

1 Introduction

1.1 Charts

Our goal is to achieve automatic chart generation for the rythm game Pump It Up. We have at our disposal a dataset which contains all the charts from the official game as SSC files (1'423 simfiles). Each chart contains multiple notes (taps and holds) on different time, taps are punctual notes and holds have a begining and an end. We have 13'598 valid charts out of 15'301 charts.

We represent them by $N \times 20$ arrays where the 10 first dimensions are used for taps and the last 10 ones are used for holds : they indicate that the corresponding arrow should be pressed. This is a simplification, we don't handle change of BPM, gimmicks, and any other complicated stuff. The same chart can also be represented as a list $l \in [0..L-1]^N$ where L denotes the number of combination of notes when it will be useful. Here we have $L = 3'622$, and therefore $L^2 = 13'118'884$.

1.2 Goal

Given a mini-chart $x \in \{0, 1\}^{K \times 20}$ we want to make a full chart $\hat{x} \in \{0, 1\}^{N \times 20}$ such that $\forall k \in [1..K], \hat{x}_k = x_k$ and \hat{x} seems like any other chart. It is difficult to define it properly but we will refine what we want later. But at least we can say that this isn't a Markov's chain because there is a heavy time dependance, but we will tackle this issue. However, we can still define $P \in [0, 1]^L$ and $T \in [0, 1]^{L \times L}$ the matrices that give the probability for each combination and couple of combinations. Right after a tap, or just after the begining of a hold, we know exactly which combination will be produced by the chart, and generating it will lead to infinite loops that we would like to avoid.

1.2.1 Tempo detector

That's why first we are going to see if we could determine how many arrows will be pressed at each timestep, using a set $S = [0..4]^2$ to represent the number of taps and holds at a given time. We can even go further and reduce S into S' the couples with a sum at most equal to 4, and $|S'| = 5 + 4 + 3 + 2 + 1 = 15$. If we want to go from $(i, j) \in S'$ to a single class $k \in [0..14]$? $k(i, j) = j + 5i - \frac{i(i-1)}{2}$.

1.2.2 Next notes (projet Alpha)

First we need to fix $n = 63$, we use n steps inside an LSTM in order to guess the next number of notes and the play style. Then we take the last $n' = 15$ steps inside a transformer in order to estimate the next notes, we give the number of notes as a hint, with the play style and the proper level of difficulty. The previous section gives us how to compute the number of notes. The level of difficulty is given by the dataset. We define the play style as the remaining information, but we want this information to be time-invariant. We sample $k \sim \mathcal{U}([1, K])$ and then we ask that $x_{1..n}$ should have the same play style than $x_{1-k..n-k}$. It would be interesting to train this playstyle through a contrastive loss.

J'ai besoin d'un dataset qui me donne $n + k + 1$ éléments : le label de la configuration, la configuration, l'élément dans S' associé, le niveau, l'information double/single,

2 Version GOLD

On se contente d'un seul transformeur mais qui serait ultra performant. On va nommer les représentations. La représentation brute c'est celle avec les 20 canaux ($x \in \{0, 1\}^{N \times 20}$). La représentation des états sera celle avec les $L = 3109$ configurations, donc $u(x) = [0..L - 1]^N$. On a également des projections, par exemple on peut compter les touches $c(x) = [0..20]^N$, mais on peut décomposer aussi en fonction du nombre de taps et de holds, $t(x) = [0..10]^N$ et $h(x) = [0..10]^N$, avec $c(x) = t(x) + h(x)$. On peut aussi projeter sur le pad, $p(x) = \{0, 1\}^{N \times 10}$.

On a enfin la représentation latente obtenue par un réseau de neurones $f_\theta : \{0, 1\}^{20} \rightarrow \mathbb{R}^d$, elle permet de reconstruire chaque projection fidèlement, par rapport à la distribution des configurations. Il sera utile de l'entraîner correctement.

Mais on peut complètement détruire la structure temporelle pour se contenter des touches. Cela nous permet d'optimiser les absences de notes.