

ANN vs FFT Project

דאטה:

$$V_1 = V_2 = \{50, 53, 56 \dots 130\}$$

$$T = \left\{ \frac{20}{1e5}, \frac{22}{1e5} \dots \frac{180}{1e5} \right\}$$

$$N = \{2^6, 2^7, 2^8 \dots 2^{12}\}$$

$$D = V_1 \times V_2 \times T \times N$$

הדאטה חולק לסיגנלים רעים וטובים כאשר סיגנל נחשב טוב אם ורק אם מתקיים:

$$\Delta F_{doppler} > threshold \cdot DopplerResolution$$

$$. threshold = 3, DopplerResolution = \frac{1}{T \cdot N}, \Delta F_{doppler} = \frac{2|v_1 - v_2|}{\lambda}$$

כאשר

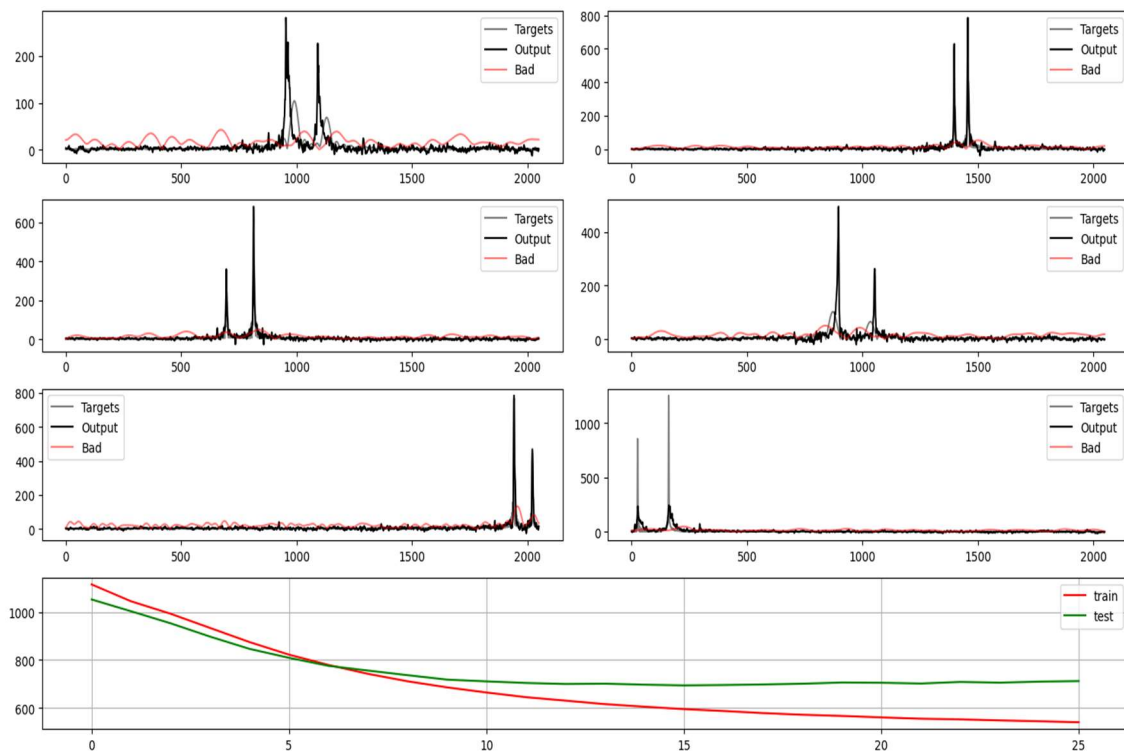
זוג סיגנלים נכנסו לרשת (הרע כ-*inputs* והטוב כ-*targets* בשביל ה-*loss*).

מודל:

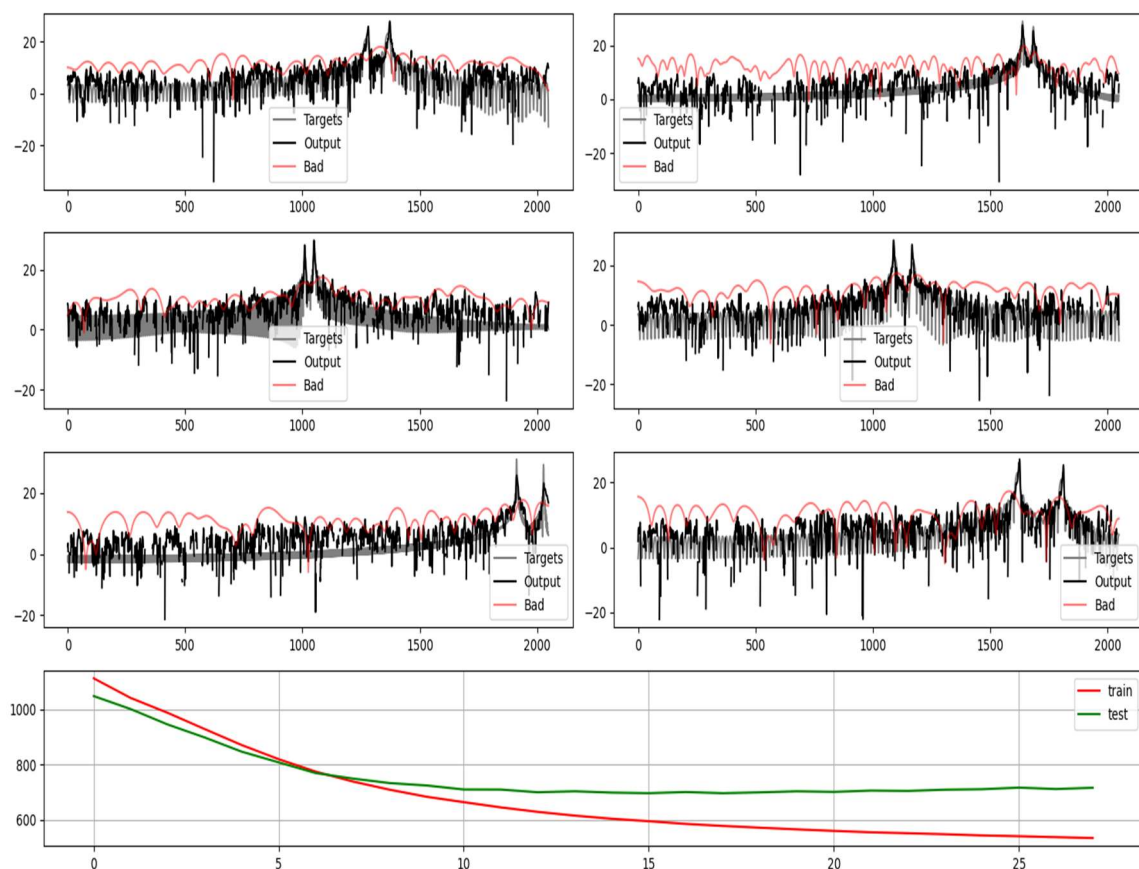
Model: "sequential"		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 4096)	16781312
batch_normalization (Batch Normalization)	(None, 4096)	16384
activation (Activation)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16781312
batch_normalization_1 (Batch Normalization)	(None, 4096)	16384
activation_1 (Activation)	(None, 4096)	0
dense_2 (Dense)	(None, 4096)	16781312
batch_normalization_2 (Batch Normalization)	(None, 4096)	16384
activation_2 (Activation)	(None, 4096)	0
dense_3 (Dense)	(None, 4096)	16781312
batch_normalization_3 (Batch Normalization)	(None, 4096)	16384
activation_3 (Activation)	(None, 4096)	0
dense_4 (Dense)	(None, 2049)	8394753
=====		
Total params: 75,585,537		
Trainable params: 75,552,769		
Non-trainable params: 32,768		

תוצאות (6 דגימות מה $test\ set$ ופונקציית $loss$ ביחס ל $epoch$):

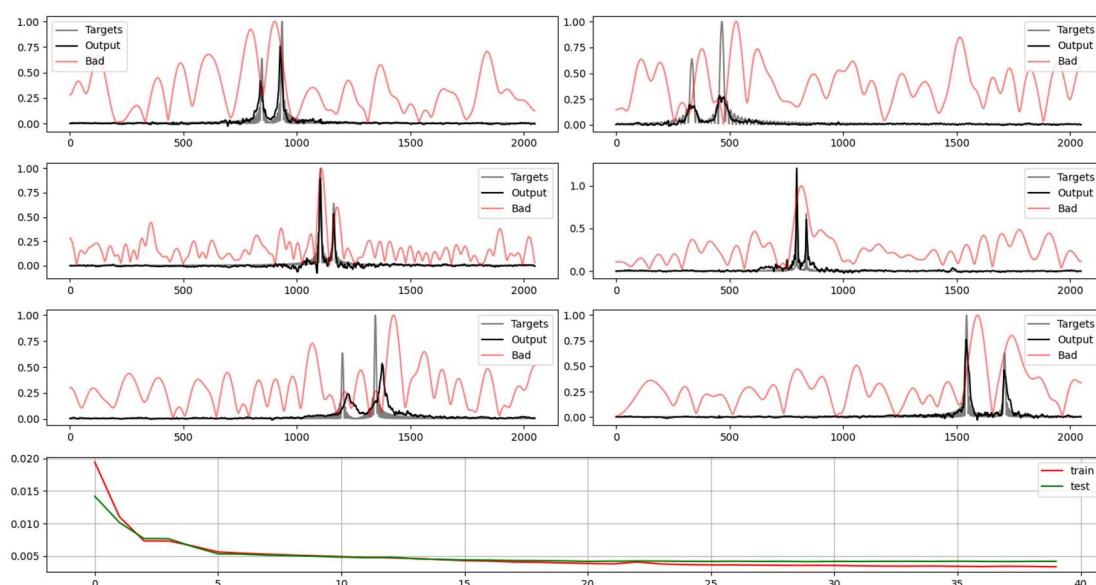
Adam optimizer, $lr = 0.001$, batch size = 32, loss = MSE, unnormalized fft



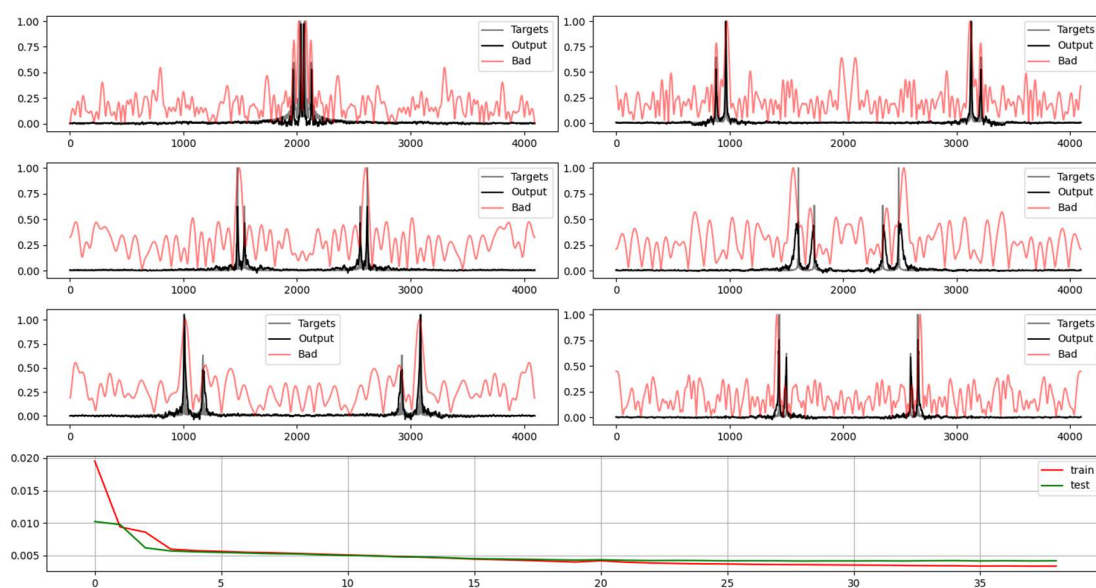
Same model but the result are in db scale



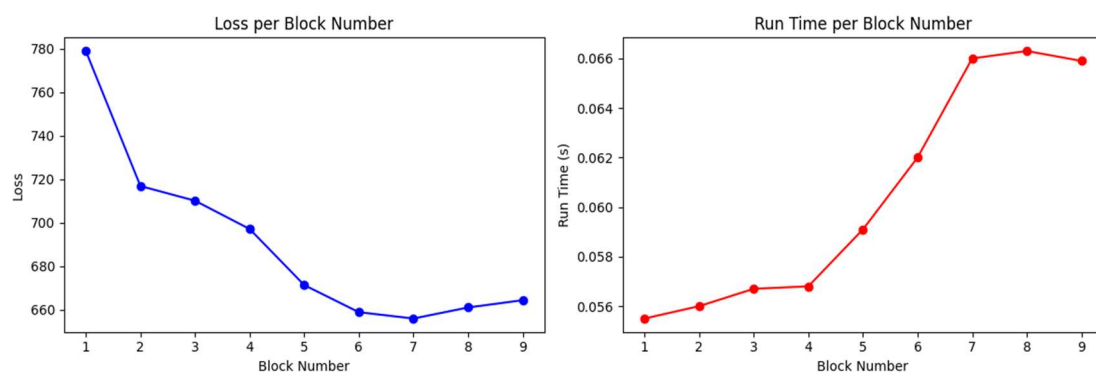
Adam optimizer, $lr = 0.001$, batch size = 32, loss = MSE, normalized fft



Like the last one but with the full fft



לכן ניתן לומר שאין כל כך הבדל בין השיטות השונות מבחינת ביצועים הרשת יודעת לעשות את ההפרדה, כרגע נשאר עם הגרסה הראשונה:



תוצאות פר מספר בלוקים (שכבה לינארית + שכבת נורמליזציה) + שכבת פלט לינארית.