The Mooderator inc.

Marshall Briggs, Katherine Cook, Noam Grebler

What is the problem

- Today most music classification => genres
- People usually turn on music to feel / experience a certain mood and not a genre.
- We feel a mood X but want to feel a mood Y.
  - 1) understand what is mood X
  - 2) **get to mood Y**

Past Literature

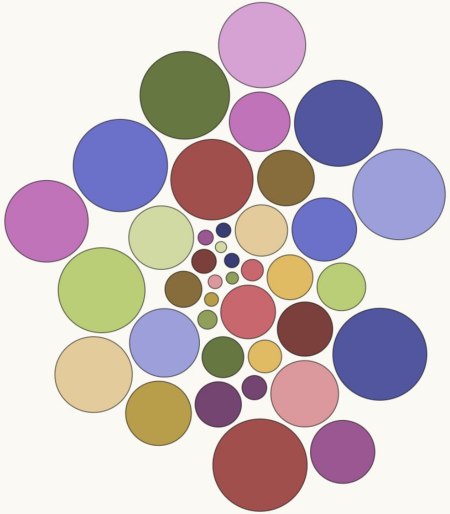
- This paper investigates a musical induction method that spontaneously leads subjects to real emotional states:
  - http://ieeexplore.ieee.org/abstract/document/5363083/
- This second experiment is trying to create an algorithm that understands the mood of songs by analyzing lyrics:
  - http://ieeexplore.ieee.org/abstract/document/4441720/

Why is interesting

- Solve the problem => classifying music by mood & ignore genre
- This could change the way we listen to music:
    - We could understand the mood that people are feeling => put music targeted at their mood
    - We could change people’s mood through this algorithm
    - It exposes people to new types of music

Our project

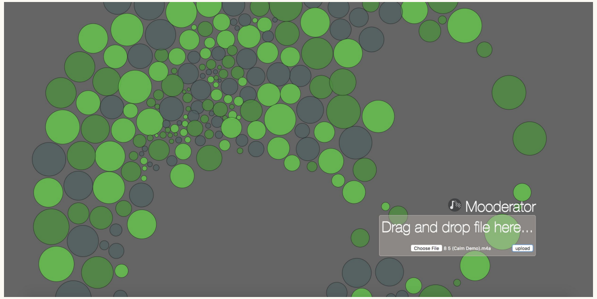
- We created an algorithm that given an audio our model identifies the mood of the song and outputs a colored data visualization.
- It's divided in different parts:
  - Front end app intakes a audio file
  - It then calls our model
  - Our model returns a mood
  - Front end app returns a graph associated with the mood
- Libraries
  - Keras
  - Librosa
  - Tensorflow
  - Numpy
  - Re (regular expression)



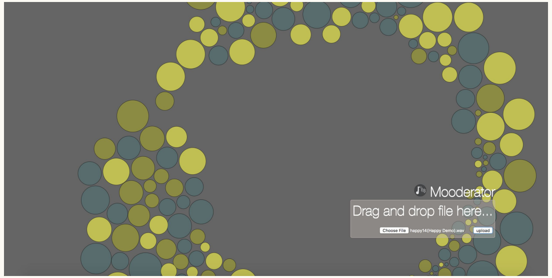
Front end

- We created a web app that generates some graphics according to the mood of the song:
  - It creates 2 groups of balls:
    - A group that shows the general mood of the song
    - A second group that shows the specific mood of the beat

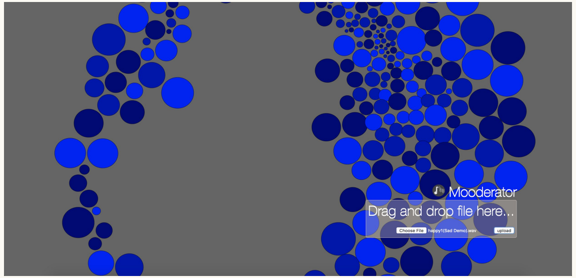
If Mooderator is calm, he becomes Green



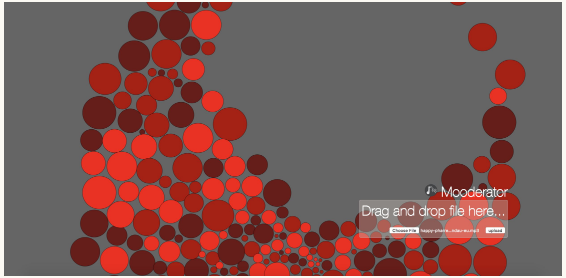
If Mooderator is happy, he becomes YELLOW



If Mooderator is sad, he becomes Blue



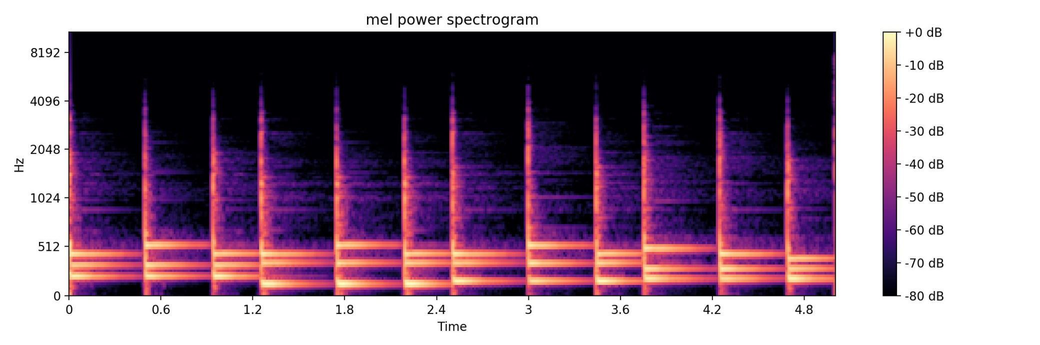
If Mooderator is aggressive, he becomes Red (don’t try this at home)



Spectral Analysis (I)

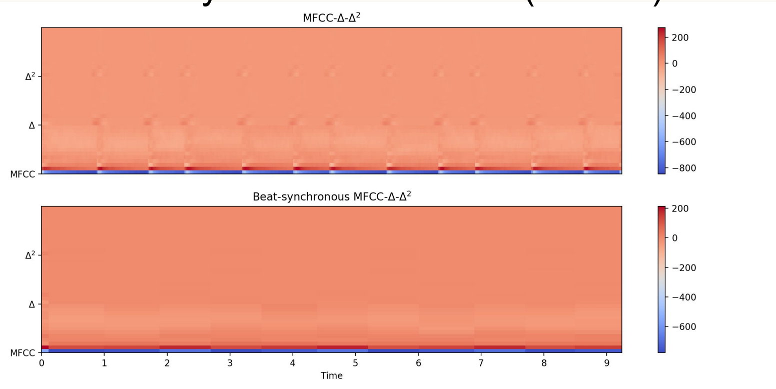
- Traditional Mel power spectogram (bottom right)
- We identified features that distinguish music of different moods (bottom left)
- Using *librosa* (Python audio analysis lib.), we extracted four identifying features from various pieces of music: pitch (Chromagram), intensity (loudness, by Constant Q-Transform), timbre (MFCC), and tempo

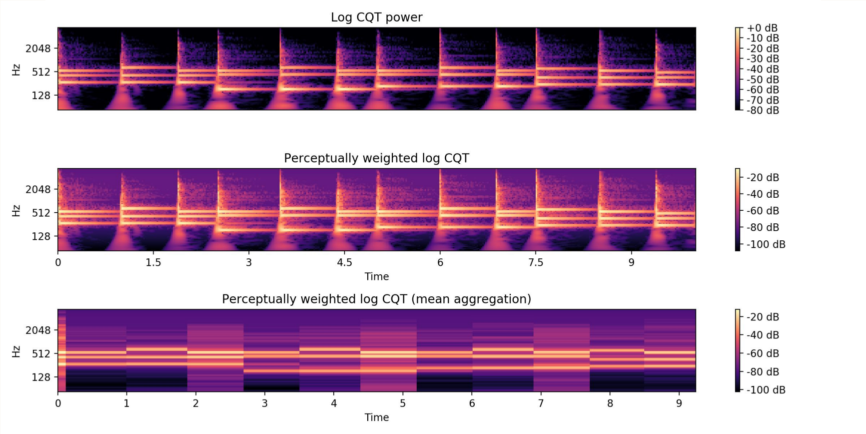
	Intensity	Pitch	Rhythm
Happy	Medium	Very High	High
Sad	Medium	Very Low	Low
Calm	Very Low	Medium	Very Low
Aggressive	High	High	High



Spectral Analysis (II)

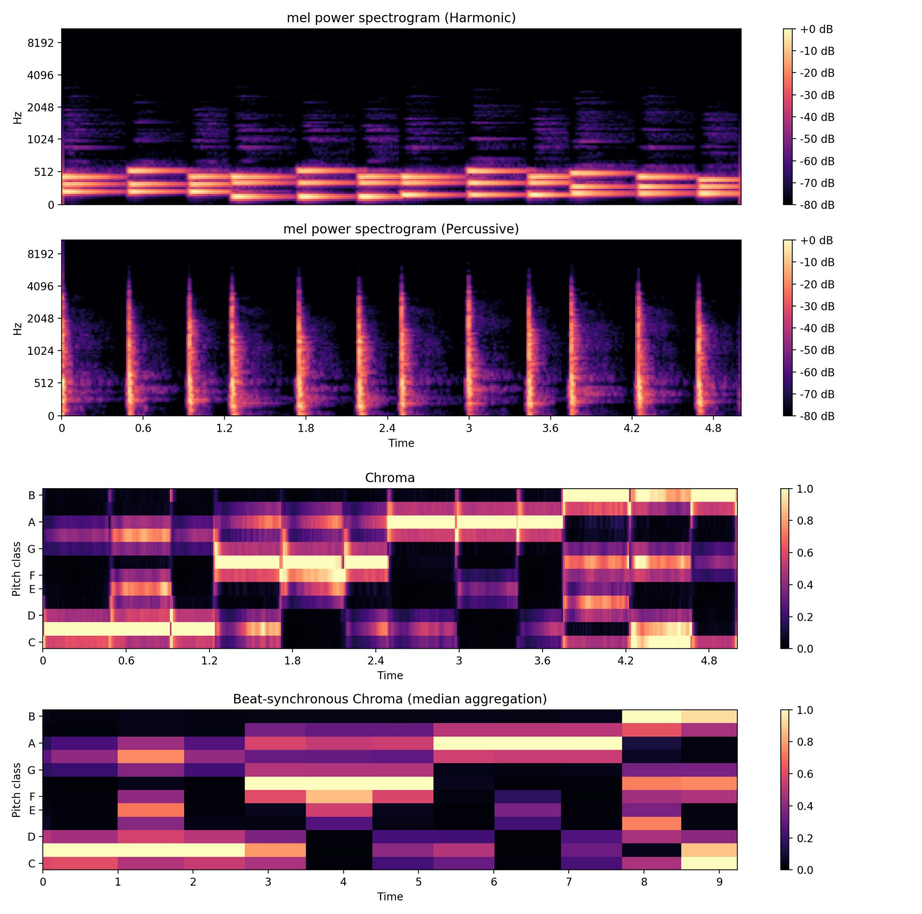
Beat-synchronous MFCC (Timbre)





Beat-synchronous CQT (Intensity)

Percussive vs. Harmonic components



Beat-synchronous Chroma (Pitch)

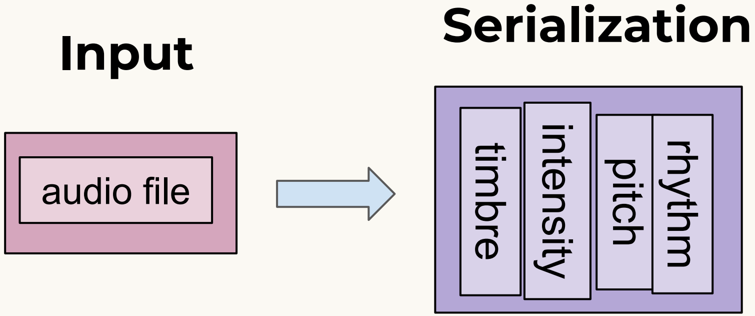
Our Approach

Input

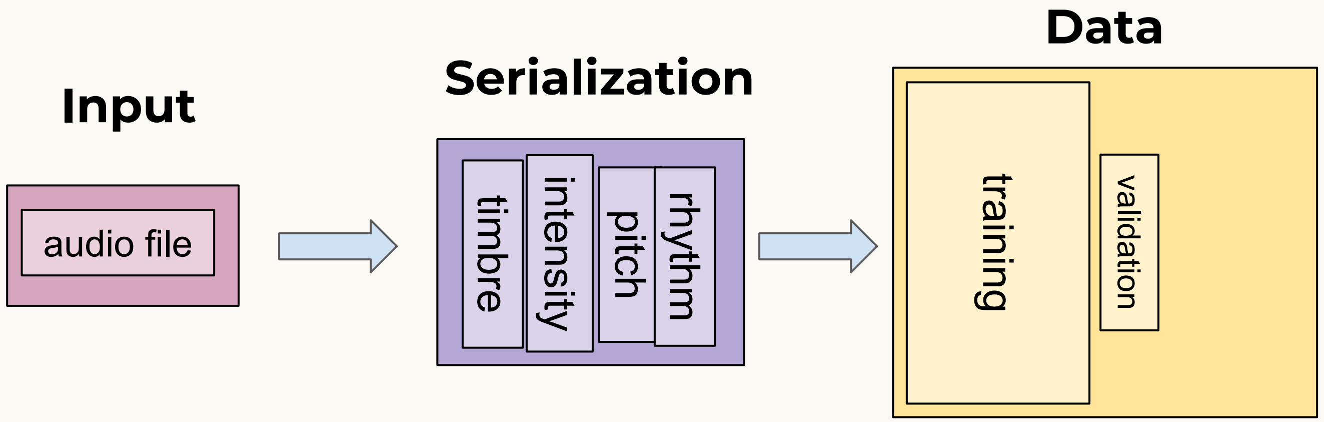
audio file*

*GTZAN Dataset

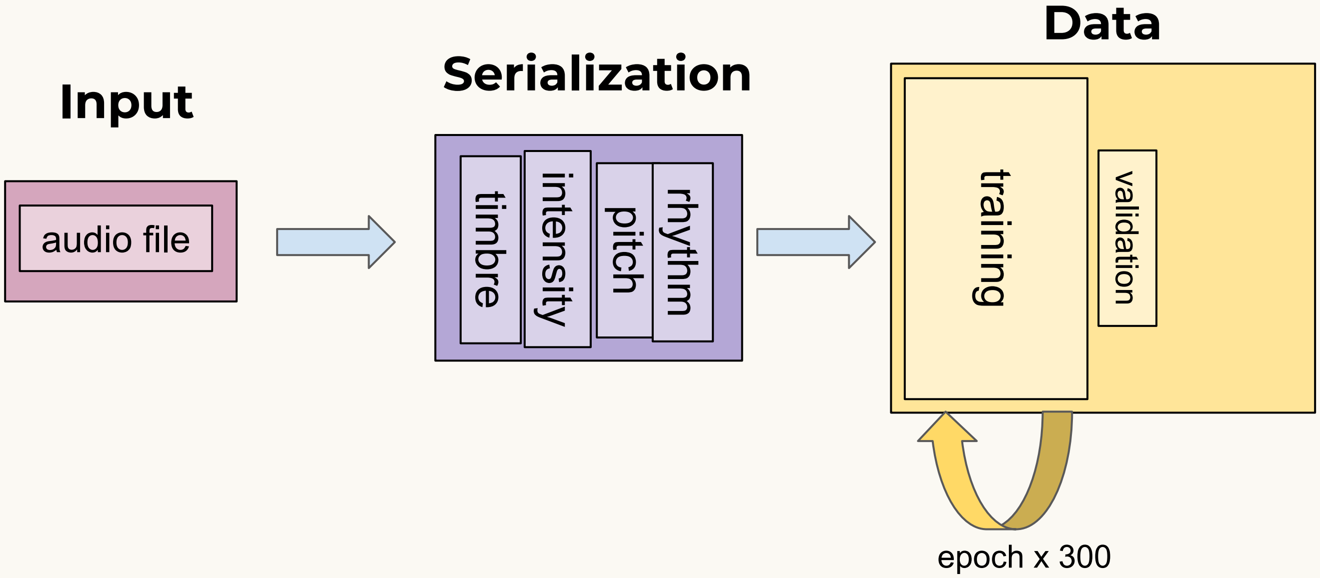
Our Approach



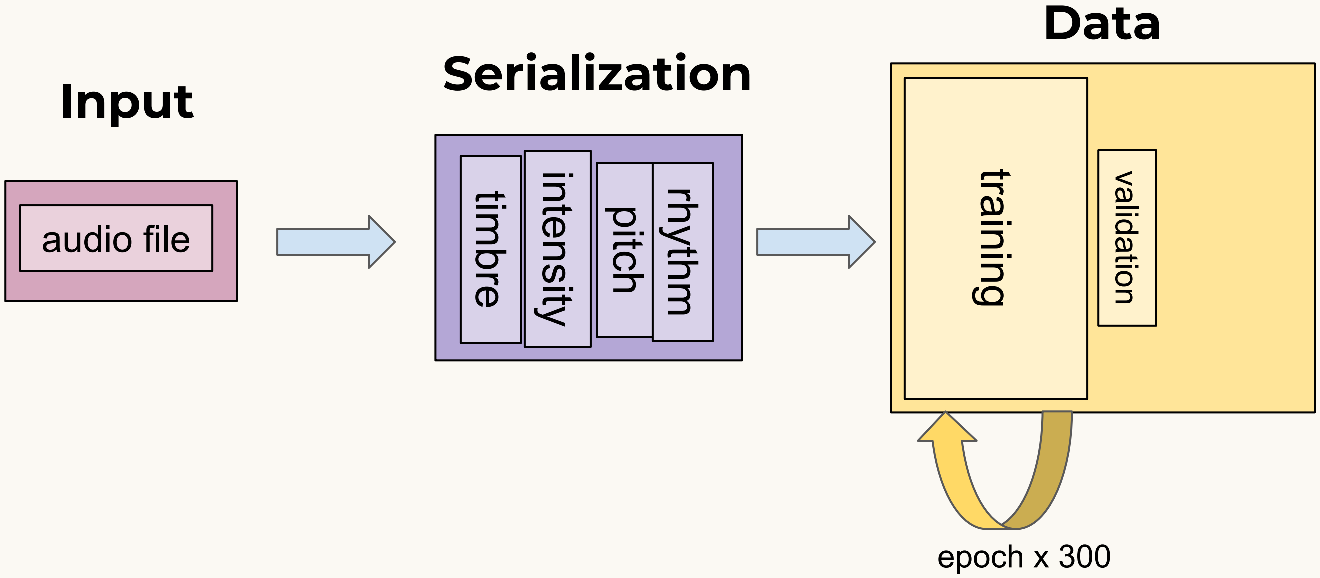
Our Approach



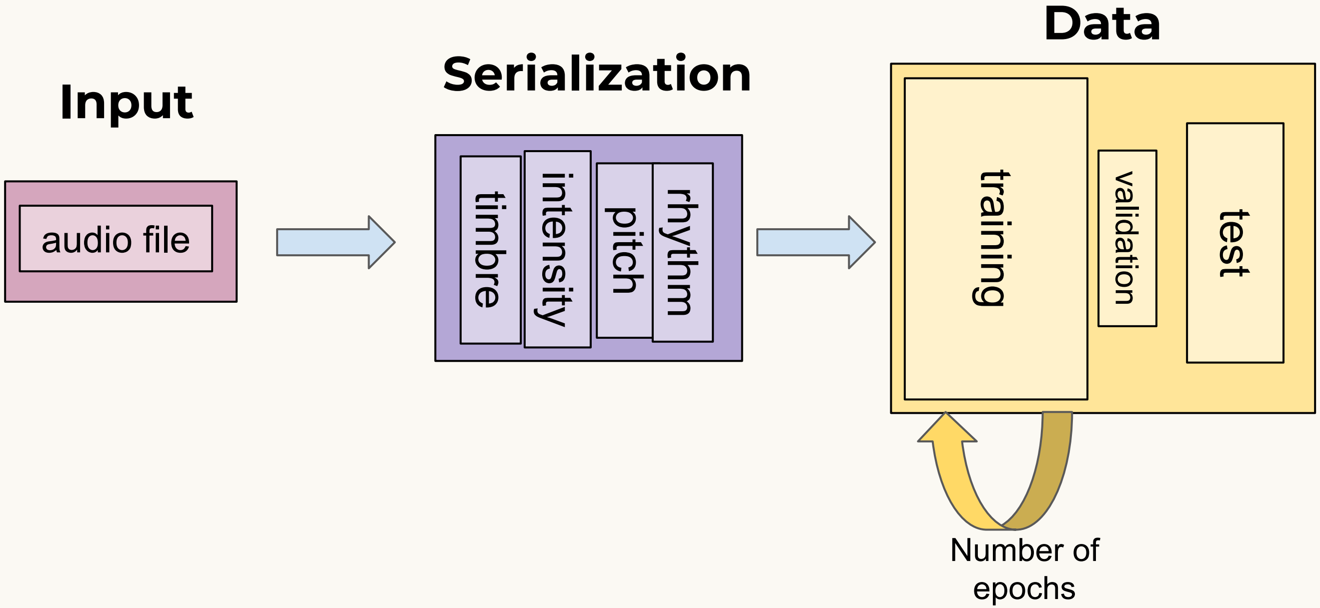
Our Approach



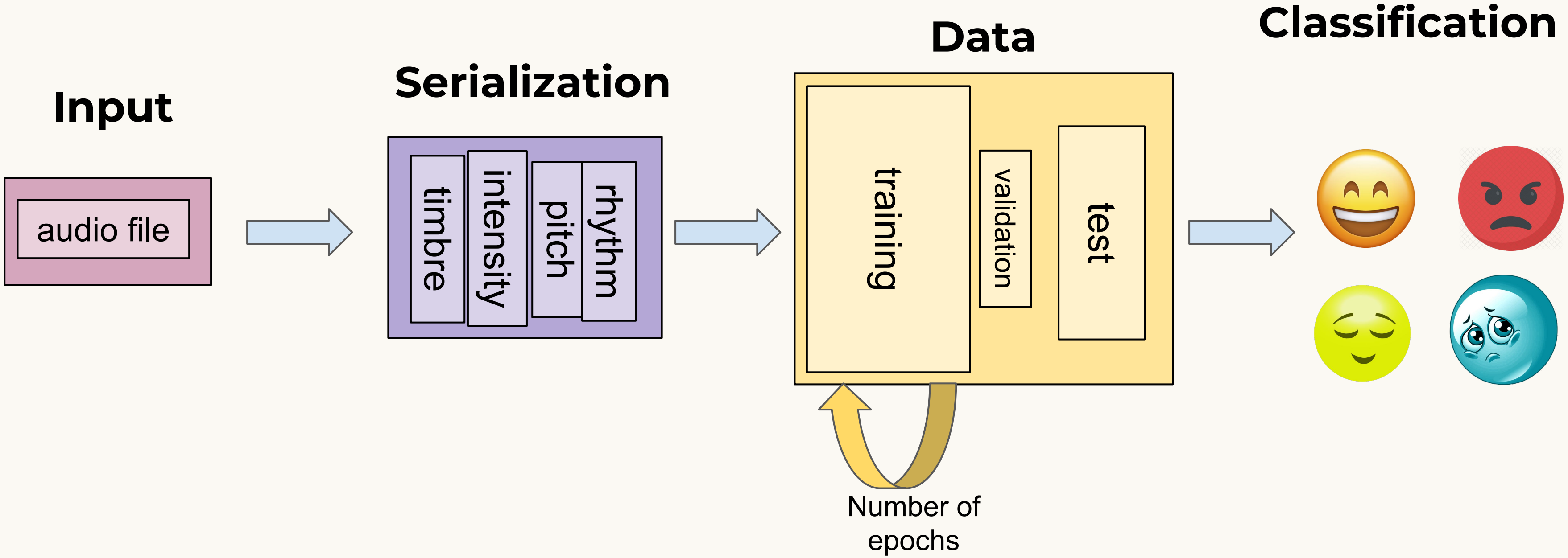
Our Approach



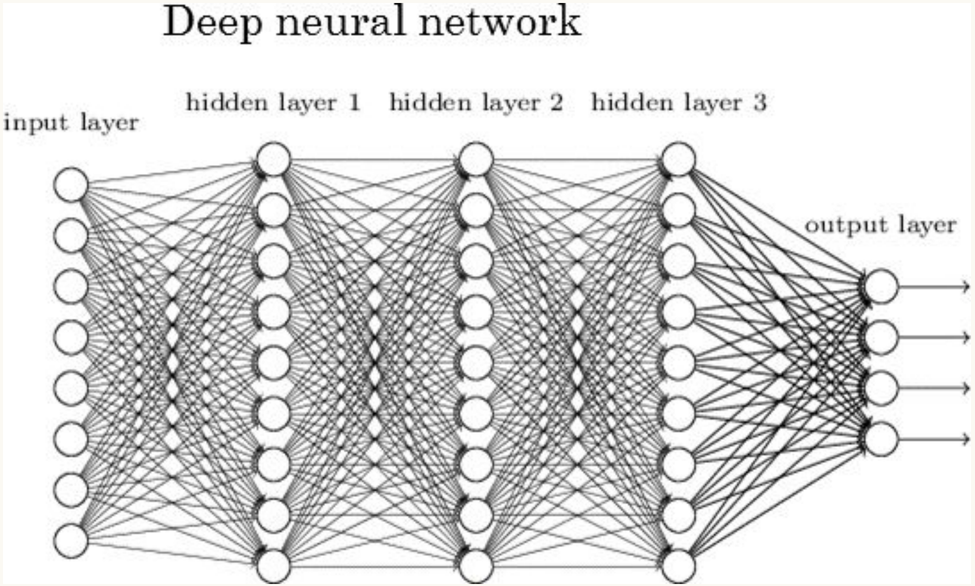
Our Approach



Our Approach



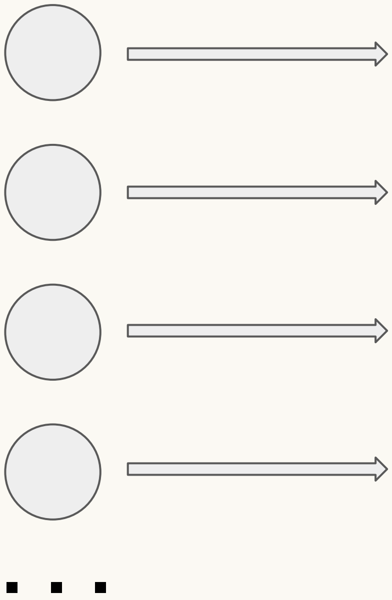
DNN Model



- Tensorflow, DNNClassifier
- Two hidden layers, each with 250 nodes
- Most widely used for image classification

DNN

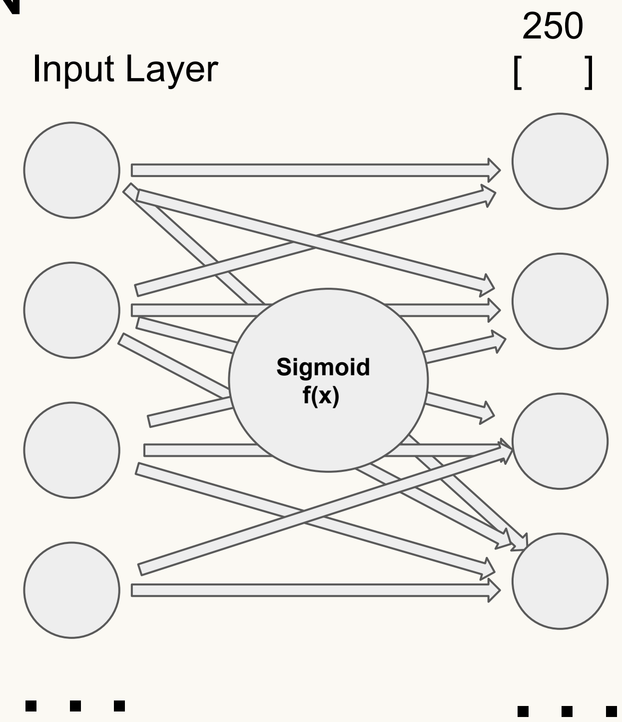
Input Layer



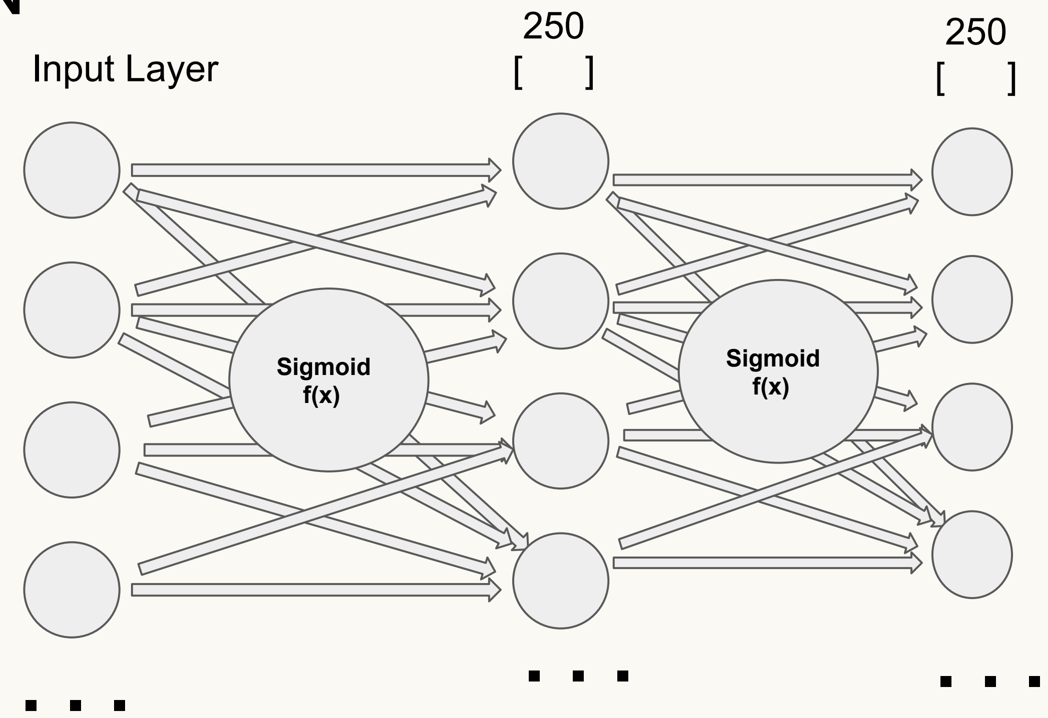
Example input:

{
aggressive.26,
aggressive,
-14.667919917705497,
-80.90939001475839,
-38.419802635449386,
-36.853654177984005,
1.0,
0.1157046766273615,
0.48945878202904797,
0.43620907774288215,
199.7710435523468,
-731.1777198817912,
-7.946742702398995,
0.007545136156390327,
0.5108390022675737,
0.16253968253968254,
0.4445214045214O454,
0.452789115646258
}

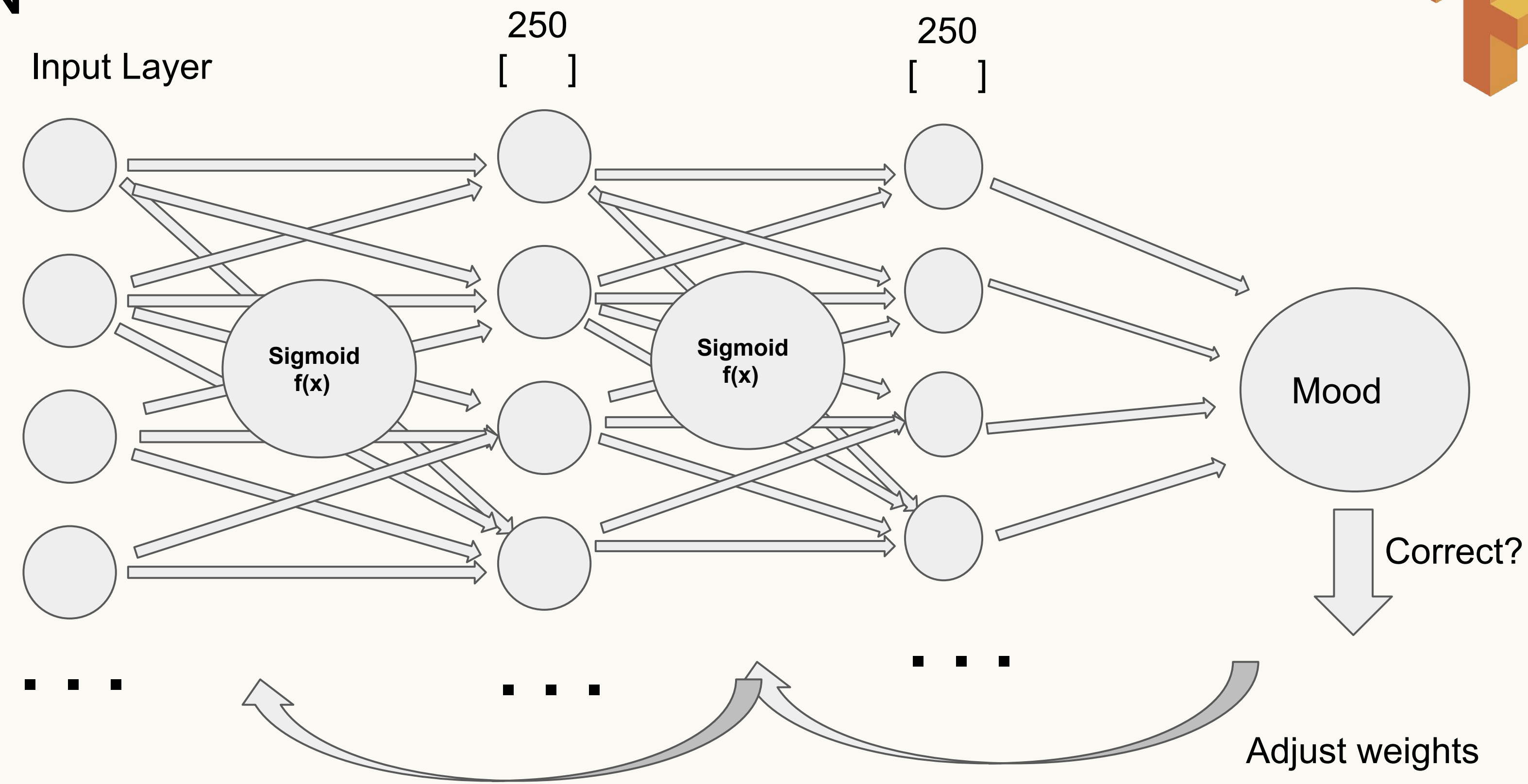
DNN



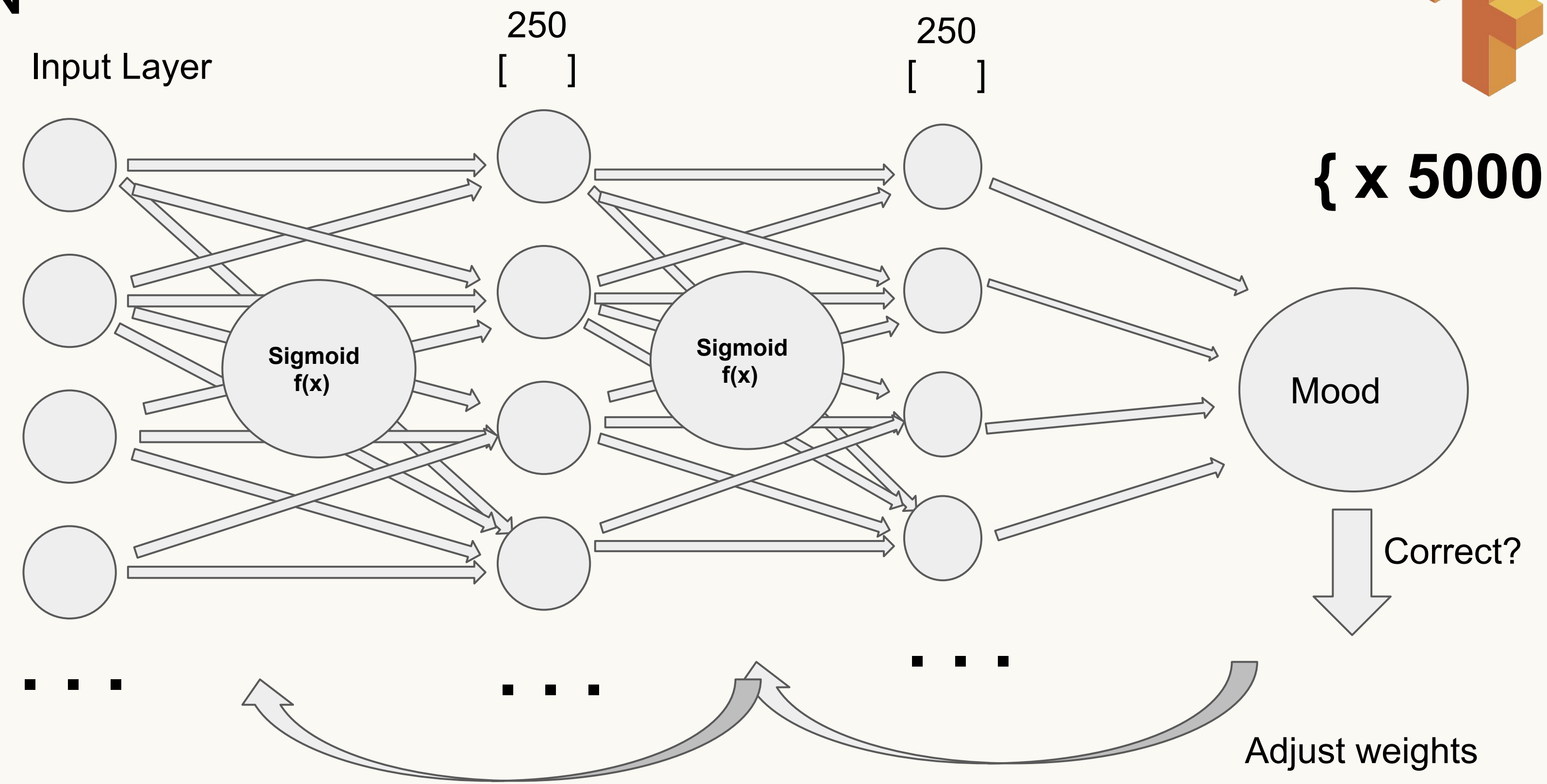
DNN



DNN


Input Layer
250
[ ]
250
[ ]
Sigmoid
f(x)
Sigmoid
f(x)
Mood
Correct?
Adjust weights

DNN



Input Layer
250
[ ]
250
[ ]
Sigmoid
f(x)
Sigmoid
f(x)
Mood
Correct?
. . .
. . .
. . .
Adjust weights

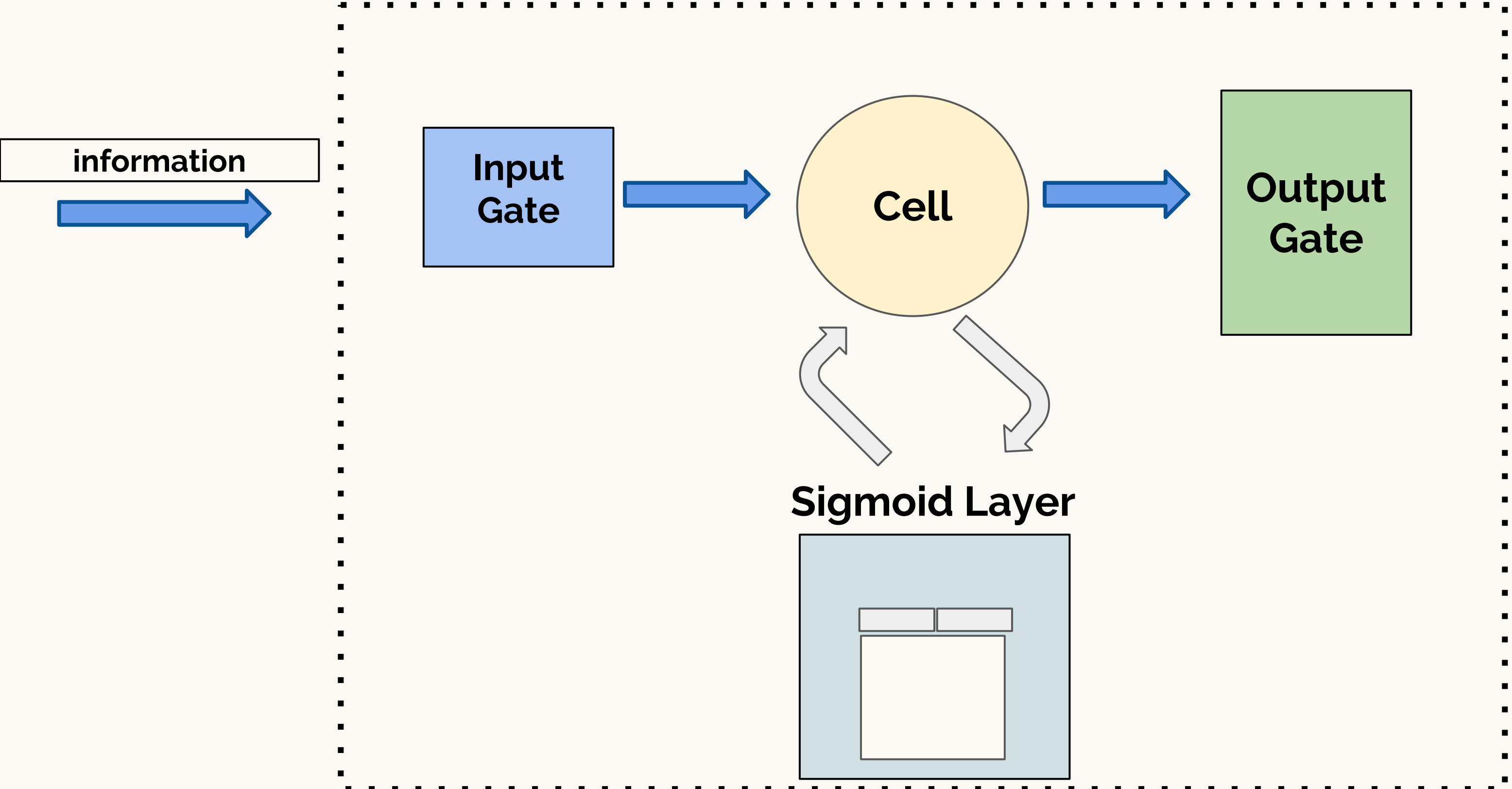
{ x 5000 }

LSTM Network

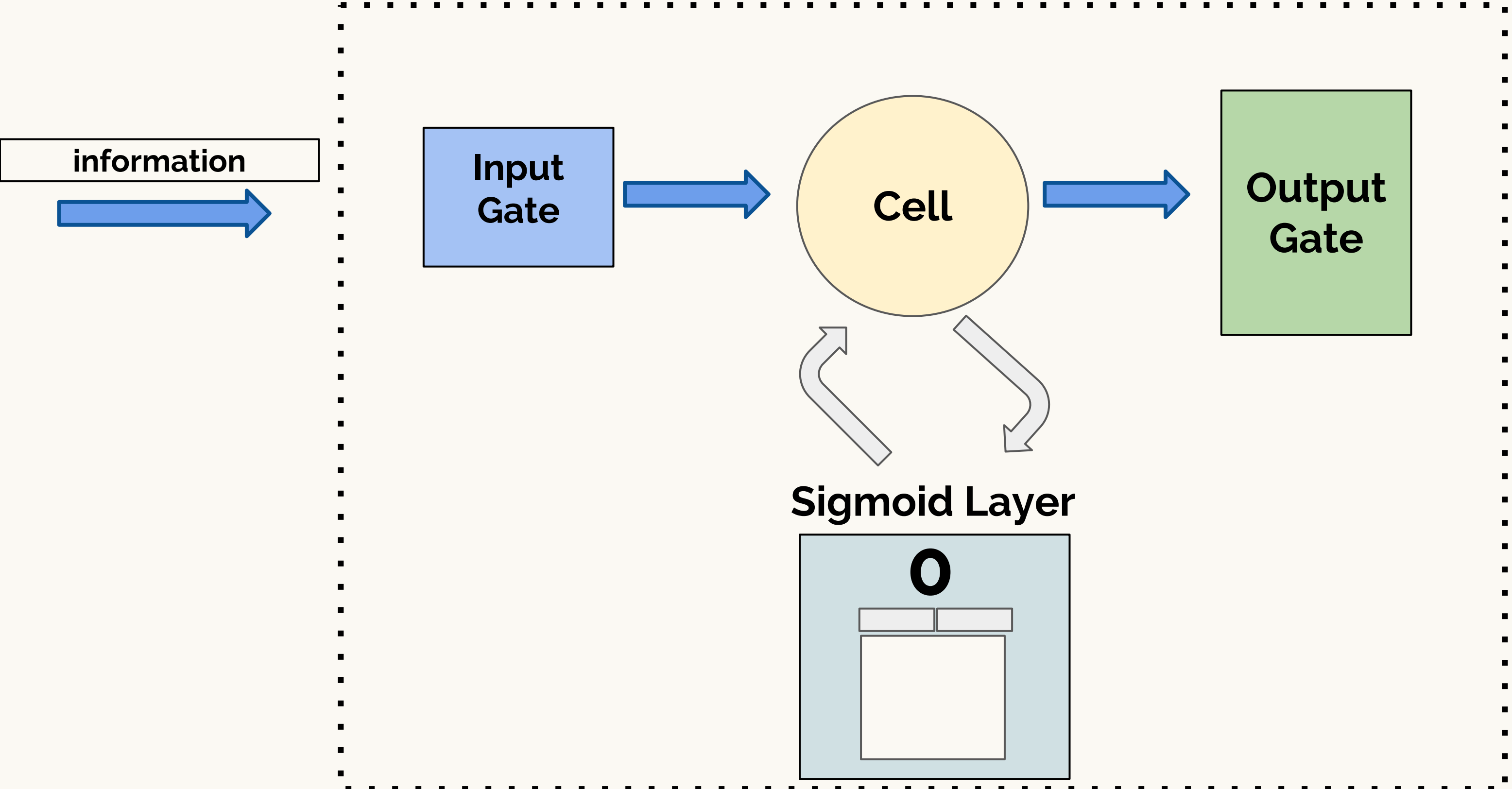
- Based on a Recurrent Neural Network
  - Useful for learning sequential data
  - Model with “memory”

- Help find relevant information within the learned information
  - Genre classification is subjective
  - Found long-term dependencies within each song

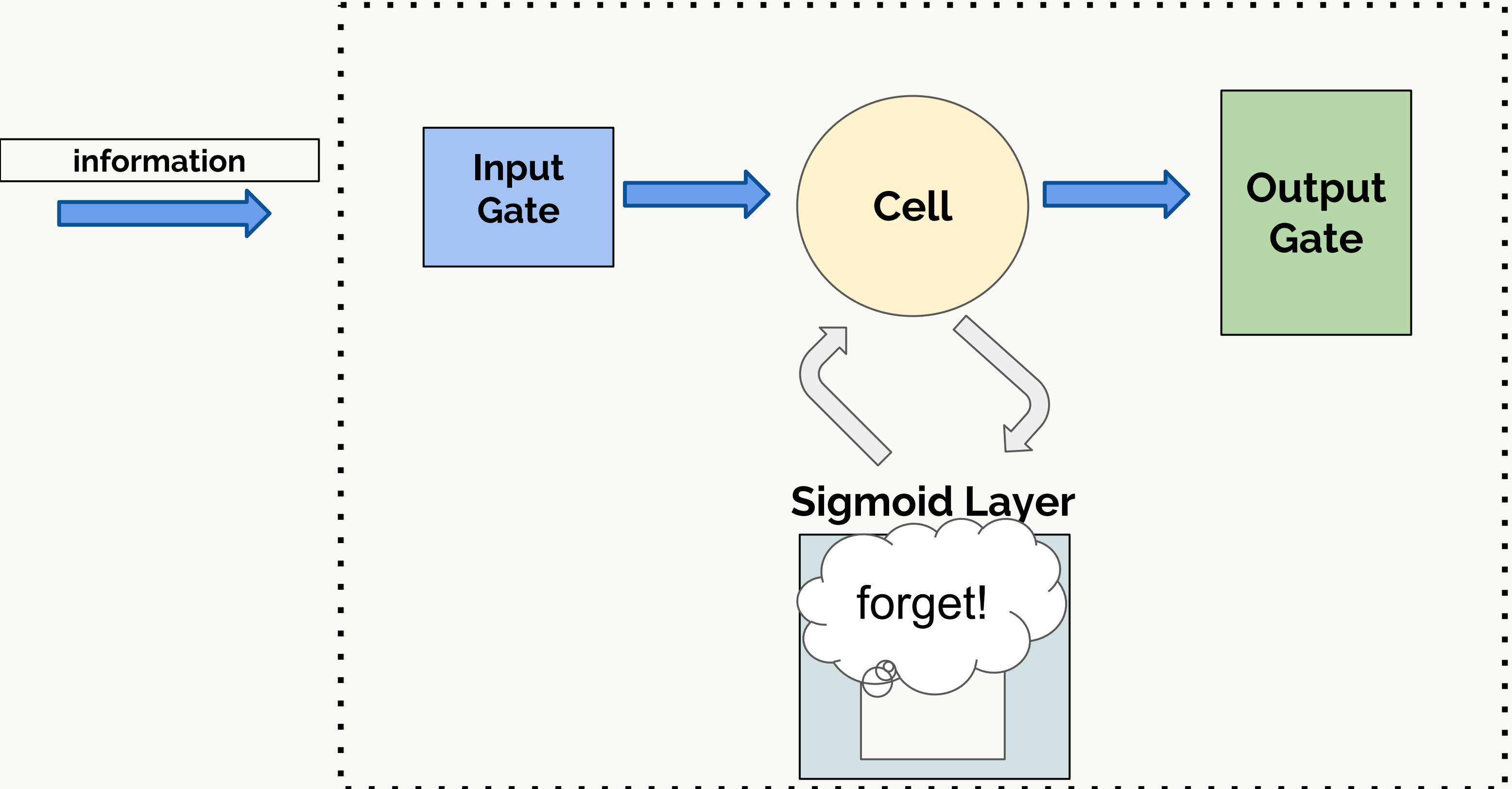
LSTM Network



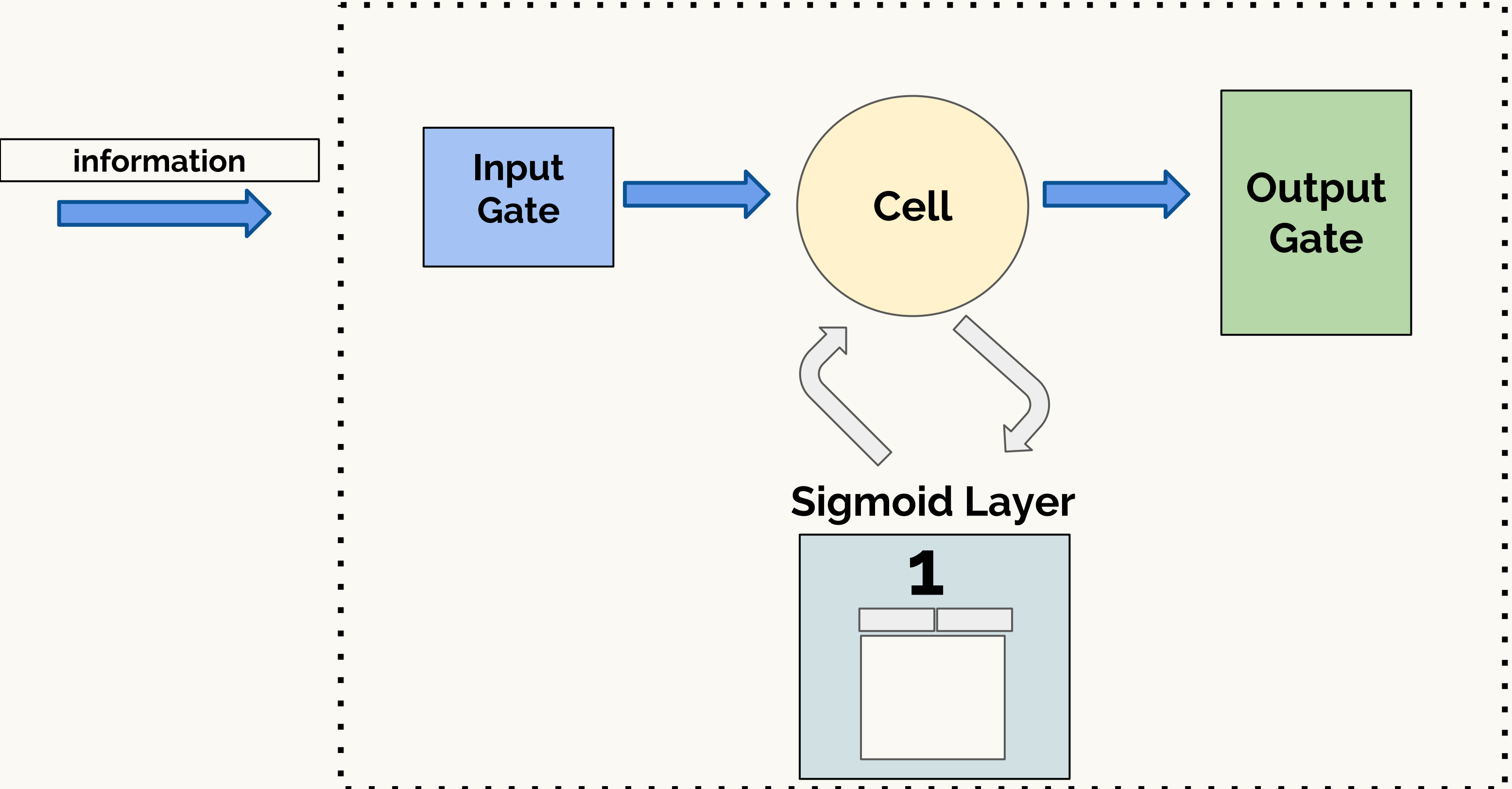
LSTM Network



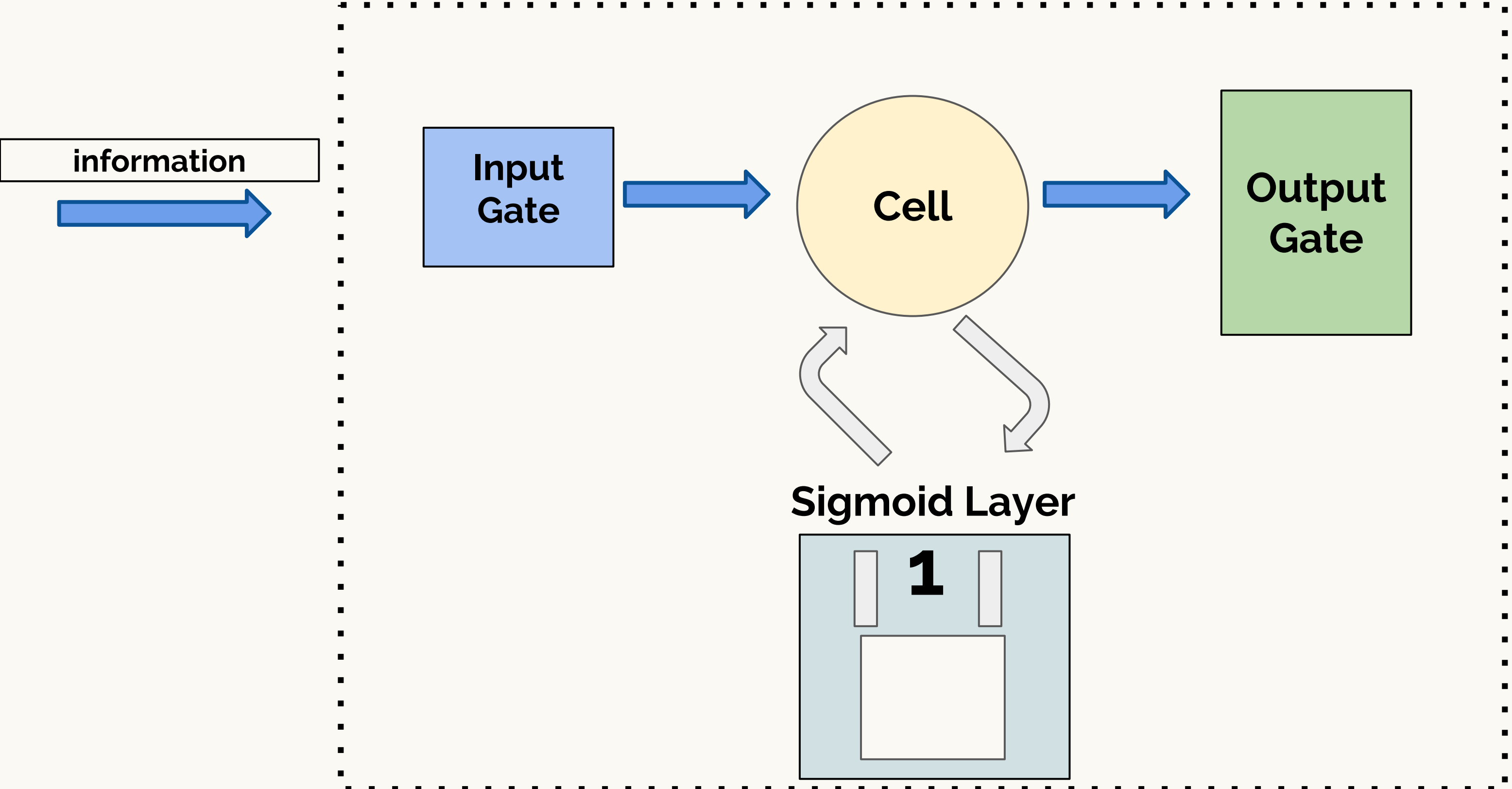
LSTM Network



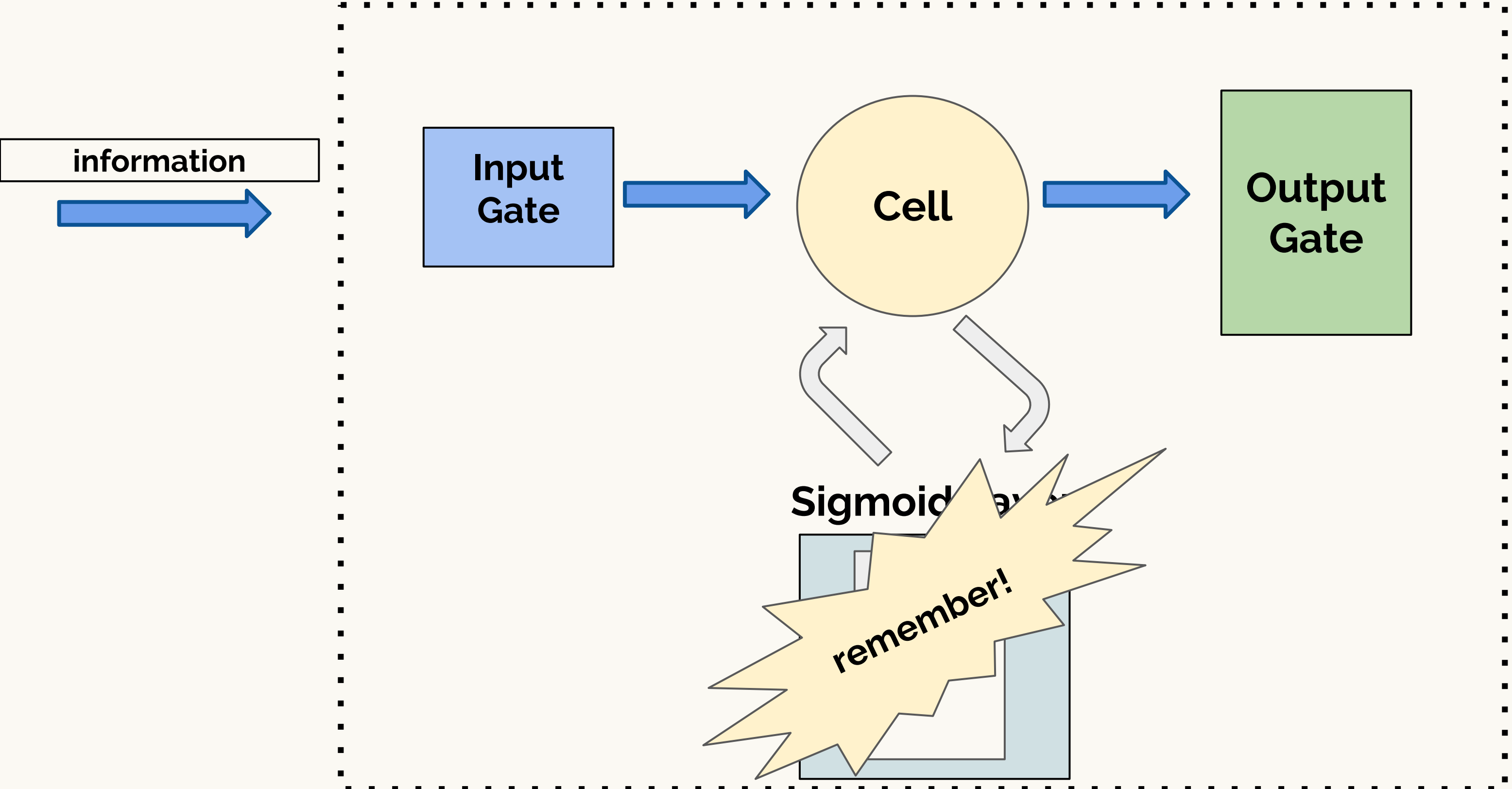
LSTM Network



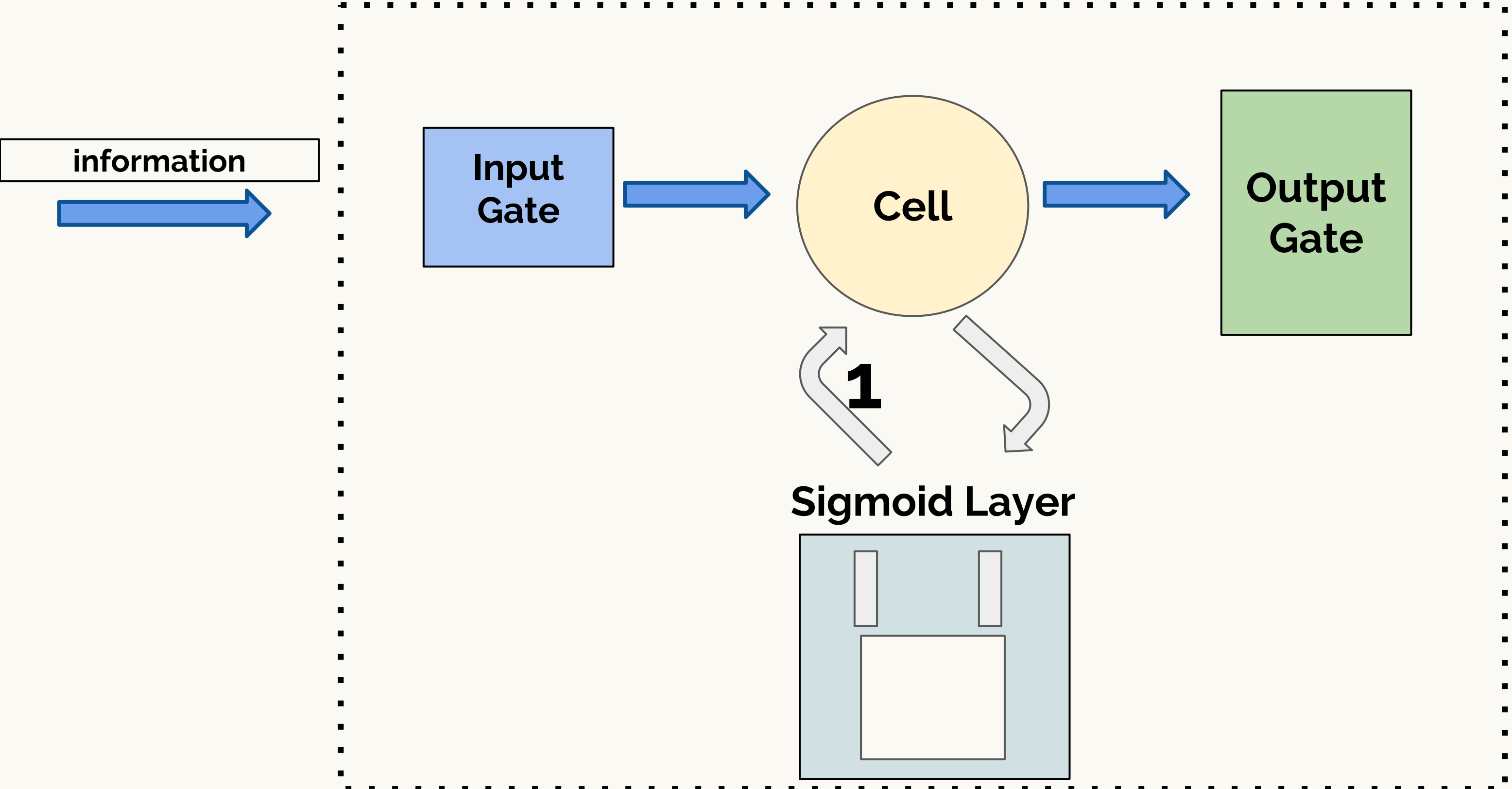
LSTM Network



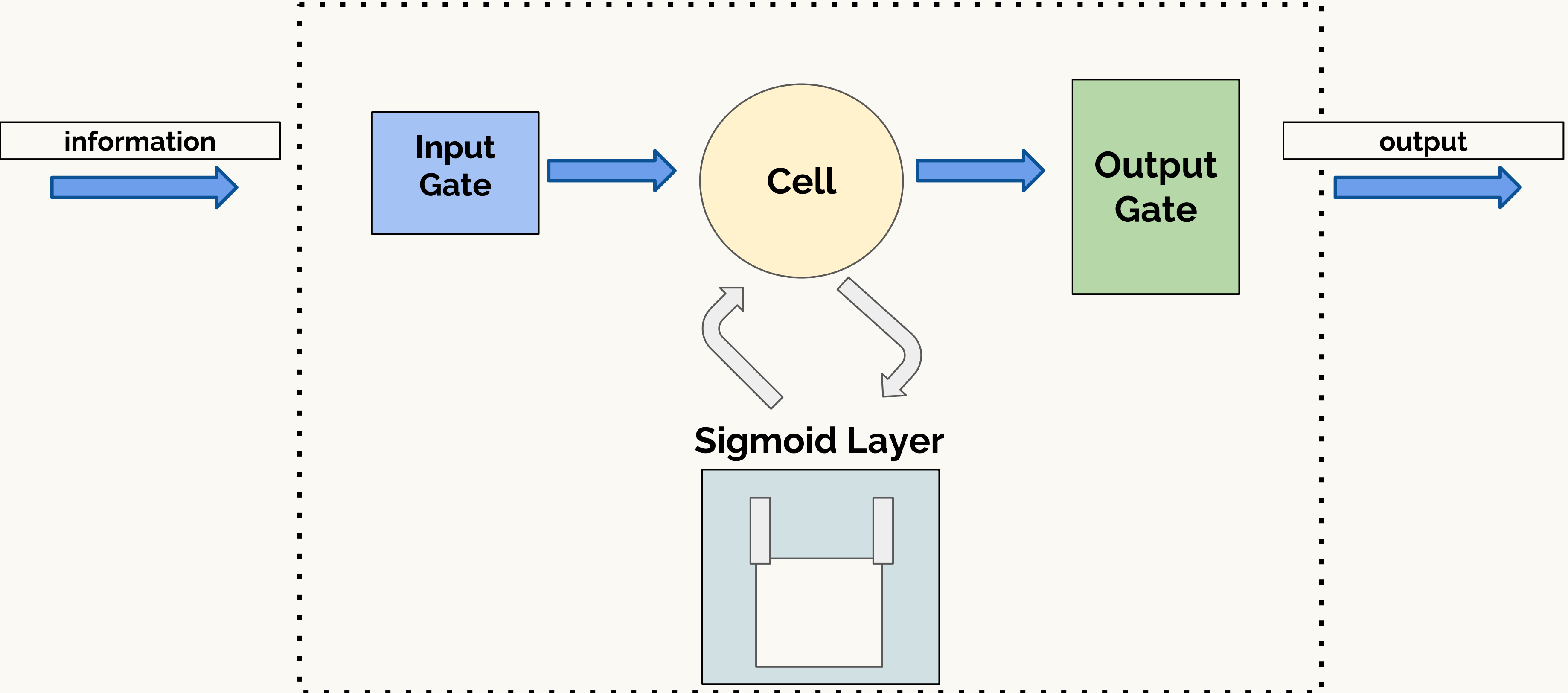
LSTM Network



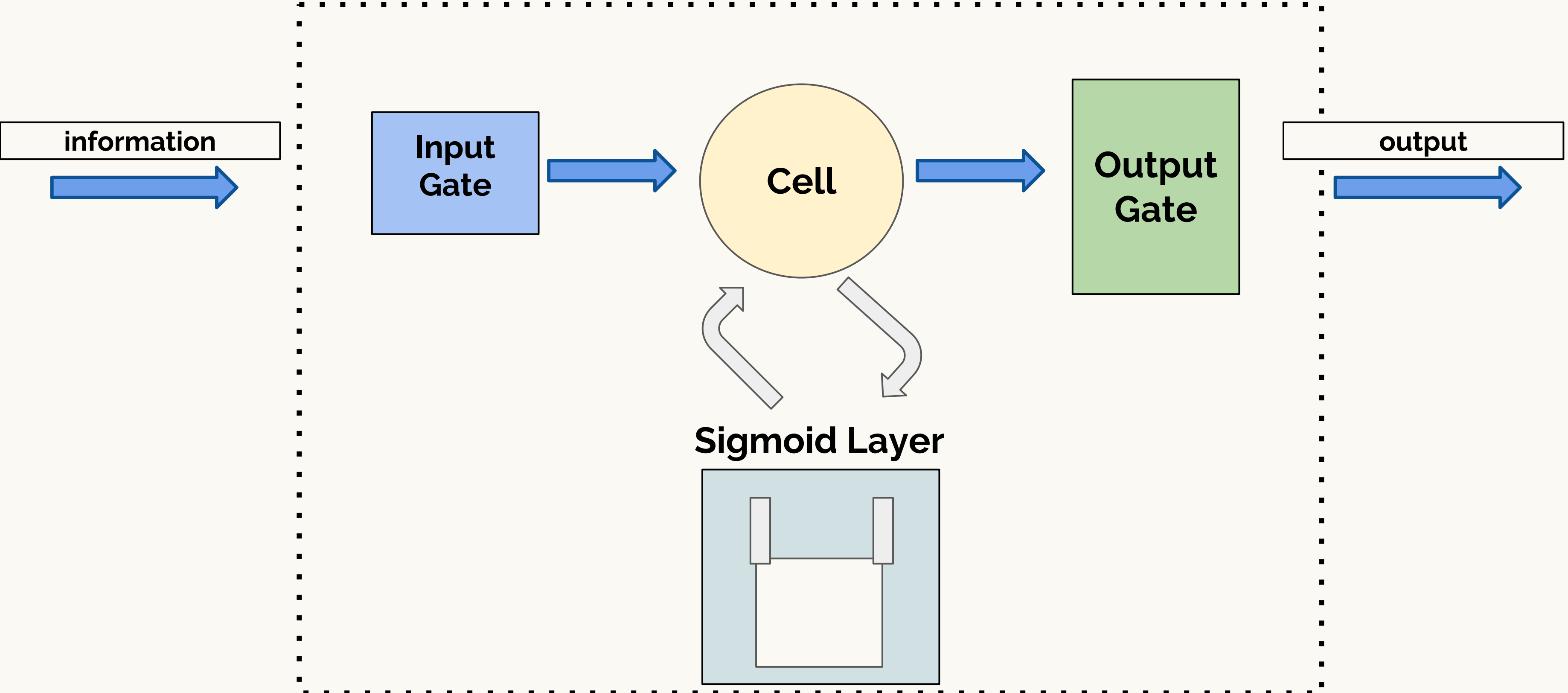
LSTM Network



LSTM Network



LSTM Network



Results: LSTM VS DNN (i)

After running the two models we found similar results:

LSTM

	Batch Size: 35	Batch Size: 100	Batch Size: 150
Training Accuracy:	- **loss**: 0.58 - **acc**: 0.76	- **loss**: 0.65 - **acc**: 0.84	- **loss**: 0.60 - **acc**: 0.79
Test Accuracy:	- **loss**: 0.65 - **acc**: 0.75	- **loss**: 0.54 - **acc**: 0.80	- **loss**: 0.58 - **acc**: 0.76

Results: LSTM VS DNN (i)

After running the two models we found similar results:

LSTM

	Batch Size: 35	Batch Size: 100	Batch Size: 150
Training Accuracy:	- **loss**: 0.58 - **acc**: 0.76	- **loss**: 0.65 - **acc**: 0.84	- **loss**: 0.60 - **acc**: 0.79
Test Accuracy:	- **loss**: 0.65 - **acc**: 0.75	- **loss**: 0.54 - **acc**: 0.80	- **loss**: 0.58 - **acc**: 0.76

Results: LSTM VS DNN (ii)

After running the two models we found similar results:

DNN:

Batch Size: 100 & Training Steps: 5000

Training Accuracy: 0.74

Test Accuracy: 0.72

Results Analysis

Both of our models LSTM & DNN return us similar accuracy levels even though our LSTM is somewhat more precise.

We can perceive that the “memory” and LSTM provides us is able to create a small difference in the precision of our model.

Limits

Main issues we faced:

- Mood Classifications
- How much mood and genre can be associated sometimes

How could we improve our model:

- More data
- More moods
- Lyrics also play a big role

Conclusions

This experiment shows that a mood classification of music is possible and can be done accurately.

The main issue is to converge in the relation between a song and a mood

We think that this experience opens the door to a new way of listening music, a personalized way…

Be happy, be mooderated!

References

- Articles:

- http://ieeexplore.ieee.org/abstract/document/5363083/
- http://ieeexplore.ieee.org/abstract/document/4441720/
- https://github.com/ruohoruotsi/LSTM-Music-Genre-Classification
- https://keras.io/
- https://www.tensorflow.org/get_started/premade_estimators
- http://marsyasweb.appspot.com/download/data_sets/


SURPRISE!
WE HAVE A LIVE DEMO