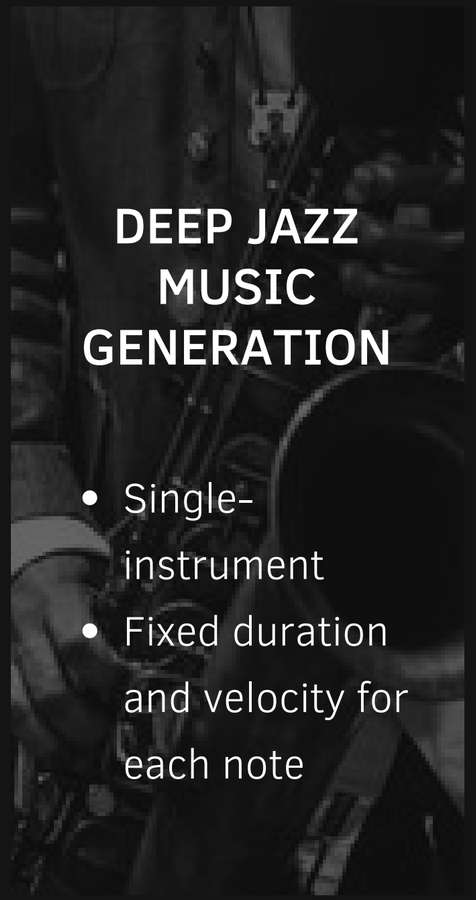
Neural Networks

Analysis and Comparison of Deep Learning Methods for Jazz Music Generation

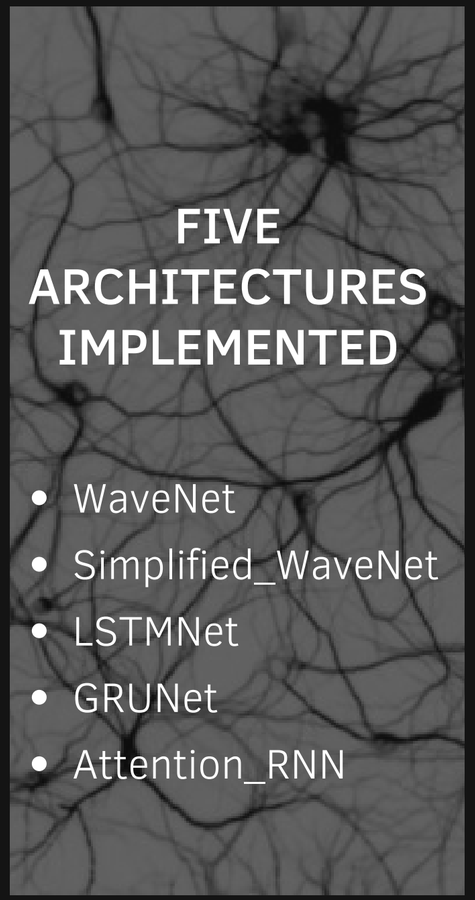
RICCARDO RATINI, 1656801
GIADA SIMIONATO, 1822614

OUR WORK



DEEP JAZZ MUSIC GENERATION

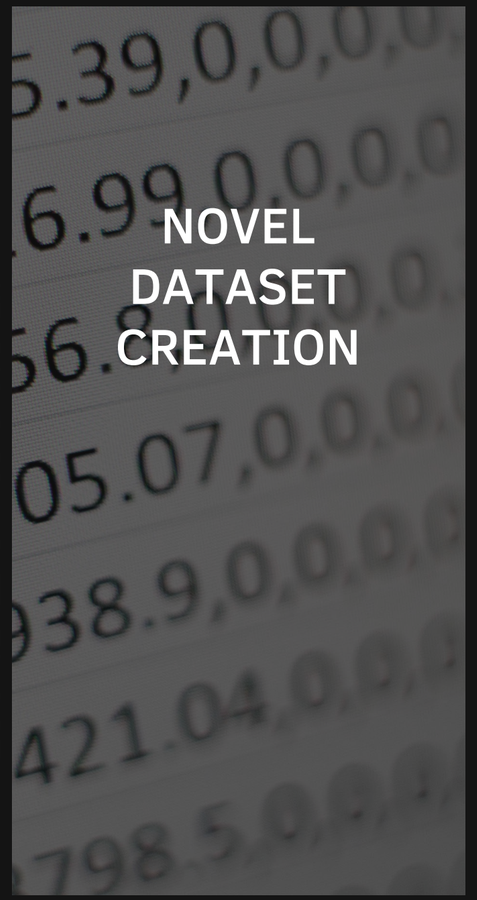
- Single-instrument
- Fixed duration and velocity for each note



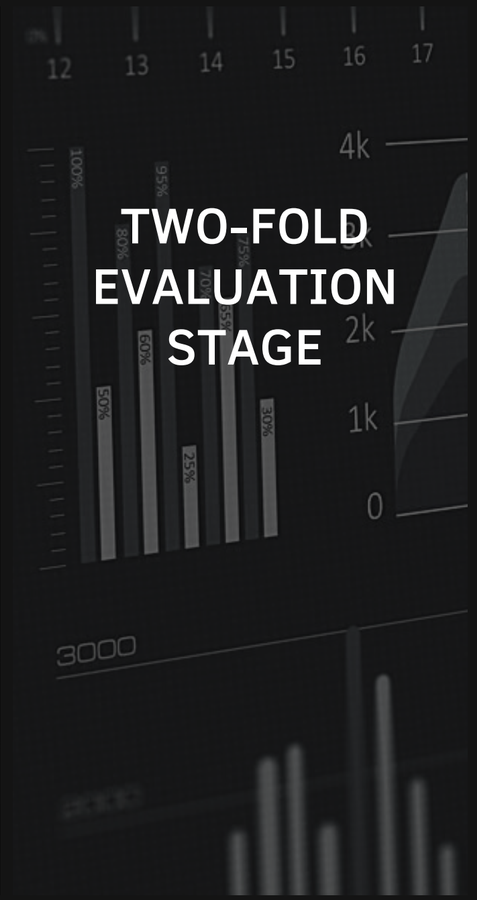
FIVE ARCHITECTURES IMPLEMENTED

- WaveNet
- Simplified_WaveNet
- LSTMNet
- GRUNet
- Attention_RNN

PRE & POST PROCESSING METHODS DEPLOYED



NOVEL DATASET CREATION



TWO-FOLD EVALUATION STAGE

DATASETS

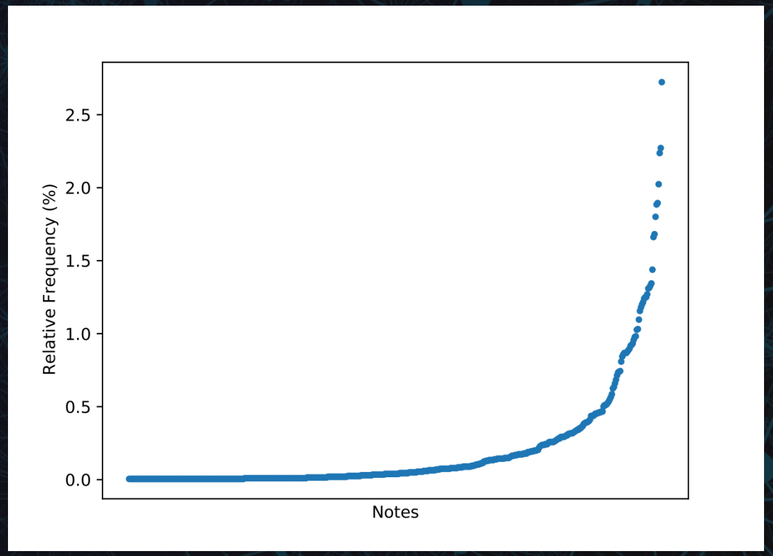
NOVEL DATASET

- Manually constructed
- 37 songs in MIDI extension
- Each song ranging from almost 2 to 4 minutes

FULL DATASET

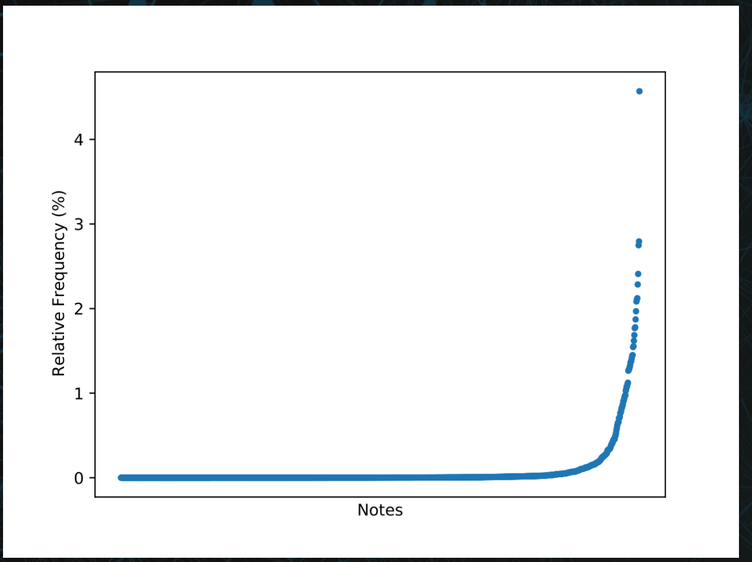
- Only jazz dataset available
- 818 songs in .csv file as list of notes
- Additional information (name and length song, set of unique notes and its size)

Analysis Dataset Composition



- Threshold: 0.1
- Vocab. size (before): 513
- Vocab. size (after): 179 (-64.11%)
- Dataset integrity: 91.49%

- Threshold: 0.1
- Vocab. size (before): 1216
- Vocab. size (after): 138 (-88.65%)
- Dataset integrity: 92.58%



PRE&POST PROCESSING

PREPROCESSING

NLP-BASED

- Applied to: WaveNet, Simplified_WaveNet, LSTMNet

1. Extract sequences of notes from songs
2. Create vocabulary (with <PAD> and <UNK> tags) to encode notes
3. Encode notes and pad/truncate sequences to match input shape
4. Take one-hot encoding of next note for each sequence as its output

PIANOROLL-BASED

- Applied to: GRUNet
- Contains the empty tag

1. Extract PianoRoll from songs
2. Convert PianoRoll to dictionary (key: temporal frame *t*, value: list of indexed notes in *t*)
3. Pad/truncate sequences to match input shape
4. Take index of next note for each sequence as its output

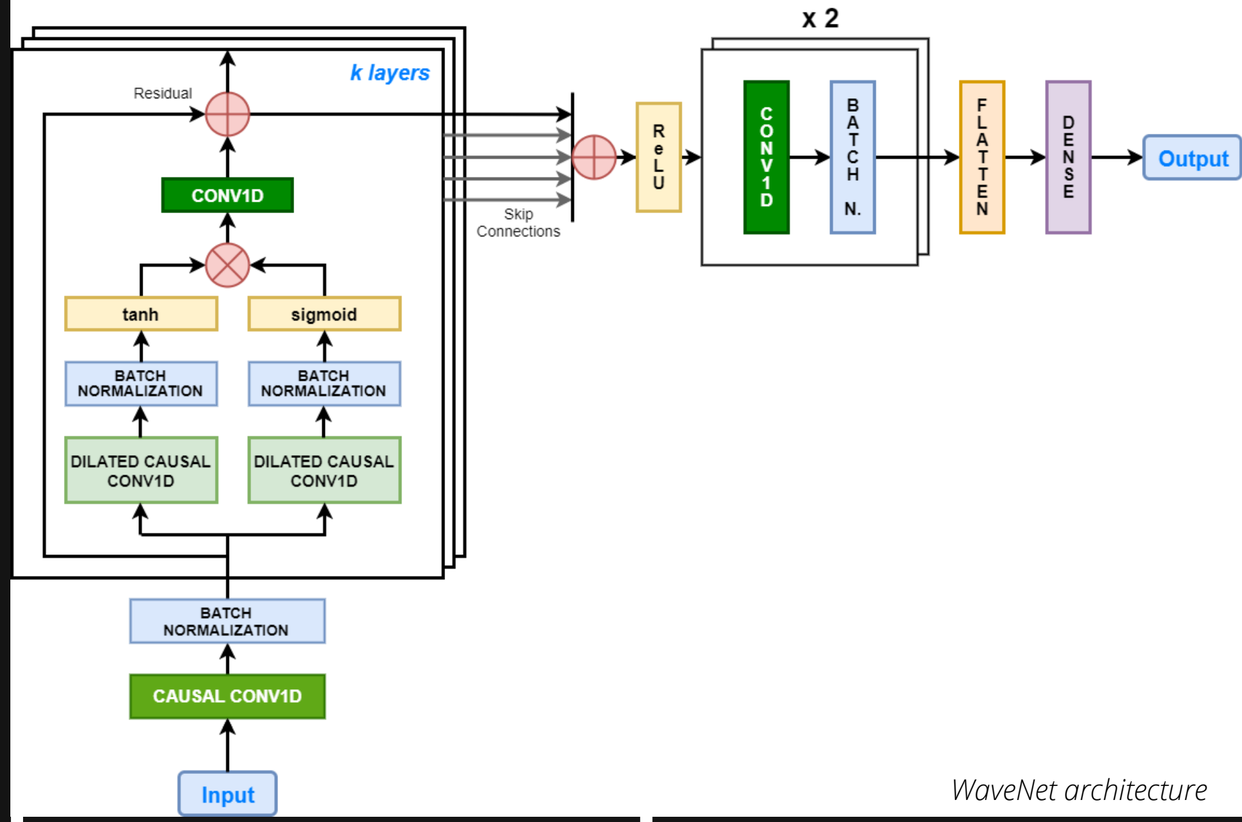
POSTPROCESSING

- Implemented second-best selection strategy when <UNK> is predicted

1. Select initial sequence
2. Feed it into predict stage
3. Took note corresponding to argmax over softmax output probability distribution
4. Concatenate predicted note to input and remove first note
5. Repeat 2-4 until limit reached
6. Convert list of generated notes to MIDI file
7. (Optional) Convert MIDI to .mp3 file

WAVENET

- Inspired by **[1]**
- Based on **dilated causal convolutional** layers
- **Dilation rate** growing exponentially until threshold, i.e. 512
- Capture **long-term relationships** due to its large receptive field
- Presence of **residual** and **skip-connections** to speed up convergence
- Trained with:
  - Only **Novel** dataset
  - Only **Full** dataset (partial)
  - **Novel+Full** datasets (partial)
- Need of **dictionary of weights** for each class in fitting stage to deal with imbalanced datasets (computed with sklearn library)
- Added **batch normalization** layers and **L2 kernel regularizers** to suppress overfitting



WaveNet architecture

HYPERPARAMETERS TUNING

- **Length sequence input:** {16, 32, 64, 128}
- **Number convolutional filters**: {64, 128, 256, 512}
- **Number residual blocks**: {3, 5, 7, 30}

BEST CONFIGURATION

- Length sequence input: **64**
- Number convolutional filters: **256**
- Number residual blocks: **5**
- **Adam** Optimizer
- **Categorical Cross Entropy** loss

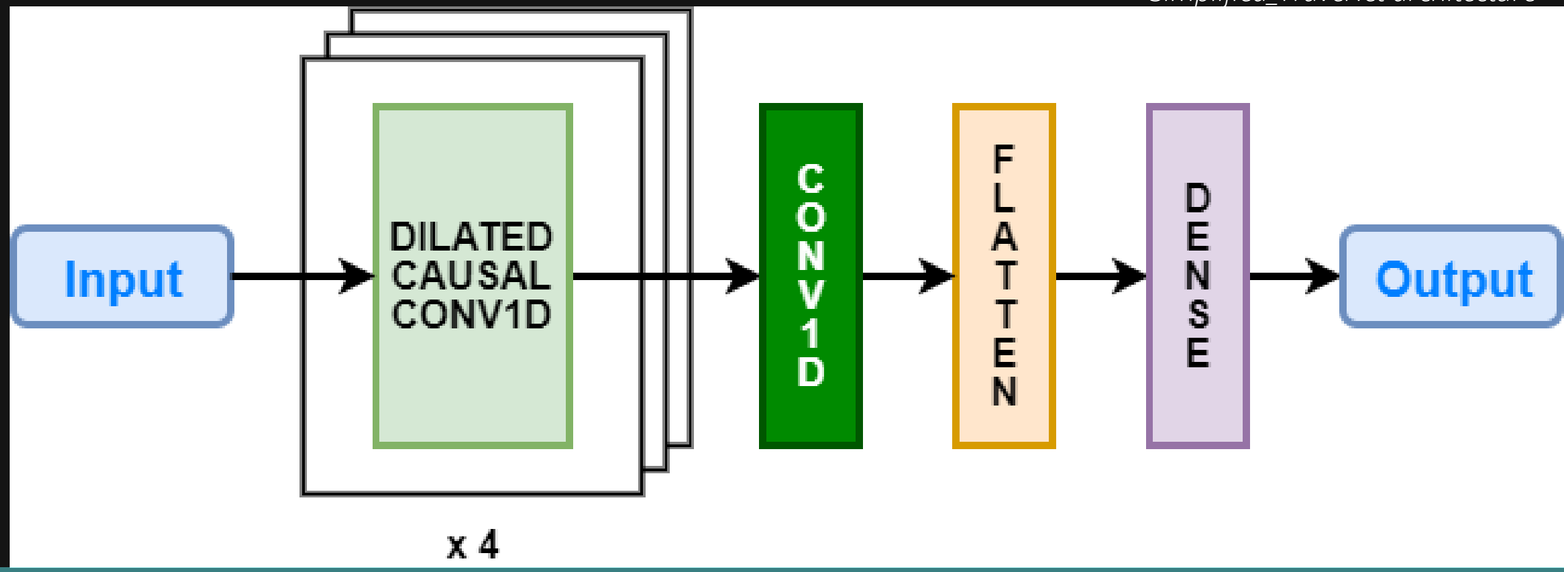
SIMPLIFIED_WAVENET

- Inspired by **[2]**
- **Simplified** version of the original WaveNet
- **No** residual and skip-connections
- Need of **dictionary of weights** for each class to avoid repeatedly generation of same note
- **Quickest to train** (averaged per epoch)

BEST CONFIGURATION

- Length sequence input: **64**
- Number dilated causal convolutional filters: **20**
- Number convolutional filters: **10**
- Kernel size: **2**
- **Adam** Optimizer
- **Mean Squared Error** loss

Simplified_WaveNet architecture



LSTMNET

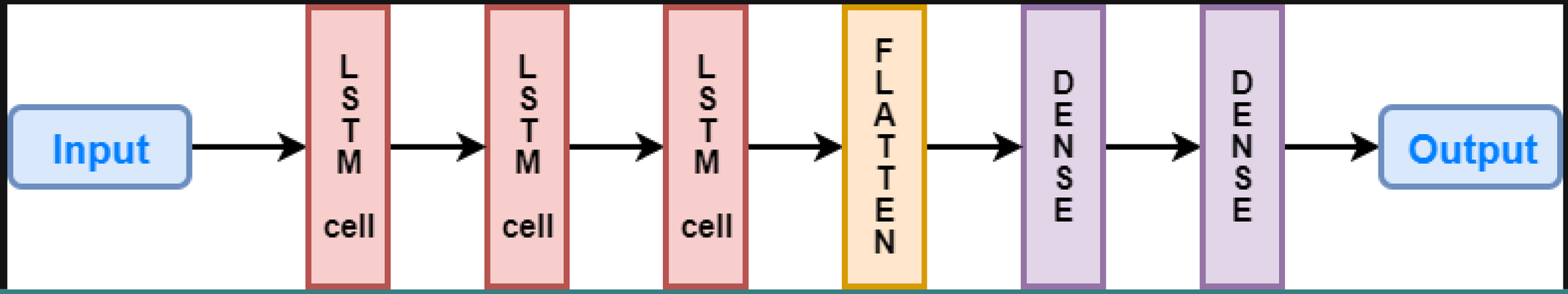
- Inspired by **[3]**
- Based on **LSTM cells**
- **No need** of dictionary of weights to deal with the imbalanced datasets
- Trained on **Novel dataset only**
- **Slowest to train** (averaged per epoch)

HYPERPARAMETERS TUNING

- **Dropout/ Recurrent Dropout**: {0.1, 0.3, 0.5}
- **Hidden units**: {100, 200, 300}
- **Optimizer**: {Adam, Nadam, SGD}

BEST CONFIGURATION

- Input sequence length: **32**
- Batch size: **64**
- Number first Dense units: **256**
- Dropout/ Recurrent Dropout: **0.3**
- Hidden units: **300**
- Optimizer: **Nadam**
- **Categorical Cross Entropy** loss



LSTMNet architecture

GRUNET

- Adapted from **[4]**
- Based on **Bidirectional GRU** cells
- Presence of **self-attention** layers
- Presence of **Embedding** layer
- **No need** of dictionary of weights to deal with the imbalanced datasets
- Trained on **Novel dataset only**

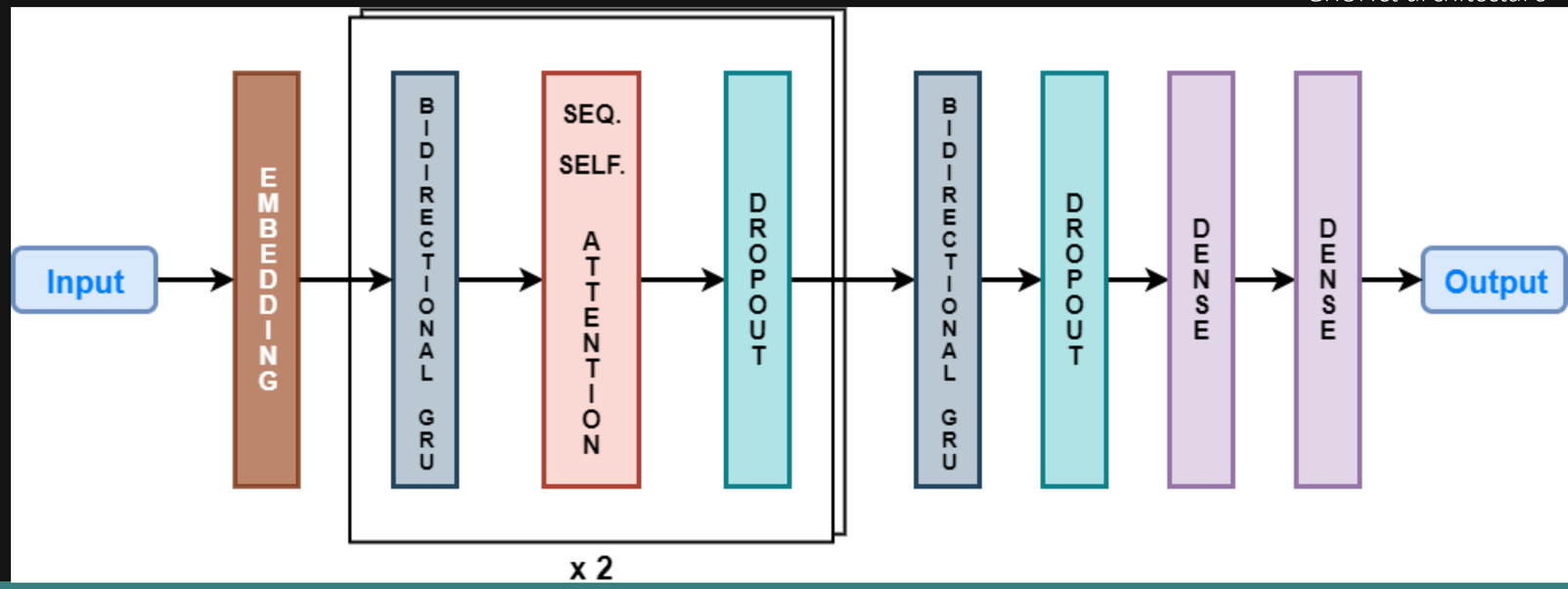
HYPERPARAMETERS TUNING

- **Input sequence length:** {16, 32, 64, 128}
- **Dropout:** {0.1, 0.3, 0.5}
- **RNN units:** {64, 128, 256}
- **Optimizer:** {Nadam, RMSProp, SGD}

BEST CONFIGURATION

- Input sequence length: **64**
- Batch size: **96**
- Dropout: **0.1**
- RNN units: **256**
- Optimizer: **Nadam**
- Embedding size: **100**
- **Sparse Categorical Cross Entropy** loss

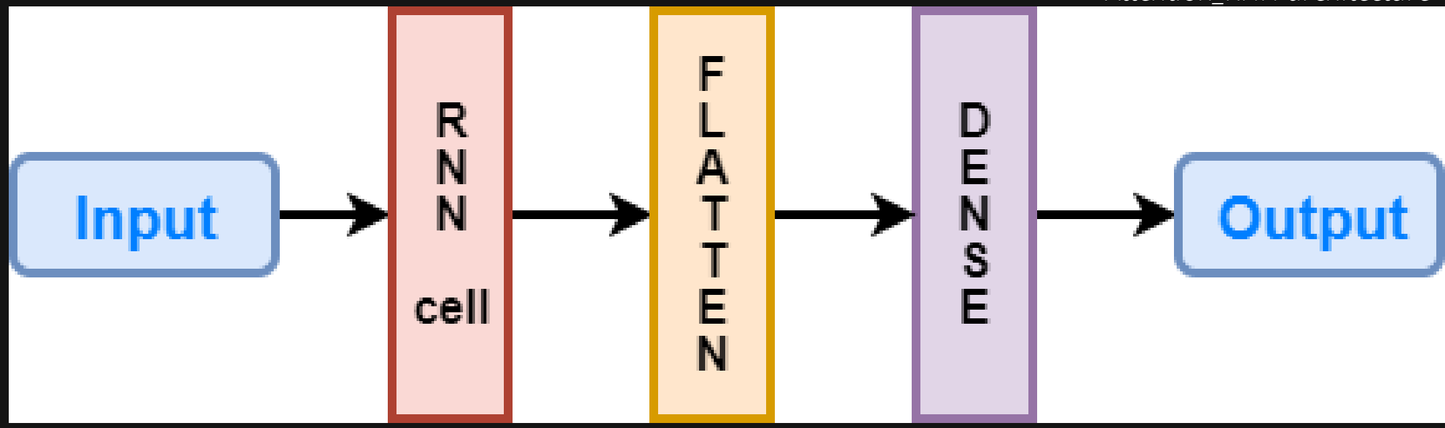
GRUNet architecture



ATTENTION_RNN

- Adapted from [5]
- Provided by Magenta library, powered by TensorFlow
- Presence of:
  - Basic_RNN
  - Mono_RNN
  - Lookback_RNN
  - Attention_RNN
- Use of pre-trained models to generate music or train models from scratch
- Based on LSTM cells
- No residual connections
- Leverage Basic_RNN with Attention mechanism
- Trained on Novel dataset only

Attention_RNN architecture



BEST CONFIGURATION

- Batch size: **128**
- RNN layer sizes: **[128, 128]**
- Dropout: **0.5**
- Attention length: **40**
- Optimizer: **Adam**
- Learning rate: **0.001**
- **Softmax Cross Entropy** loss

EVALUATION: SUBJECTIVE GROUPED COMPARISON

- Inspired by **[1,7]**
- Experiment involved **13** persons
- Drafted using **Google Form** service
- Guarantees **unbiased** responses
- **WaveNet** and **GRUNet** preferred
- **Small difference** w.r.t. real data

MODALITY

- Six links to six .mp3 files with encoded names
  - 5 generated tracks, one for each model
  - 1 real track, extracted from the dataset
- Compose top-3 chart in order of preference
- Preferences based on the concept of naturalness
- For each network sum (the higher the better):
  - Number of first places multiplied by 5
  - Number of second places multiplied by 3
  - Number of third places multiplied by 1

	Architecture					
	WaveNet	Simplified_WaveNet	Attention_RNN	LSTMNet	GRUNet	Dataset
SGC score	**28**	16	14	11	27	**30**

Cumulative SGC score for each architecture

EVALUATION: MEAN OPINION SCORE

- Inspired by **[1,7]**
- Experiment involved **14** persons
- Drafted using **Google Form** service
- Guarantees **unbiased** responses
- **GRUNet** and **WaveNet** preferred
- **Small difference** w.r.t. real data

MODALITY

- 18 links to 18.mp3 files with encoded names
  - 15 generated tracks, three for each model
  - 3 real tracks, extracted from the dataset
- Rate on five-point Likert scale each audio (i.e. 1: Bad, 2: Poor, 3: Fair, 4: Good, 5: Excellent)
- Ratings based on the concept of naturalness
- For each network:
  - Compute mean value of ratings of all the three songs
  - Compute the standard deviation of ratings of all the three songs

Mean Opinion Score for the different architectures

	Architecture					
	WaveNet	Simplified_WaveNet	Attention_RNN	LSTMNet	GRUNet	Dataset
Mean	3.33333	2.51282	2.69231	3.30769	**3.46154**	**3.97436**
St. Dev.	1.34425	0.82308	1.10391	0.76619	0.85367	0.98641

CONCLUSION

WHAT WAS DONE

- Focus on **deep generation of jazz music**
- **MIDI Jazz songs dataset** created and made available
- **5 architectures** implemented, trained and tuned
- **Pre & post processing** methods implemented
- **WaveNet** and **GRUNet** preferred
- **Small difference** between generated and real data

WHAT WAS DONE

- Implement **independent networks** for **velocity** and **duration** of notes and combine results, as in [3]
- Implement **networks** for other **instruments** to make multi-instrumental compositions, as in [3, 8]
- **Fine tune** Magenta **pre-trained** models, as suggested in [9]

REFERENCES

[1] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. Proc. 9th ISCA Speech Synthesis Workshop, 2016, pp. 125-125.

[2] A. Géron. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow. 2019, O'Reilly, 2nd Edition, pp. 521-523.

[3] J. M. Simões, P. L. Machado, A. C. Rodrigues. Deep Learning for Expressive Music Generation. In Proceedings of the 9th International Conference on Digital and Interactive Arts (ARTECH 2019). Association for Computing Machinery, New York, NY, USA, Article 14, 1–9. DOI:https://doi.org/10.1145/3359852.3359898.

[4] https://github.com/haryoa/note_music_generator

[5] https://github.com/magenta/magenta/tree/master/magenta/models/melody_rnn

[6] https://www.kaggle.com/saikayala/jazz-ml-ready-midi

[7] A. Huang, R. Wu. Deep Learning for Music. 2016, arxiv: http://arxiv.org/abs/1606.04930.

[8] H. W. Dong, W. Y. Hsiao, L. C. Yang, Y. H. Yang. MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[9] H. Hung, C. Wang, Y. Yang and H. Wang, "Improving Automatic Jazz Melody Generation by Transfer Learning Techniques," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp. 339-346, doi: 10.1109/APSIPAASC47483.2019.9023224.