2018 Fall

CTP431: Music and Audio Computing

Automatic Music Generation

Graduate School of Culture Technology, KAIST
Juhan Nam

Outlines

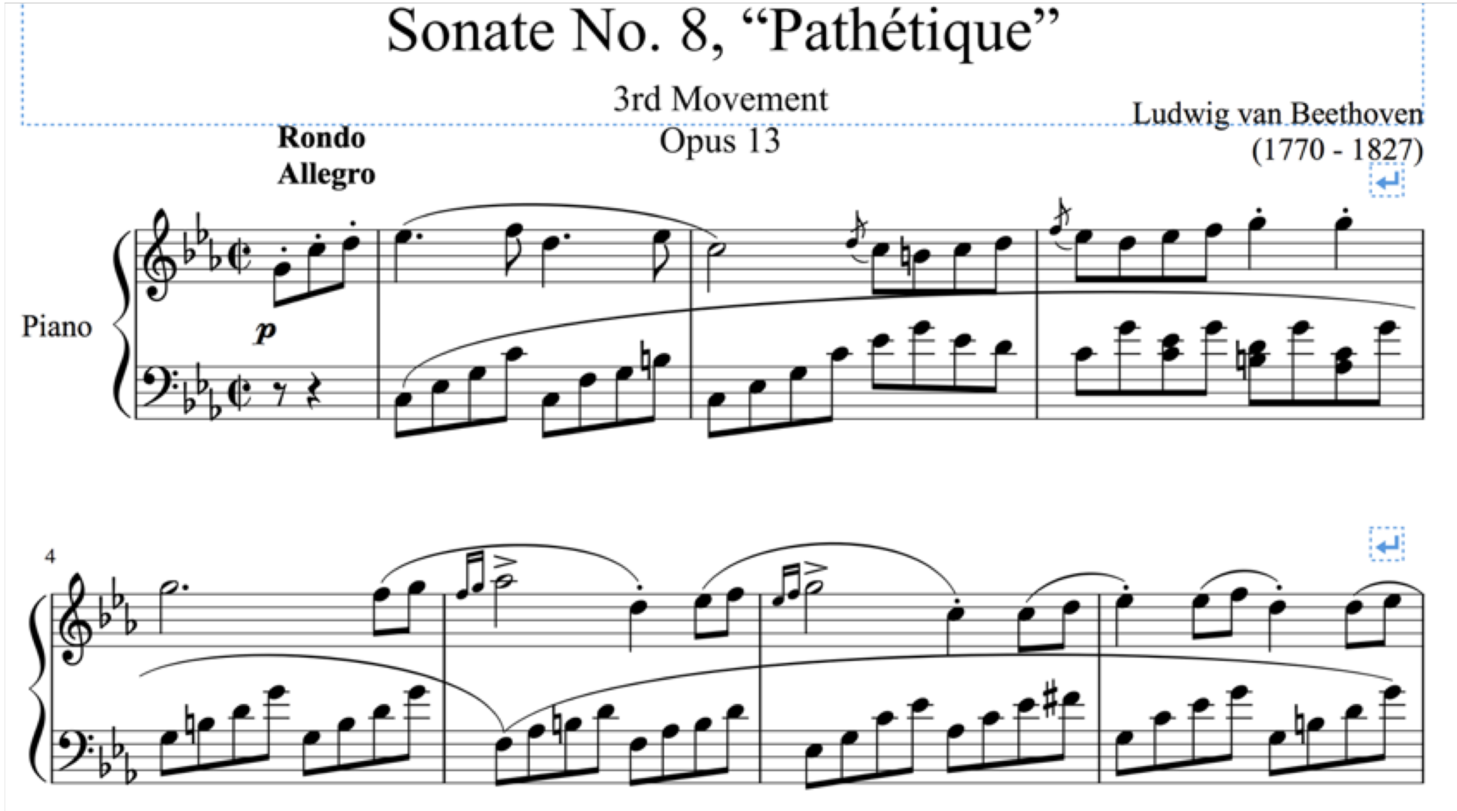
- Early Approaches
  - Markov Models
  - Recombinant Models
  - Cellular Automata
  - Genetic Algorithm

- Recent Advances
  - Neural Networks

- Interactive music generation

Symbolic Music

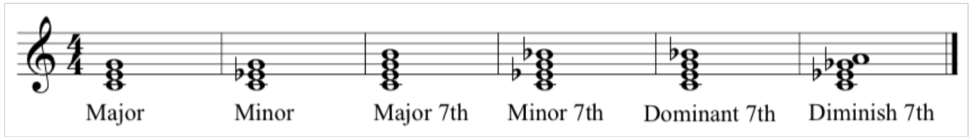
- Symbolic music is represented as a sequence of notes



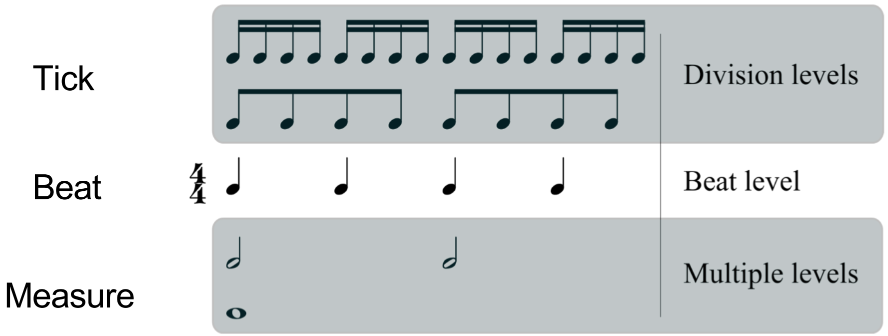
Symbolic Music

- Music is structured sequential data

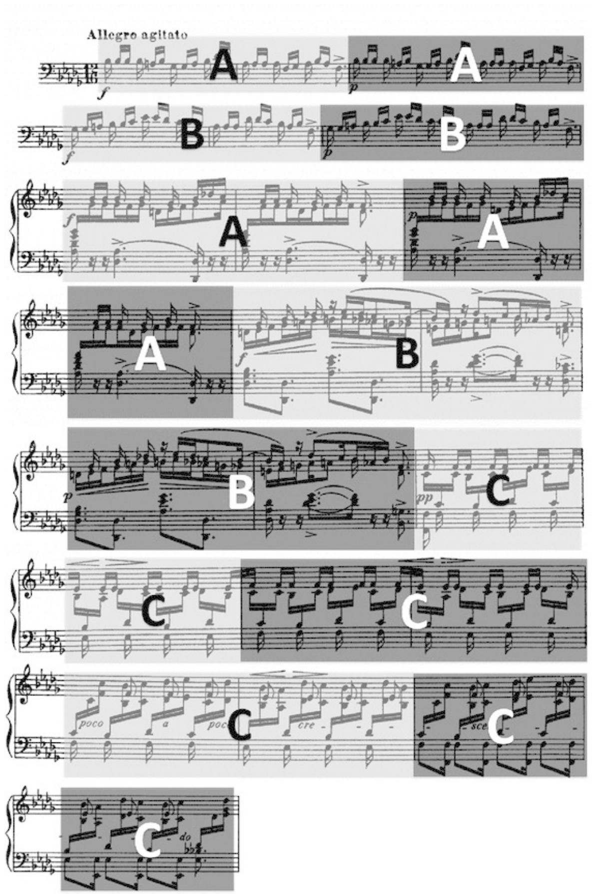
Scale



Harmony



Rhythm



Form

Li et. al, “The Clustering of Expressive Timing Within a Phrase in Classical Piano Performances by Gaussian Mixture Models”, CMMR, 2015

Symbolic Music

- Musical notes are temporally dependent
  - Note-level
  - Beat-level
  - Measure-level

Markov Model

- A random variable q has N states $(S_1, S_2, \ldots, S_N)$ and, at each time step, one of the states are randomly chosen: $q_t \in \{S_1, S_2, \ldots, S_N\}$

- The probability distribution for the current state is determined by the previous state(s)
  - The first-order Markov model: $P(q_t|q_1, q_2, \ldots, q_{t-1}) = P(q_t|q_{t-1})$
  - The second-order Markov model: $P(q_t|q_1, q_2, \ldots, q_{t-1}) = P(q_t|q_{t-1}, q_{t-2})$

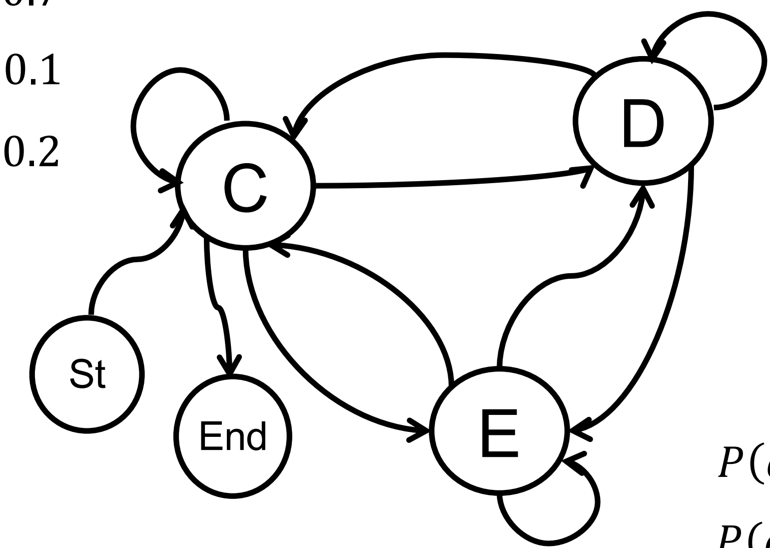
Markov Model

- Example: simple melody generation
  - $q_t \in \{C, D, E\}$
  - The transition probability matrix 3 by 3

$P(q_t = C | q_{t-1} = C) = 0.7$

$P(q_t = D | q_{t-1} = C) = 0.1$

$P(q_t = E | q_{t-1} = C) = 0.2$



$P(q_t = C | q_{t-1} = D) = 0.2$

$P(q_t = D | q_{t-1} = D) = 0.6$

$P(q_t = E | q_{t-1} = D) = 0.2$

$P(q_t = C | q_{t-1} = E) = 0.3$

$P(q_t = D | q_{t-1} = E) = 0.1$

$P(q_t = E | q_{t-1} = E) = 0.6$

Markov Model

- The transition matrix can be learned from data
  - Dancing Markov Gymnopédies: https://codepen.io/teropa/pen/bRqYVj/

- Generated music
  - Learned with Satie's "Gymnopédies" and "Trois Gnossiennes”
  - https://www.youtube.com/watch?v=H3xgdDTvvlc
  - Learned with Bach's "Toccata and Fugue in D minor" (BWV 565)
  - https://www.youtube.com/watch?v=IIOiAK0x4vA

Example: Illiac Suite

- The first computer-generated composition (1956)
  - Lejaren Hiller and Leonard Issacson
  - They used Markov models of variable order to select notes with different lengths

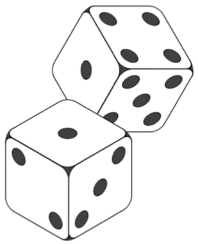
- Music
  - https://www.youtube.com/watch?v=n0njBFLQSk8&list=PLIVblwUBdcStsNpl0v4OCbC5k-mIDcyaR

Recombinant Music

- Musical Dice Game
  - Generate from pre-composed small pieces by random draws
  - The table of me preserves musical “style”

Erster Theil. Premiere Partie.

	A	B	C	D	E	F	G	H
2	96	22	141	41	105	122	11	30
3	32	6	128	63	146	46	134	81
4	69	95	158	13	153	55	110	24
5	40	17	113	85	161	2	159	100
6	148	74	163	45	80	97	36	107
7	104	157	27	167	154	68	118	91
8	152	60	171	53	99	133	21	127
9	119	84	114	50	140	86	169	94
10	98	142	42	156	75	129	62	123
11	3	87	165	61	135	47	147	33
12	54	130	10	103	28	37	106	5



Zweiter Theil. Seconde Partie.

	A	B	C	D	E	F	G	H
2	70	121	26	9	112	49	109	14
3	117	39	126	56	174	18	116	83
4	66	139	15	132	73	58	145	79
5	90	176	7	34	67	160	52	170
6	25	143	64	125	76	136	1	93
7	138	71	150	29	101	162	23	151
8	16	155	57	175	43	168	89	172
9	120	88	48	166	51	115	72	111
10	65	77	19	82	137	38	149	8
11	102	4	31	164	144	59	173	78
12	35	20	108	92	12	124	44	131

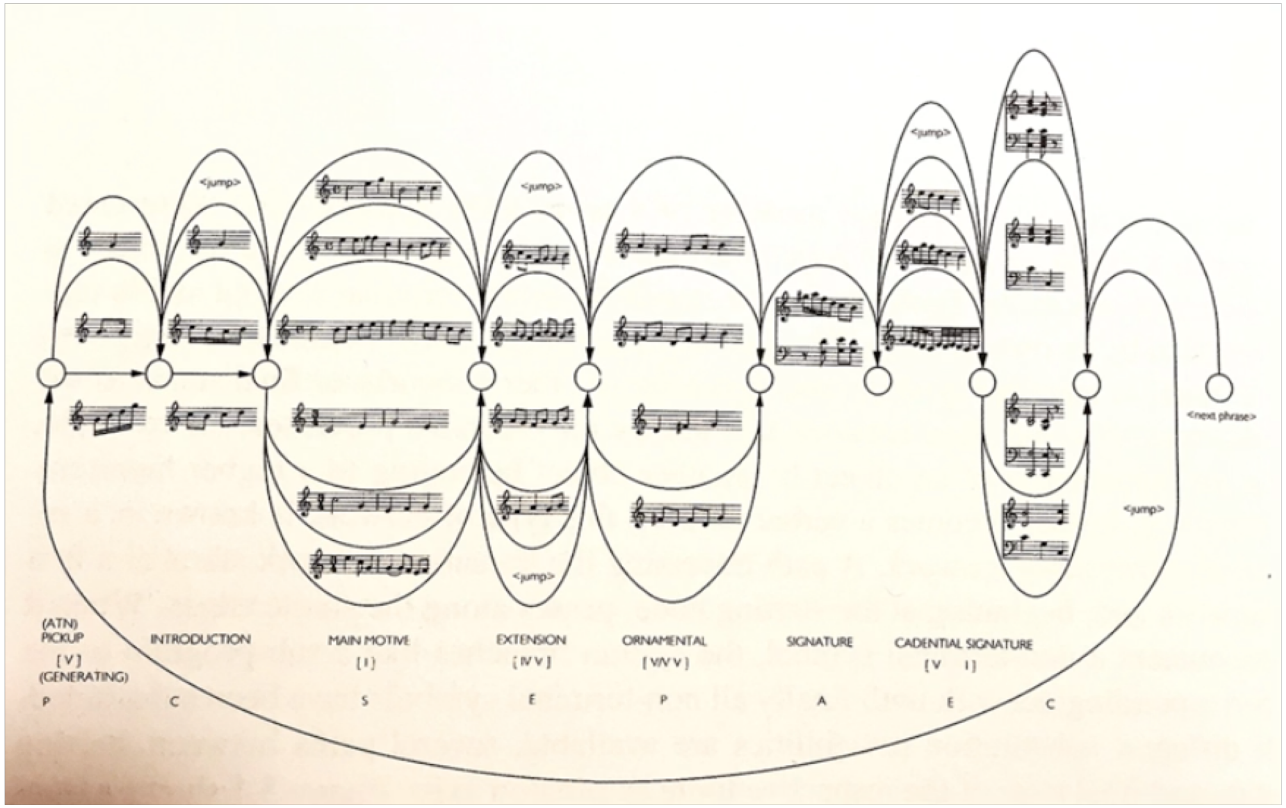
11^{16} = 45,949,729,863,572,161 variations

https://imslp.org/wiki/Musikalisches_W%C3%BCrfelspiel,_K.516f_(Mozart,_Wolfgang_Amadeus)

Mozart K. 516F

Recombinant Music

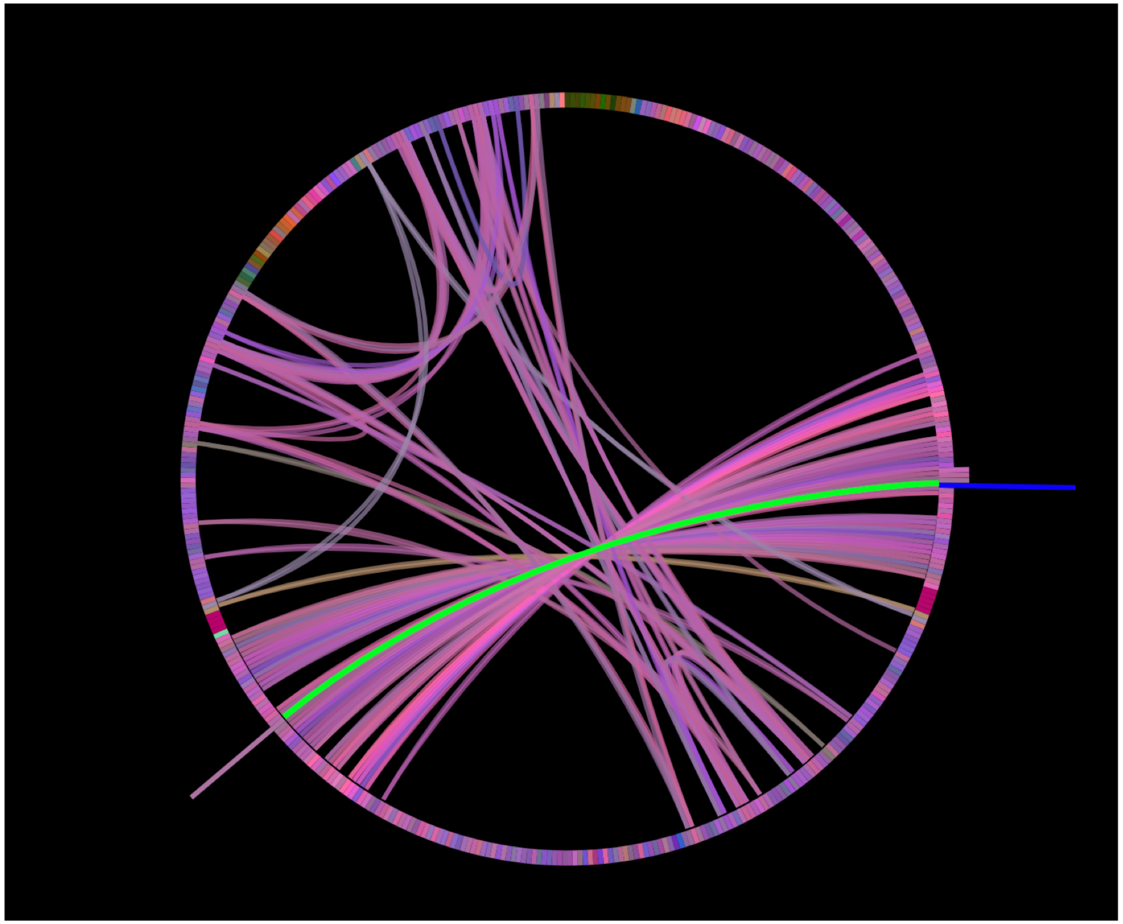
- David Cope's Experiments in Musical Intelligence (EMI)
  - Segment and reassemble existing pieces of music by pattern matching
  - Create a new piece of music that preserves the style of the original



Augmented Transition Networks (David Cope)

Infinite Jukebox

- Music mash-up using beat-level self-similarity within a song



http://infinitejukebox.playlistmachinery.com/

"In C"

- Ted Riley's ensemble music
  - Also called "Minimal music"

"In C"
by Terry Riley

Instruction for beginners

1 Any number of people can play this piece on any instrument or instruments (including voice).

2 The piece consists of 53 melodic patterns to be repeated any amount of times. You can choose to start a new pattern at any point. The choice is up to the individual performer! We suggest beginners are very familiar with patterns 1-12.

3 Performers move through the melodic patterns in order and cannot go back to an earlier pattern. Players should try to stay within 2-3 patterns of each other.

4 If any pattern is too technically difficult, feel free to move to the next one.

5 The eighth note pulse is constant. Always listen for this pulse. The pulse for our experience will be piano and Orff instruments being played on the stage.

6 The piece works best when all the players are listening very carefully. Sometimes it is better to just listen and not play. It is important to fit into the group sound and understand how what you decide to play affects everybody around you. If you play softly, other players might follow you and play soft. If you play loud, you might influence other players to play loud.

7 The piece ends when the group decides it ends. When you reach the final pattern, repeat it until the entire group arrives on this figure. Once everyone has arrived, let the music slowly die away.

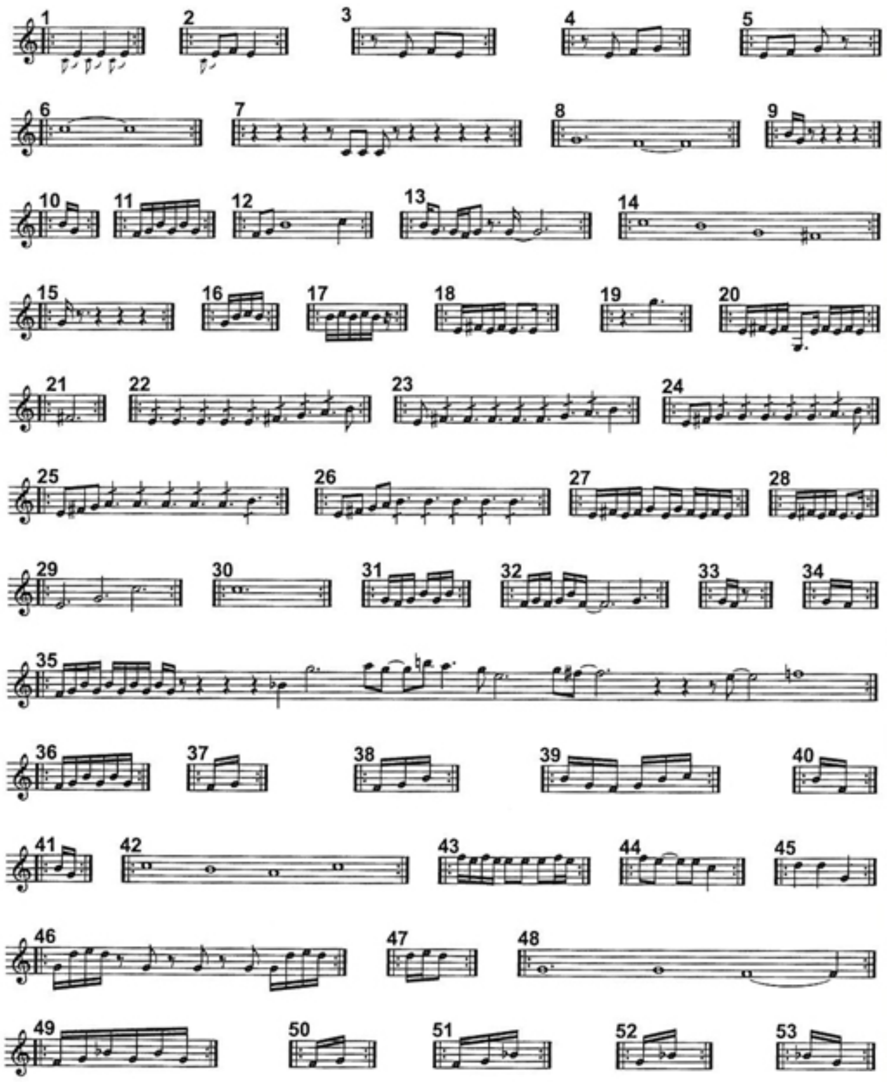


Figure 1.1. Score of *In C* (copyright Terry Riley, 1964).

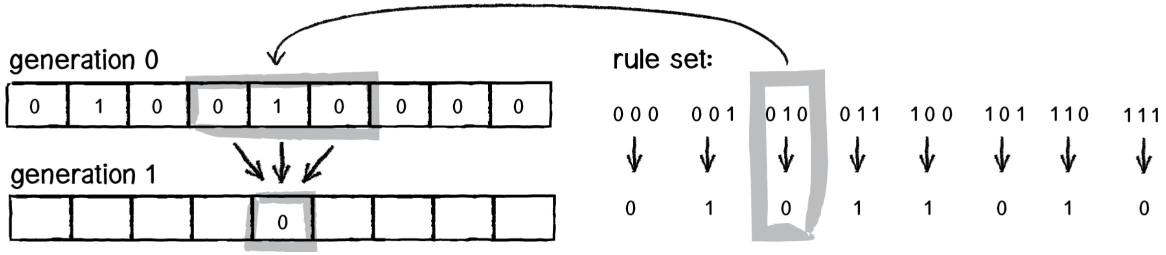
Source: https://www.musicinst.org/sites/default/files/attachments/In%20C%20Instructions%20for%20Beginners.pdf

Source: https://nmbx.newmusicusa.org/terry-rileys-in-c/

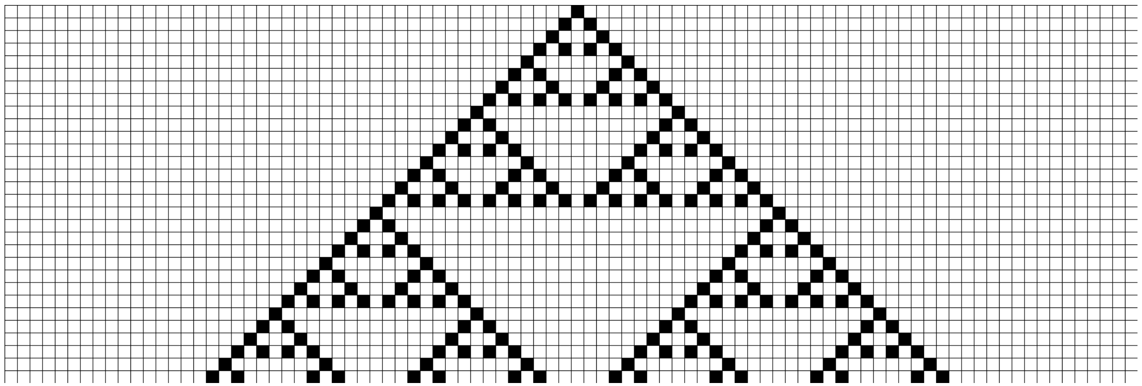
https://www.youtube.com/results?search_query=Terry+Riley+In+C

Cellular Automata

- A cell-based state evolution model
  - Determines the **state** of each **cell** using **neighbors** and **a rule set**
  - A Wolfram model example:



"Rule 90"

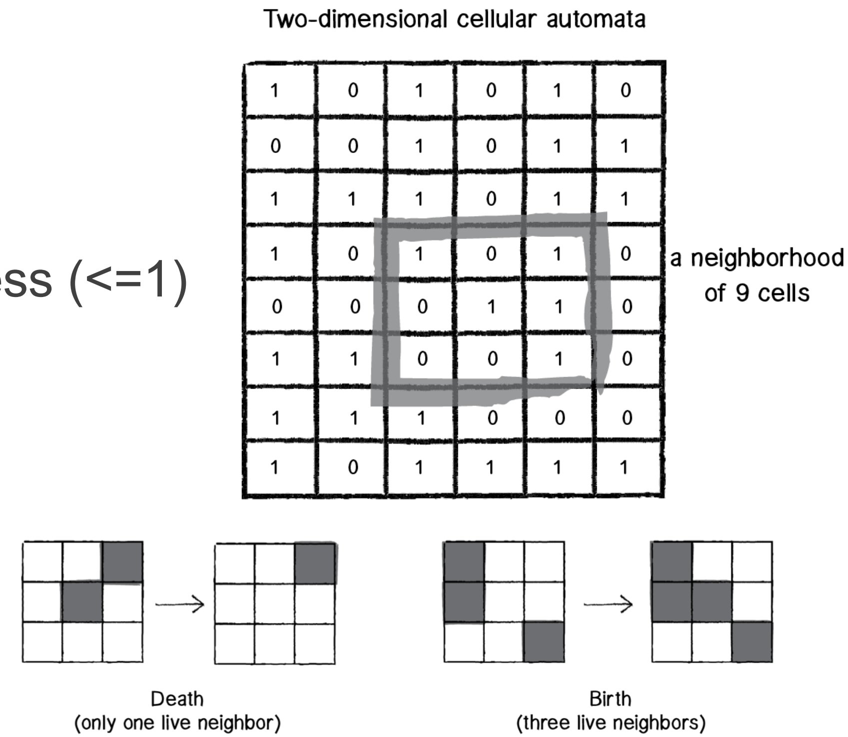


  - Related to self-replicating patterns in biology

Source: https://natureofcode.com/book/chapter-7-cellular-automata/

Conway’s Game of Life

- 2D cellular automata
  - Rules of life
    - Death (1→0) : overpopulation (>=4) or loneliness (<=1)
    - Birth (0→1) : 3 neighbors are alive
    - Otherwise, stay in the same state

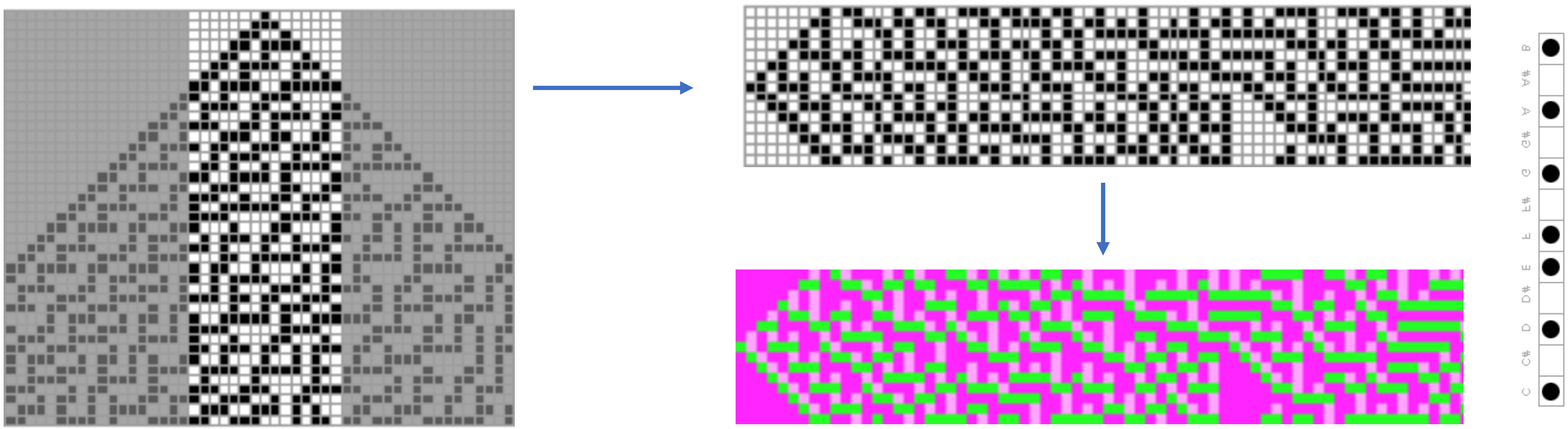


Source: https://natureofcode.com/book/chapter-7-cellular-automata/

- Demos:
  - http://www.cappel-nord.de/webaudio/conways-melodies/
  - http://nexusosc.com/gameofreich/
  - http://blipsoflife.herokuapp.com/

WolframTones

- Automatic music generation system based on cellular automata



Mapping to musical notes by rules

- Demo: http://tones.wolfram.com/generate

Statistical Models

- As aforementioned, music is highly structure sequence data. Thus, we can model the sequence using **an auto-regressive model**.

q_1 q_2 q_3 q_4 ...

$$\boxed{p(q_t|q_1, \dots, q_{t-1})}$$

q_t: note features

- In the first-order Markov model, it was simplified to $p(q_t|q_{t-1})$
  - However, it explains only short-term relations among notes

- Can we model the long-term relations using more complicated model?

Toy Example

$$3 + 5 = 18$$

$$4 + 4 = 20$$

$$6 + 7 = 48$$

$$8 + 9 = 80$$

$$9 + 10 = ?$$

Note that “+” is not addition here

Toy Example

3 + 5 = 18

4 + 4 = 20

6 + 7 = 48

8 + 9 = 80

9 + 10 = ?

$$y = f(x_1, x_2)$$

$$y = x_1 \times (x_2 + 1)$$

Note that “+” is not addition here

Toy Example

$$2 + 2 = 6$$

$$3 + 6 = 12$$

$$4 + 5 = 19$$

$$6 + 10 = 40$$

$$7 + 18 = ?$$

Note that “+” is not addition here

Toy Example

2 + 2 = 6

3 + 6 = 12

4 + 5 = 19

6 + 10 = 40

7 + 18 = ?

$$y = f(x_1, x_2)$$

$$y = \sqrt{x_1 + x_2} + {x_1}^2$$

Note that "+" is not addition here

Neural Network

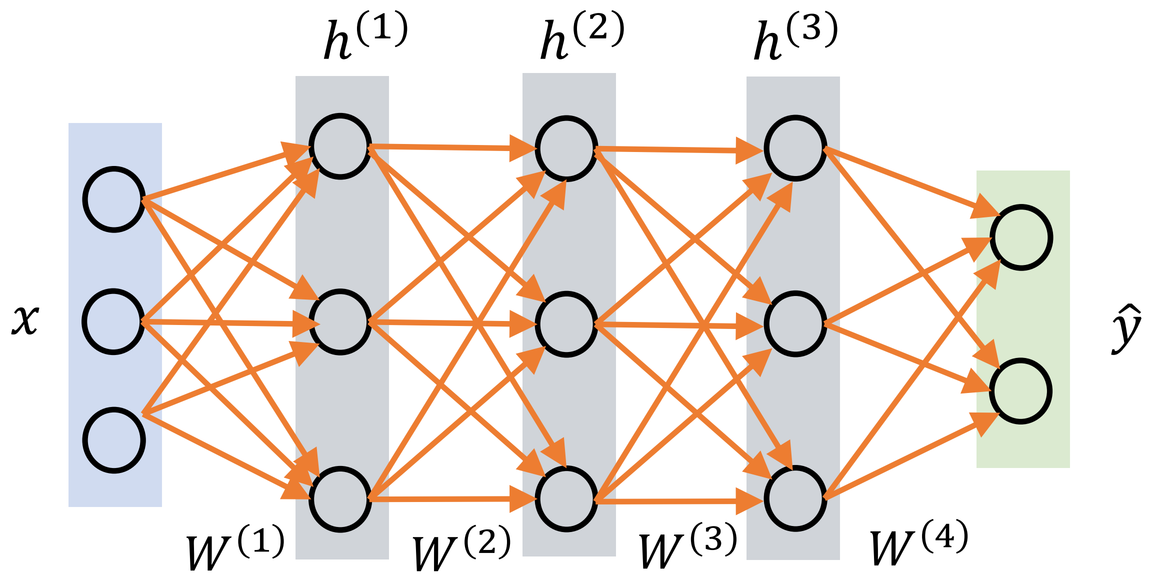
- A learning model based on multi-layered networks
  - The basic model (MLP) is composed of linear transforms and element-wise nonlinear functions

$$h^{(1)} = g^{(1)}(W^{(1)}x + b^{(1)})$$

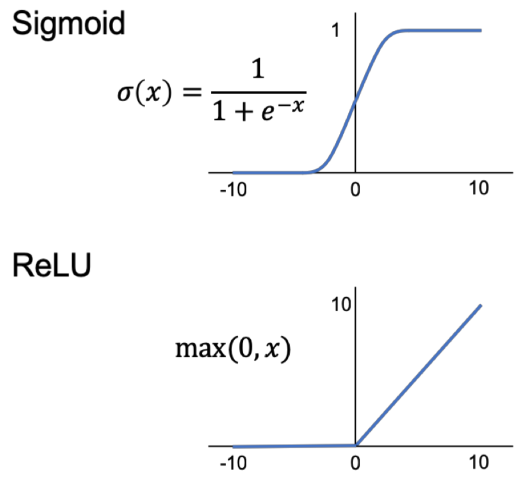
$$h^{(2)} = g^{(2)}(W^{(2)}h^{(1)} + b^{(2)})$$

$$h^{(3)} = g^{(3)}(W^{(3)}h^{(2)} + b^{(3)})$$

$$\hat{y} = \sigma(h^{(3)})$$



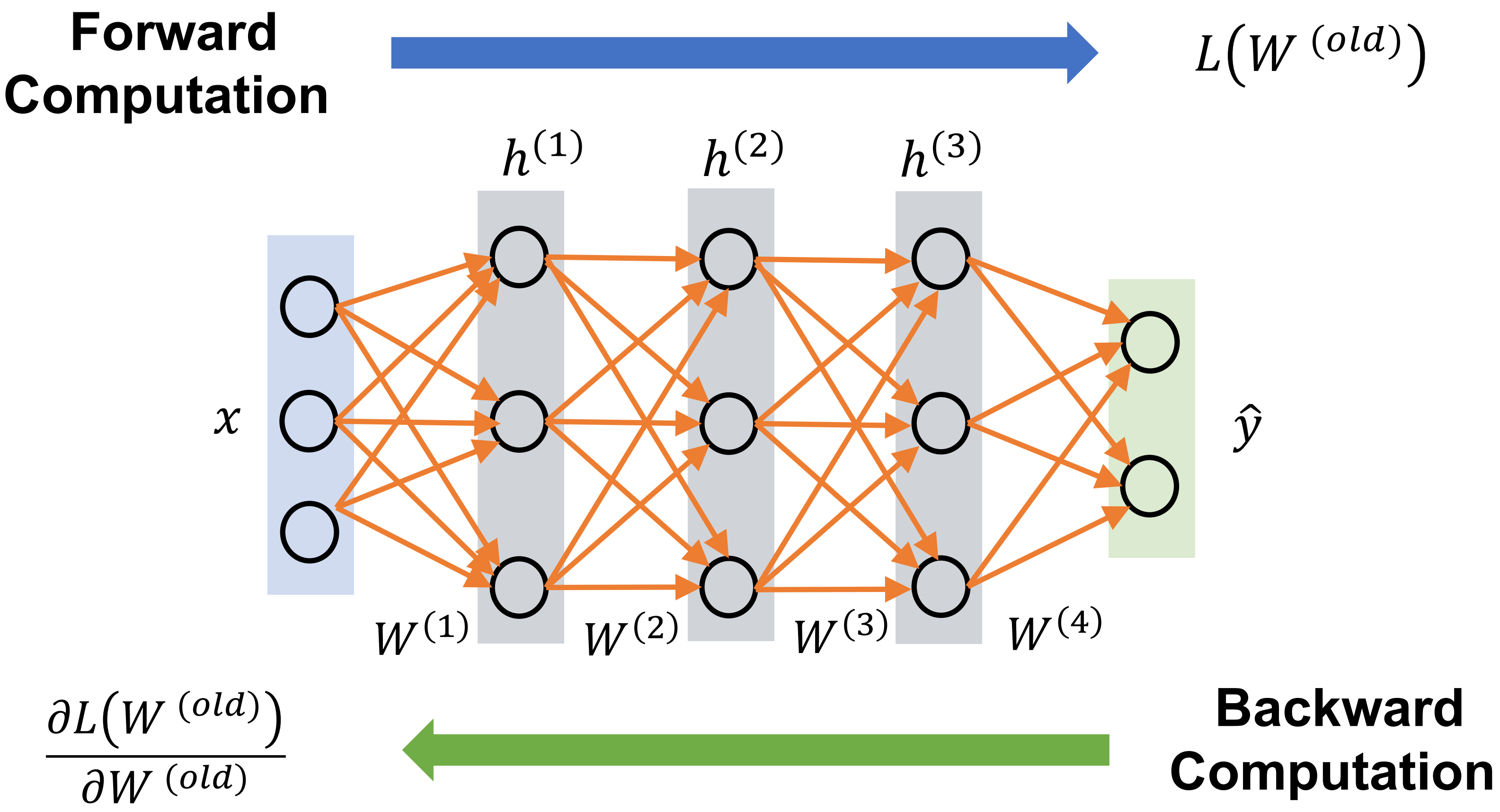
Multi-Layer Perceptron (MLP)



Non-linear functions

Neural Network

- The Neural network is trained via error back-propagation



Gradient descent

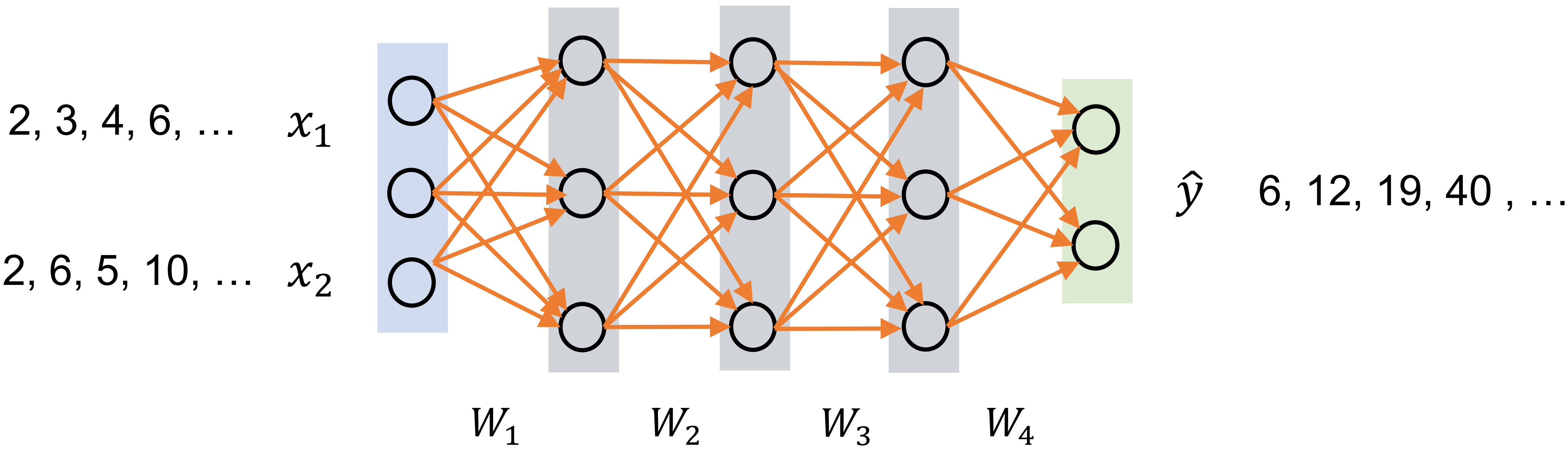
$$\boxed{W_{ij}^{(new)} = W_{ij}^{(old)} - \mu \frac{\partial L\left(W_{ij}^{(old)}\right)}{\partial W_{ij}^{(old)}}}$$

MLP Demo and visualization

- https://playground.tensorflow.org

The Toy Example

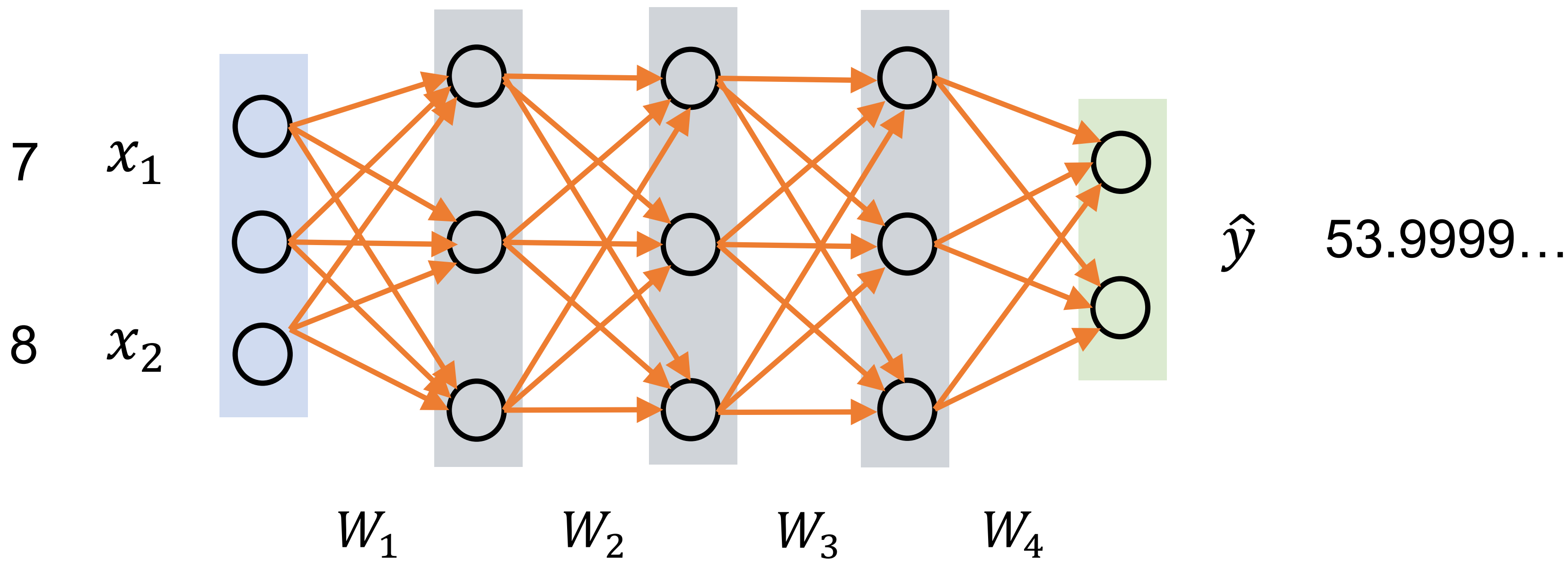
- The neural network can learn highly complicated relations between input and output



$$\boxed{W_i^{\,new} \leftarrow W_i^{\,old} - \mu \frac{\partial \|\hat{y} - y\|}{\partial W_i}}$$

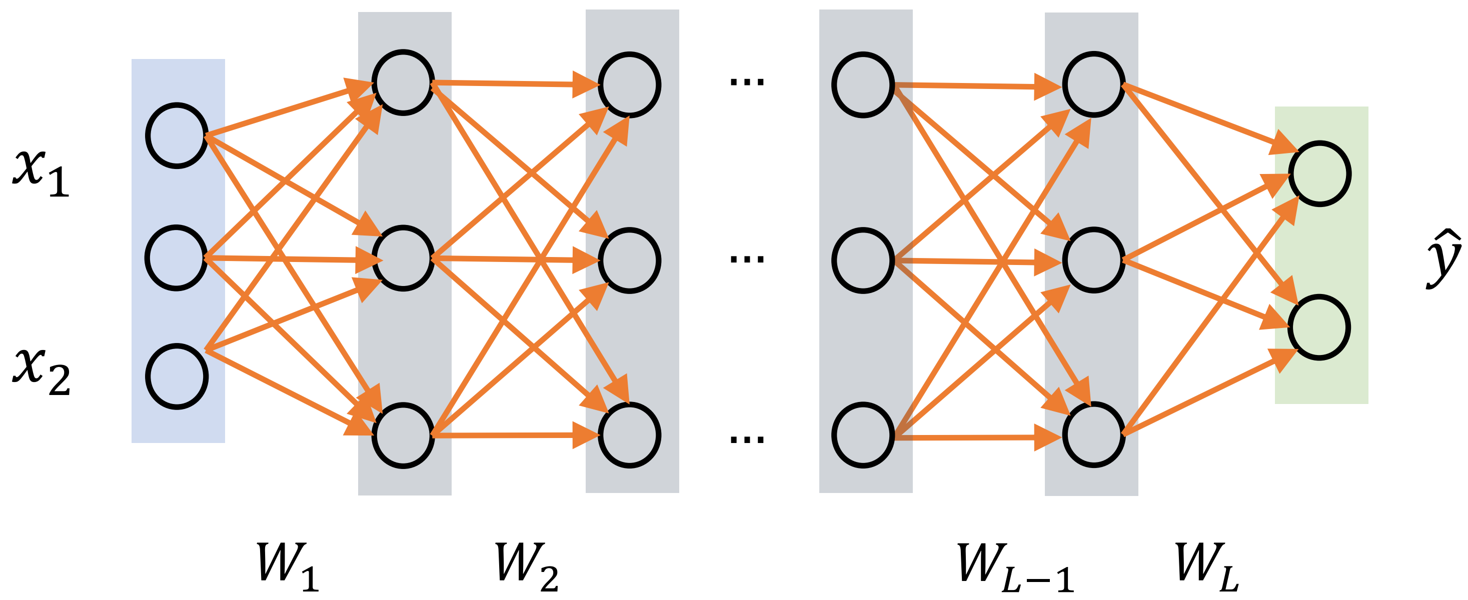
The Toy Example

- The neural network can learn highly complicated relations between input and output



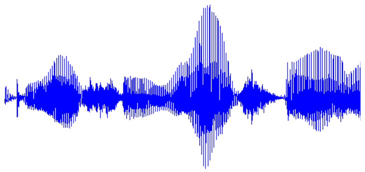
Deep Neural Network

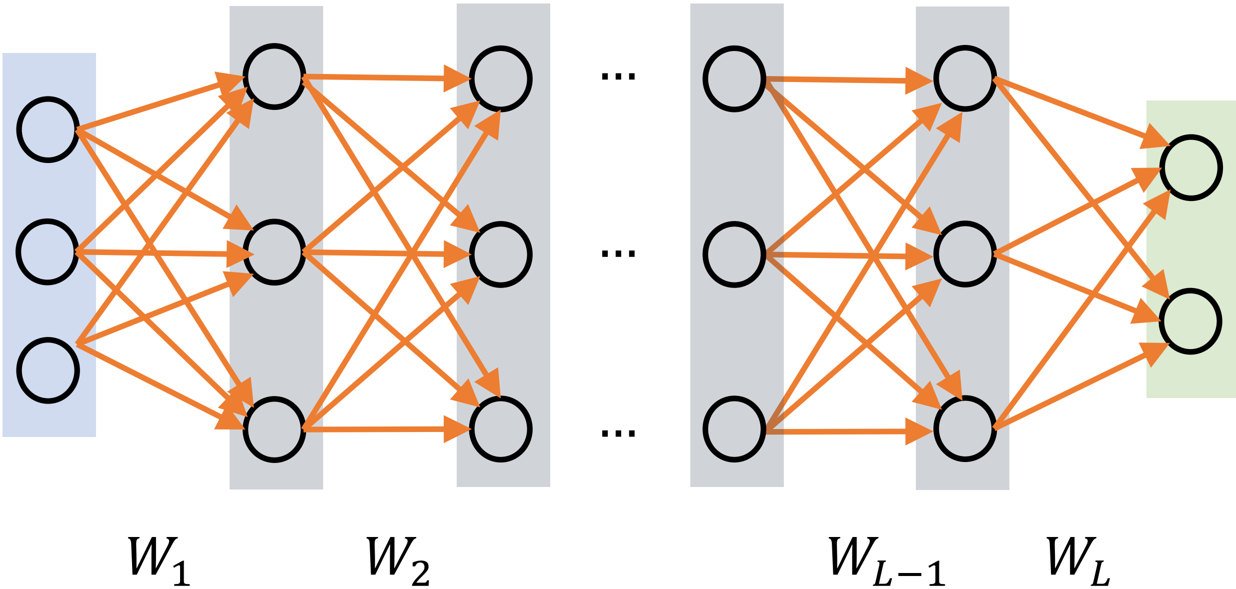
- Use “deep” layers
  - Many parameters to explain the data distribution
  - Need more data and fast computation (e.g. GPU)
  - Many efficient training techniques



Deep Neural Network

- Universal model regardless of the domain (image, audio, text, …)





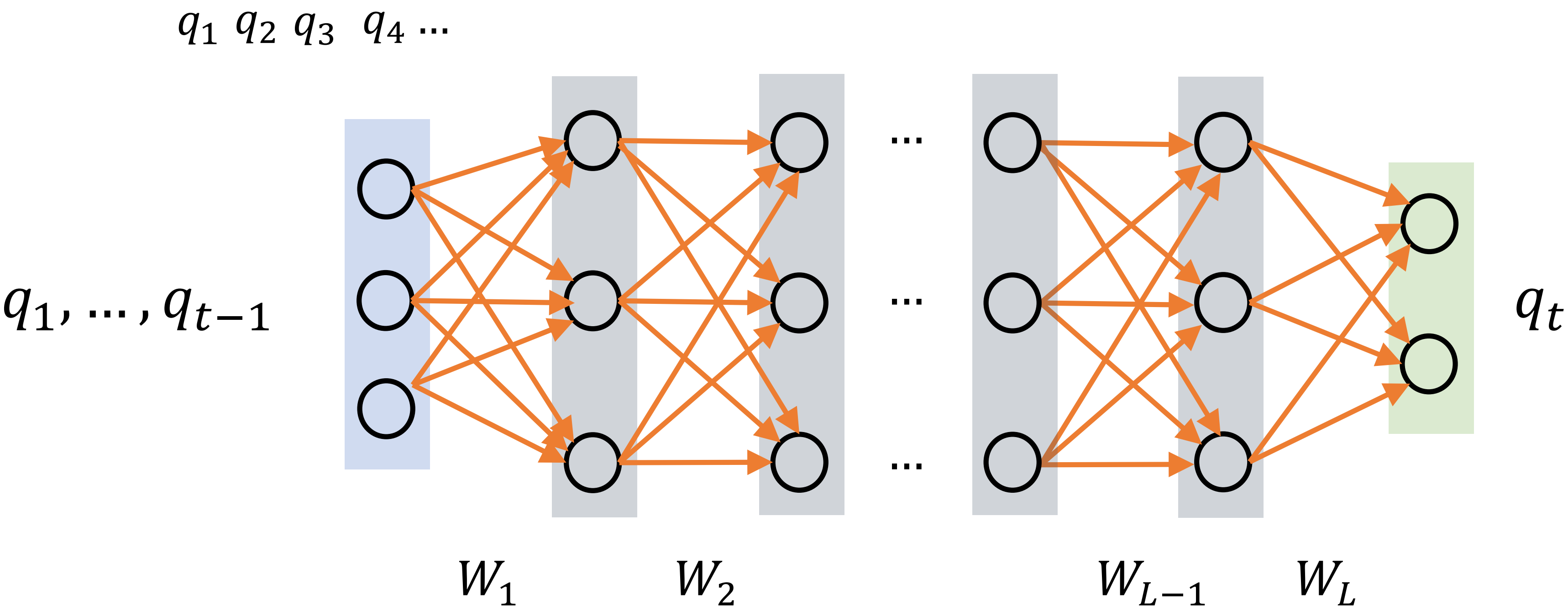
"motor-bike"

"I love coffee"

"오늘 남북정상이 만나…"

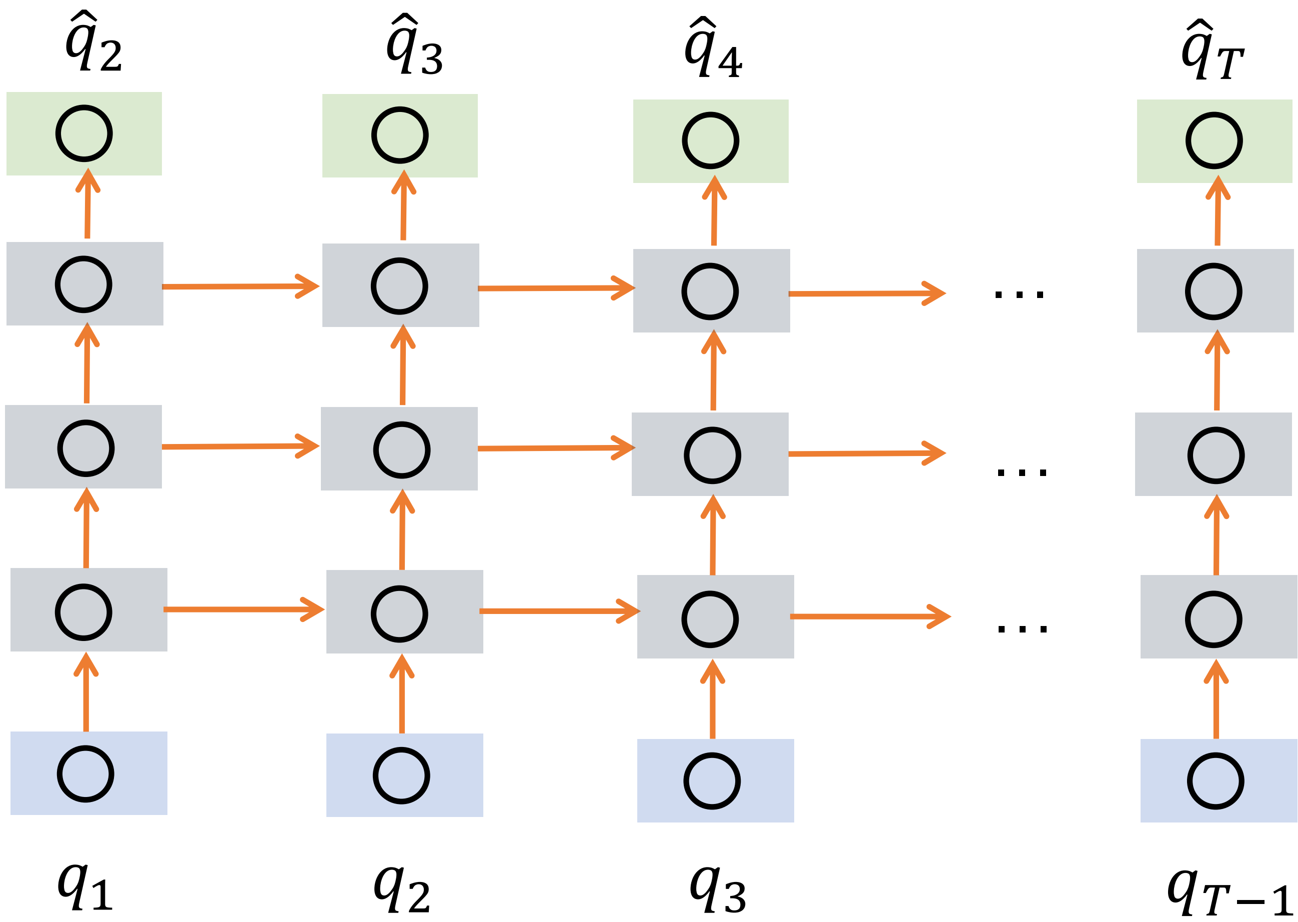
Deep Neural Network

- Thus, we can apply the model to music!
  - However, we need to handle long sequences and variable lengths



Recurrent Neural Networks (RNN)

- Sequence-to-sequence modeling

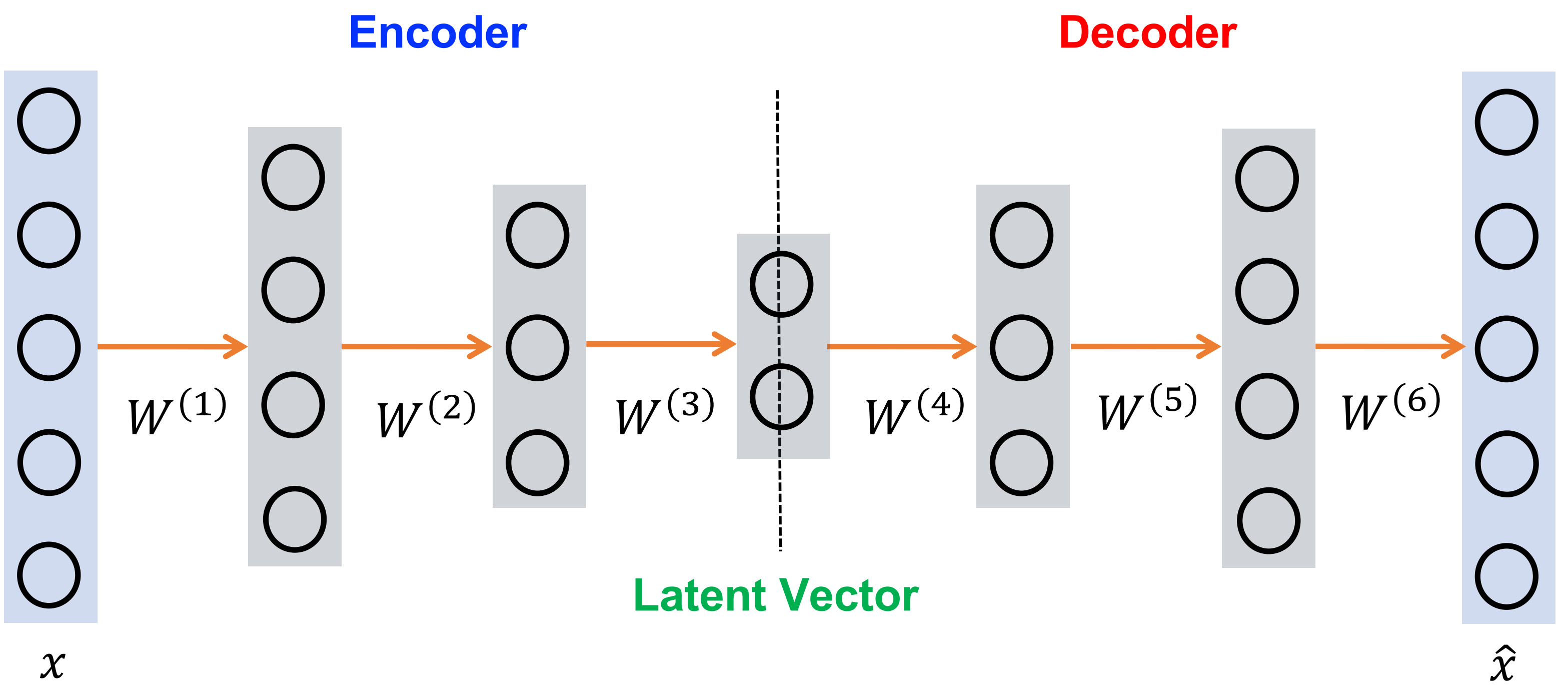


Examples

- FolkRNN
  - https://folkrnn.org/
- DeepBach
  - Four-part chorales
  - https://www.youtube.com/watch?v=QiBM7-5hA6o
- DeepJazz
  - https://deepjazz.io/
- PerformanceRNN
  - https://magenta.tensorflow.org/performance-rnn

Auto-Encoder

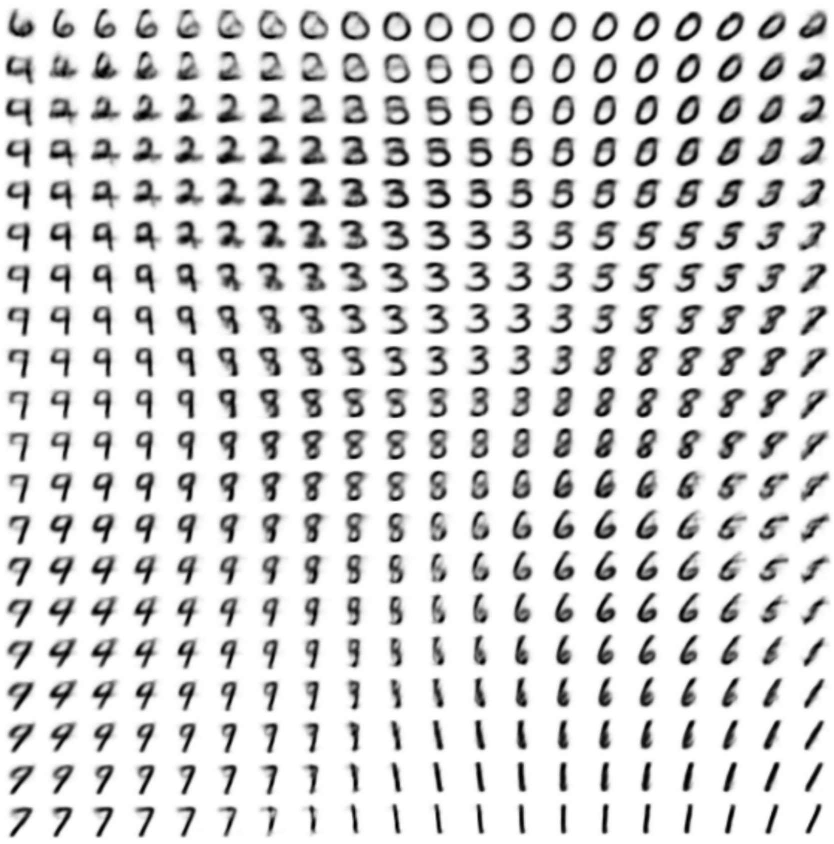
- Neural networks configured to reconstruct the input
  - The latent vector contains compressed information of the input
  - The decoder can be used to generate data: Variational Auto-Encoder (VAE) is more often used



Train to minimize the reconstruction error: $L(W;x) = \|x - \hat{x}\|^2$

Generation Examples

- Interpolation from the latent space



(Auto-Encoding Variational Bayes, Kingma and Welling, 2014)

Music Examples

- Music-VAE
  - https://magenta.tensorflow.org/music-vae