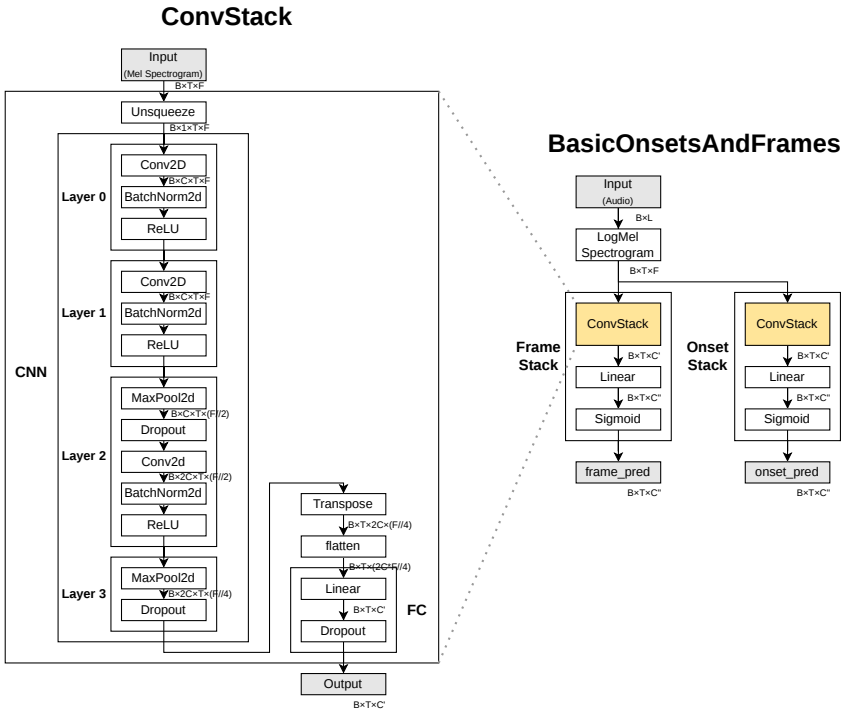
HW3

20244418 임주은

Question 1: Review the Baseline Model

Architecture



Parameters	B=16	L=102400	T=200	F=229	C=24	C′=64	C″=88
Variable name	batch_size	sequence_length	-	N_MELS	cnn_unit	fc_unit	-

Thorough Review of the Process:

Explanations are omitted for repeated blocks.

1. **Input (B×L)**

   Randomly cropped waveform with batch size B=16 and length L=102400.

2. **LogMelSpectrogram (B×T×F)**

   T=200 is audio length L=102400 divided by hop size=512, and F=229 is the number of mel bins.

3. **Frame Stack or Onset Stack (B×T×C′)**

   a. **ConvStack**

      i. **Unsqueeze (B×1×T×F)**

Treats the mel spectrogram as an image with a single channel.

ii. **CNN**

- **Layer0**

① **Conv2D (B×C×T×F)**

Applies 24 kernels to produce C = 24 output channels. The kernel size is 3×3, hop size is 1, and zero-padding of size 1 is applied to all four edges, ensuring the output shape is preserved: X//hop - (kernel-1) + padding = X.

② **BatchNorm2D**

Normalizes across the channel dimension, independently normalizing the output of each kernel.

③ **ReLU**

Element-wise activation function to give non-linearity.

- **Layer1**

Identical to layer0.

- **Layer2**

① **MaxPool2D (B×C×T×F//2)**

The kernel size is (1, 2) which reduces only the frequency domain by half, leaving the maximum value of every two adjacent values. The time domain is unchanged because the prediction is made on each input frame, thus must be one-to-one matched.

② **Dropout**

Randomly sets elements to 0.

③ **Conv2D (B×2C×T×F//2)**

Applies 48 kernels to yield 2C output channels. Otherwise identical to the previous layer.

④ **BatchNorm2D**

⑤ **ReLU**

- **Layer3**

① **MaxPool2D (B×2C×T×F//4)**

② **Dropout**

iii. **Transpose (B×T×2C×F//4)**

Transpose T and C in order to flatten last two dimensions.

iv. **Flatten (B×T×(2C*F//4))**

Regard channel and frequency dimension as feature attributes and combine them to one dimension.

v. **FC**

- **Linear (B×T×C′)**

Projects to another space.

- **Dropout**

b. **Linear (B×T×C″)**

Projects to the prediction space. In this case, 88 corresponds to the number of keys.

c. **Sigmoid**

Map the output to a value between 0 and 1, representing the probability of the prediction.

4. **Output (B×T×C″)**

Either frame_pred or onset_pred. Frame_pred predicts whether the key is being played in that frame, while onset_pred predicts whether the key started playing in that frame.

Question 2: Run the Baseline Model

Experiments

1. **Dataset and Model Hyperparameters**

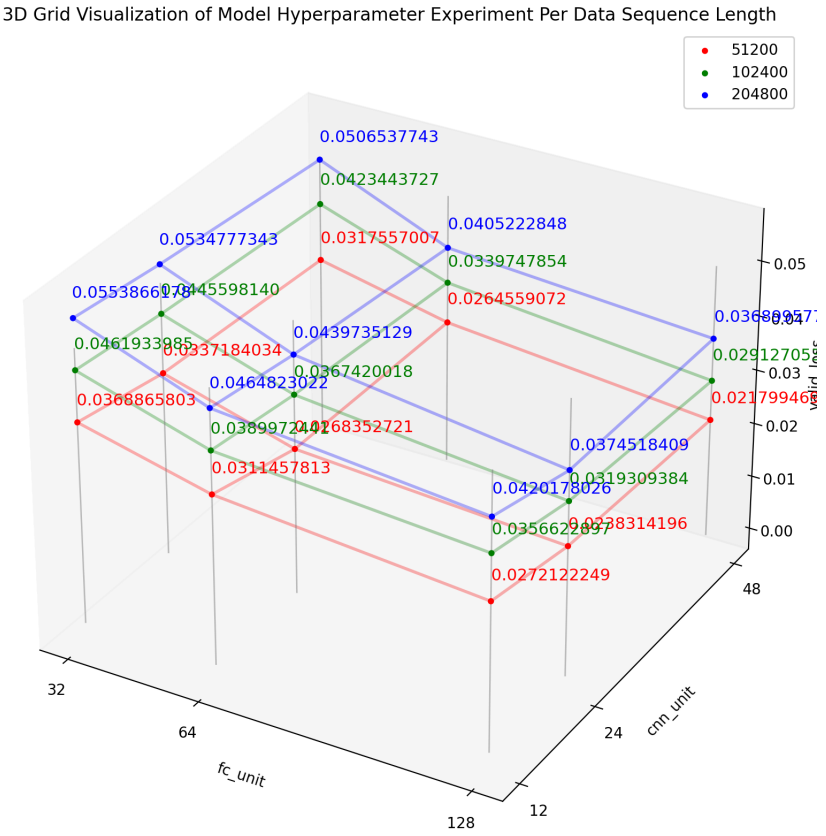
```
# Variables
sequence_lengths=[51200,102400,204800]
cnn_units=[12,24,48]
fc_units=[32,64,128]

# Fixed
batch_size = 16
iterations=3000
validation_interval=1000
learning_rate=1e-3
weight_decay=0
NUM_EPOCHS = 10
```

2. **Training Hyperparameters**

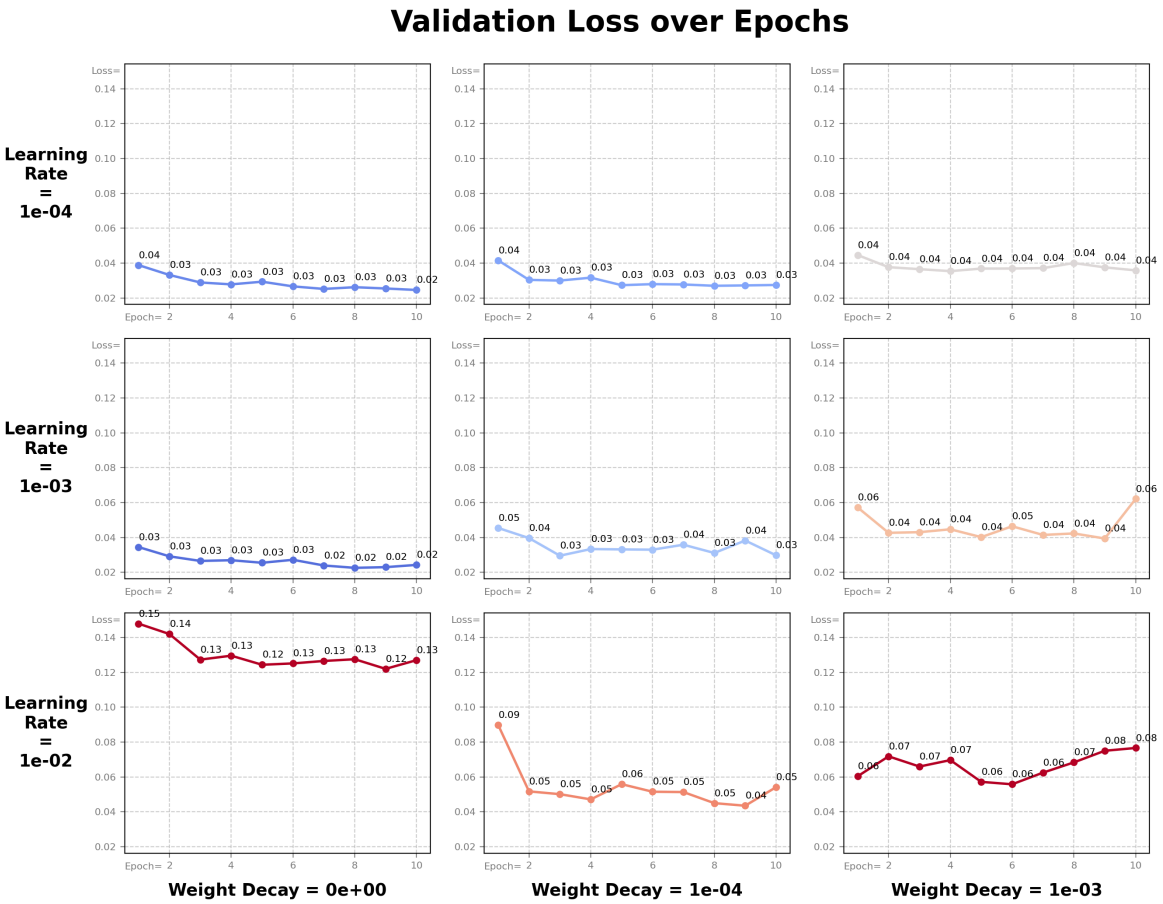
```
# Variables
learning_rates=[1e-4,1e-3,1e-2]
weight_decays=[0.0,1e-4,1e-3]

# Fixed (Best value from the previous experiment)
batch_size = 16
iterations=3000
validation_interval=1000
sequence_length=51200 # From previous experiment result
cnn_unit=48 # From previous experiment result
fc_unit=128 # From previous experiment result
NUM_EPOCHS = 10
```

Results

1. **Dataset and Model Hyperparameters**



2. **Training Hyperparameters**



Discussions

1. **Dataset and Model Hyperparameters**
   - Shorter data sequence result in higher performance.

     → Shorter frames capture frequent changes more effectively, whereas longer frames tend to average out the information, potentially distorting it and making it more vague.
   - Increasing any model dimension improves performance.

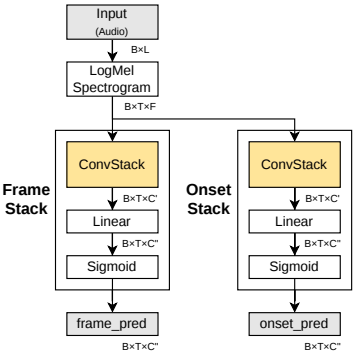
     → A more complex model provides additional capacity, allowing it to better learn and adapt to the data distribution.
2. **Training Hyperparameters**
   - A learning rate of 0.001 and a weight decay of 0 yielded the best performance.

     → While a trend is showing, the results appear to be more sensitive to the specific context and the combination of hyperparameters used. Needs more extensive experiment to prove the trend.

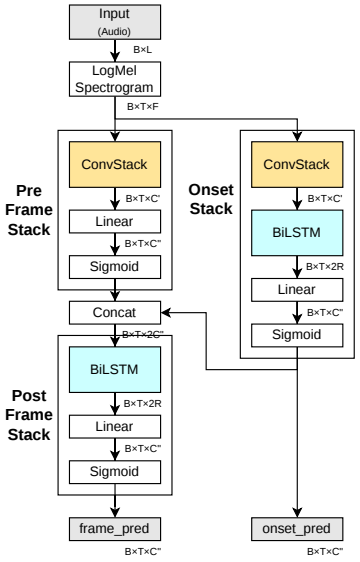
Question 3: Implement the Onsets and Frames Model

Architecture

BasicOnsetsAndFrames



OnsetsAndFrames



Parameters	B=16	L=102400	T=200	F=229	C′=64	C″=88	R=64
Variable name	batch_size	sequence_length	-	N_MELS	fc_unit	-	rnn_unit

Experiments

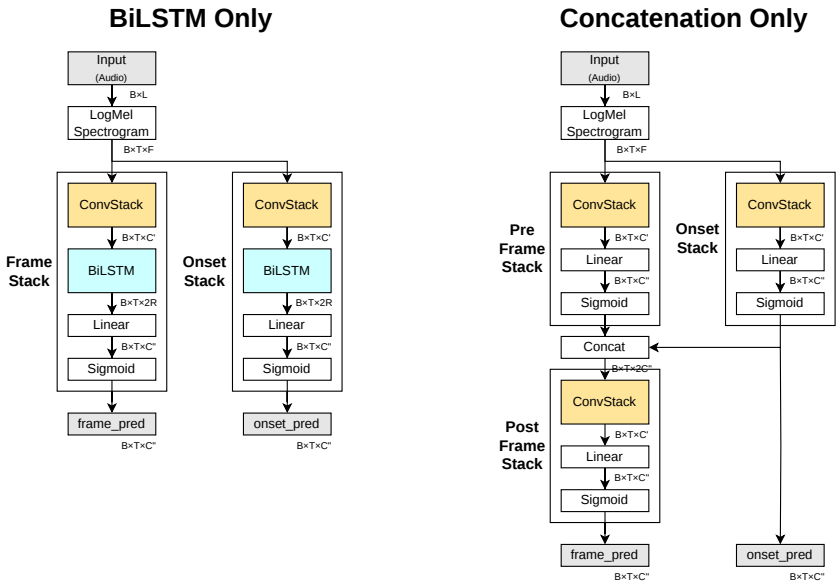
1. **RNN Unit Comparison**

```
# Variables
rnn_units=[32, 64, 128]

# Fixed (Best value from the previous experiment)
batch_size = 16
iterations=3000
validation_interval=1000
sequence_length=51200
cnn_unit=48
fc_unit=128
learning_rate = 1e-3 # From previous experiment result
weight_decay = 0.0 # From previous experiment result
NUM_EPOCHS = 20 # Increased since the model compexity is higher, and needs more training to converge.
```

2. **Ablations**

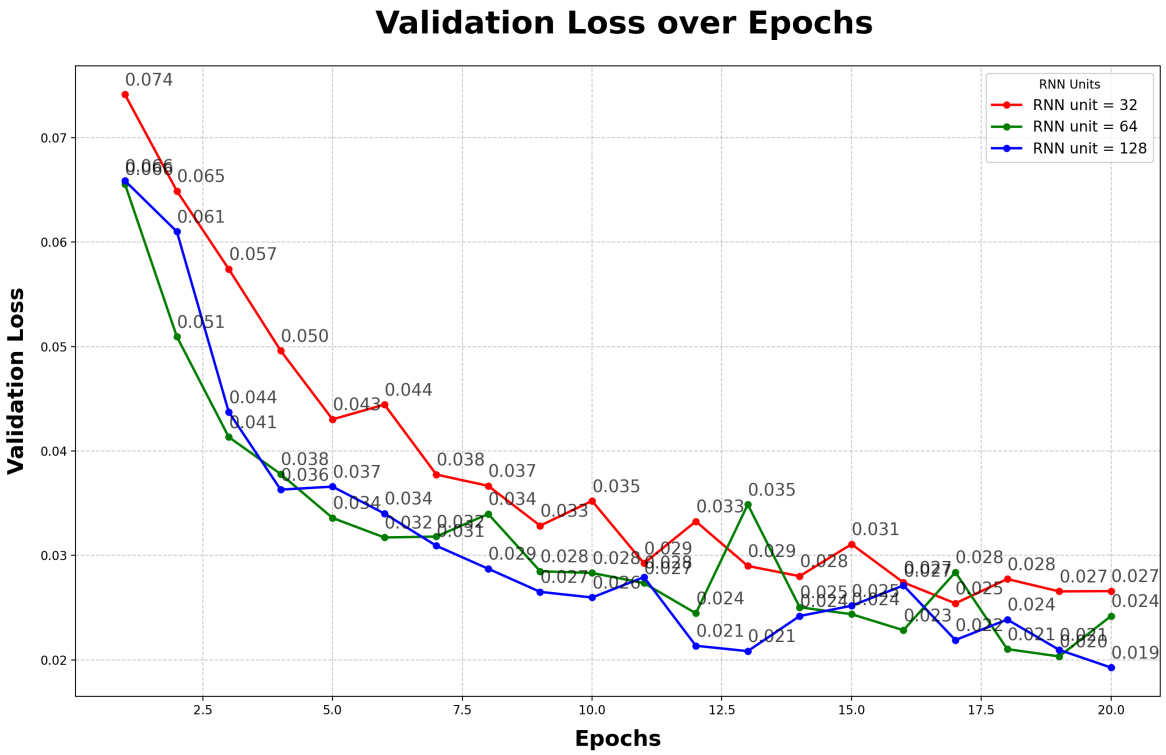
```
# Fixed (Best value from the previous experiment)
batch_size = 16
iterations=3000
validation_interval=1000
sequence_length=51200
cnn_unit=48
fc_unit=128
rnn_units=128 # From previous experiment result
learning_rate = 1e-3
weight_decay = 0.0
NUM_EPOCHS = 20

# Variables
models = {
  'BasicOnsetsAndFrames': BasicOnsetsAndFrames(cnn_unit=cnn_unit, fc_unit=fc_unit),
  'BiLSTMOnly': BiLSTMOnly(cnn_unit=cnn_unit, fc_unit=fc_unit, rnn_unit=rnn_unit),
  'ConcatOnly': ConcatOnly(cnn_unit=cnn_unit, fc_unit=fc_unit),
```

```
  'OnsetsAndFrames': OnsetsAndFrames(cnn_unit=cnn_unit, fc_unit=fc_unit, rnn_unit=rnn_unit)
}
```

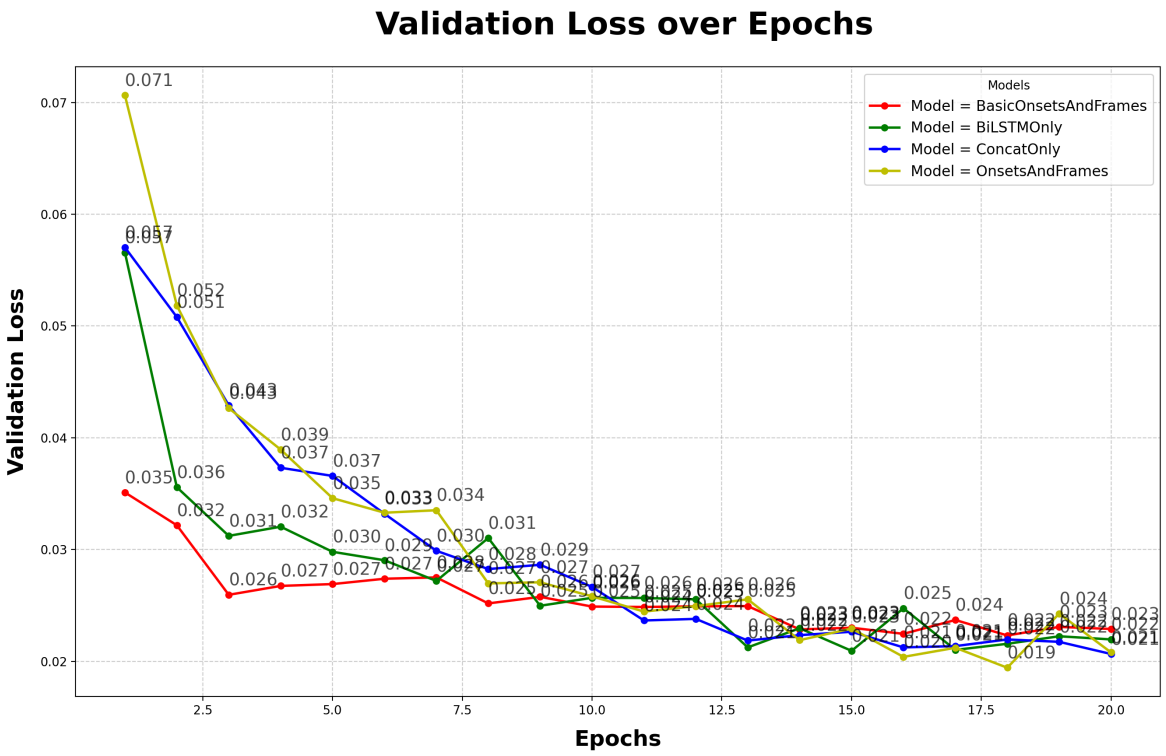


Results

1. **RNN Unit Comparison**



2. **Ablations**



Discussions

1. **RNN Unit Comparison**
   - Increasing the number of RNN units improves performance.

     → This aligns with the findings from the model hyperparameter experiment on the base model: a higher complexity model has more capacity to learn. However, the performance gain starts to saturate as the number of units increases, potentially due to overfitting or structural limitations.
2. **Ablations**
   - OnsetsAndFrames outperforms all other models, while the baseline model performs the worst.

     → All features are proved to be beneficial.
   - There is a marginal gap between concatenated group and non-concatenated group.

     → Concatenation proves to be more effective than replacing CNN with BiLSTM. This suggests that sharing onset prediction information with the frame prediction is more beneficial than merely increasing model complexity, and indicates the presence of redundant information within the two independent stacks.