

```
AttributeError: module 'tensorflow._api.v2.compat.v1.nn' has no attribute 'min_pool'
```

It looks like we've hit another error in the code execution. The error message indicates that TensorFlow doesn't have a function called `min_pool` in its `tf.nn` module.

Let's look at the issue more closely:

Analysis of the Error

The error `AttributeError: module 'tensorflow._api.v2.compat.v1.nn' has no attribute 'min_pool'` occurs because:

1. In the `OnsetNet_Gina5` class, there's code that attempts to use `tf.nn.min_pool` for the pooling operation when `pooling_method` is set to 'min'.
2. However, TensorFlow only provides `tf.nn.max_pool` and `tf.nn.avg_pool` functions natively, but not `tf.nn.min_pool`.

Solution

We need to modify the `OnsetNet_Gina5` class to handle the 'min' pooling method differently, since there's no direct `min_pool` function in TensorFlow. Here are a few ways to implement this:

Option 1: Implement a custom min pool function

The best approach would be to implement min pooling using TensorFlow operations:

```
python

def min_pool(input, ksize, strides, padding):
    """Custom implementation of min pooling using TensorFlow operations."""
    # Negate the input tensor, apply max pooling, then negate again
    return -tf.nn.max_pool(-input, ksize=ksize, strides=strides, padding=padding)
```

Option 2: Modify the pooling_method selection in the `OnsetNet_Gina5` class

Find the sections in `OnsetNet_Gina5` that use `tf.nn.min_pool` and replace them with the custom implementation above. Look for something like:

python

```
if pooling_method == 'max':
    pooled = tf.nn.max_pool(...)
elif pooling_method == 'avg':
    pooled = tf.nn.avg_pool(...)
elif pooling_method == 'min':
    pooled = tf.nn.min_pool(...) # This is the line causing the error
```

And change it to:

python

```
if pooling_method == 'max':
    pooled = tf.nn.max_pool(...)
elif pooling_method == 'avg':
    pooled = tf.nn.avg_pool(...)
elif pooling_method == 'min':
    # Custom min pooling implementation
    pooled = -tf.nn.max_pool(-input_tensor, ksize=pool_shape, strides=pool_shape, padding='SAME')
```