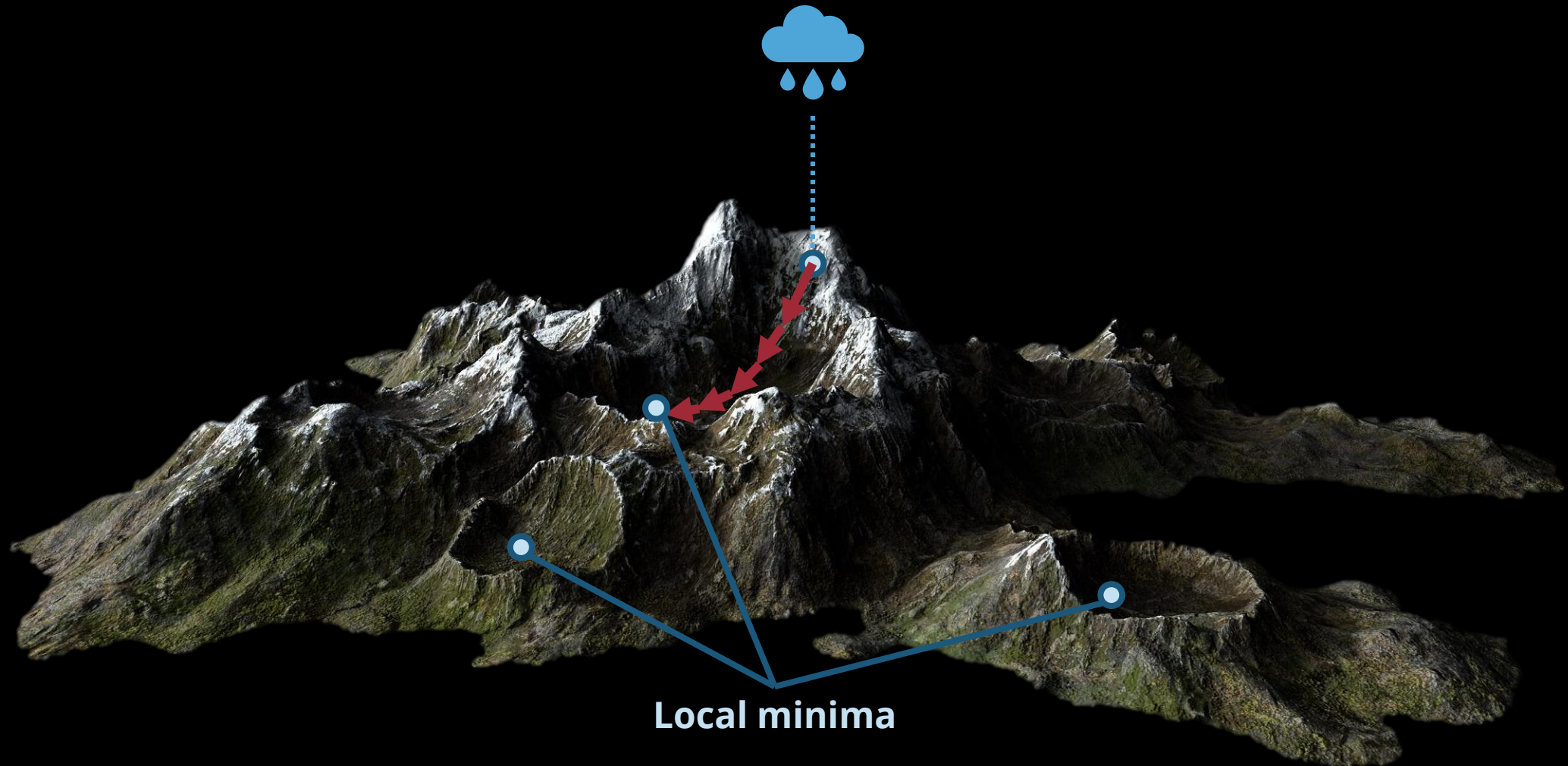
$$Loss(\boldsymbol{\theta}) = \sum_k^N L(\hat{\mathbf{y}}_k, \mathbf{y}_k)$$

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

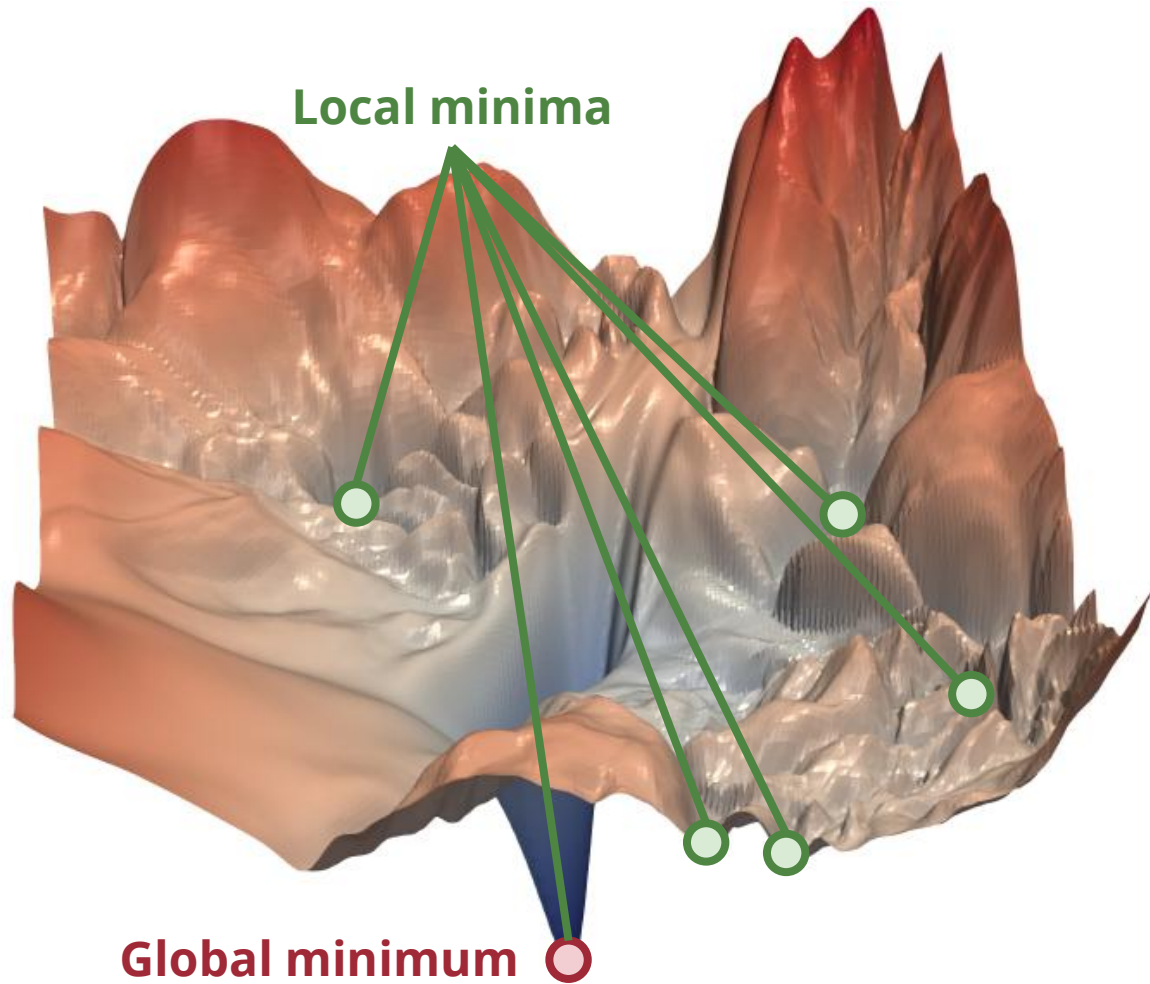
Gradient Descent Finds a Local Minimum



| Gradient Descent Finds a Local Minimum



Local Minima in Complex Loss Landscape



Solution 1

Use an optimizer with adaptive learning rate

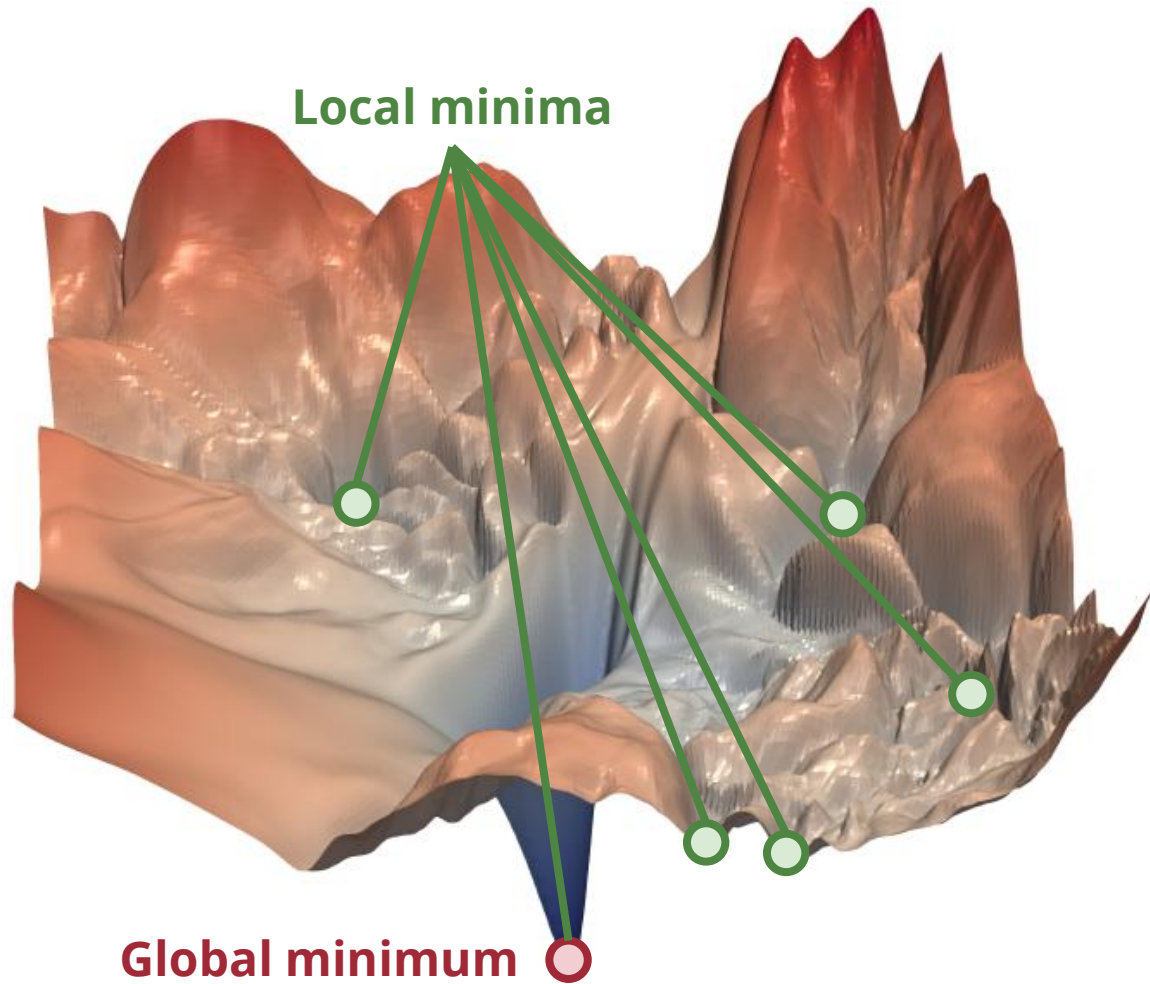
Solution 2

Use a stochastic optimizer

Solution 3

Make the loss landscape smoother

Local Minima in Complex Loss Landscape

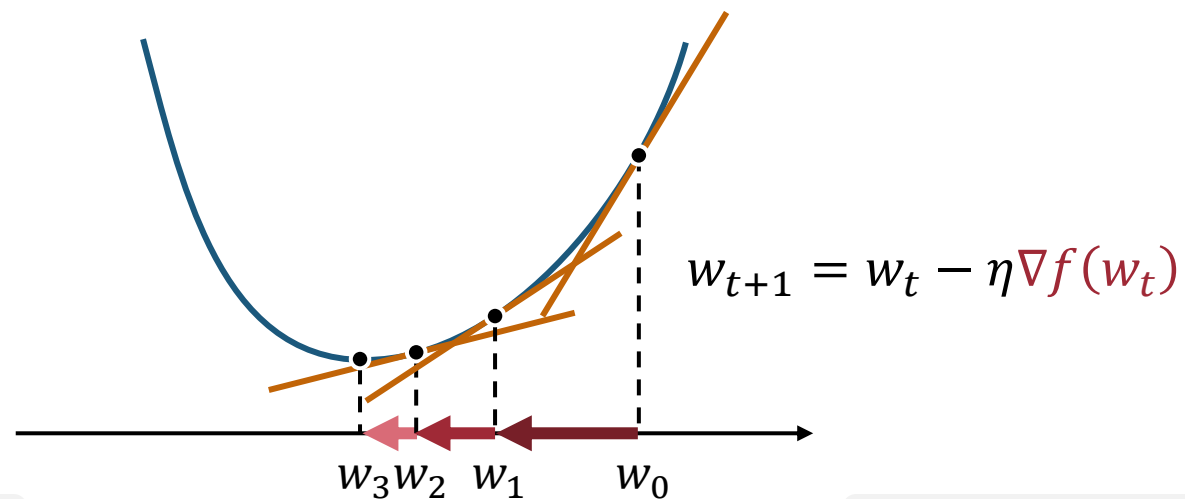


Solution 1
Use an optimizer with
adaptive learning rate

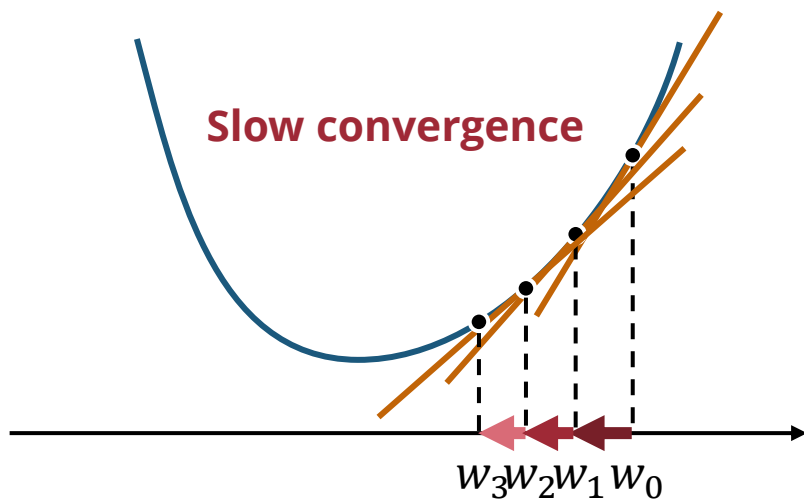
Solution 2
Use a stochastic
optimizer

Solution 3
Make the loss
landscape smoother

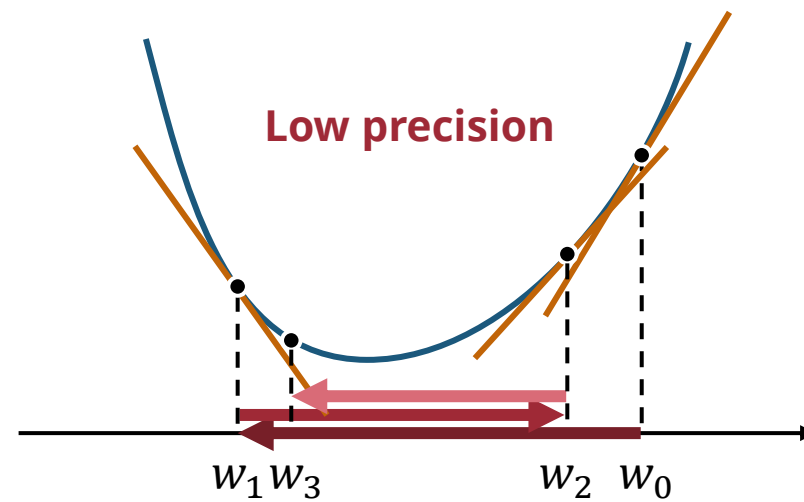
Learning Rate in Gradient Descent



Smaller learning rate

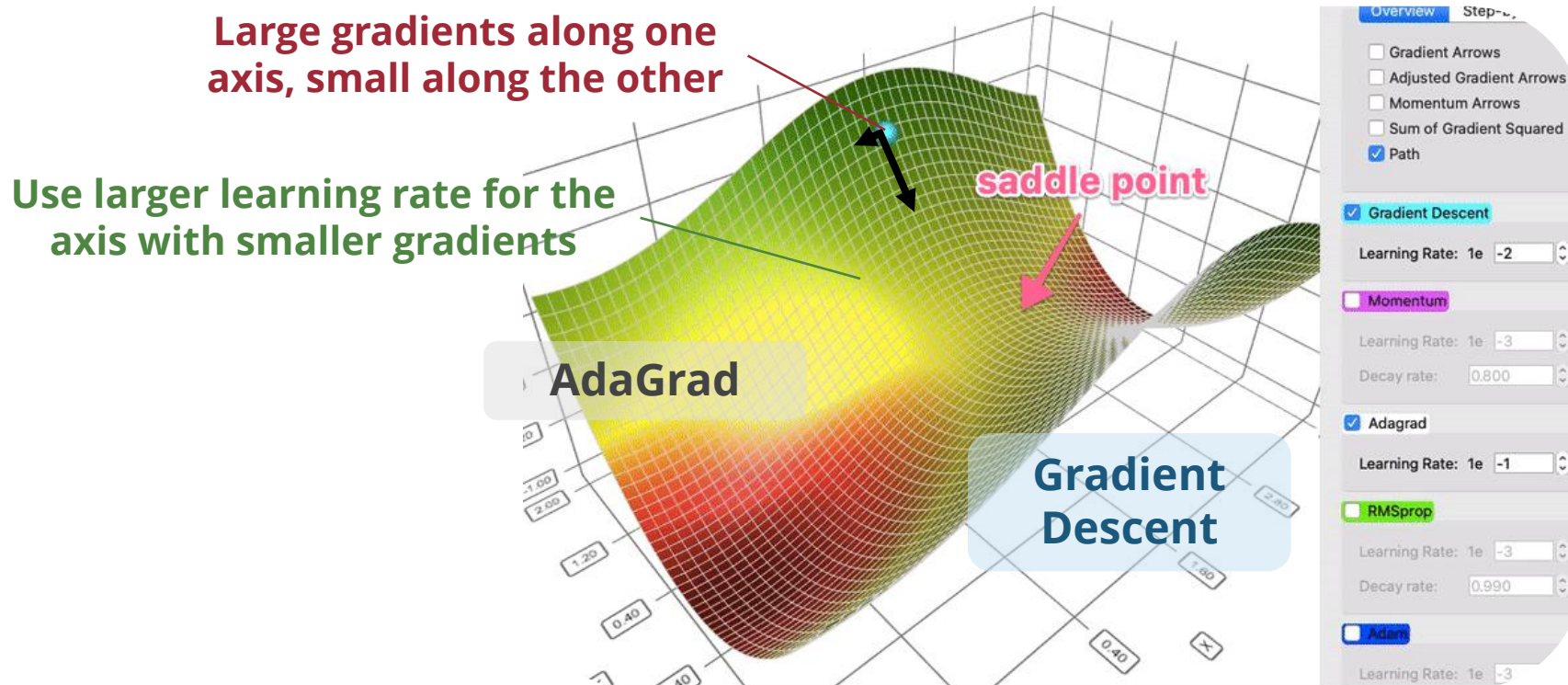


Larger learning rate



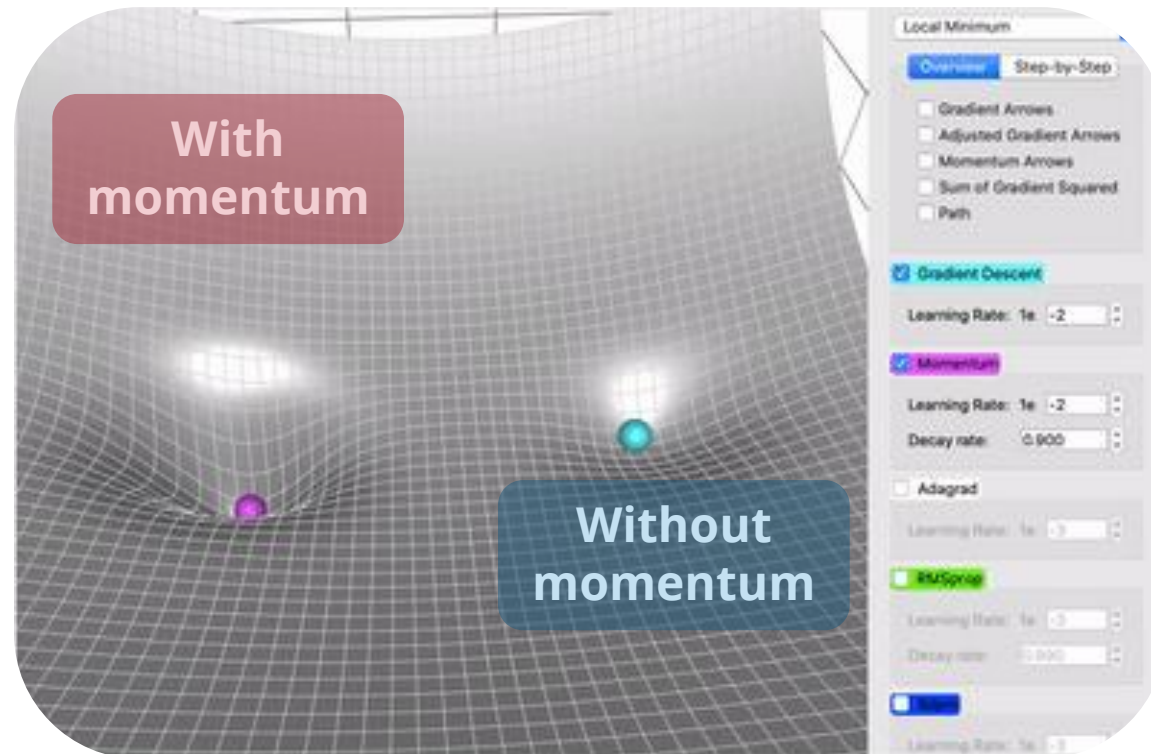
Gradient-based Adaptive Learning Rate

- **Intuition: Compensate axis that has little progress** by comparing the current gradients to the previous gradients

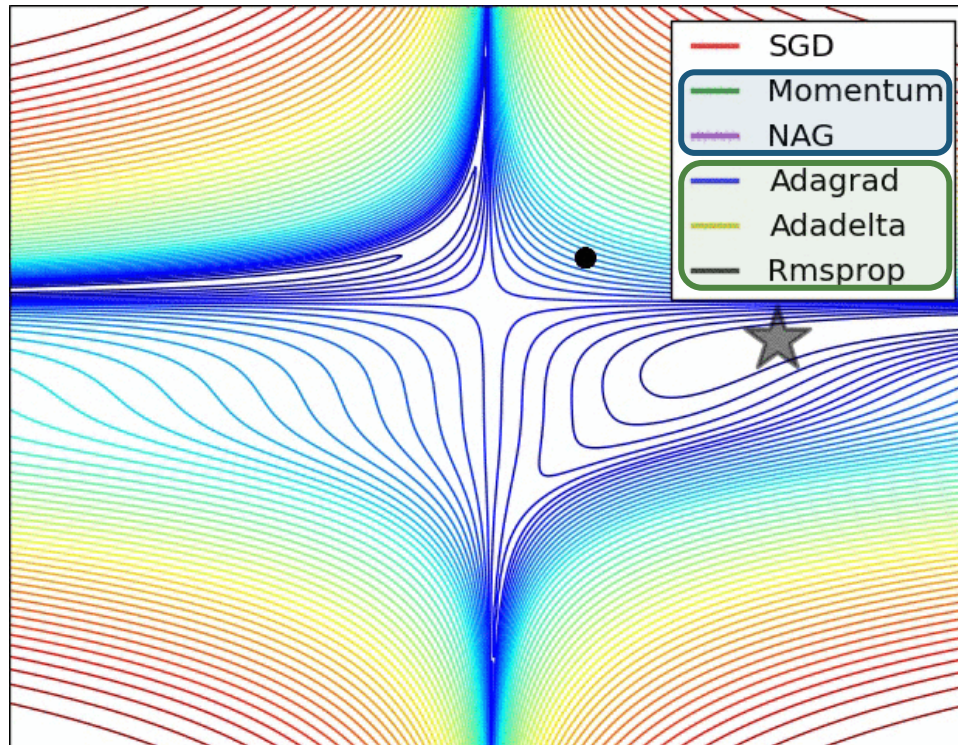


Momentum

- **Intuition:** Maintain the momentum to **escape from local minima**

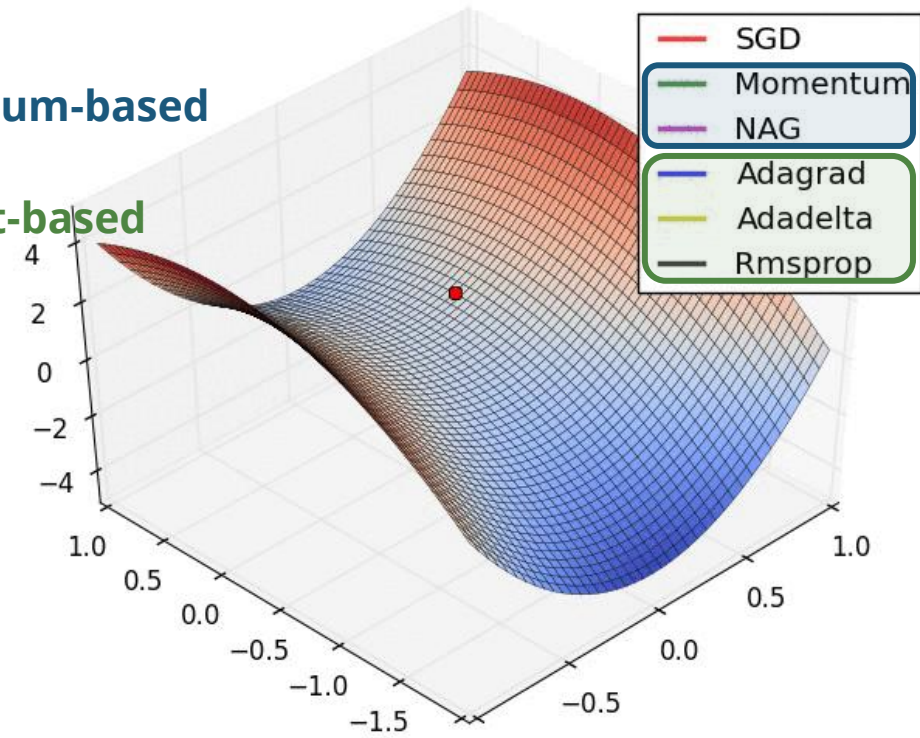


Comparison of Optimizers



Momentum-based

Gradient-based



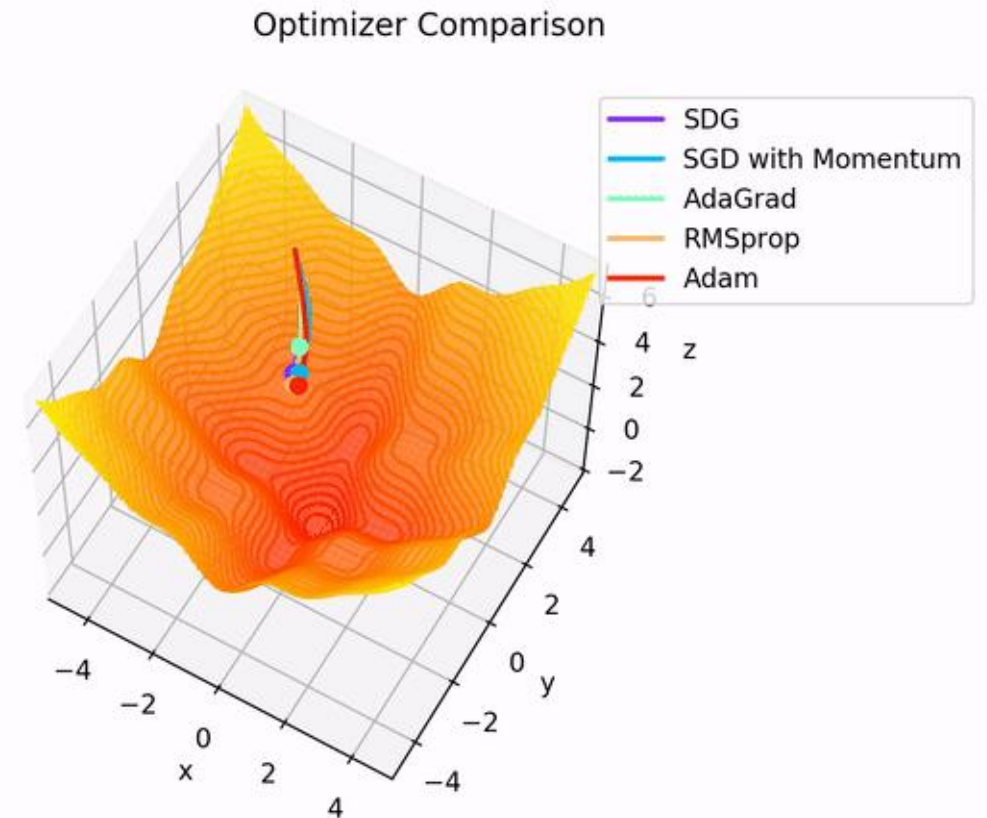
Comparison of Optimizers

- **Momentum**

- Gets you out of spurious local minima
- Allows the model to explore around

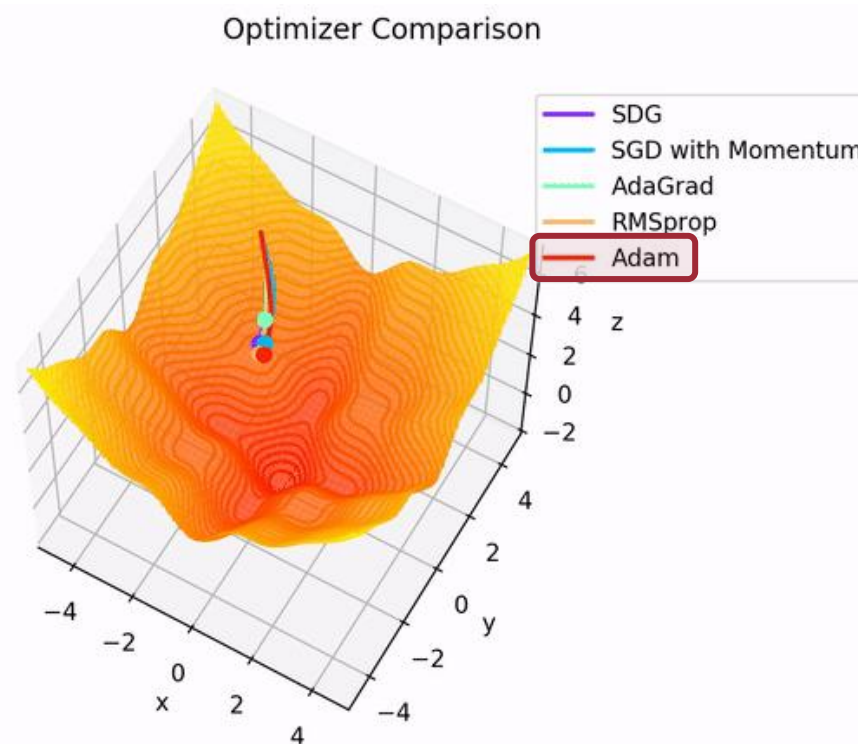
- **Gradient-based adaption**

- Maintains steady improvement
- Allows faster convergence

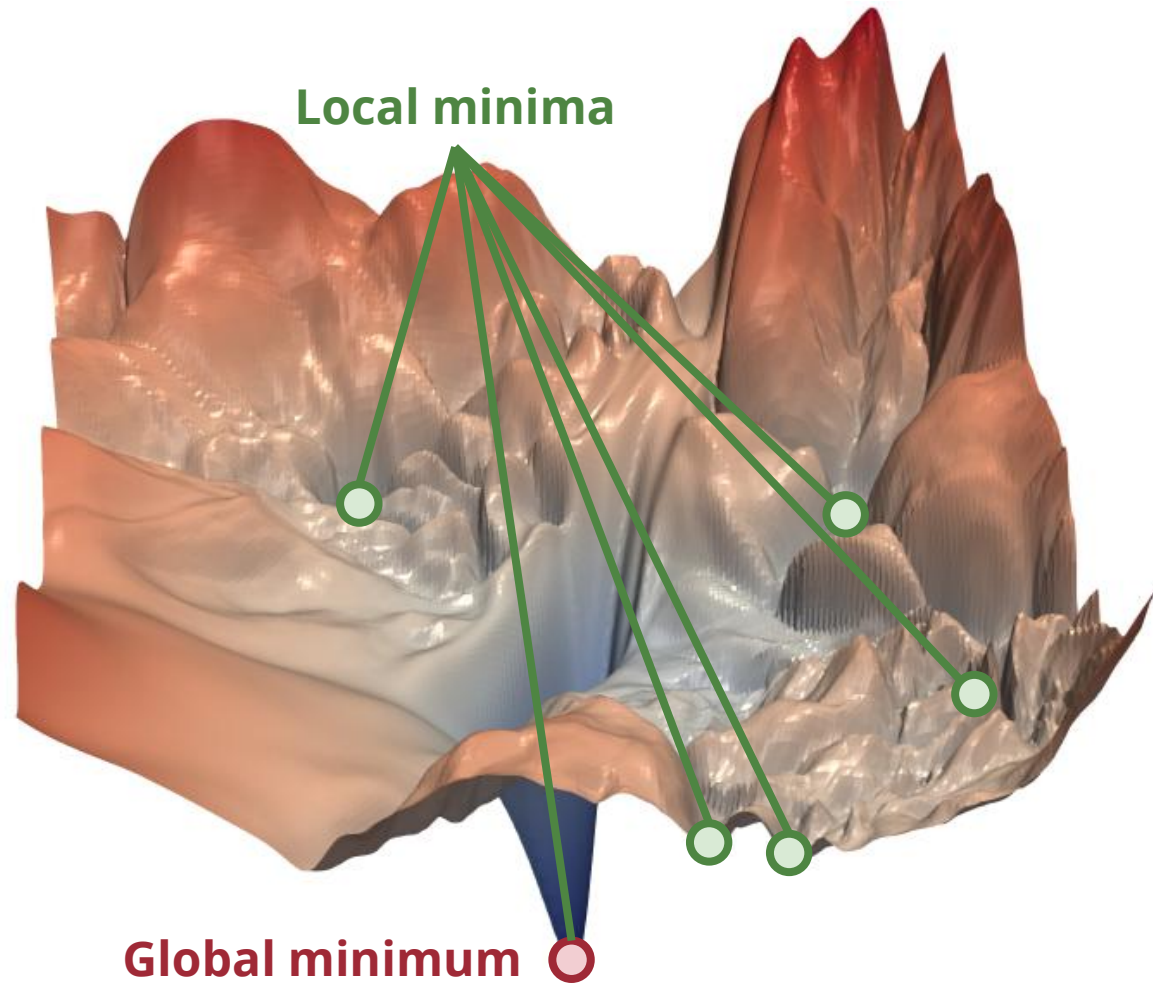


| Adam Optimizer

- Combine the idea of **adaptive learning rate** and **momentum**
- Work **empirically well** in complex neural network
- The **go-to choice** for most cases



Local Minima in Complex Loss Landscape



Solution 1
Use an optimizer with
adaptive learning rate

Solution 2
Use a stochastic
optimizer

Solution 3
Make the loss
landscape smoother

Batch Gradient Descent

- How to aggregate the gradients obtained from different training samples?
- Batch gradient descent computes the mean gradients **over the whole training set**

MSE loss

$$Loss(\boldsymbol{\theta}) = \sum_k^N L(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{n} \sum_k^N \sum_i^n \left(\hat{y}_i^{(k)} - y_i^{(k)} \right)^2$$

Binary cross entropy

$$Loss(\boldsymbol{\theta}) = \sum_k^N L(\hat{\mathbf{y}}, \mathbf{y}) = \sum_k^N -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Cross entropy

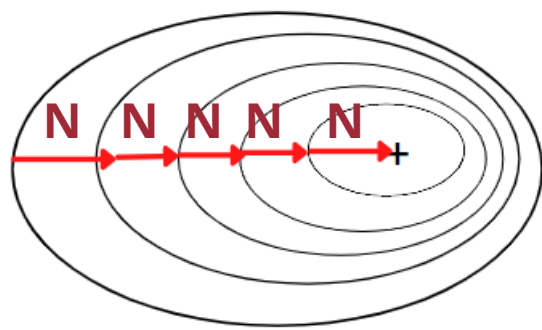
$$Loss(\boldsymbol{\theta}) = \sum_k^N L(\hat{\mathbf{y}}, \mathbf{y}) = - \sum_k^N \sum_i^n y_i \log \hat{y}_i$$

Stochastic Gradient Descent (SGD)

- **Intuition:** **Estimate** the gradient using **one random training sample**
- Benefits
 - **Speed up** the computation of the gradient **N computations $\rightarrow$ 1 computation**
 - Add some **randomness** to the gradient descent algorithm **Help escape spurious local minima**

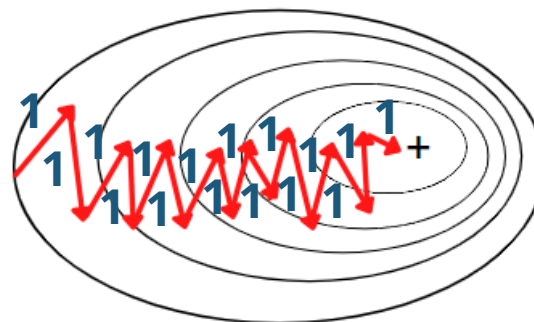
Gradient descent

5N gradient
computations



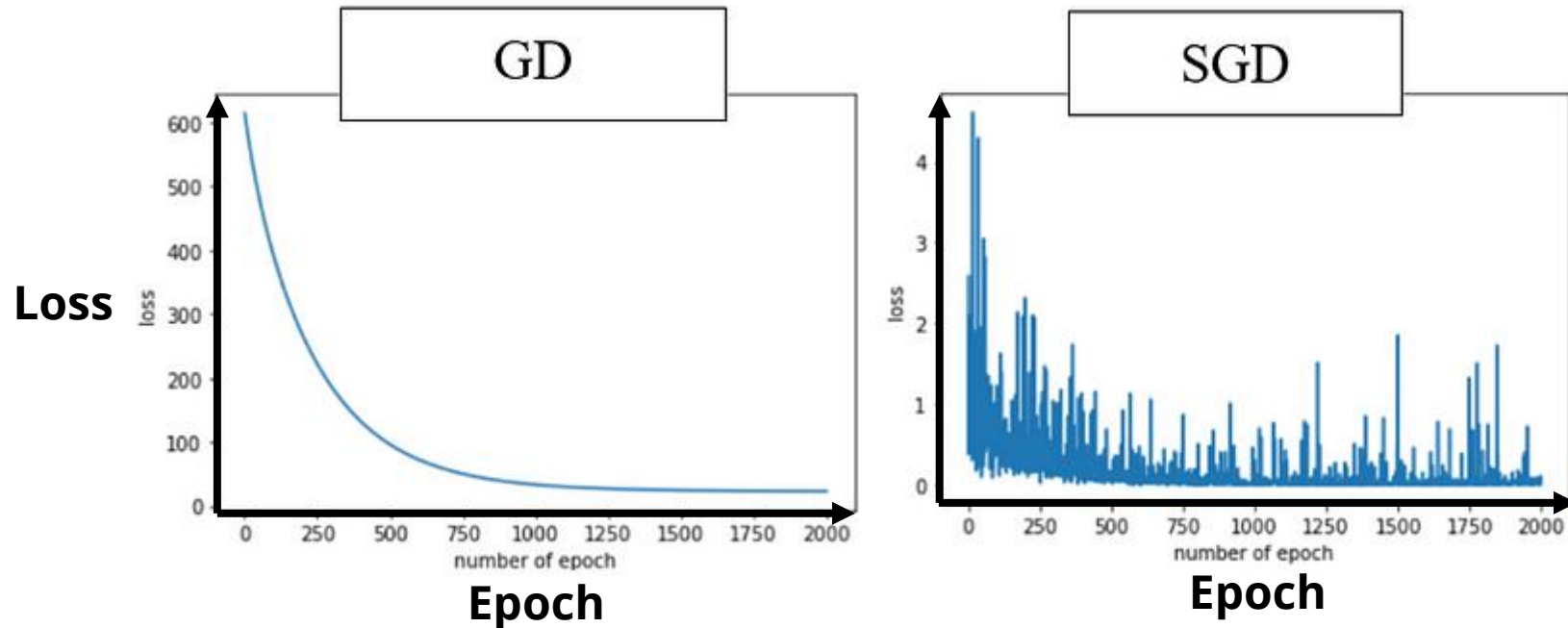
Stochastic gradient descent

16 gradient
computations



Stochastic Gradient Descent is Noisy and Unstable

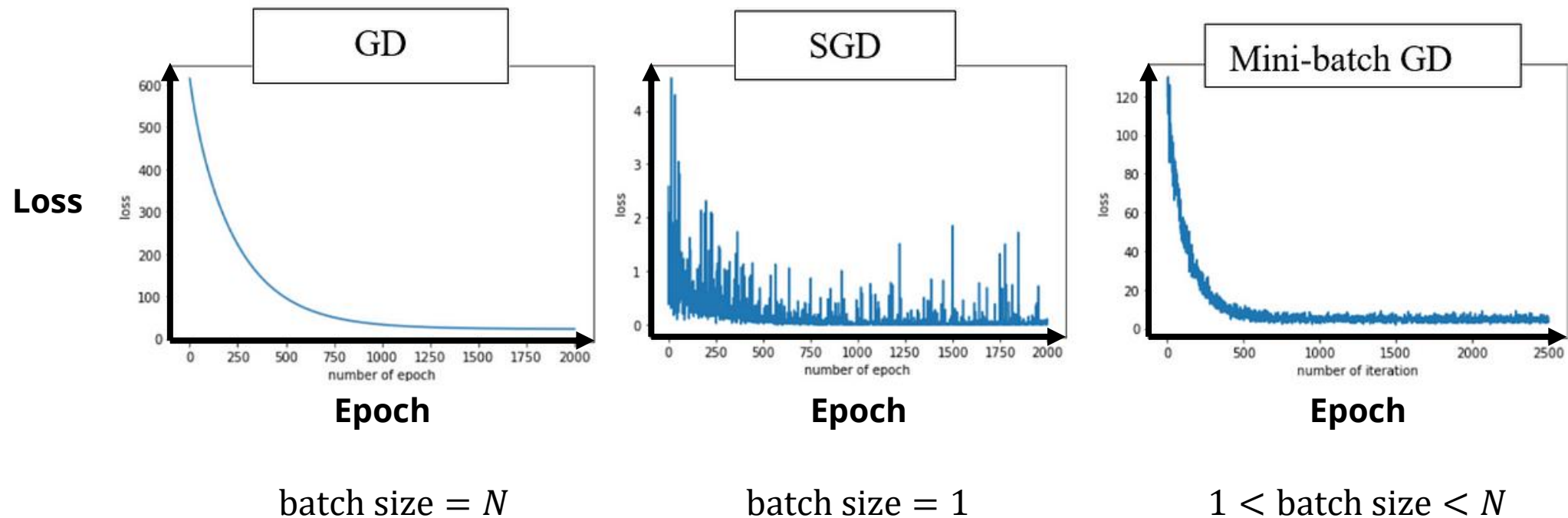
- Gradient estimate using one single sample can be unreliable



How about we use more samples to estimate the gradient?

Mini-batch Gradient Descent

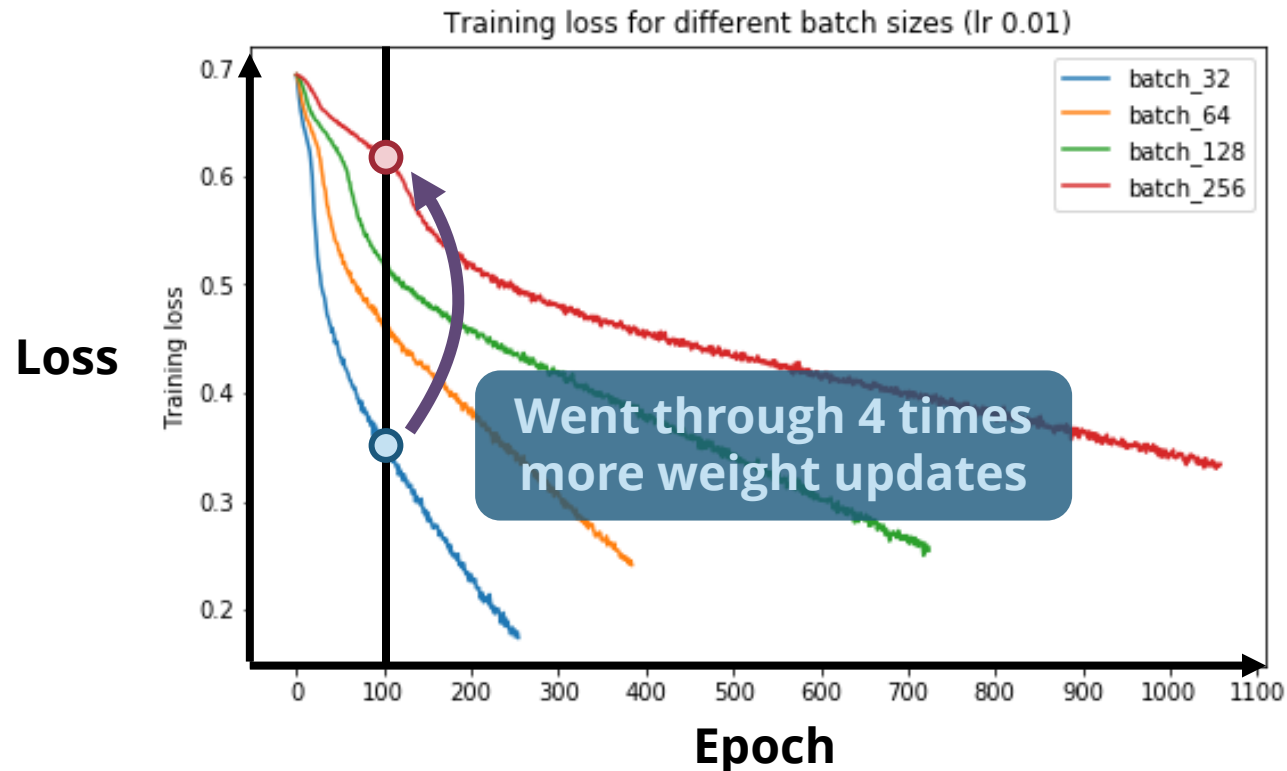
- **Intuition:** **Estimate** the gradient using **several random training samples**



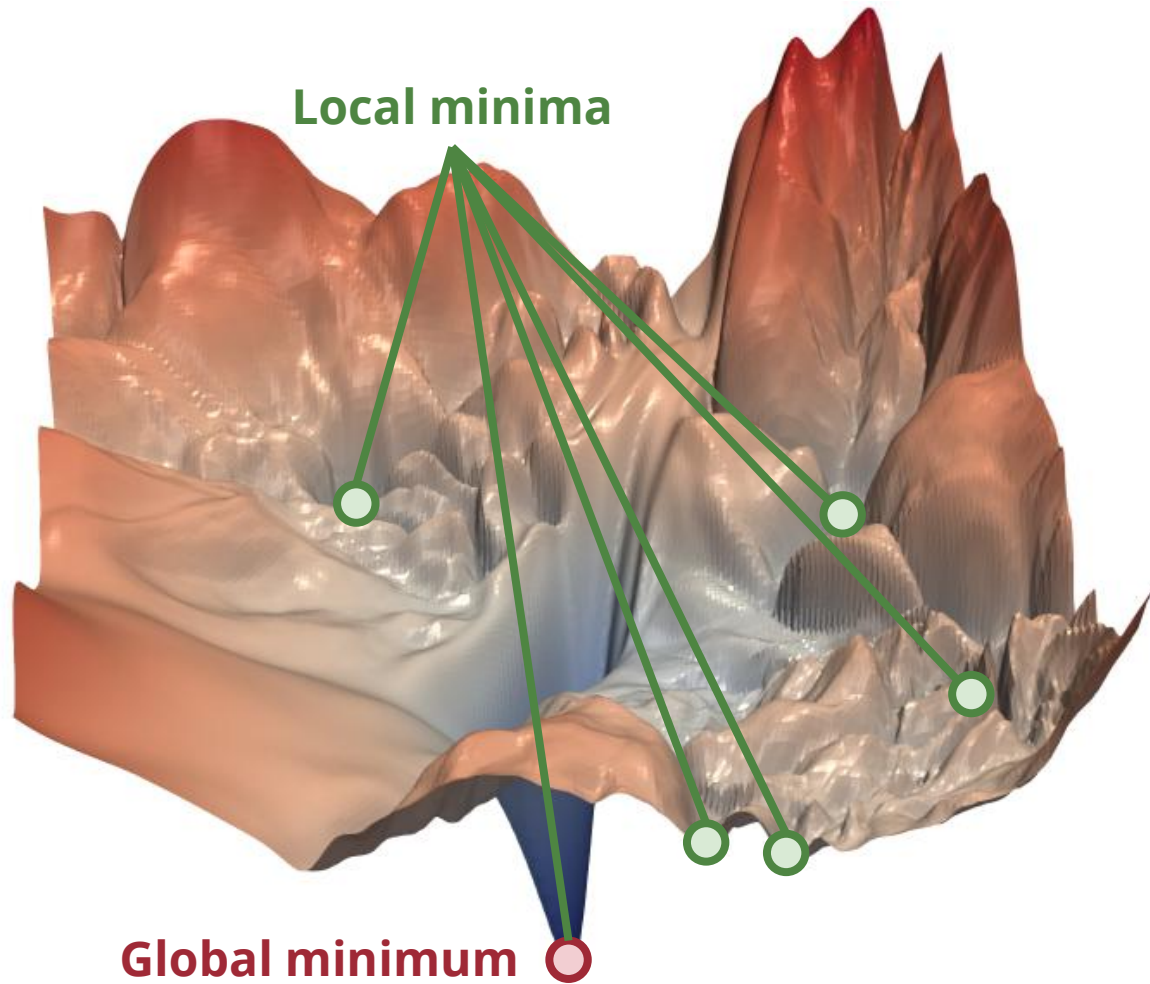
Effects of Batch Size

- An **epoch** is a full run of the whole dataset
- Steps per epoch depends on the batch size

$$\#(\text{steps}) = \frac{\#(\text{training samples})}{\text{batch size}}$$



Local Minima in Complex Loss Landscape



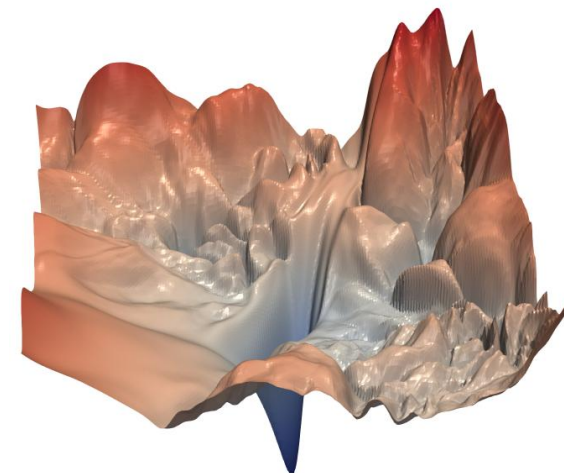
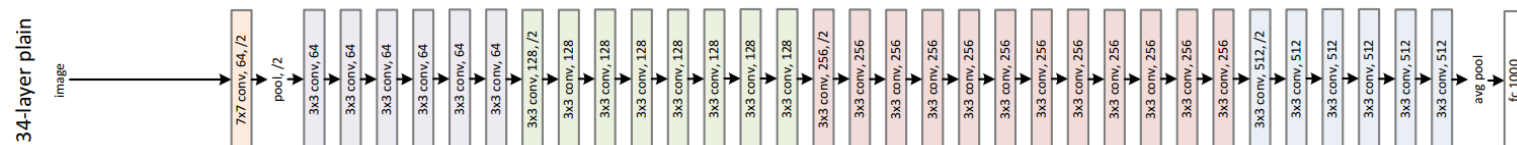
Solution 1
Use an optimizer with
adaptive learning rate

Solution 2
Use a stochastic
optimizer

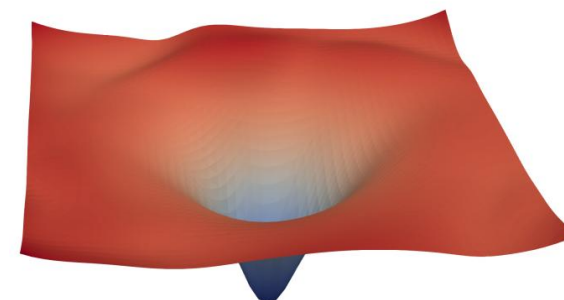
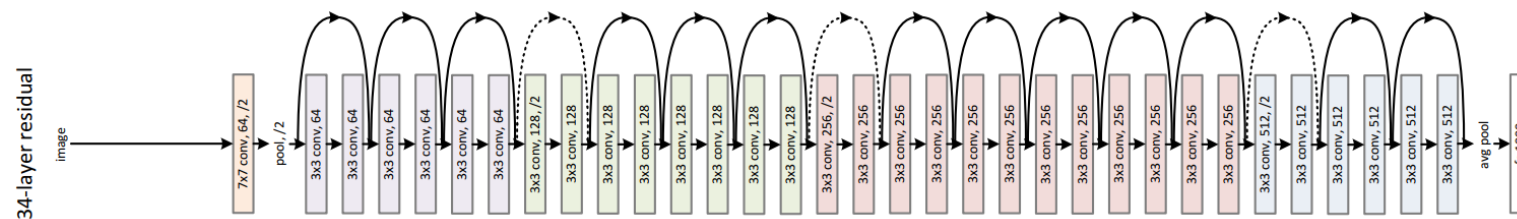
Solution 3
Make the loss
landscape smoother

Skip Connections

Without skip connections



With skip connections



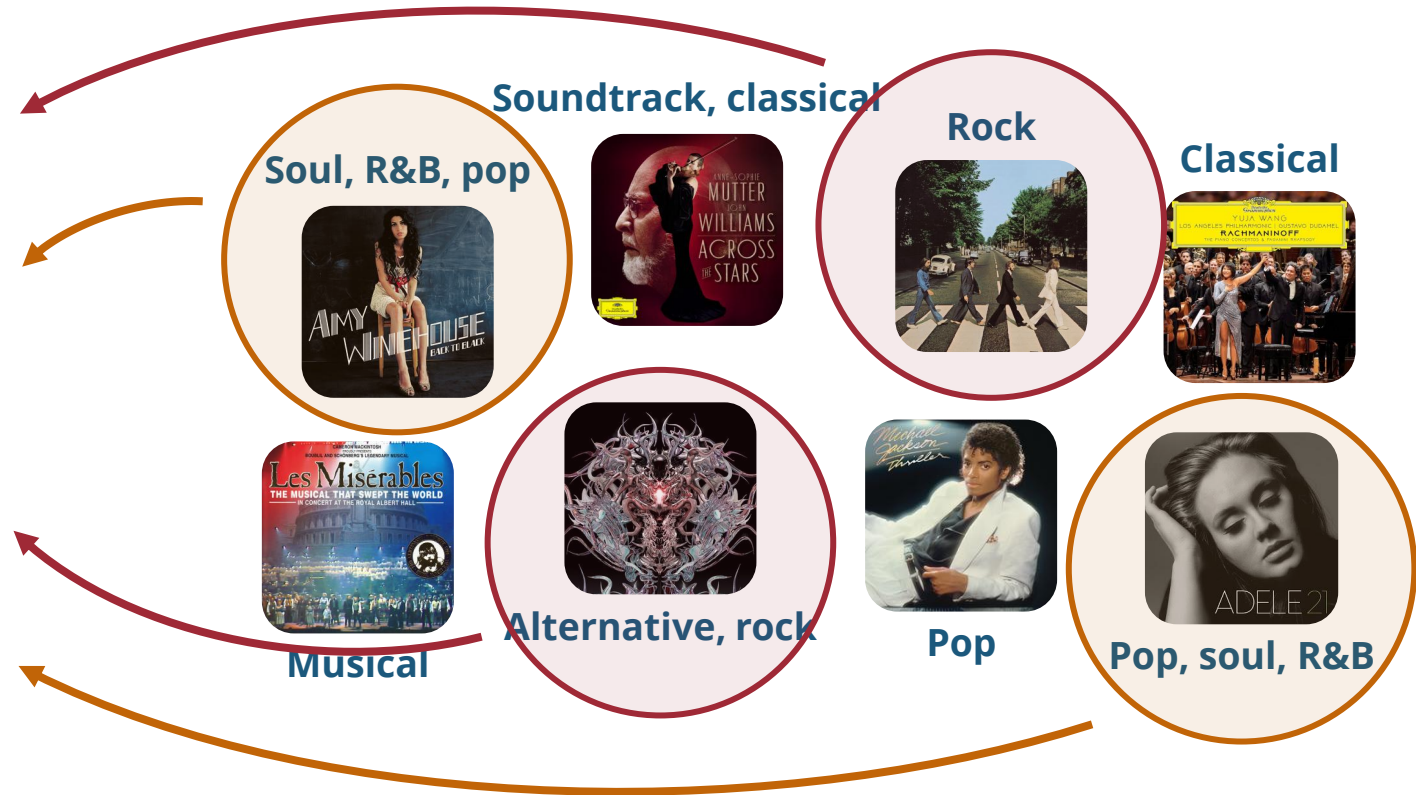
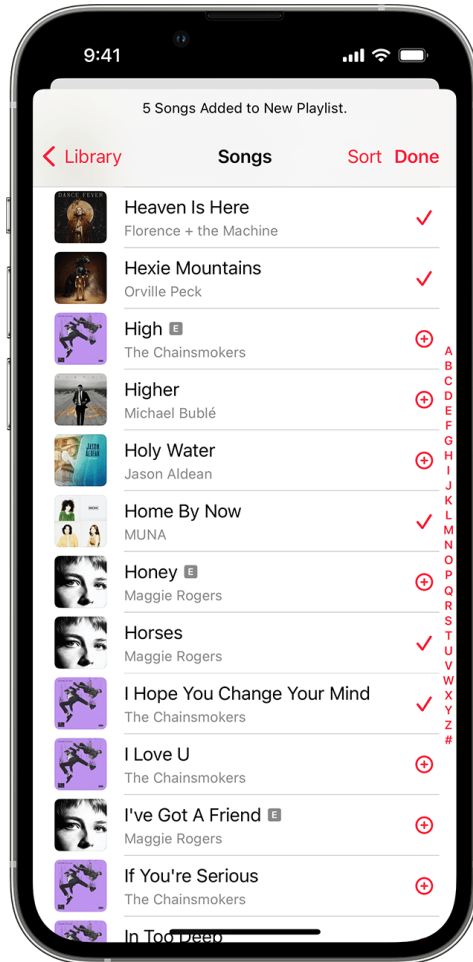
Recap

Music Classification for Recommendation

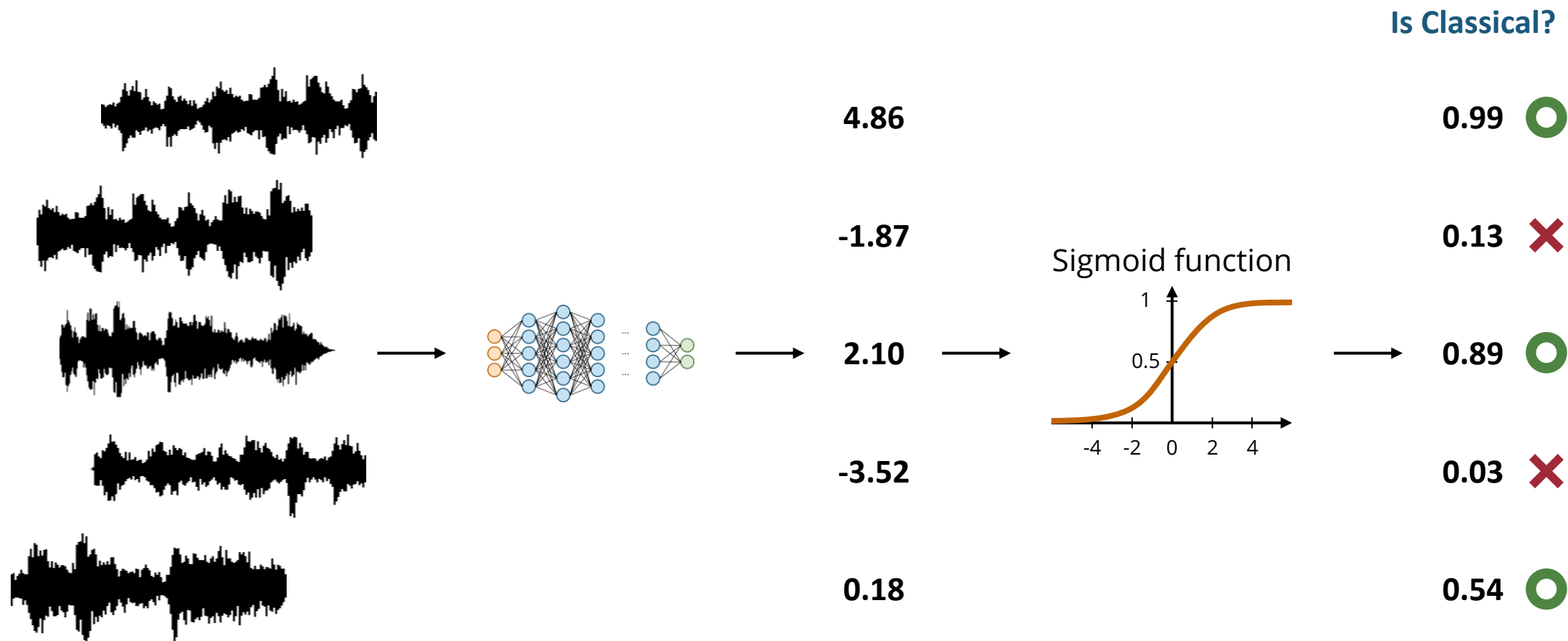
What to play next?



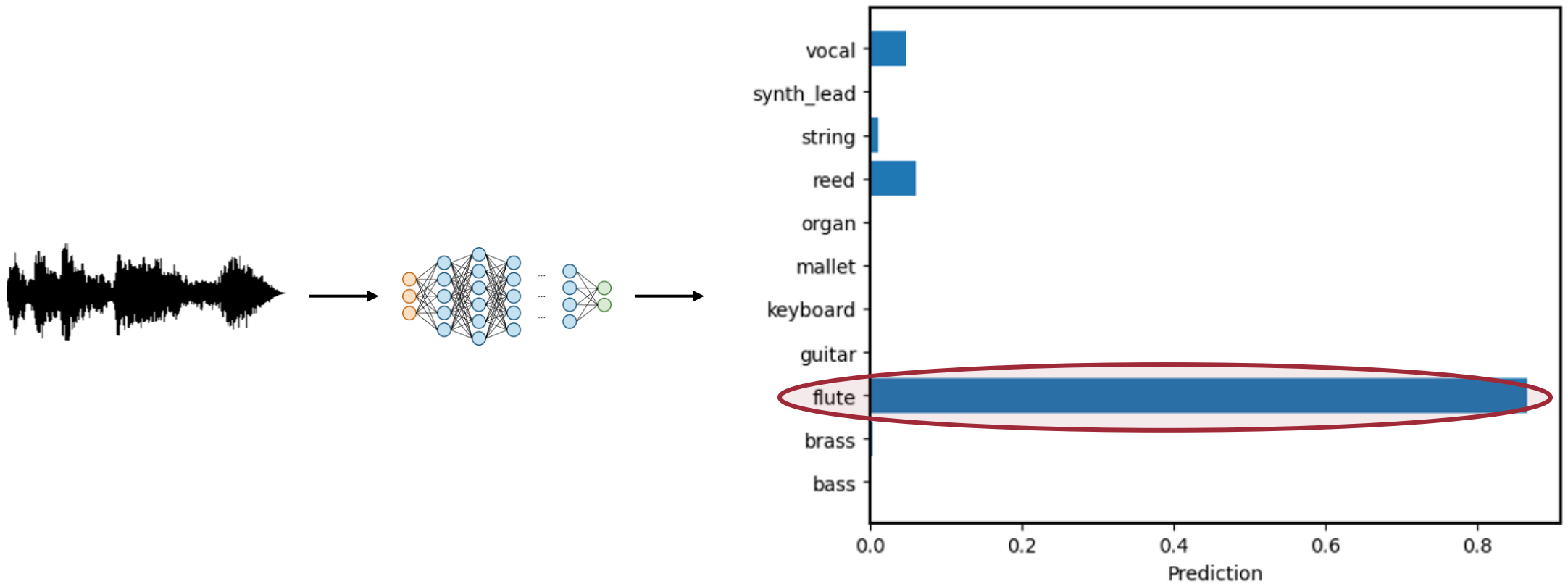
Music Classification for Playlist Generation



Binary Classification



Multiclass Classification



Multi-label Classification



Soul, R&B, pop



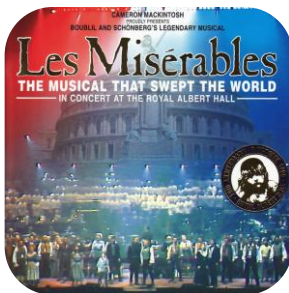
Soundtrack, classical



Rock



Classical



Musical



Alternative, rock

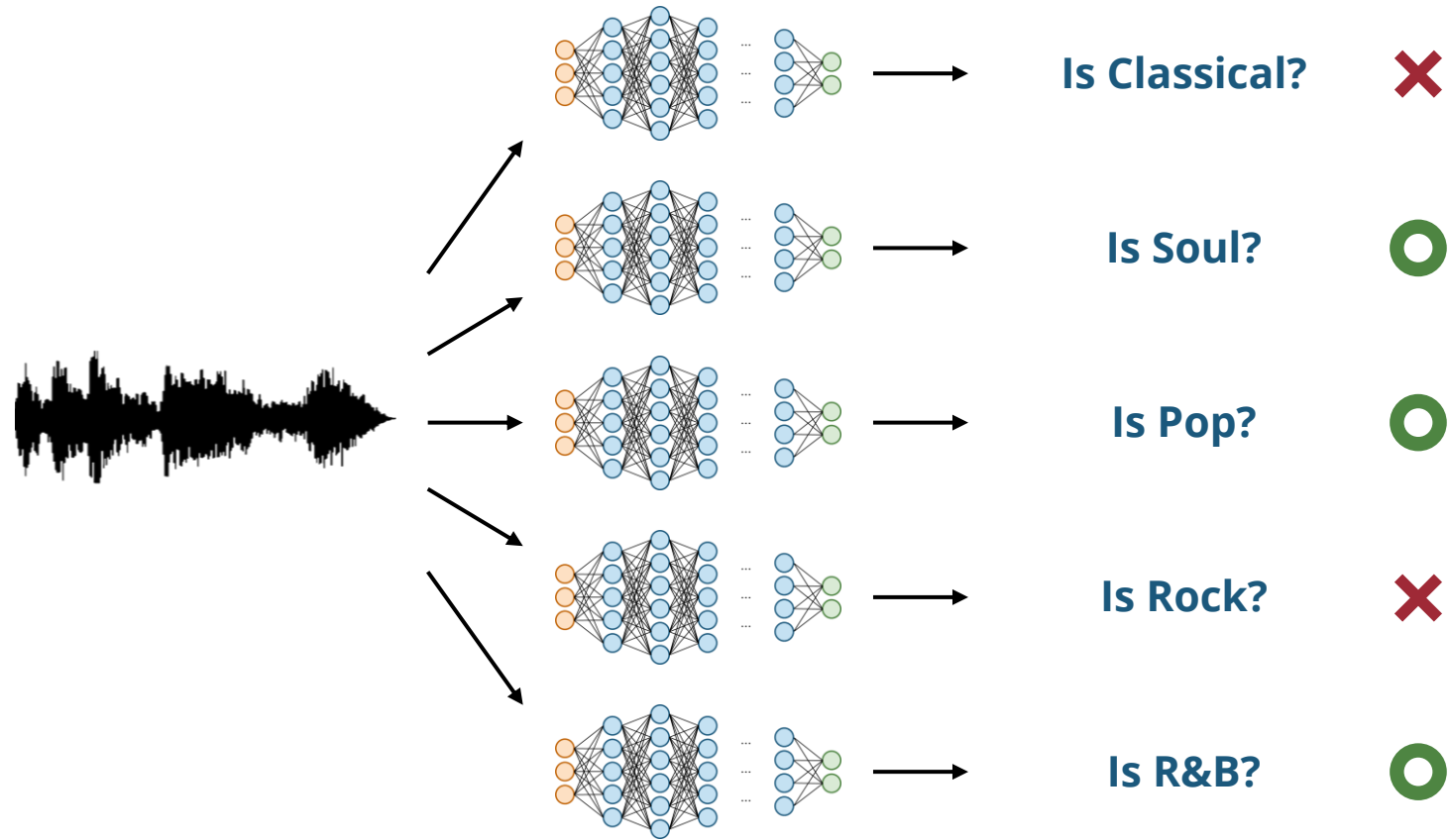


Pop

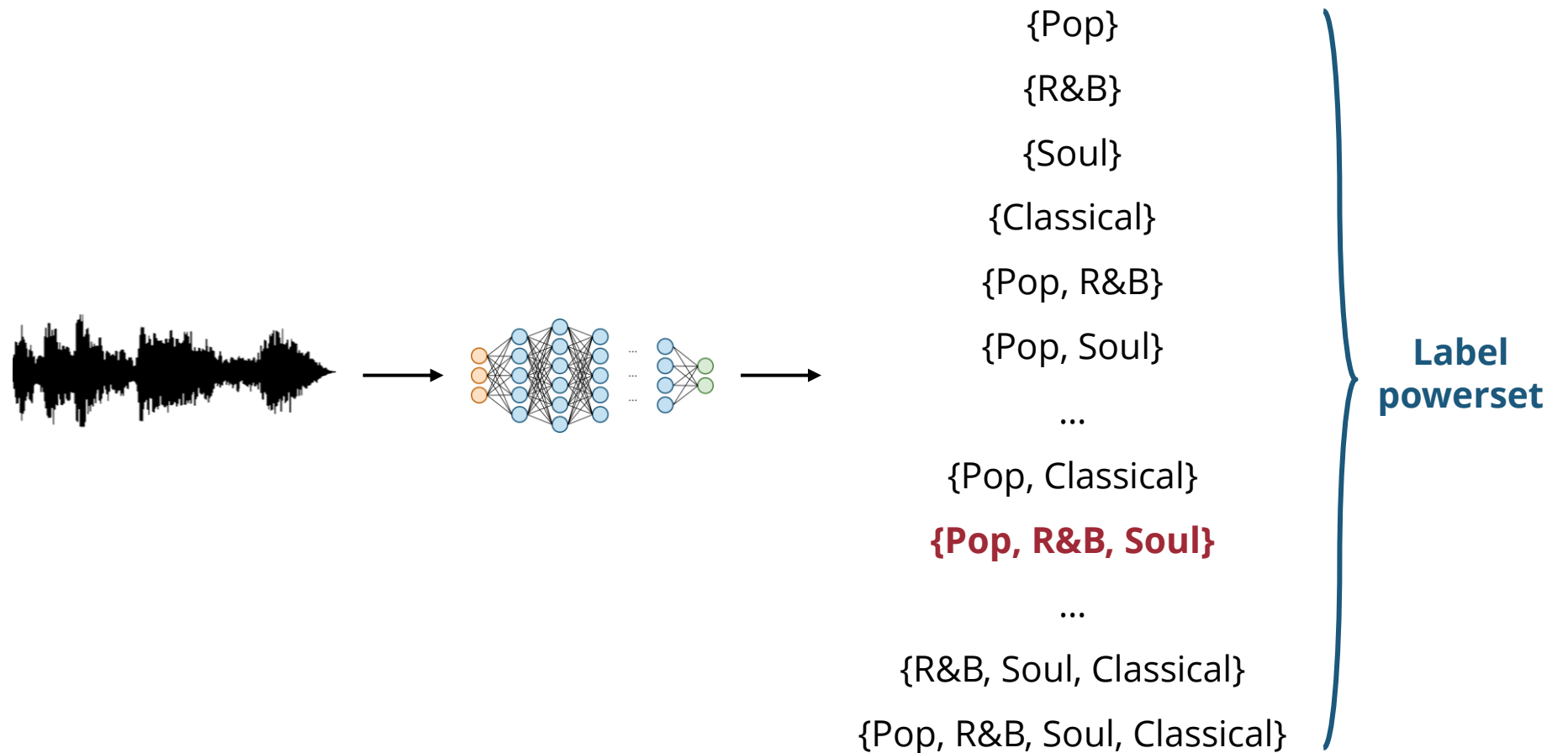


Pop, soul, R&B

Multi-label Classification as Binary Classification

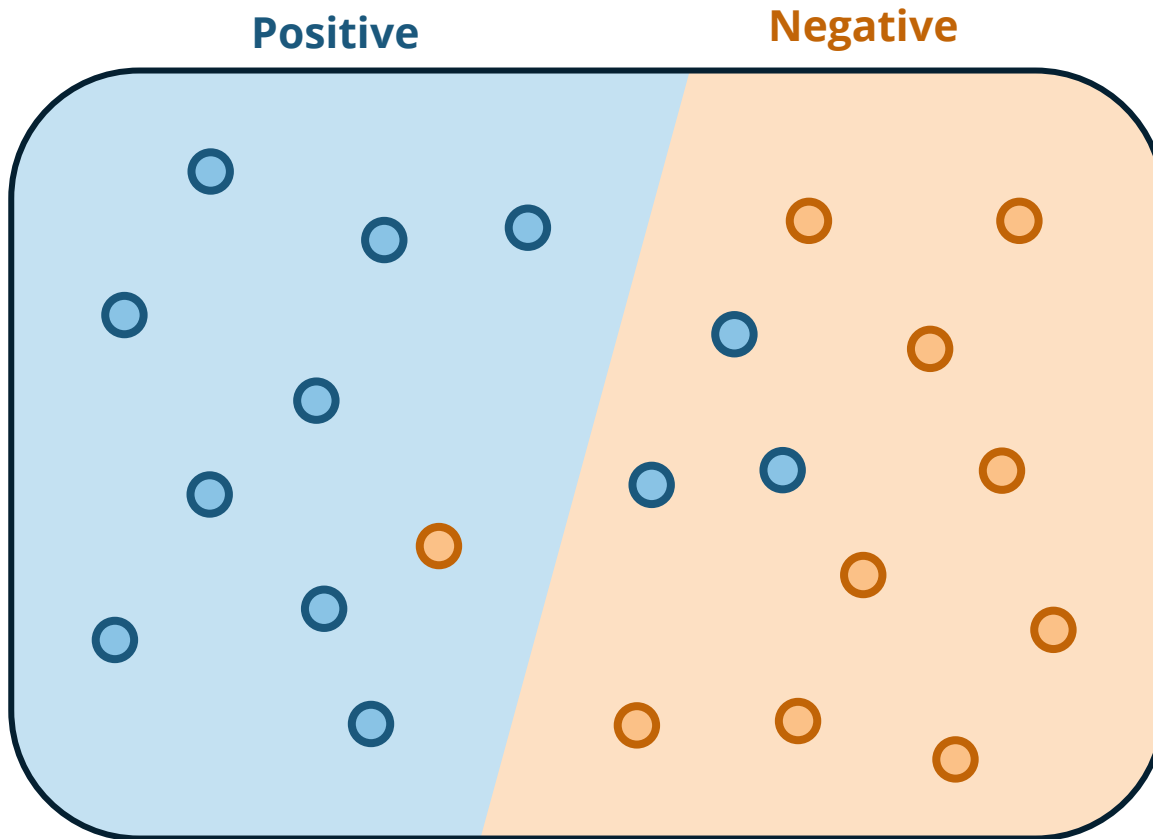


Multi-label Classification as Multi-class Classification



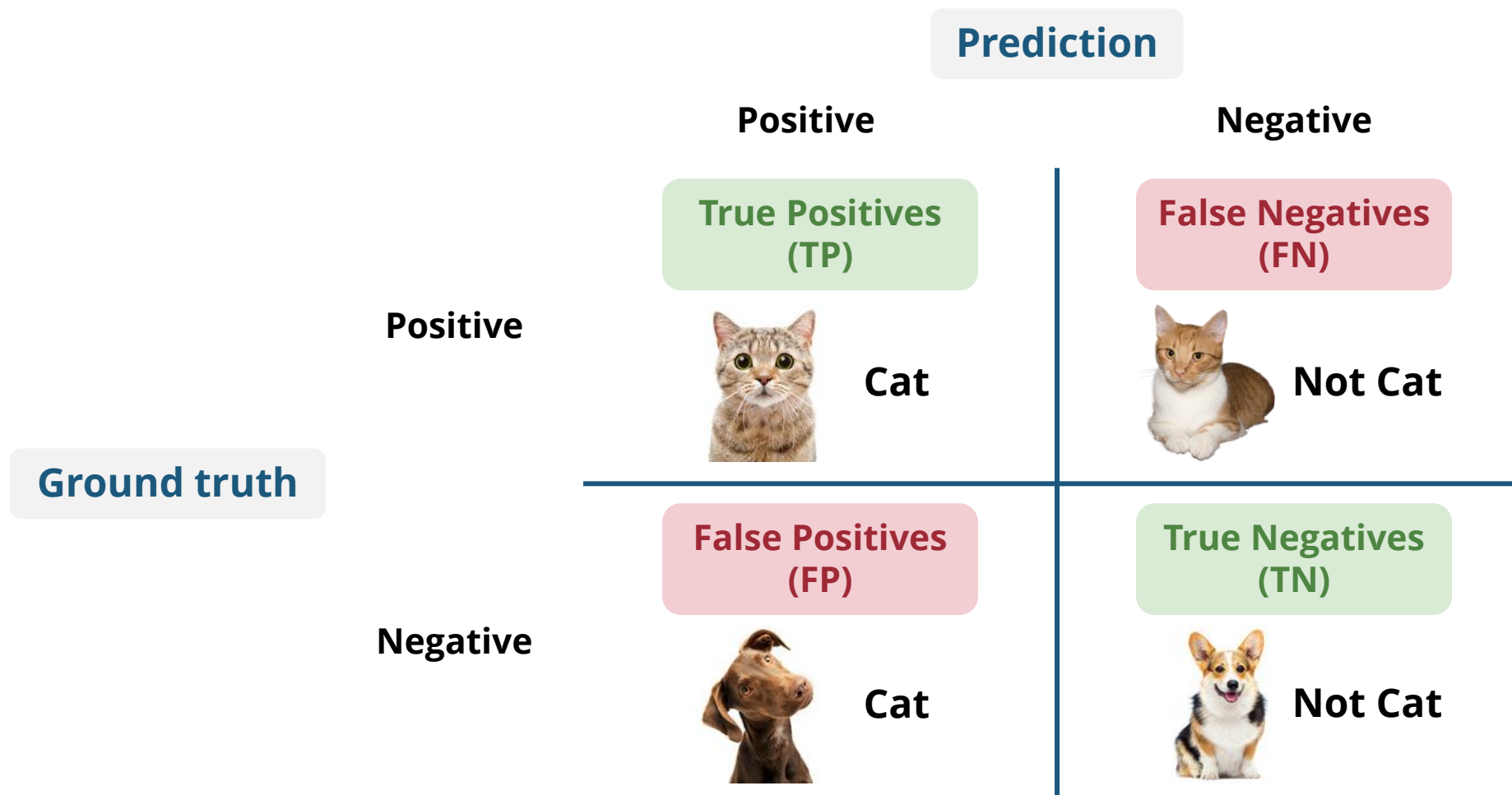
Accuracy

- **Definition:** Percentage of correct predictions across all classes



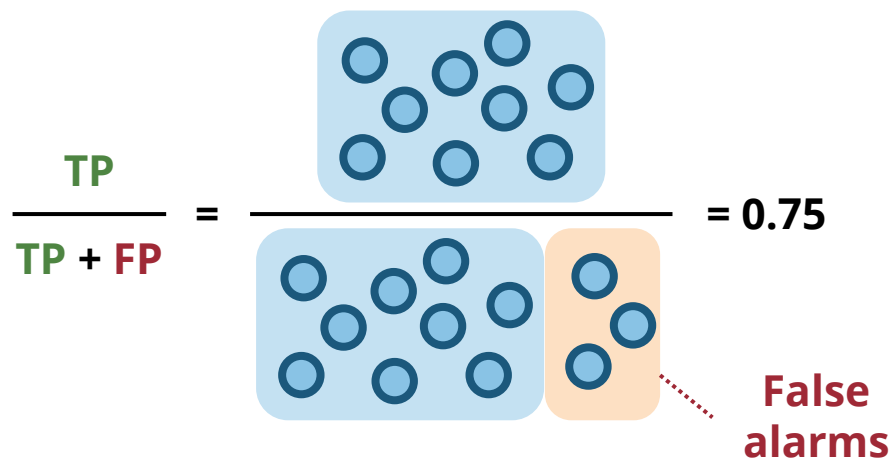
$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{12 + 12}{12 + 1 + 1 + 12} = 0.82$$

Confusion Matrix for Binary Classification



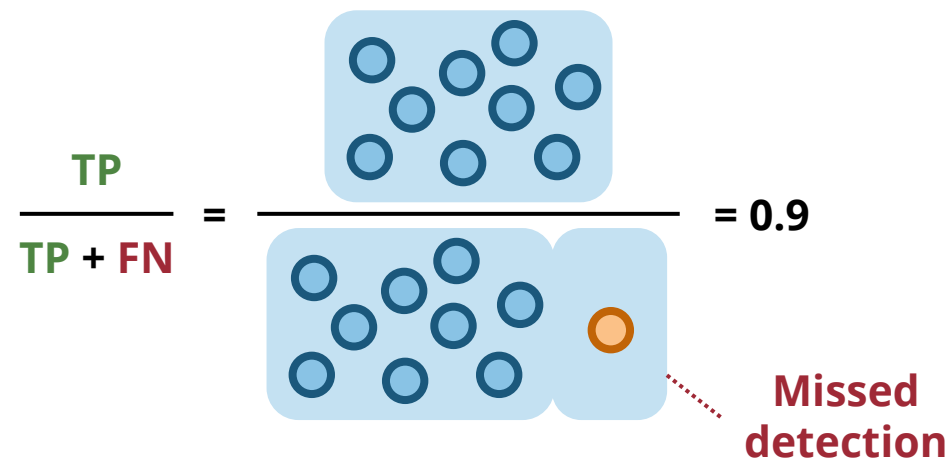
Precision vs. Recall

Precision



How often predictions for the positive are correct

Recall



How well the model finds all positive instances in the dataset

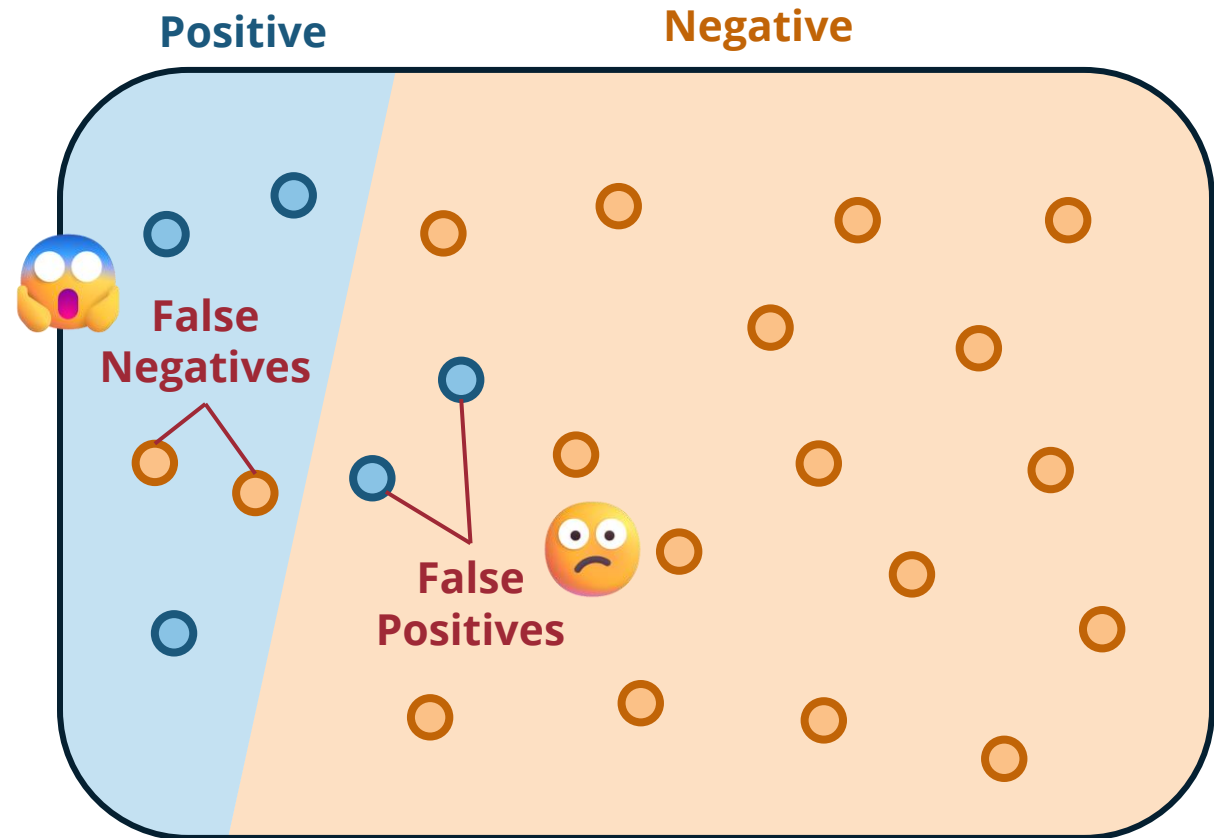
When should we care about Precision & Recall?

Rare cancer detection



Aim for high precision or high recall?

High recall ensures most cancer cases are identified.



False alarms vs Missed detections

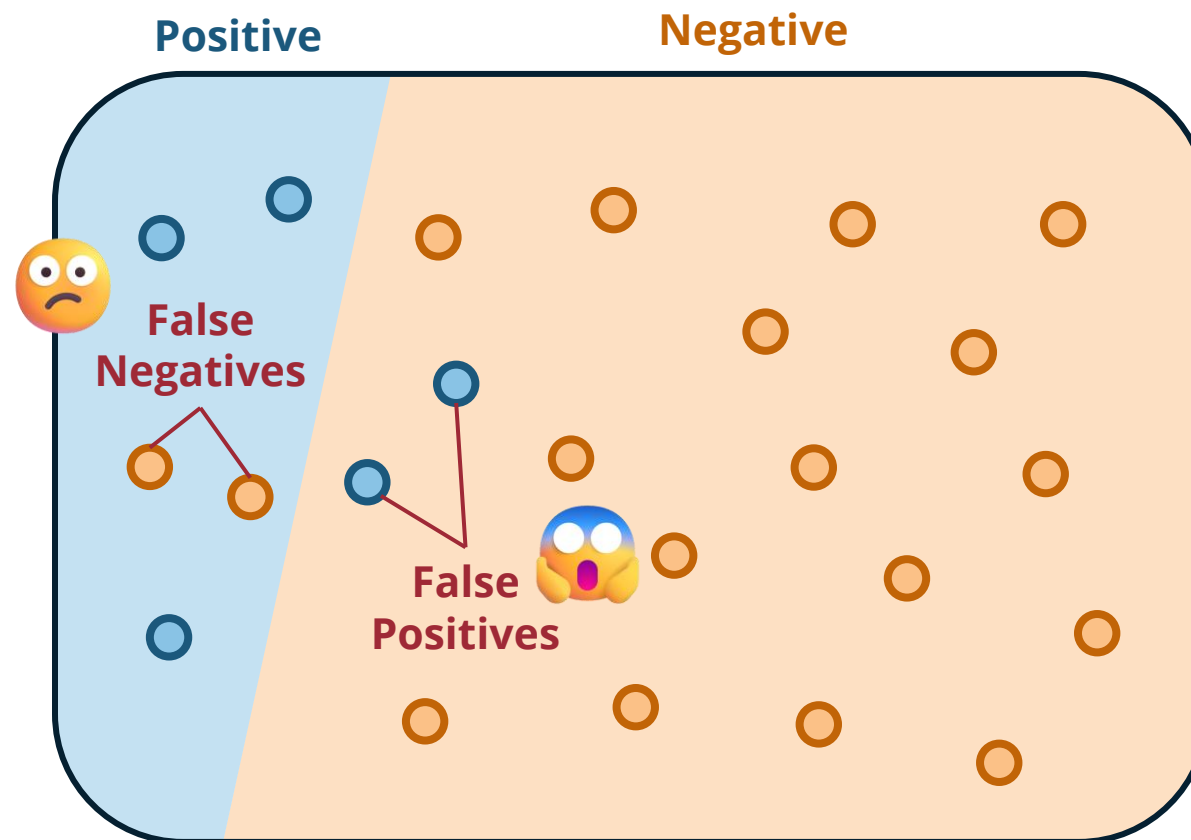
When should we care about Precision & Recall?

Music recommendation



Aim for high precision or high recall?

High precision ensures that the model won't recommend irrelevant items.



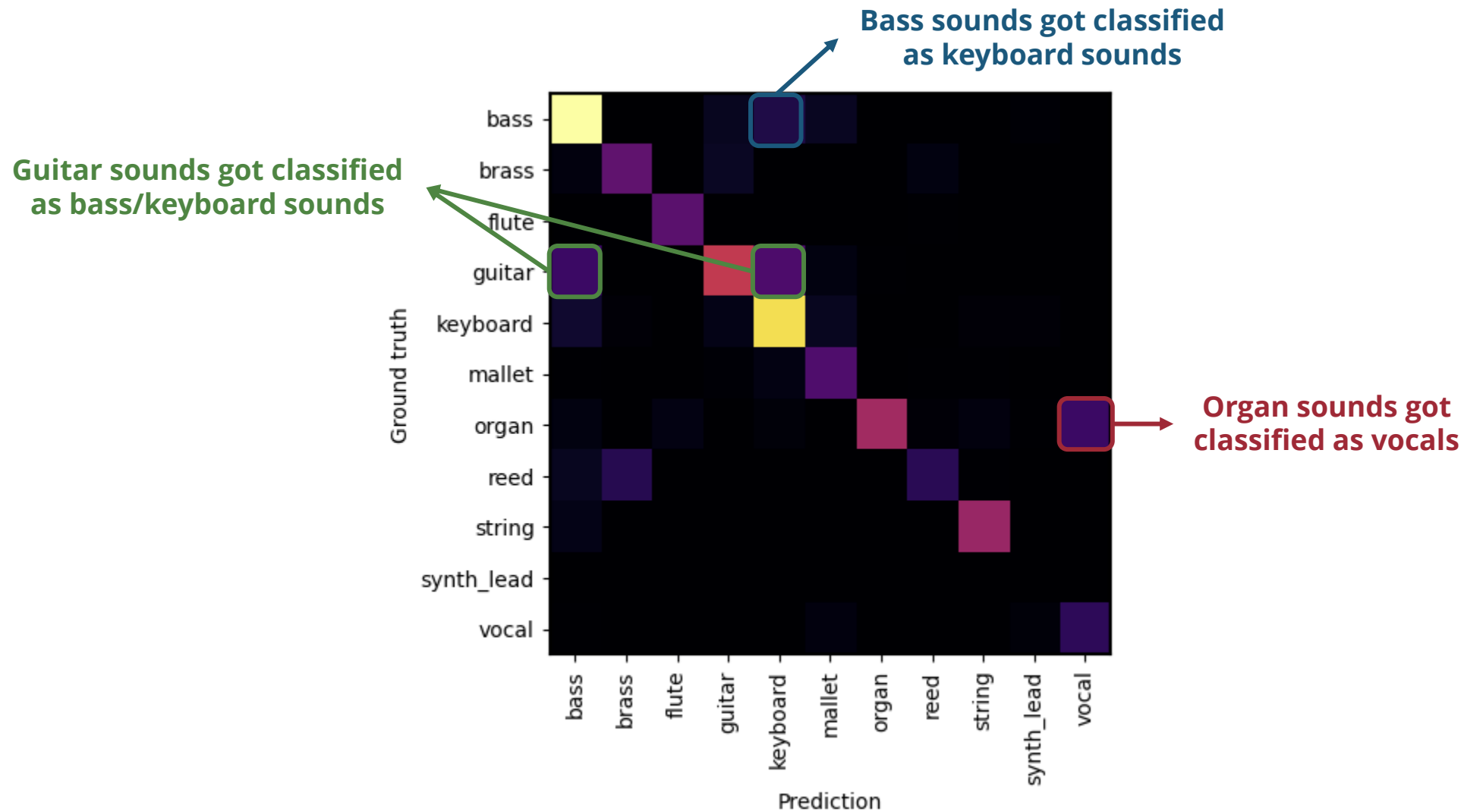
False alarms vs Missed recommendations

F1 Score: Considering both Precision & Recall

- Particularly useful for **imbalanced datasets**
 - Work better than accuracy when the dataset is imbalanced
 - For example, **music search, retrieval, and recommendation**

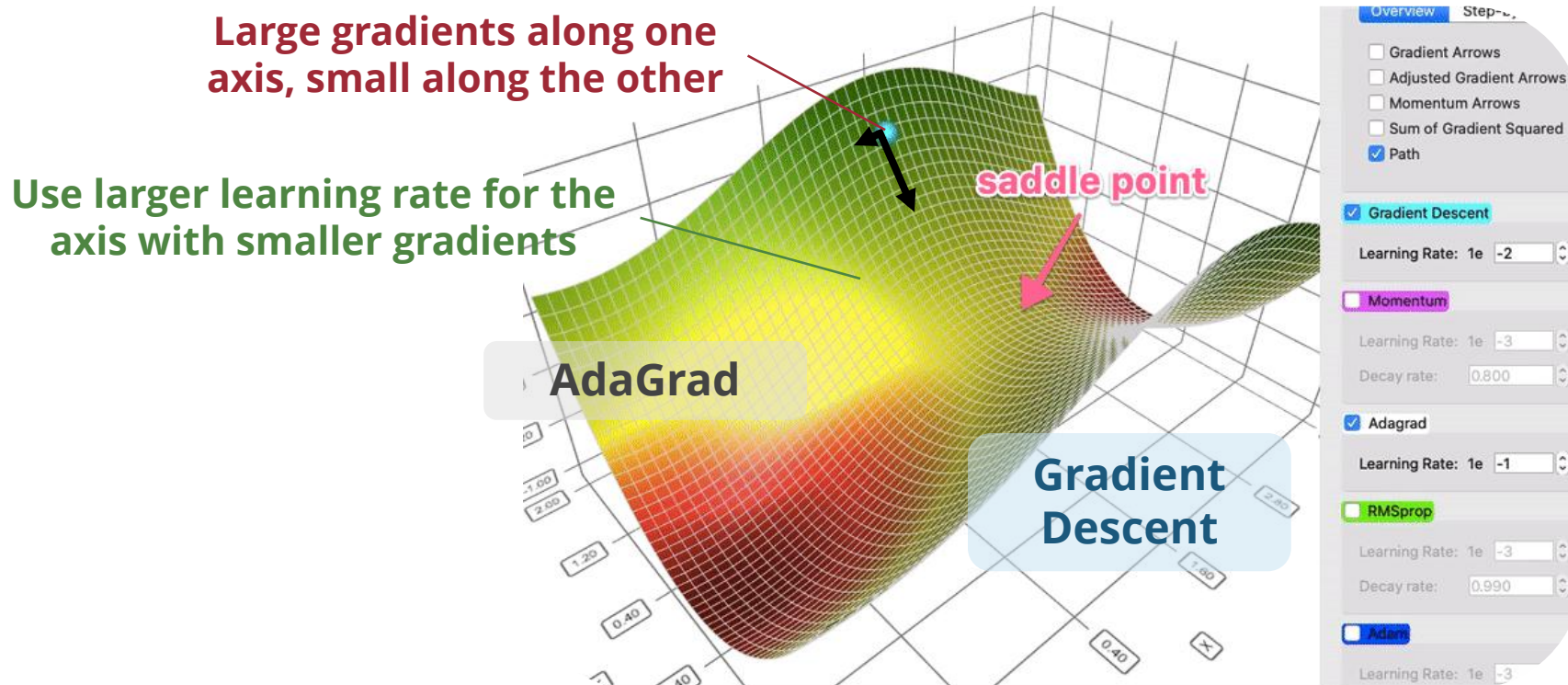
$$\begin{aligned} F_1 &= \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \\ &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \end{aligned}$$

Confusion Matrix for Multiclass Classification



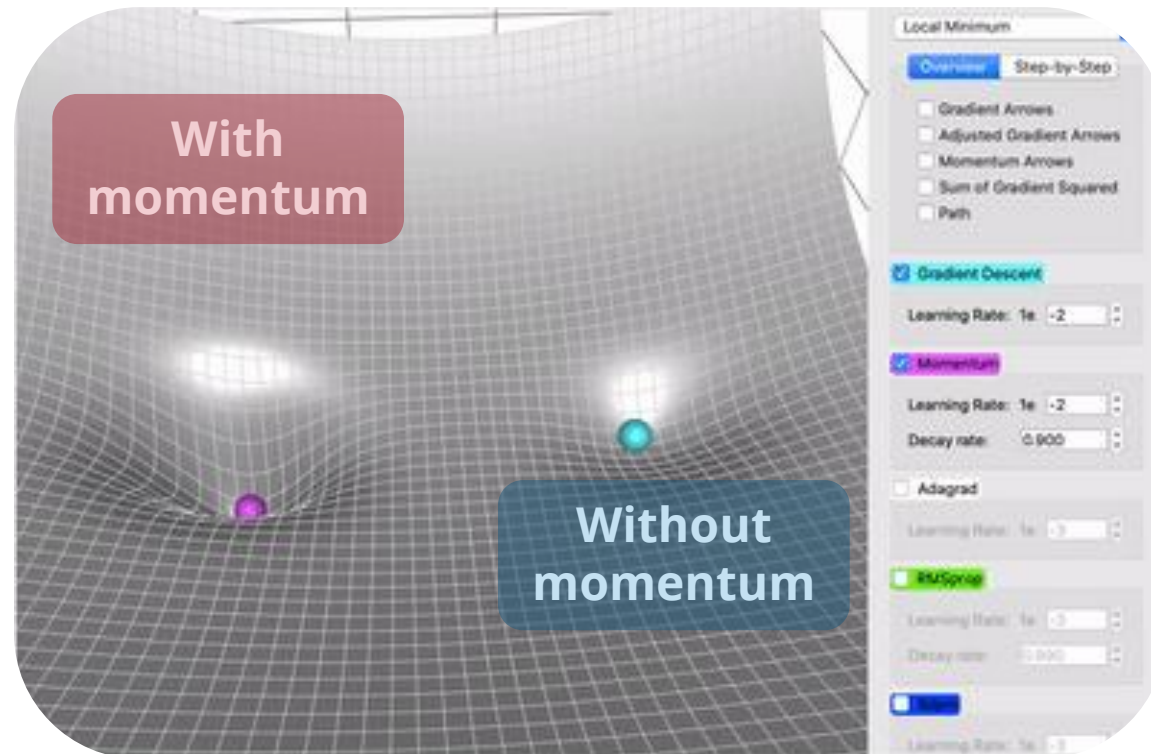
Gradient-based Adaptive Learning Rate

- **Intuition: Compensate axis that has little progress** by comparing the current gradients to the previous gradients



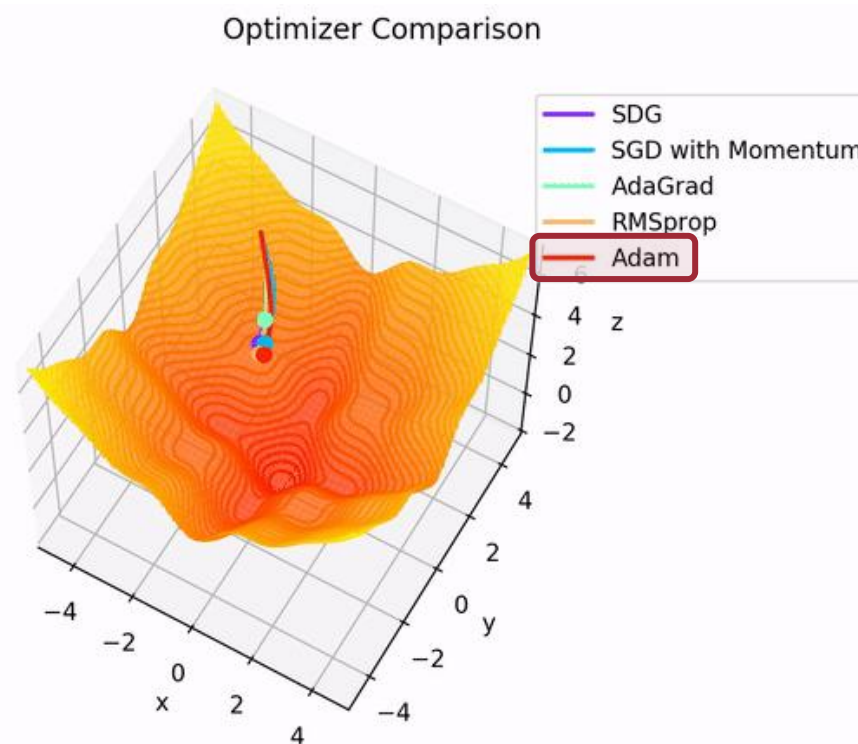
Momentum

- **Intuition:** Maintain the momentum to **escape from local minima**



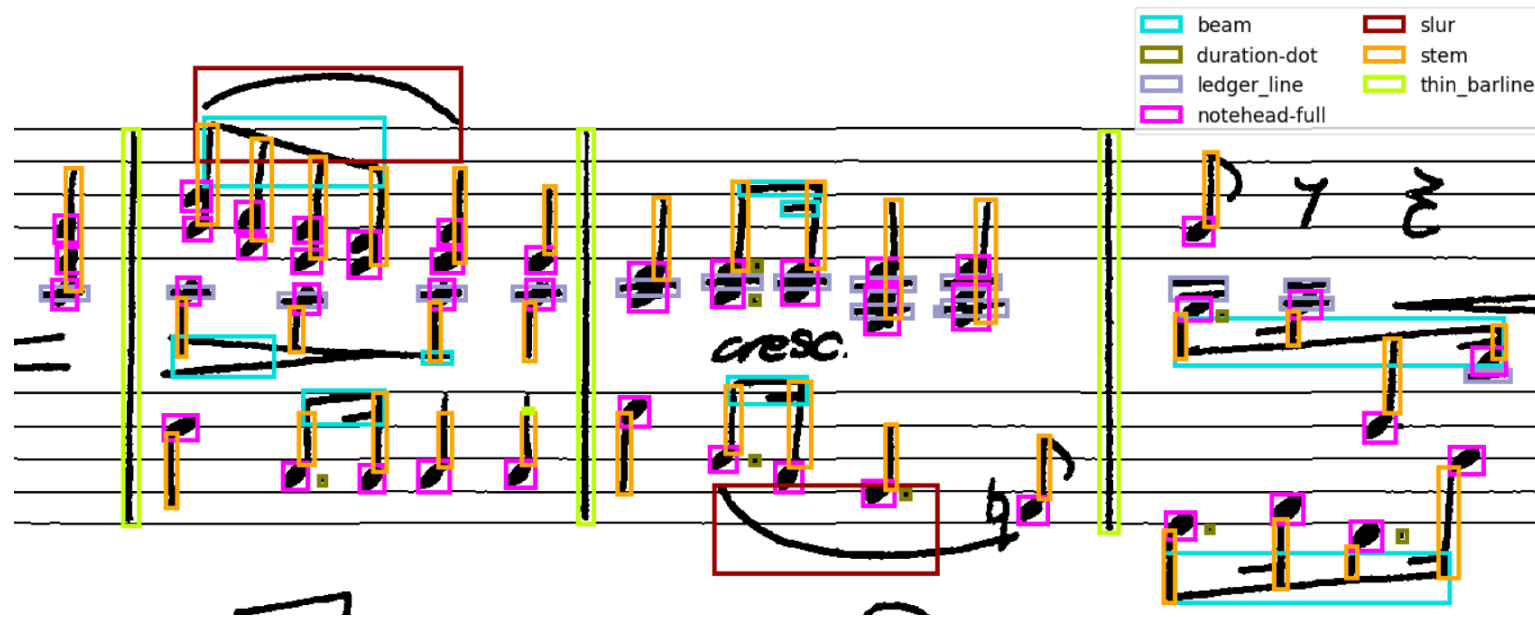
| Adam Optimizer

- Combine the idea of **adaptive learning rate** and **momentum**
- Work **empirically well** in complex neural network
- The **go-to choice** for most cases



Next Lecture

Optical Music Transcription



(Source: Pacha et al., 2018)