Temperature Scaling $\left[\frac{p_i^{1/T}}{\sum_j p_j^{1/T}} \right]$

for $T = 0.5$, $[0.6, 0.3, 0.1] \longmapsto [0.78, 0.2, 0.02]$

$0.6^{1/0.5} = 0.36 \mid \sum p_j^{1/0.5} = 0.78$

$0.3^{1/0.5} = 0.09 \mid \sum p_j^{1/0.5} = 0.20$

$0.1^{1/0.5} = 0.01 \mid \sum p_j^{1/0.5} = 0.02$

for $T = 2$, $[0.6, 0.3, 0.1] \longmapsto [0.48, 0.32, 0.2]$

$0.6^{1/2} = 0.77 \mid \sum p_j^{1/2} = 0.48$

$0.3^{1/2} = 0.55 \mid \sum p_j^{1/2} = 0.32$

$0.1^{1/2} = 0.32 \mid \sum p_j^{1/2} = 0.2$

~/xmlMusicGen — -zsh

```
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A3', 'A4', 'B2', 'B4', 'E4b')
Now playing: ('A4',)
Now playing: ('A4',)
Now playing: ('A4',)
Now playing: ('B3', 'B4', 'D4', 'E3', 'G3', 'G4')
Now playing: ('B3', 'B4', 'D4', 'E3', 'G3', 'G4')
Now playing: ('B3', 'B4', 'D4', 'E3', 'G3', 'G4')
Now playing: ('B3', 'B4', 'D4', 'E3', 'G3', 'G4')
Now playing: ('B3', 'B4', 'D4', 'E3', 'G3', 'G4')
Now playing: ('B3', 'B4', 'D4', 'E3', 'G3', 'G4')
Now playing: ('B3', 'B4', 'D4', 'E3', 'G3', 'G4')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('A2', 'A4', 'C4#', 'E4', 'G3')
Now playing: ('G4',)
Now playing: ('G4',)
Now playing: ('A3', 'A4', 'C4#', 'D3', 'F3', 'F4')
Now playing: ('B3', 'D4', 'F3', 'G2', 'G4')
Now playing: ('B3', 'D4', 'G4')
Now playing: ('B3', 'D4', 'G1', 'G4')
Now playing: ('B3', 'D4', 'G1', 'G4')
Now playing: ('B3', 'D4', 'G1', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
Now playing: ('C2', 'C4', 'E4', 'G4')
^CPlayback interrupted by user (Ctrl+C).
Fluidsynth resources cleaned up.
(fooMUSIC) ethansie@Ethans-MacBook-Pro xmlMusicGen %
```

readXML.py | xmlGenera... | matrixMusi... | requiremen... | refactored... | db.py | mxlConvert... | musicGen.py | log.txt | debug.py | forgetTest....

```python
import xmlGenerate
import random
import numpy as np
import fluidsynth
import time

matrix = xmlGenerate.getMatrix()
chord_list = xmlGenerate.getChordList()
chord_index = xmlGenerate.getChordIndex()

# MATRIX MANIPULATION
# Higher = more entropy, Lower = less changes
# Emphasizes/Minimizes the existing row probability vectors
def scale_temperature(matrix, temperature=1.0):
    assert temperature > 0, "Temperature must be positive"
    log_matrix = np.log(matrix + 1e-9)  # Avoid log(0)
    scaled = np.exp(log_matrix / temperature)
    scaled = np.maximum(scaled, 0)
    scaled /= scaled.sum(axis=1, keepdims=True)
    return scaled

def inject_noise(matrix, epsilon=0.01):
    noisy = matrix + epsilon * np.random.rand(*matrix.shape)
    noisy /= noisy.sum(axis=1, keepdims=True)  # Renormalize rows
    return noisy



matrix = scale_temperature(matrix, 0.5)
# matrix = inject_noise(matrix, 0.0001)


# --------------- TRAJECTORY THROUGH THE ROW STOCHASTIC MATRIX ----------
initial = random.choice(chord_list)
generated = [initial]

for _ in range(1000):
    i = chord_index[initial]
    probs = matrix[i]
    j = np.random.choice(len(chord_list), p=probs)
```

matrixMusic.py 29:37 LF UTF-8 Python main Fetch GitHub Git (1)

Moon River (low temp)