Seminar Presentation

Research Center for Technology and Art

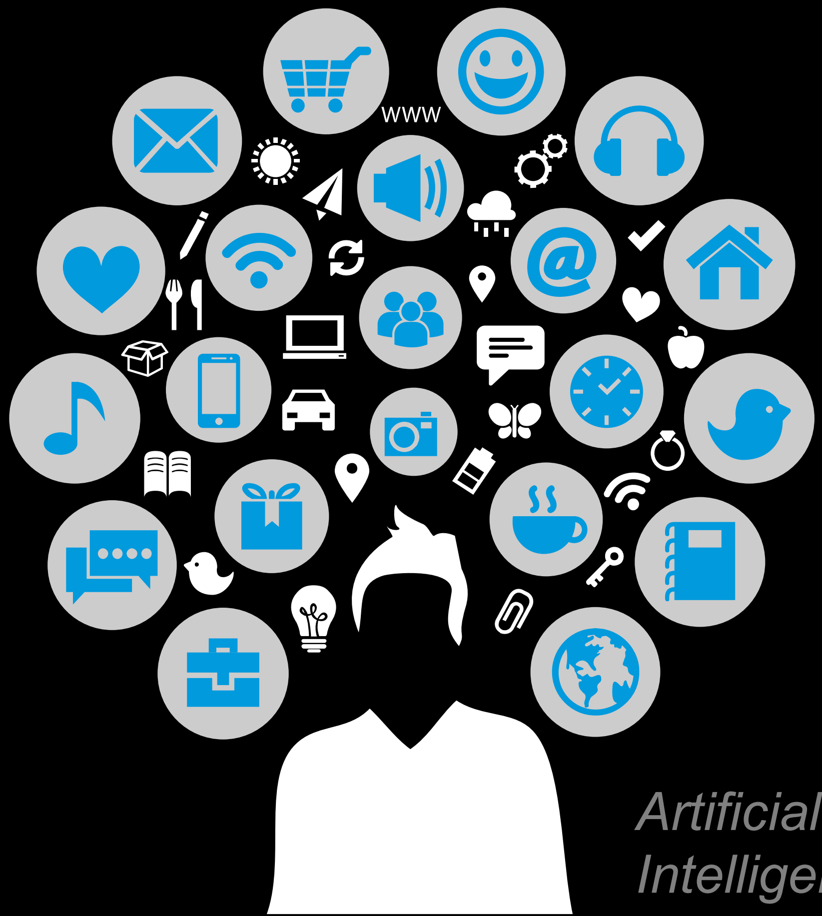
ARS Electronica Festival 2019

“ Dear Glenn “
– Yamaha AI Project

Reporter: IPHD, YuanFu Yang

https://ars.electronica.art/outofthebox/de/glenn/

- **01** Background
- **02** Art Statement
- **03** Connection
- **04** Demo



Artificial Intelligence Art

Glenn Gould

Born in Toronto, Canada in 1932, Glenn Gould was a legendary pianist who passed away in 1982 at the young age of 50. Gould has received extraordinary praise and is known for his masterful performances of J.S. Bach's music, beginning with his debut album Bach: The Goldberg Variations, which was released in 1956. In 1964, Gould announced the end of his concert career and began to concentrate on recording, devoting himself to digital media releases. Gould was also known for his unconventional and unique performance habits, which included sitting on a low chair and leaning over the piano keyboard, as well as humming while playing, even during recordings. In his later years, Gould recorded three albums, including Bach: The Goldberg Variations, on a Yamaha concert piano.

Background

The AI system consists of a player piano and the AI software, which instantly generates playing data that incorporates the unique touch, pacing, and other stylistic traits of **Glenn Gould** and then provides that data to the player piano. The concert was held at St. Florian Monastery on September 7, the third day of the Ars Electronica festival.

Neither of the ensemble pieces performed were included in the machine learning data, so audience members listened with great interest to see how well the AI system could reproduce **Gould**'s musicality without any recording data to rely on and how well it could cooperate and interact with human players while playing together in ensemble.

Background

The AI performed **Glenn Gould's** masterful J.S. Bach's Goldberg Variations (BWV 988) and some pieces. These were not included in its machine learning data. The audience listened intently to see how well it would reproduce the artist's musicality.

In addition to a piano solo, The system also performed together with renowned contemporary pianist Francesco Tristano and members of the Bruckner Orchester Linz (violin and flute) for a performance that "**transcended space and time**."

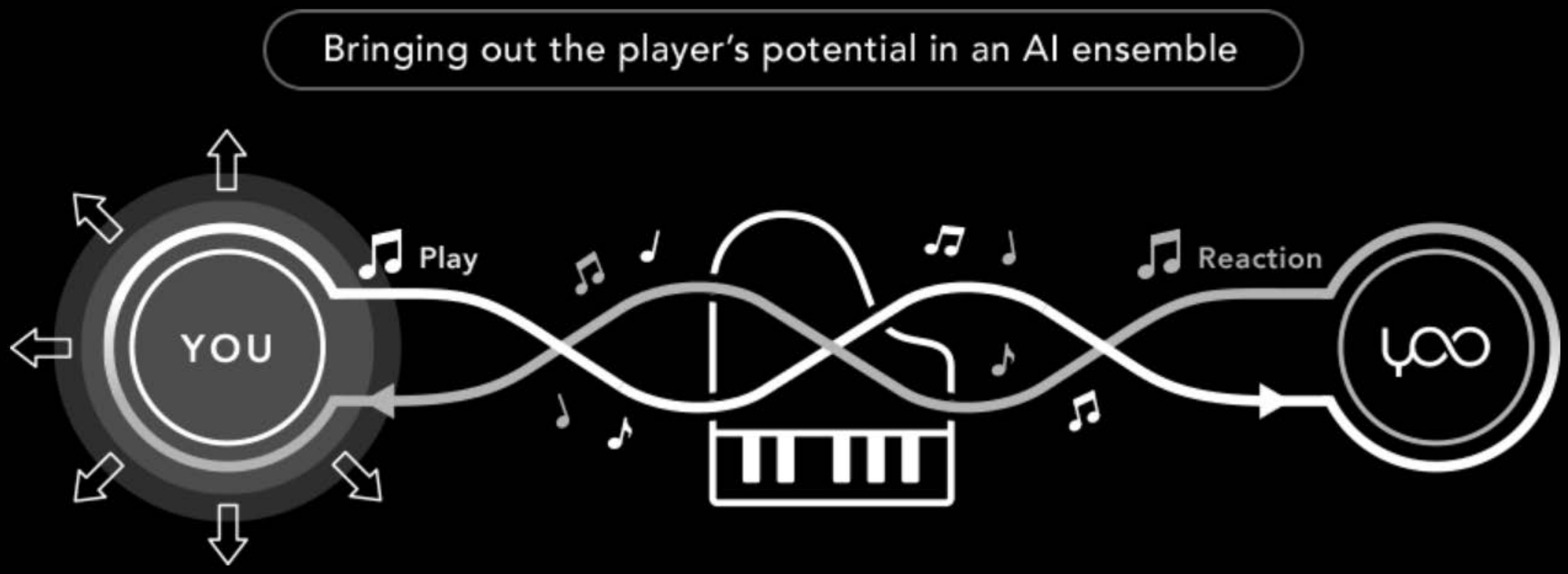

GLENN
GOULD
AS A.I.
IN ARS ELECTRONICA

Art Statement

- Yamaha develops world's first*1 AI which can play any piece of music in the style of legendary pianist Glenn Gould.
- The concert footage shows that the AI system playing songs never performed by Gould and playing together with renowned performers of today
- Discussion explored the possibilities of co-creation between AI and humans and how musical performance might be affected in the future

Art Statement

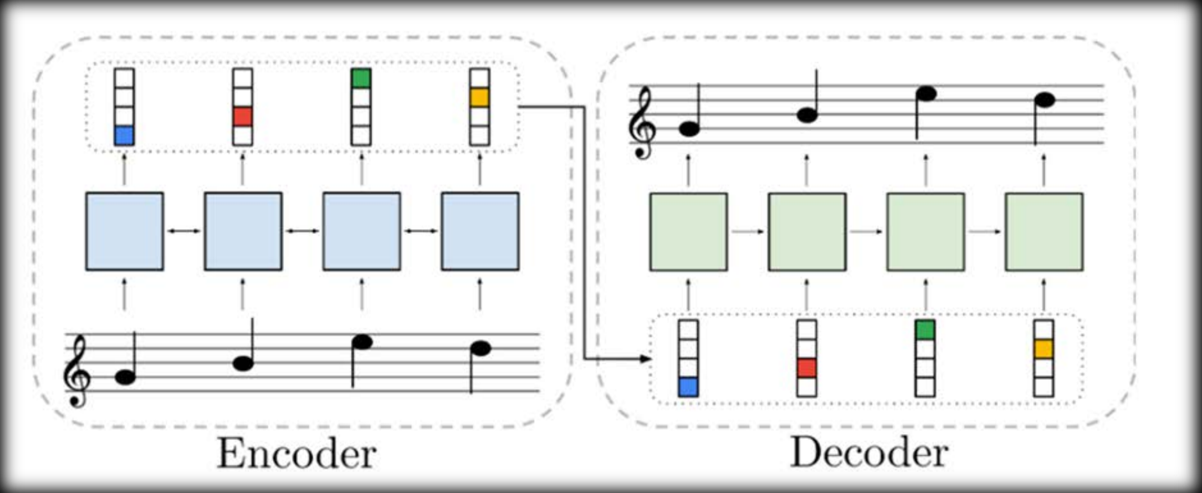
Dear Glenn features an Artificial Intelligence Music Ensemble System developed by Yamaha that analyzes the performance of players, and predicts the players' timing and tempo to control a **Disklavier** piano to play in synchrony— all in real-time. It's like playing with a human partner.

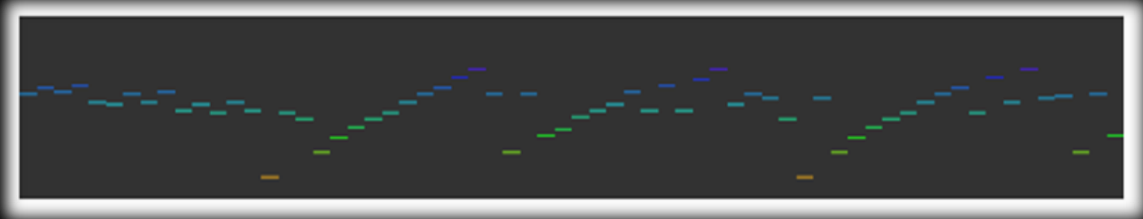


Art Statement

Yamaha AI restrict ourselves to 1-to-1 mappings between button presses and notes, giving the user precise control over timing. For example, the 8 buttons could map to a fixed scale over a single octave. Instead of using such a fixed mapping, we learn a time-varying mapping using a discrete autoencoder architecture trained on a set of existing piano performances.

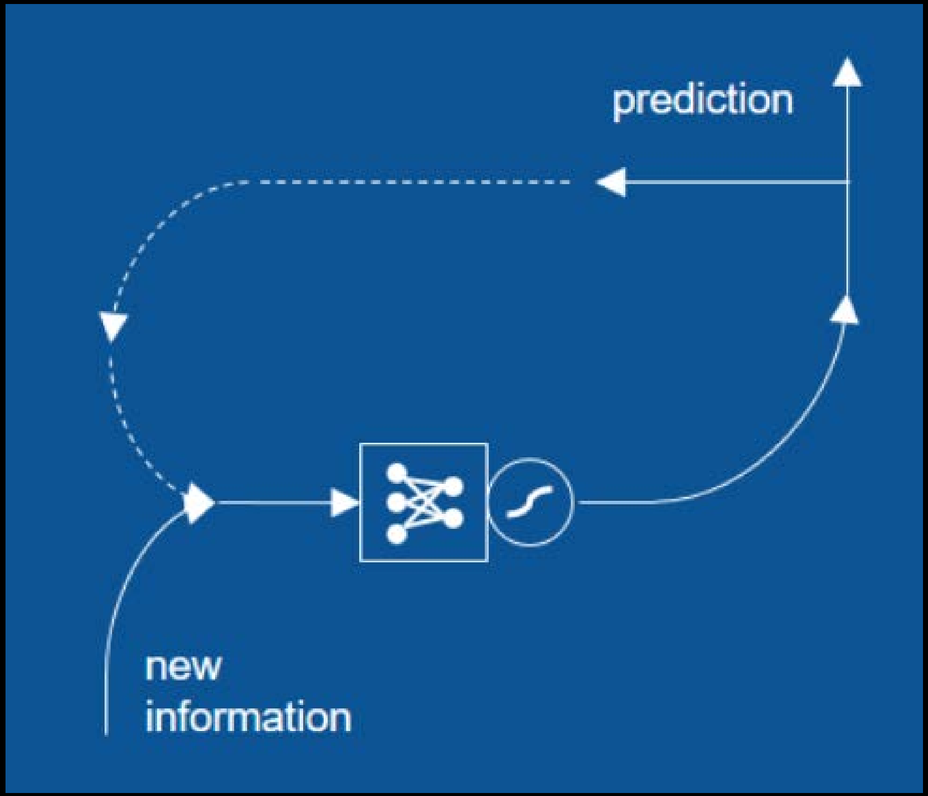
A bidirectional **LSTM** encoder maps a sequence of piano notes to a sequence of controller buttons. A unidirectional LSTM decoder then decodes these controller sequences back into piano performances. After training, the encoder is discarded and controller sequences are provided by user input.



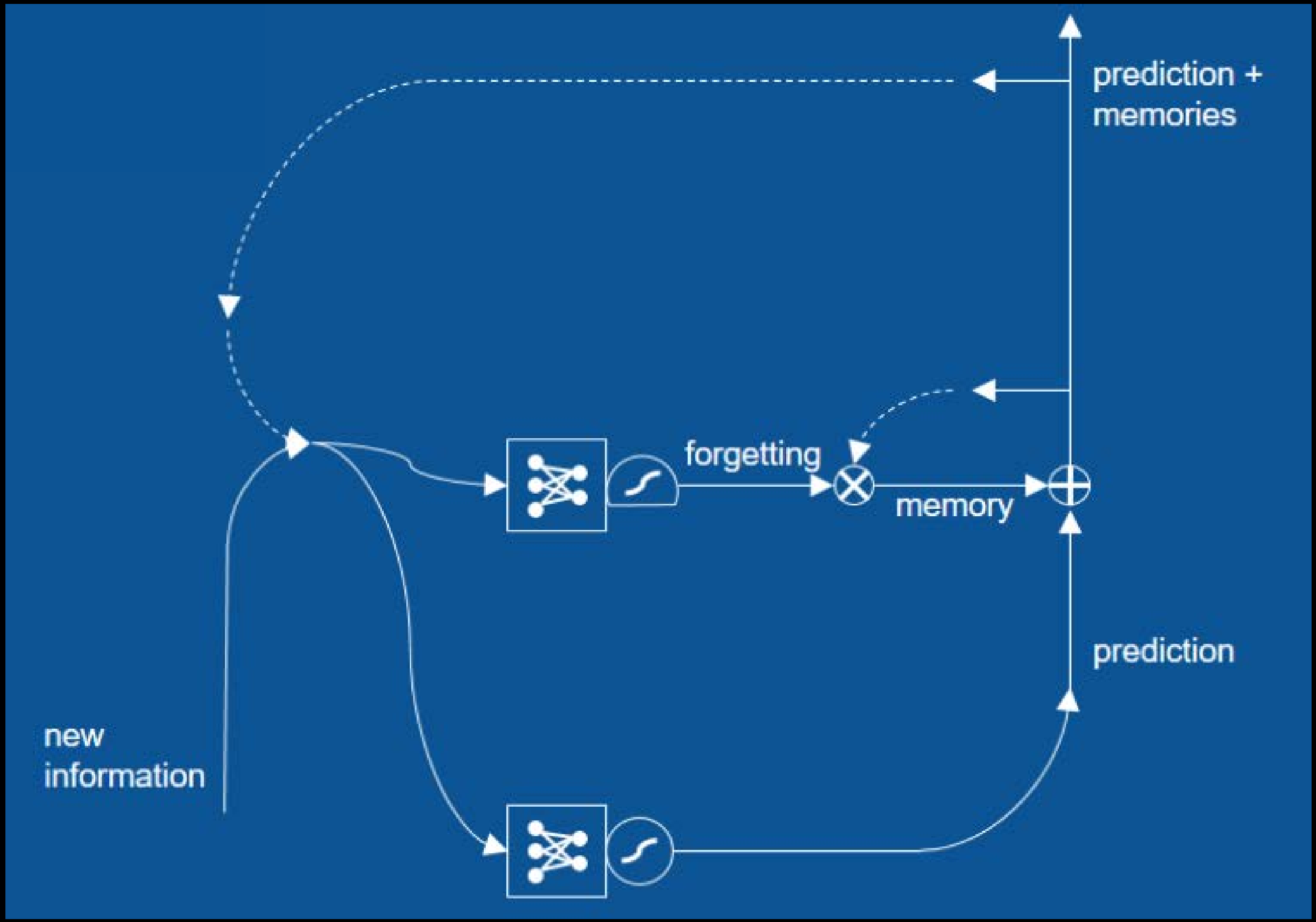


Art Statement

Recurrent Neural Network (RNN)



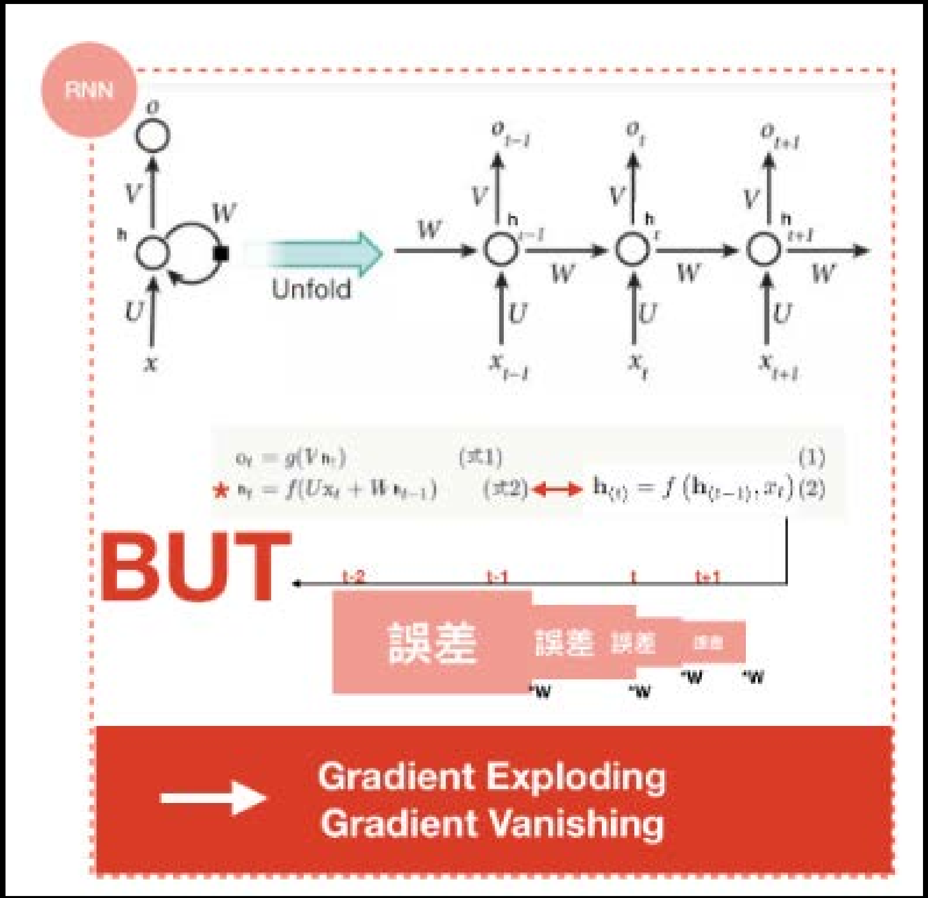
Long Short-Term Memory Network (LSTM)



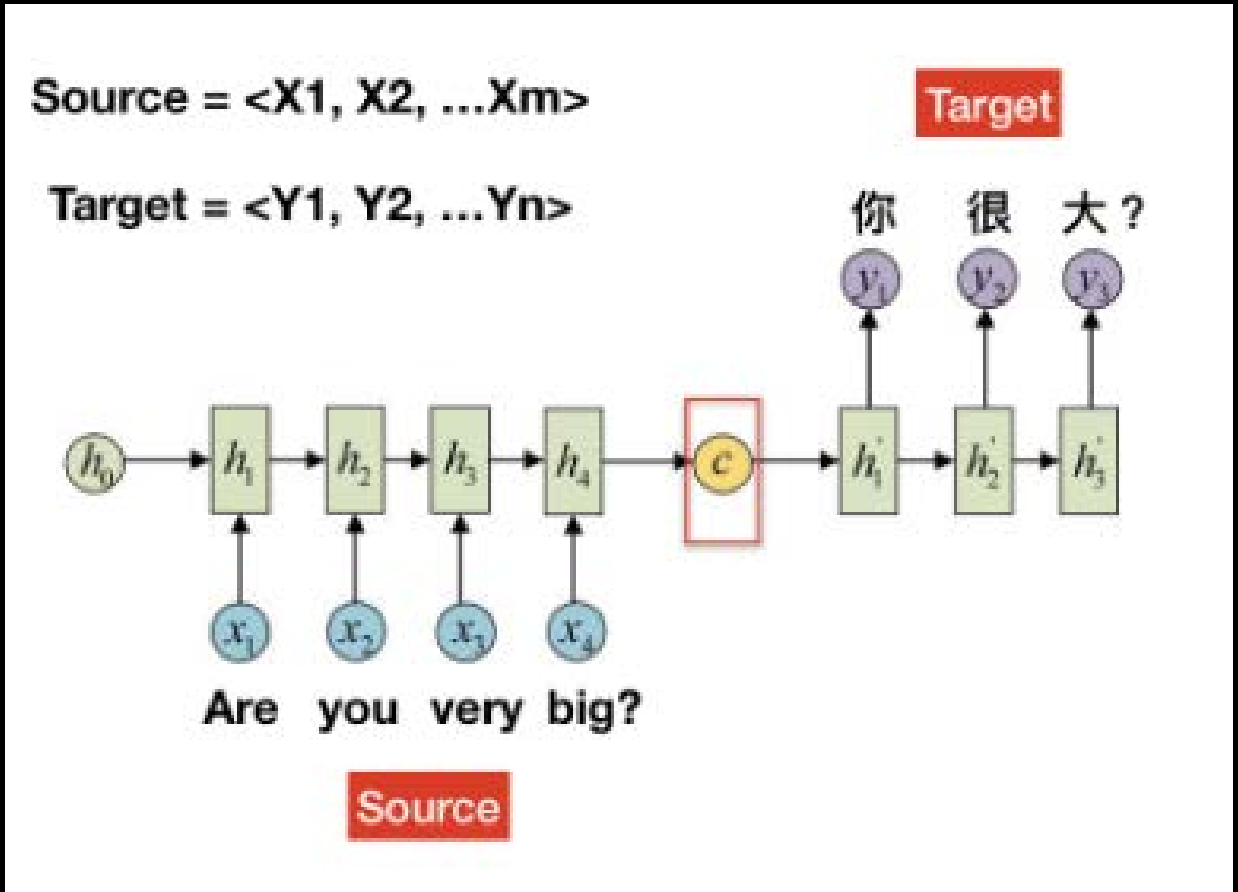
Reference: https://towardsdatascience.com/how-to-implement-seq2seq-lstm-model-in-keras-shortcutnlp-6f355f3e5639

Art Statement

Recurrent Neural Network (RNN)



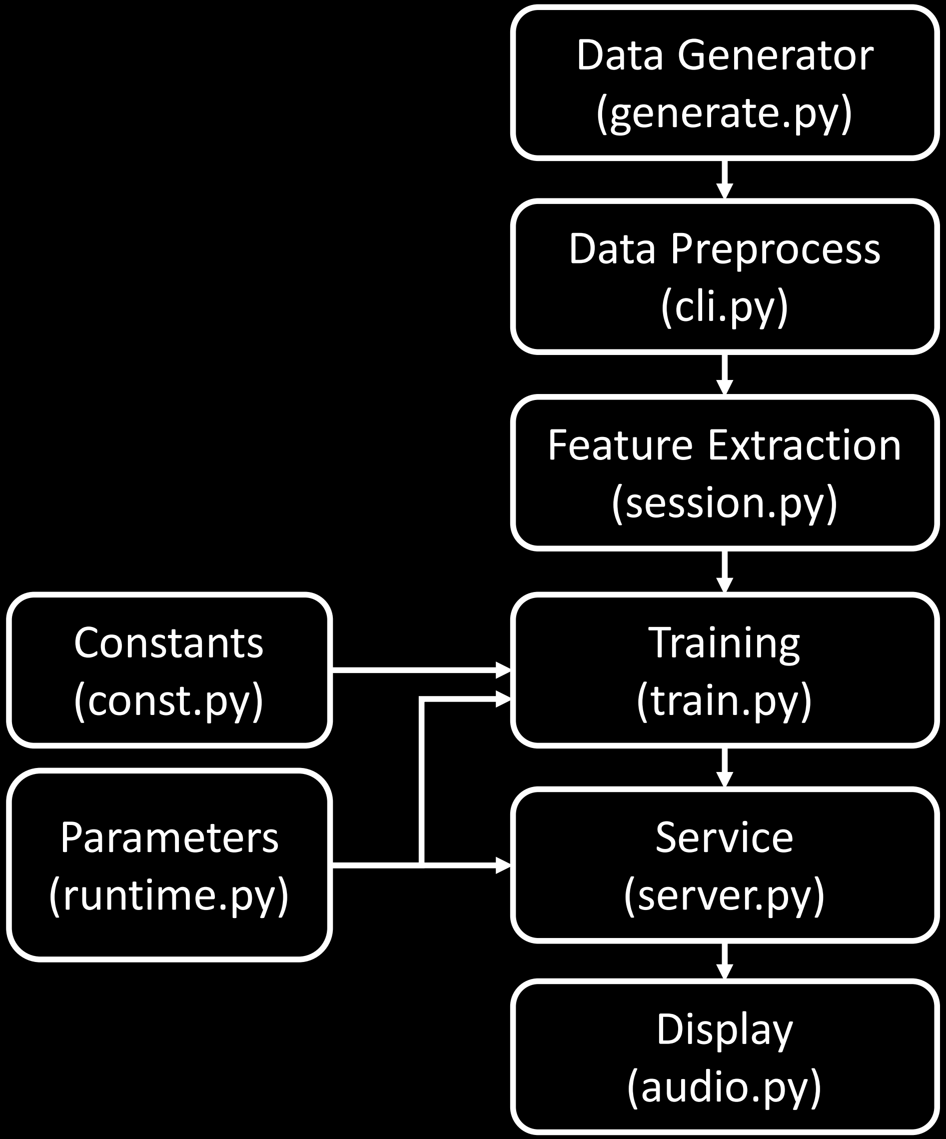
Long Short-Term Memory Network (LSTM)



Reference: https://towardsdatascience.com/how-to-implement-seq2seq-lstm-model-in-keras-shortcutnlp-6f355f3e5639

Art Statement

Architecture of Yamaha A.I. (Dear Glenn)



Feature Extraction

Step1: Pre-emphasis

s2(n) = s(n) - a*s(n-1)
a: 0~1

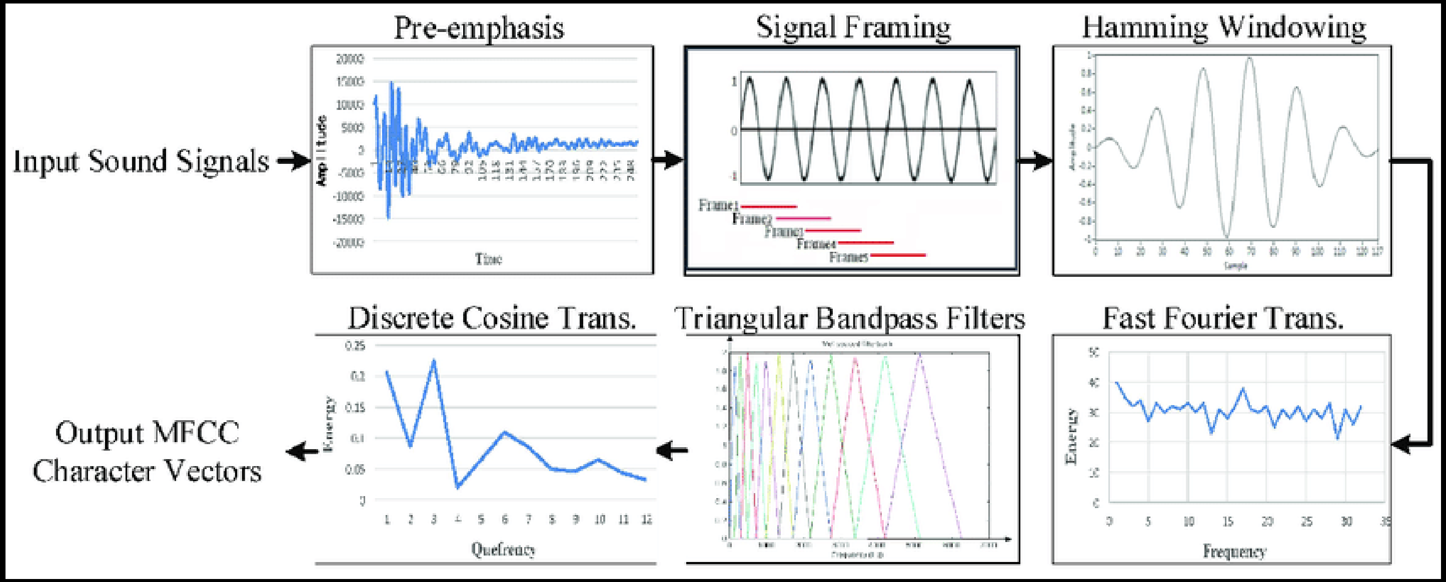
Step2: Signal Framing

N: 256 or 512 (set of sample)
M: N/3 (set of overlap sample)

Step3: Hamming Windowing

W(n, a) = (1 - a) - a cos(2pn/(N-1)),
$0 \leqq n \leqq N-1$

MFCC (Mel-Frequency Cepstral Coefficients)



Step6: Discrete Cosine Trans.

$C_m = \Sigma_{k=1}^{N} \cos[m*(k-0.5)*\pi/N]*E_k$,
m=1,2, ..., L

Step5: Triangular Bandpass Filters

mel(f)=2595*log10(1+f/700)

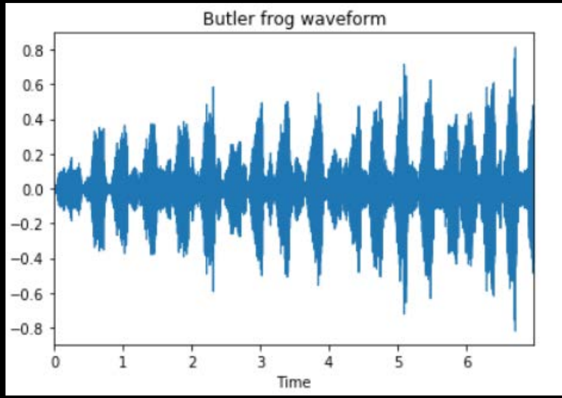
Step4: Fast Fourier Transform

$F(x) = \sum_{n-0}^{N-1} f(n) e^{-i2\pi(x\frac{n}{N})}$

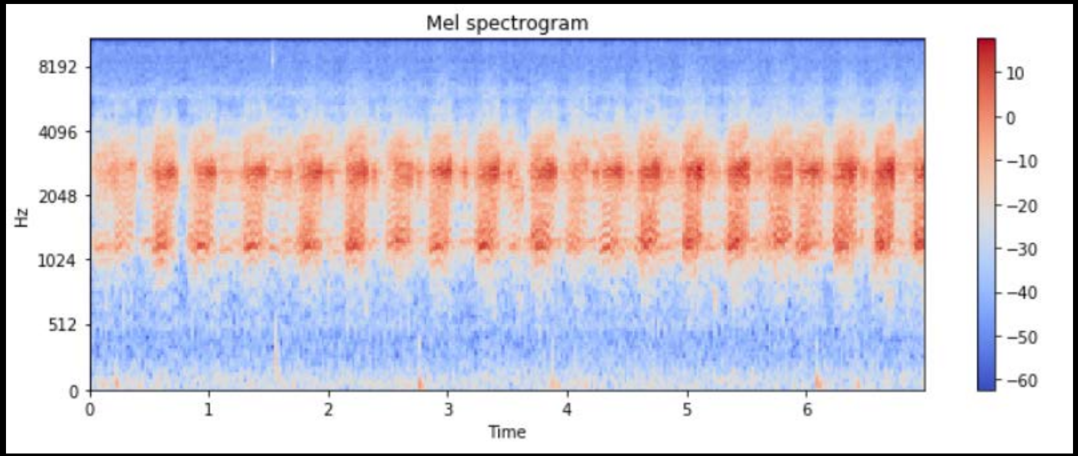
$f(n) = \frac{1}{N} \sum_{n-0}^{N-1} F(x) e^{i2\pi(x\frac{n}{N})}$

Feature Extraction

MFCC (Mel-Frequency Cepstral Coefficients) - Taiwan's Frog Sounds Classification



MFCC



Feature

	0	1	2	3	4	5	6	7	8	9	...	200	201	202	
0	-185.626883	107.772851	-9.412268	46.641109	-31.252501	26.115870	0.743796	16.494876	12.095498	-9.613577	...	0.430407	51.993177	21.858606	7.6
1	-195.073173	99.788199	-11.642600	49.751983	-28.479792	22.598996	1.311489	12.465497	14.098283	-8.353313	...	0.632943	49.751168	19.857528	4.0
2	-195.435764	95.834447	-9.763945	50.097418	-31.189562	25.782336	-0.952654	14.067396	13.743476	-8.822381	...	0.482606	43.245856	20.145879	6.8
3	-351.180482	92.030936	3.365378	27.404095	1.362222	15.229861	-5.240146	26.568555	-8.659325	16.669641	...	1.358107	46.366270	16.790002	6.8
4	-335.960816	97.861751	-1.966739	19.760014	0.298002	16.324501	-10.484625	28.557586	-5.481404	15.082006	...	0.825509	38.168811	16.063708	6.6

5 rows × 210 columns

Train

Model	Initial Model	Model turning	Testing
SVM	0.8	0.94	0.99
Random Forst	0.6	0.90	0.95
XGBoost	0.75	0.85	0.75
LightGBM	0.85	0.85	0.85

Reference: https://github.com/Yfyangd/frog

LSTM Model

4 LSTM with 4 dropout function, 1 dense layer (256 to 99 classes). Param count: 2,152,035

```
model = Sequential()
model.add(layers.Embedding(input_dim=num_classes,
                           output_dim=num_units,
                           input_length=seq_len))
for n in range(num_layers - 1):
    model.add(layers.LSTM(num_units, return_sequences=True))
    if dropout > 0.0:
        model.add(layers.Dropout(dropout))
model.add(layers.LSTM(num_units))
if dropout > 0.0:
    model.add(layers.Dropout(dropout))
model.add(layers.Dense(num_classes, activation='softmax'))

model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['acc'])

model.summary()
```

num_layers = 5
num_classes = 99

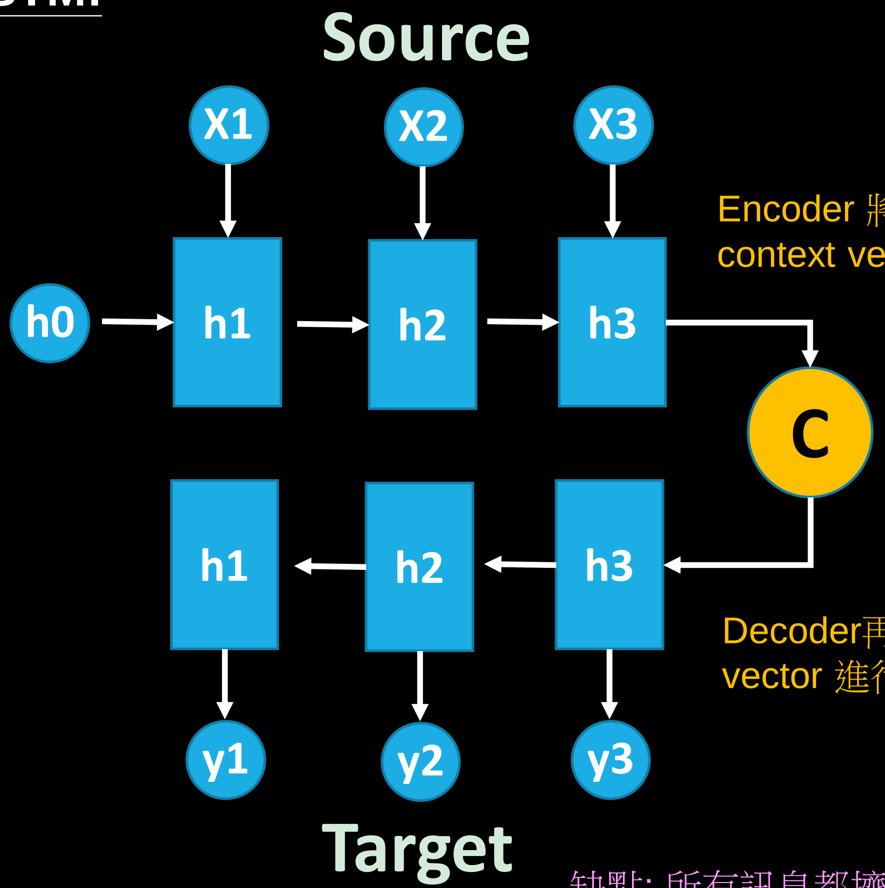
Layer (type)	Output Shape	Param #
embedding_5 (Embedding)	(None, 3, 256)	25344
lstm_7 (LSTM)	(None, 3, 256)	525312
dropout_6 (Dropout)	(None, 3, 256)	
lstm_8 (LSTM)	(None, 3, 256)	525312
dropout_7 (Dropout)	(None, 3, 256)	
lstm_9 (LSTM)	(None, 3, 256)	525312
dropout_8 (Dropout)	(None, 3, 256)	
lstm_10 (LSTM)	(None, 3, 256)	525312
dropout_9 (Dropout)	(None, 3, 256)	
dense_3 (Dense)	(None, 3, 99)	25443
Total params: 2,152,035		2,152,035
Trainable params: 2,152,035		2,152,035
Non-trainable params: 0		0

Connection

- YAMAHA A.I. project is not only a musical experiment with a non-human performer, but also an undertaking to make computer culture "audible."
- The performance raises questions about the logic and politics of computers in relation to human culture.
- The Artificial Intelligence (AI) "learns" the artist's individual style via voice recordings and directly confronts him with the newly generated material. Their joint performance shows how interactive technology and AI can influence a (vocal) style. However, this dialogue also makes clear that the artist will always be more creative and unpredictable than his mechanical counterpart.
- The state of the art method: Attention Mechanism

LSTM vs Attention Mechanism

LSTM:



Encoder 將所有輸入存成 context vector

Decoder再藉由context vector 進行資訊傳導

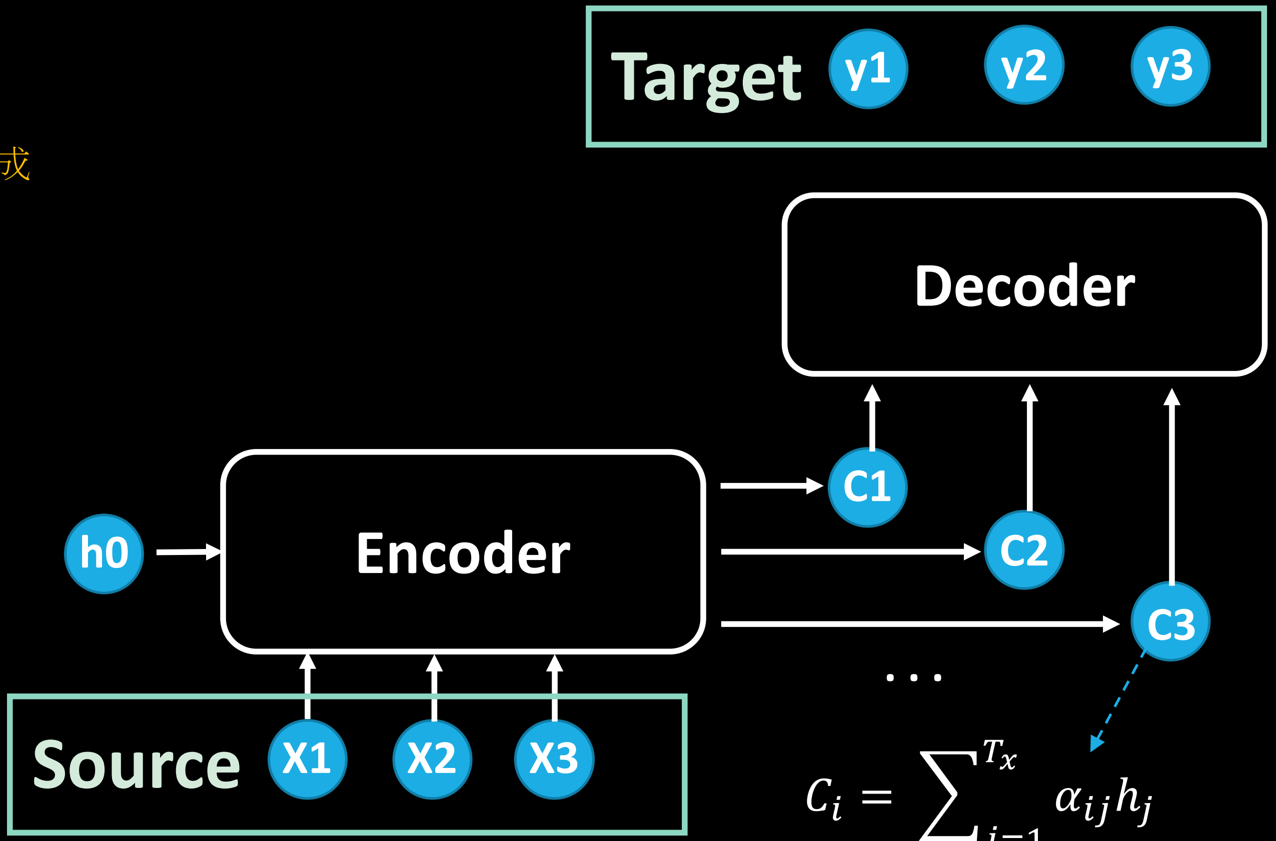
$$C = F(X_1 + X_2 + \ldots + X_m)$$

$$Y_i = G(C, Y_1, Y_2, \ldots, Y_{i-1})$$

缺點: 所有訊息都擠壓縮成 one context vector, 造成:

(1) 無法表達序列訊息
(2) 句子太長會丟失訊息

Attention Mechanism:



$$C_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{jk})}$$

Reference: Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473

Demo - A.I. Music