NuDot Unet

Time sensitive pulse extraction

Masooma Sarfraz

April, 28, 2023

Formatted input

Before training and test run

- Still needs modification to limit the wave generation between 0-250 samples.
- Time_values in the following format:

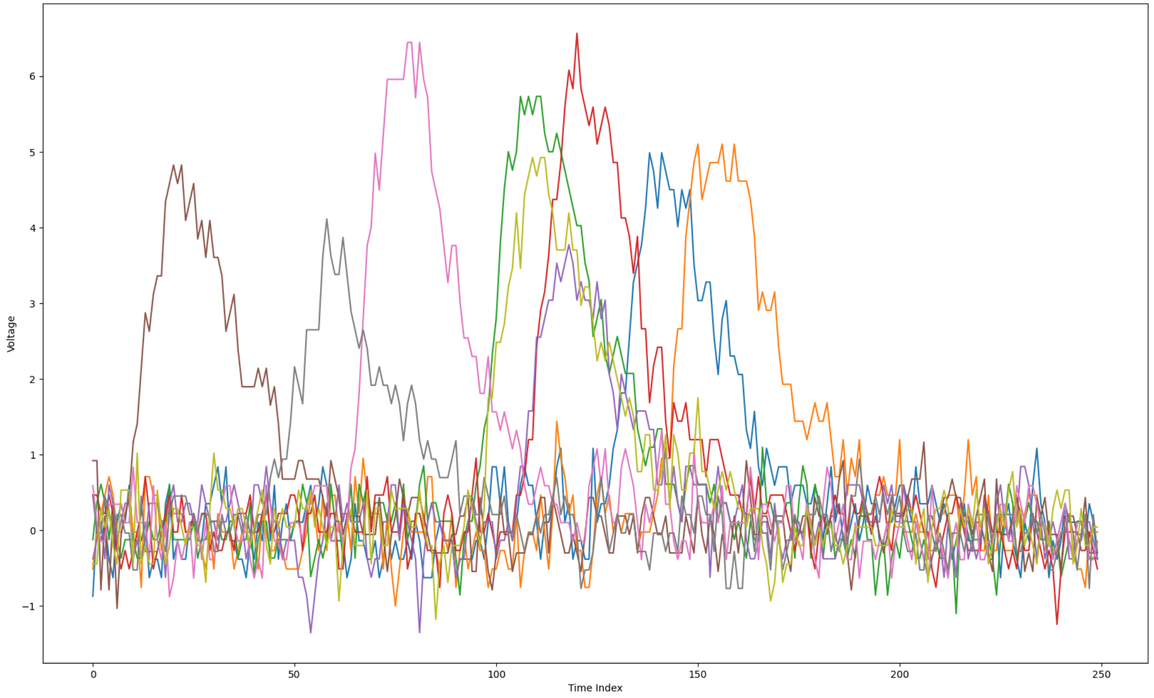
time for 2 waves :-

[[[0 1], [0 1] ..... [1 0], [1 0], [1 0] ..... [0 1]],

[[0 1], [0 1], [0 1] ..... [1 0], [1 0], [1 0] ..... [0 1]]

instead of

[[20 ns], [15 ns]]

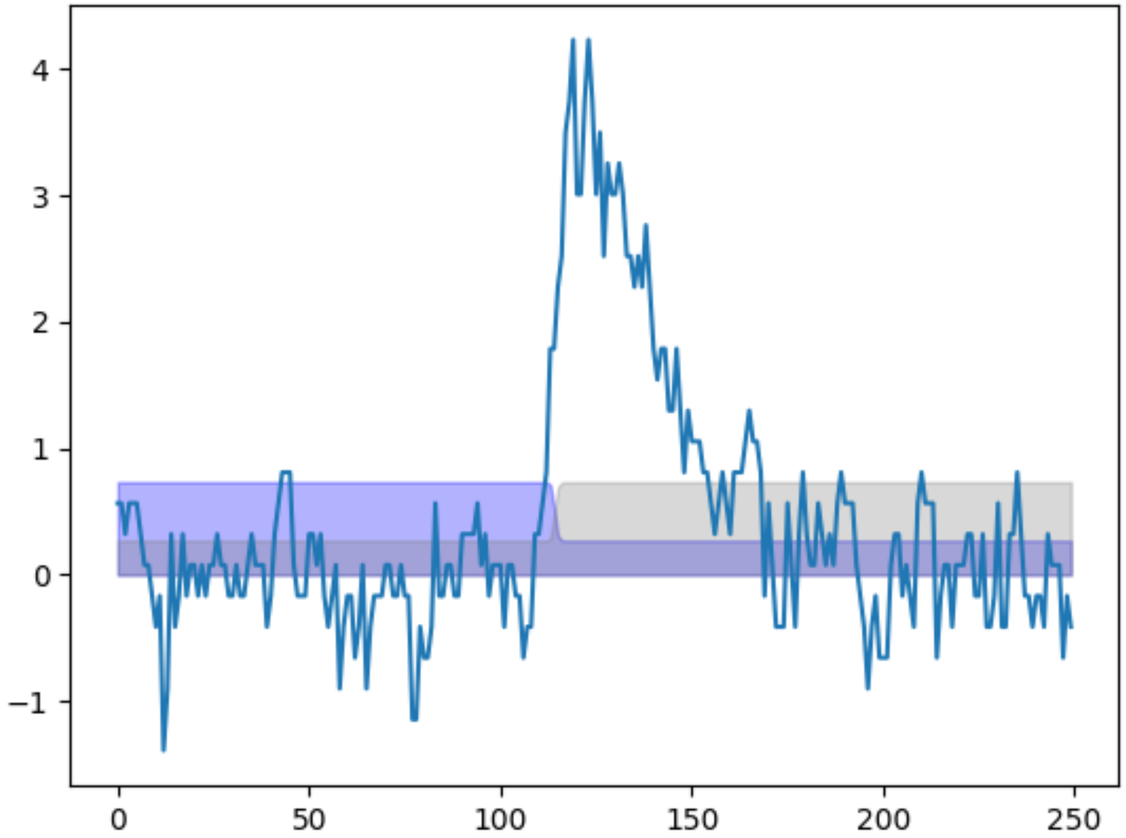


Unet Model

Training and Testing

- Training the waveforms along with the truth, we got this kind of output:
- Condition set to >2mV.

```
[0.7309, 0.2691],
[0.7309, 0.2691],
[0.7309, 0.2691],
[0.7310, 0.2690],
[0.7310, 0.2690],
[0.7310, 0.2690],
[0.7310, 0.2690],
[0.7310, 0.2690],
[0.7304, 0.2696],
[0.7103, 0.2897],
[0.5234, 0.4766],
[0.3073, 0.6927],
[0.2713, 0.7287],
[0.2692, 0.7308],
[0.2690, 0.7310],
[0.2690, 0.7310],
[0.2691, 0.7309],
[0.2692, 0.7308],
[0.2693, 0.7307],
[0.2693, 0.7307],
```



True_Times

Formatting into 0s and 1s

- Insert this data in ML code with an additional dimension for waveforms.

```
length = 753
[[-0.02509523  0.08788292  0.11919804 ...  0.16718494  0.14869807
   0.15670731]
 [ 0.02785448  0.04291223  0.11287905 ... -0.04830048  0.08278078
  -0.00976977]
 [ 0.16850564 -0.01312803  0.1591846  ... -0.07343087  0.05831218
  -0.05325997]
 ...
 [-0.1014102  -0.09217753 -0.03834512 ... -0.13617991 -0.02111919
   0.04802738]
 [-0.05077456 -0.05660413 -0.1128265  ...  0.02546225  0.03628132
   0.09512475]
 [-0.08835234 -0.02359074 -0.05920883 ... -0.22766901 -0.04950122
  -0.00197119]]
time bins
753 [[[0 1]
  [0 1]
  [0 1]
  ...
  [1 0]
  [1 0]
  [1 0]]

 [[0 1]
  [0 1]
  [0 1]
  ...
  [1 0]
  [1 0]
  [1 0]]

 [[0 1]
  [0 1]
  [0 1]
  ...
  [1 0]
  [1 0]
  [1 0]]
```

Explanation of time bins

What do 0s and 1s refer to?

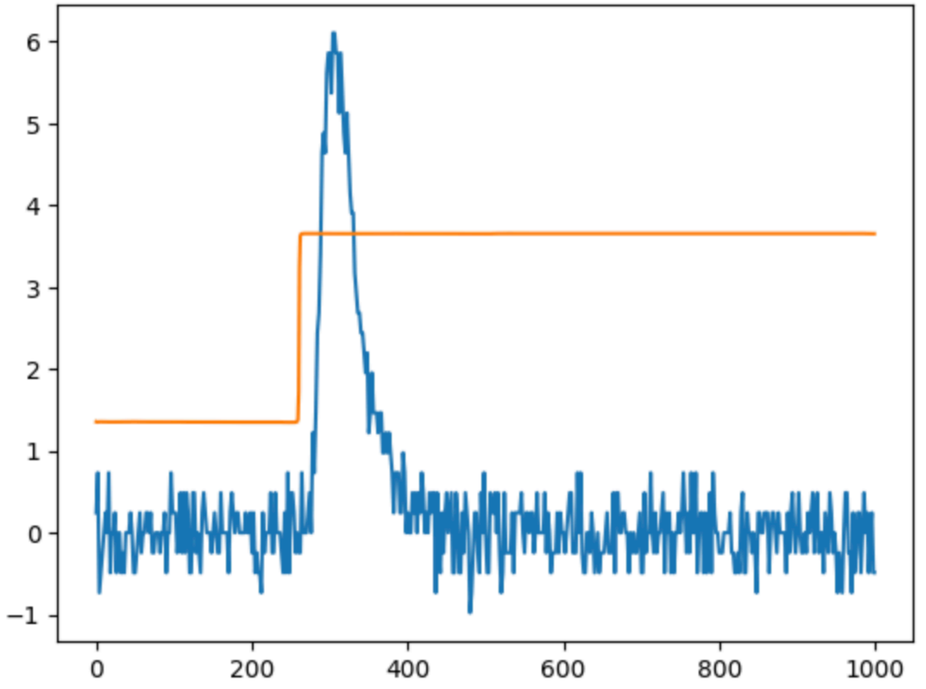
How to extract charge from these pulses?

Latest Results:

Train Waveform with the truth

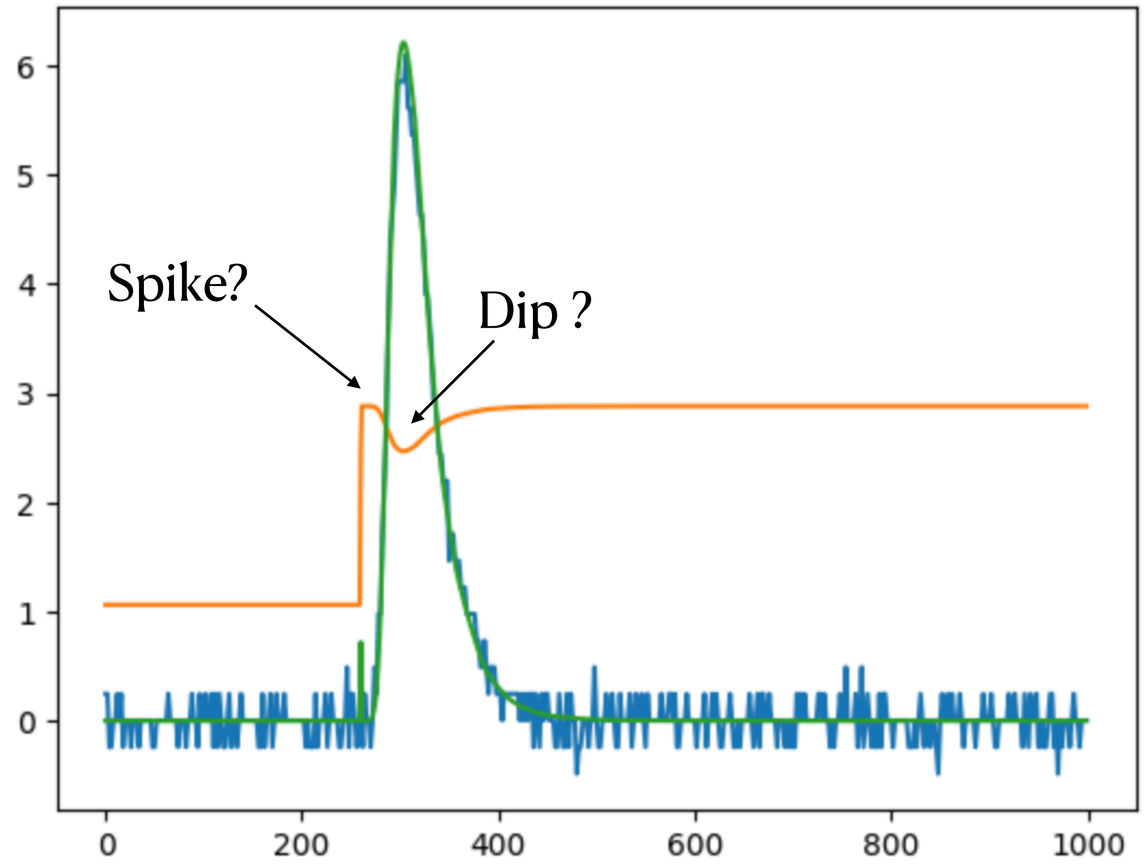
```
tensor([[[0.2710, 0.7290],
         [0.2702, 0.7298],
         [0.2705, 0.7295],
         ...,
         [0.7305, 0.2695],
         [0.7301, 0.2699],
         [0.7300, 0.2700]]],
```

Instead of 0 and 1?



Adding charge ?

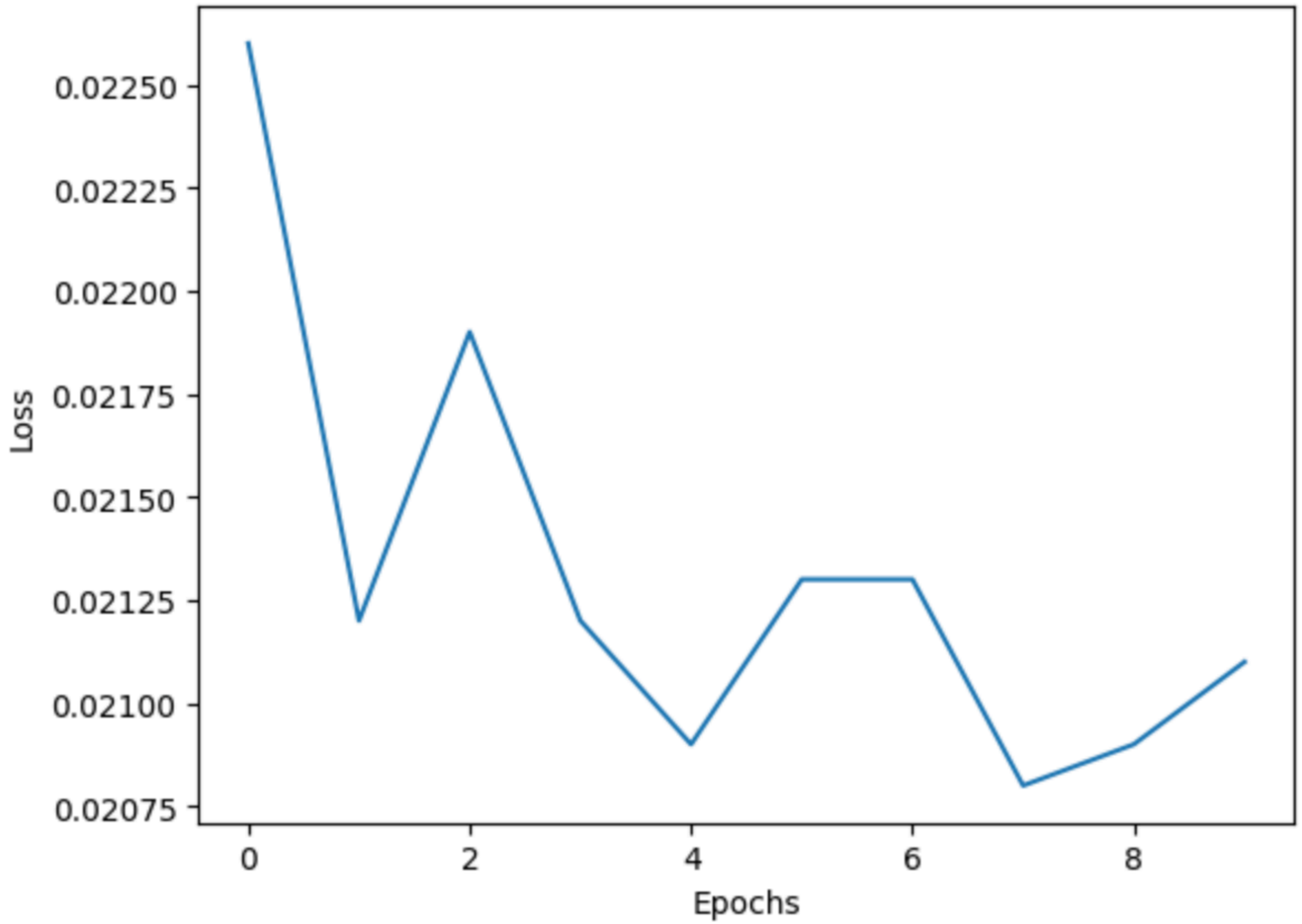
```
tensor([[[0.2120, 0.5760, 0.2120],
         [0.2119, 0.5761, 0.2119],
         [0.2119, 0.5761, 0.2119],
         ...,
         [0.5761, 0.2119, 0.2119],
         [0.5761, 0.2119, 0.2119],
         [0.5761, 0.2119, 0.2119]]], grad_fn=<SoftmaxBackward0>)
[array([1.0599922, 2.880231 , 1.0597771], dtype=float32)]
```



Evaluation

Efficiency of training

- Epochs vs. Loss function
- Lower loss —> better performance on the task



```
class UNet(nn.Module):
    def __init__(self):
        super(UNet, self).__init__()
        self.bilinear = True

        multi = 40

        self.inc = DoubleConv(1, multi) #1 instead of 12 for feeding 1 wave form at a time --> input channels
        self.down1 = Down(multi, multi*2)
        self.down2 = Down(multi*2, multi*4)
        self.down3 = Down(multi*4, multi*8)
        factor = 2 if self.bilinear else 1
        self.down4 = Down(multi*8, multi*16 // factor,pool=False)
```

Multi =40

Effects the number of filters in the network

Changing the value to any higher value might lead to overfitting ?

Changing it to lower value may lead to underfitting?