





دانشگاه شهید بهشتی  
دانشکده برق

پروژه نهایی کارشناسی مهندسی برق

## پردازش سیگنال نوار قلبی (ECG) بوسیله شبکه های عصبی در پایتون

استاد راهنما: دکتر علیرضا یزدی زاده

نگارش: سجاد رضوانی خالدي

زمستان ۱۴۰۲



83  
P2-4AC

15.0 cm MI 0.9  
Gen TIs 1.1  
[2d] G78/71 d  
FA4/P90  
HAR/FS10  
[C] G50/0.80 kHz  
FA5/F1/B  
TDI

3

Time (s)

# مقدمه پژوهش

# بیان مسئله

- بررسی سیگنال های قلبی که در حال حاضر در مراکز درمانی توسط پزشکان صورت می گیرد، می تواند با کمک هوش مصنوعی، با دقت، سرعت و دسترسی بیشتری صورت گیرد.
- یکی از اولین روش های تشخیص اولیه بیماری ها در اورژانس، گرفتن نوار قلب و بررسی آن توسط پزشک متخصص است.
  - همواره پزشکانی با دانش و تخصص بالا در دسترس نیستند.
  - بررسی تعداد زیادی نوار قلب برای پزشک متخصص در زمان محدودی که دارد، زمان بر است.
  - پزشکان دارای خطای انسانی هستند.

# اهداف پروژه

هدف از این پروژه طراحی مدلی بر پایه شبکه عصبی است که در ورودی ۱۲ سیگنال نوار قلبی ساجکت را می گیرد و با دقت خوبی در خروجی شبکه، کلاس بیماری قلبی را در یکی از ۵ کلاس مشخص شده تعیین می کند.

با یاری گرفتن از این سیستم ها در حوزه پزشکی می توان بر همه ی نواقص سیستم های تمام انسانی که ذکر شد از جمله موارد زیر غلبه کرد:

- سیستم کامپیوتری می تواند همواره در دسترس باشد و مانند سیستم های انسانی خسته نمی شود و به استراحت نیاز ندارد.
- با استفاده از سیستم های کامپیوتری می توان به صورت مرتب و در زمان های مشخص دیتا لحظه ای افراد را پردازش کرد.
- سیستم های کامپیوتری بخصوص هوش مصنوعی ثابت کرده اند که می توانند در تسک های ریاضیاتی و پردازش سیگنالی سریع تر و دقیق تر از انسان ها باشند.

# اهمیت و ضرورت پژوهش

- بیماری های قلبی یکی از موارد شایع مرگ میر در میان افراد مخصوصا در سنین بالاتر و علت اصلی مرگ و میر در بین مردان، زنان هستند.
- هر ۳۳ ثانیه یک نفر در ایالات متحده به علت بیماری قلبی عروقی جان خود را از دست می دهد.
- حدود ۶۹۵۰۰۰ نفر در ایالات متحده در سال ۲۰۲۱ از بیماری قلبی جان خود را از دست دادند که این به معنی یک نفر در هر پنج نفر است.

(۱) مقاله ECG Signal Classification Using Deep Learning Techniques Based on the PTB-XL Dataset

کاری از Sandra ´Smigiel, Krzysztof Pałczy ´nski, and Damian Ledzi ´nski

Table 3. The results of the convolutional network.

Number of Classes	Acc
2	0.882
5	0.72
20	0.589

Table 4. The results of SincNet.

Number of Classes	Acc
2	0.858
5	0.73
20	0.593

Table 5. The results of the convolutional network with entropy features.

Number of Classes	Acc
2	0.892
5	0.765
20	0.698

# پیشینه پژوهش

**Table 7. Results for five-class classification.**

Name	Acc
QRS, Raw signal, Raw signal entropy	79.1–74.9%
QRS, QRS entropy	78.0–75.2%
QRS, QRS entropy, Raw signal	77.7–73.6%
QRS, QRS entropy, Raw signal entropy	77.4–75.2%
QRS entropy, Raw signal, Raw signal entropy	77.2–75.3%
Raw signal	77.2–74.0%
QRS	77.1–75.1%
QRS, QRS entropy, Raw signal, Raw signal entropy	76.9–74.8%
QRS, Raw signal	76.7–74.9%
QRS, Raw signal entropy	76.5–73.5%
QRS entropy, Raw signal	76.5–74.7%
Raw signal, Raw signal entropy	76.2–73.9%
QRS entropy, Raw signal entropy	70.5–68.2%
QRS entropy	70.0–68.0%
Raw signal entropy	65.1–63.7%

مقاله (۲) Deep Learning Techniques in the Classification of ECG Signals  
Using R-Peak Detection Based on the PTB-XL Dataset

کاری از Sandra ´Smigiel, Krzysztof Pałczy ´nski, and Damian Ledzi ´nski



# پیشینه پژوهش

Table 4. Results for five-class classification

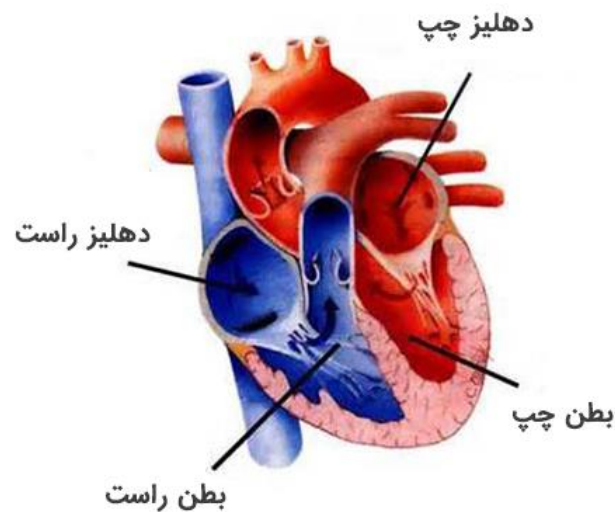
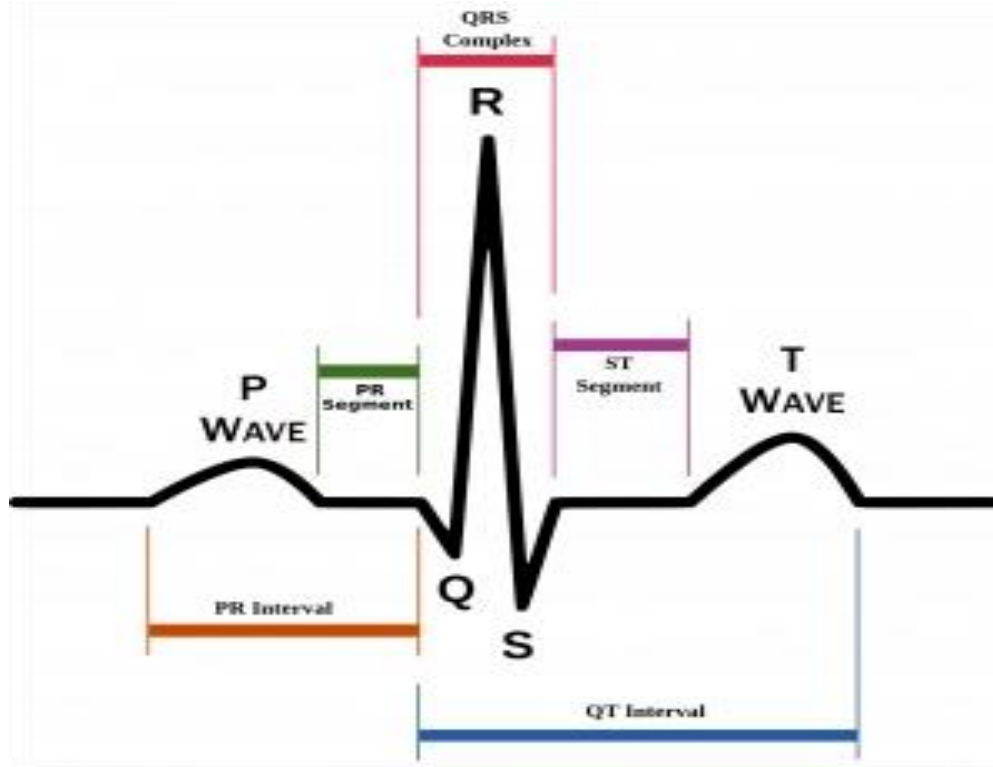
Technique	Acc
FSL proximity-based	69.8–74.2%
Softmax-based classification	75.1–77.1%
FSL + XGBoost	74.8–76.1%
FSL + Random Forest	75.2–77.7%
FSL + Decision Tree	67.0–68.5%
FSL + KNN – 5 neighbors	74.4–76.7%
FSL + KNN – 20 neighbors	77.3–79.5%
FSL + SVM with linear kernel	77.0–79.8%
FSL + SVM with polynomial kernel	74.5–76.9%
FSL + SVM with RBF kernel	77.9–80.2%
FSL + SVM with Sigmoid kernel	64.4–76.6%

۳) مقاله Study of the Few-Shot Learning for ECG Classification Based on the PTB-XL Dataset از Krzysztof Pałczyński و همکاران که با ترکیب روش ها کلاس بندی SVM و KNN با تکنیک های few shot learning به دقت های مقابل رسیدند:

# سیگنال قلبی ECG (نوار قلب)

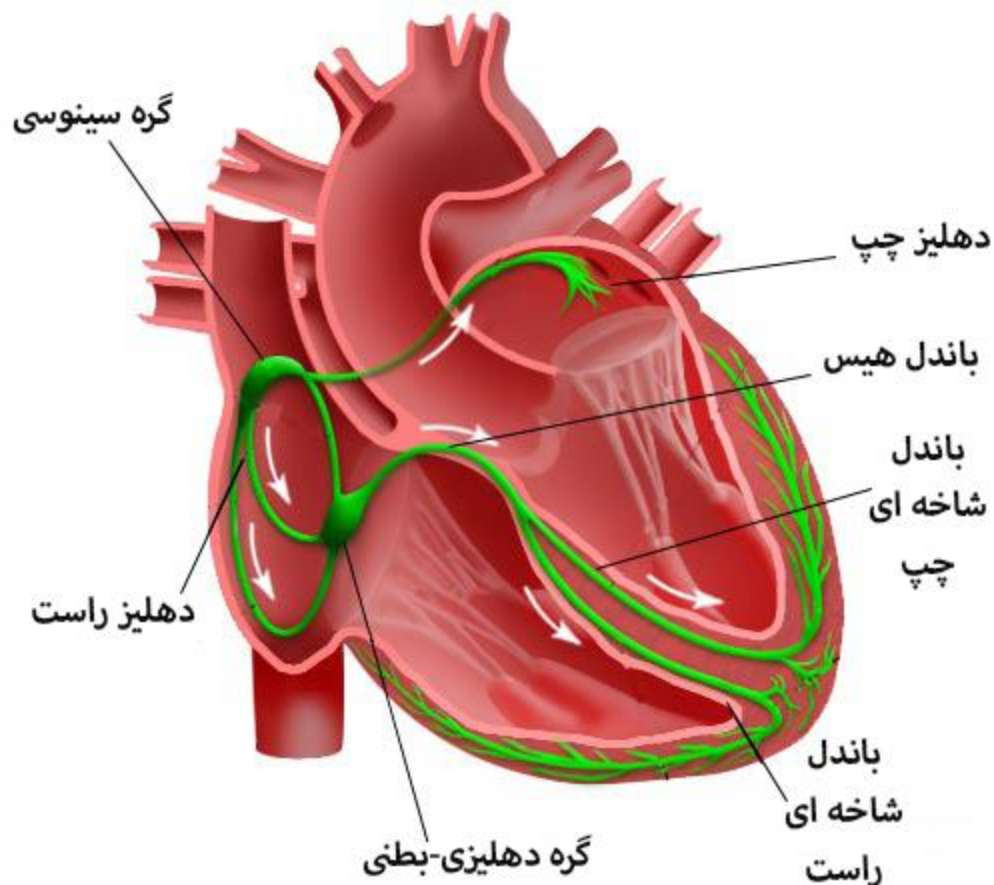
نوار قلب یک نمودار دوبعدی است که شدت الکتریسیته را برحسب زمان نشان میدهد. هر نوار ECG شامل چندین بخش اصلی است که در شکل زیر می‌بینید:

- موج P: زمانی که دهلیزها برای پمپاژ خون به داخل بطن‌ها منقبض می‌شوند.
- کمپلکس QRS: با دیپولاریزاسیون بطن‌های راست و چپ که همان انقباض بطن‌ها است، مطابقت دارد.
- موج T: نشان دهنده رپلاریزاسیون بطنی است، یعنی زمانی که بطن‌ها شل می‌شوند.



# سیگنال قلبی ECG (نوار قلب)

قلب برای کارکرد صحیح به سیستم الکتریکی منظم نیاز دارد. این سیستم هماهنگی انقباضات ماهیچه ای را ایجاد می کند که نیروی لازم برای حرکت خون را فراهم می کنند. دیاگرام این سیستم را در شکل روبه رو مشاهده می کنید:



# روش ضبط سیگنال

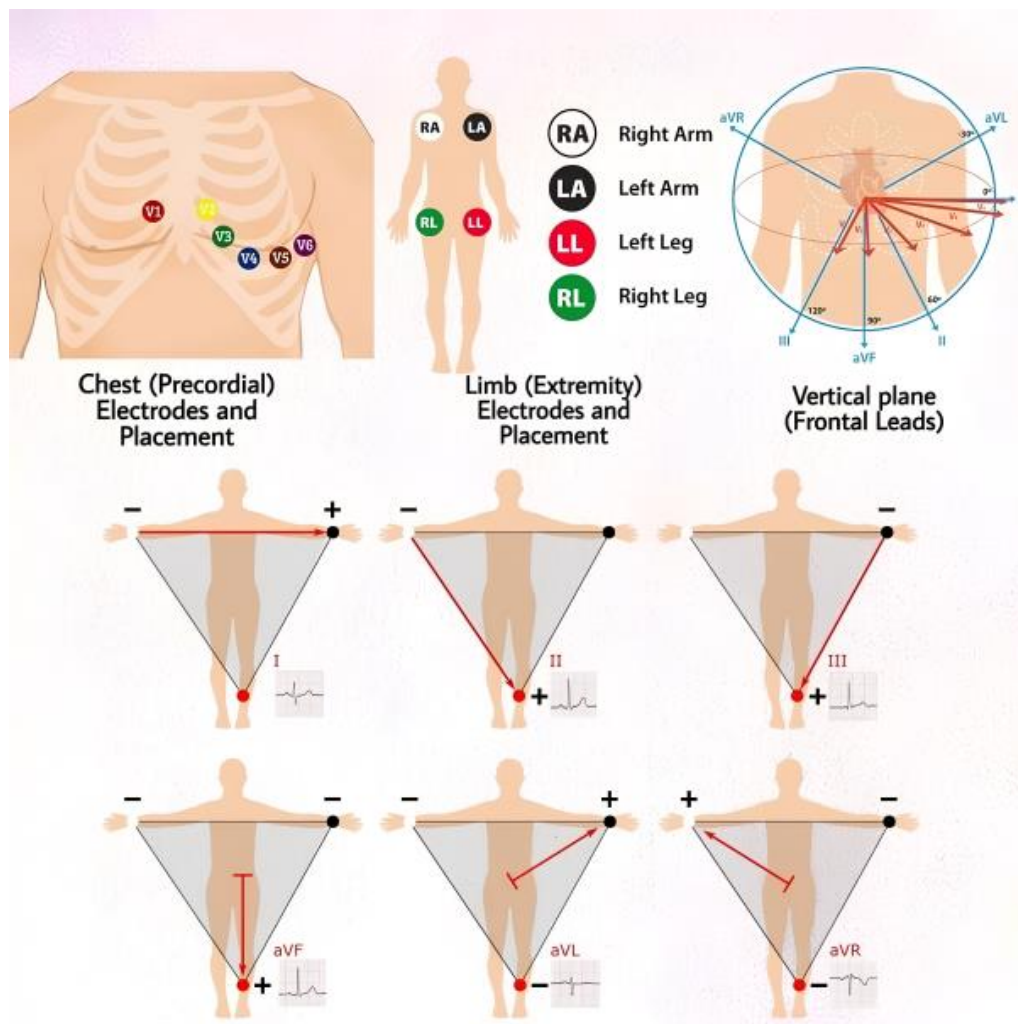
نوار قلب معمولاً از ۱۲ کانال ضبط می‌شود که این کانال‌ها در نوارهای ECG به شرح زیر هستند و محل قرارگیری آنها را در شکل روبه رو می‌بینید:

(۱) Lead I, II, III, aVR, aVL, aVF:

شش کانال اصلی که از الکترودها که به لیدهای اندامی شناخته می‌شوند.

(۲) V1 تا V6:

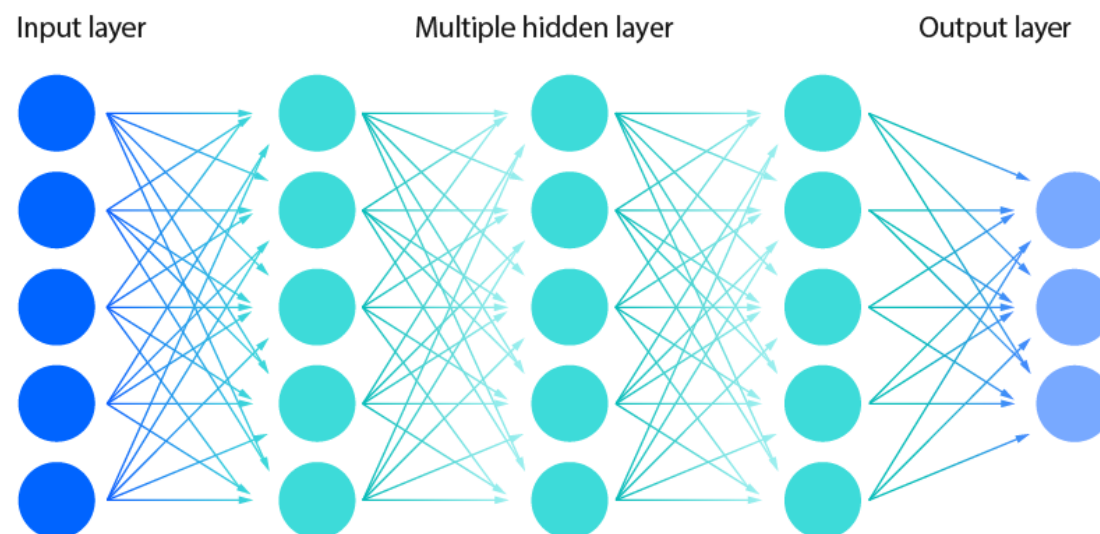
شش کانال دیگر که الکترودها روی قفسه سینه قرار دارند.



# هوش مصنوعی و شبکه های عصبی

در دهه اخیر هوش مصنوعی و شبکه های عصبی تحول زیادی کردند و کاربرد آنها در بسیاری از زمینه ها از جمله پزشکی افزایش یافته است. شبکه های عصبی که قسمتی از هوش مصنوعی بنام یادگیری عمیق هستند تاثیرات گسترده ای در تشخیص، پیش بینی و درمان بیماری ها داشته اند. در شکل زیر شمای کلی یک شبکه عصبی را مشاهده می کنید:

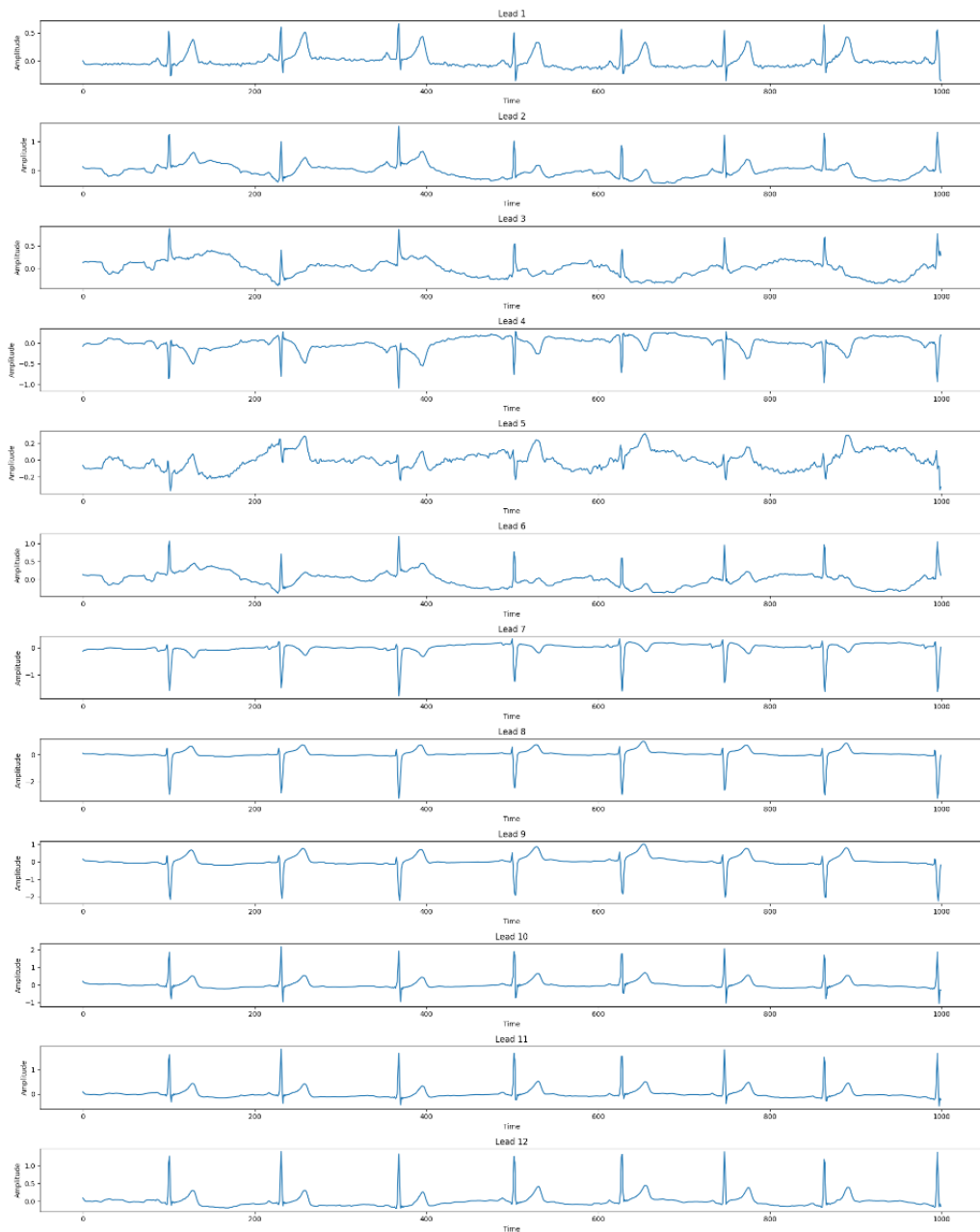
## Deep neural network



# بررسی مجموعه داده ها

# دیتاست

مجموعه داده ECG PTB-XL یک مجموعه داده بزرگ از  
۲۱۷۹۹ داده از ۱۲ کانال ECG بالینی از ۱۸۸۶۹ بیمار با طول  
سیگنال ۱۰ ثانیه است.



# کلاس بندی لیبل های دیتا

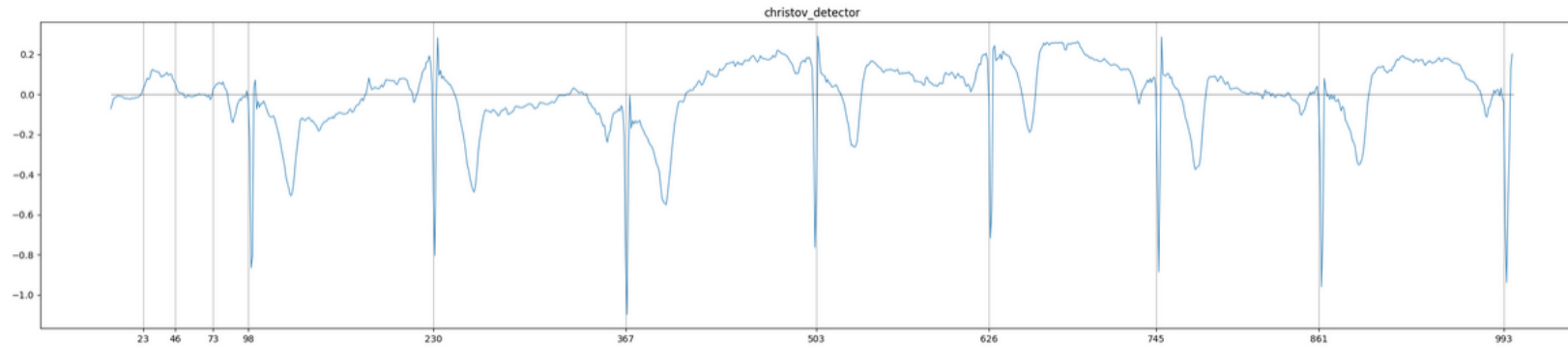
Records	Superclass	Description	کلاس
9514	NORM	Normal ECG	نرمال
5469	MI	Myocardial Infarction	سکته قلبی
5235	STTC	ST/T Change	تغییرات سگمنت ST
4898	CD	Conduction Disturbance	اختلال هادی
2649	HYP	Hypertrophy	بزرگ شدن عضله قلب



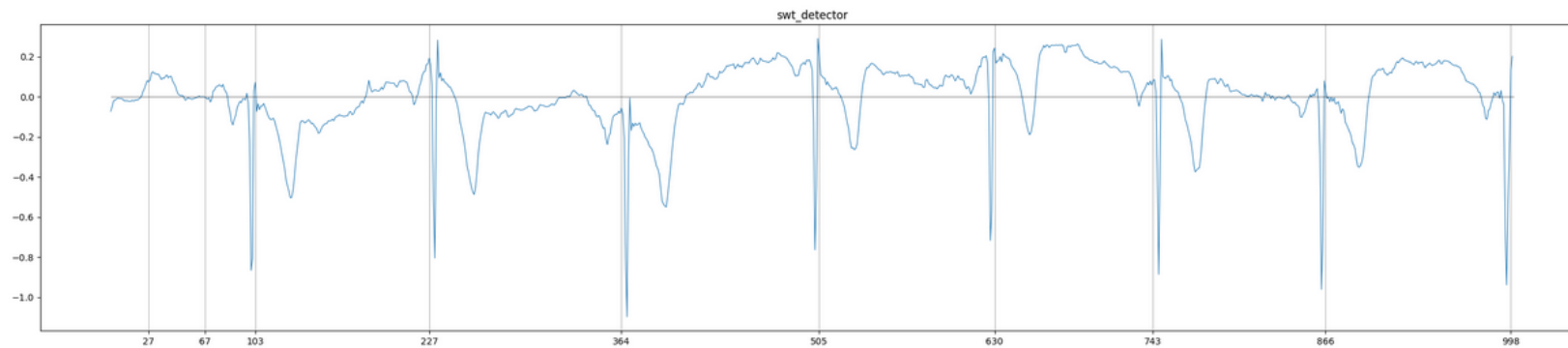
# محاسبه دوره تناوب دیتا

با استفاده از پکیج `ecgdetectors` با استفاده از دو الگوریتم `swt_detector` و `christov_detector` که با آزمون و خطا متوجه بهترین عملکرد از آن ها شدیم پیک های سیگنال را بدست آورده و میانگین آن را حساب می کنیم که جلوتر در قسمت `batch_size` از آن استفاده می کنیم:

```
Rpeaks: [ 23  46  73  98 230 367 503 626 745 861 993]
RR_intervals: [132, 137, 136, 123, 119, 116, 132]    ave: 127.85714285714286
Beeb per min: 76.71428571428572    freq: 1.2785714285714287
```



```
Rpeaks: [ 27  67 103 227 364 505 630 743 866 998]
RR_intervals: [124, 137, 141, 125, 113, 123, 132]    ave: 127.85714285714286
Beeb per min: 76.71428571428572    freq: 1.2785714285714287
```



روش شناسی

# معماری پایه LSTM

ساختار معماری شبکه پایه LSTM:

Layer (type)	Output Shape	Param #
lstm_5 (LSTM)	(None, 64)	19712
dense_8 (Dense)	(None, 5)	325
Total params: 20037 (78.27 KB)		
Trainable params: 20037 (78.27 KB)		
Non-trainable params: 0 (0.00 Byte)		

همانطور که مشاهده می کنید این مدل اصلا قابلیت یادگیری ویژگی ها را ندارد و از دقت بسیار پایینی برخوردار است:

```
Epoch 20/50
54/54 [=====] - 34s 631ms/step - loss: 1.9021 - accuracy: 0.4990 - val_loss: 2.0159 - val_accuracy: 0.4422
Epoch 21/50
54/54 [=====] - 34s 632ms/step - loss: 1.9024 - accuracy: 0.4993 - val_loss: 1.9980 - val_accuracy: 0.4422
Epoch 22/50
54/54 [=====] - 35s 646ms/step - loss: 1.9035 - accuracy: 0.4991 - val_loss: 1.9924 - val_accuracy: 0.4422
Epoch 23/50
54/54 [=====] - 39s 726ms/step - loss: 1.9020 - accuracy: 0.4991 - val_loss: 2.0084 - val_accuracy: 0.4422
Epoch 24/50
54/54 [=====] - 41s 766ms/step - loss: 1.9010 - accuracy: 0.4991 - val_loss: 2.0113 - val_accuracy: 0.4422
Epoch 25/50
54/54 [=====] - 37s 687ms/step - loss: 1.9011 - accuracy: 0.4990 - val_loss: 2.0090 - val_accuracy: 0.4422
Epoch 26/50
54/54 [=====] - 35s 647ms/step - loss: 1.9034 - accuracy: 0.4990 - val_loss: 2.0021 - val_accuracy: 0.4416
Epoch 27/50
54/54 [=====] - 39s 712ms/step - loss: 1.9017 - accuracy: 0.4991 - val_loss: 1.9909 - val_accuracy: 0.4416
Epoch 28/50
54/54 [=====] - 35s 656ms/step - loss: 1.9014 - accuracy: 0.4991 - val_loss: 2.0090 - val_accuracy: 0.4422
```

# افزایش پیچیدگی شبکه LSTM

ساختار معماری شبکه جدید LSTM:

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 64)	19712
dropout_4 (Dropout)	(None, 64)	0
batch_normalization_4 (Batch Normalization)	(None, 64)	256
dense_4 (Dense)	(None, 64)	4160
dropout_5 (Dropout)	(None, 64)	0
batch_normalization_5 (Batch Normalization)	(None, 64)	256
dense_5 (Dense)	(None, 5)	325

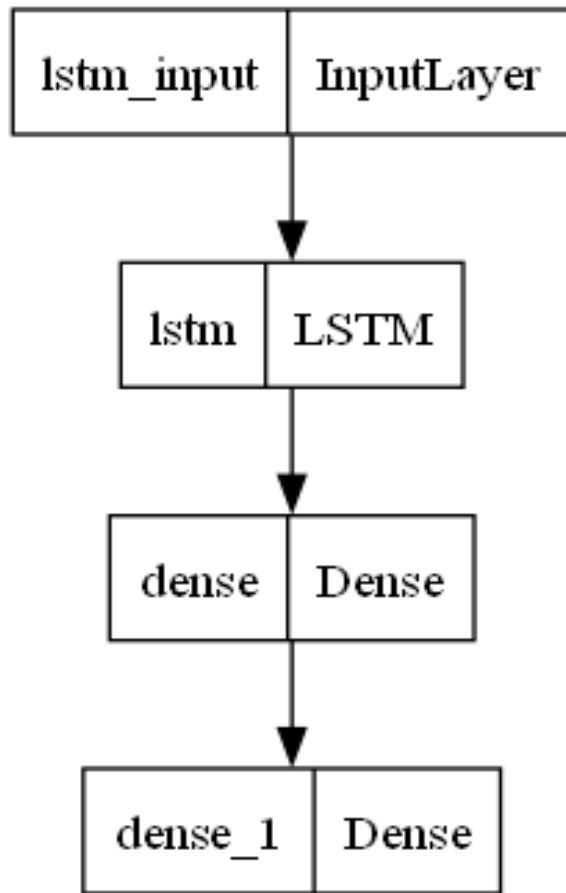
=====  
Total params: 24709 (96.52 KB)  
Trainable params: 24453 (95.52 KB)  
Non-trainable params: 256 (1.00 KB)

همانطور که مشاهده می کنید این مدل هم اصلا قابلیت یادگیری ویژگی ها را ندارد و از دقت بسیار پایینی برخوردار است:

```
54/54 [=====] - 28s 523ms/step - loss: 2.1061 - accuracy: 0.4416 - val_loss: 2.1261 - val_accuracy: 0.4416
Epoch 22/50
54/54 [=====] - 26s 477ms/step - loss: 2.1204 - accuracy: 0.4258 - val_loss: 2.1177 - val_accuracy: 0.4422
Epoch 23/50
54/54 [=====] - 34s 631ms/step - loss: 2.1270 - accuracy: 0.4362 - val_loss: 2.1108 - val_accuracy: 0.4439
Epoch 24/50
54/54 [=====] - 35s 646ms/step - loss: 2.1324 - accuracy: 0.4116 - val_loss: 2.1220 - val_accuracy: 0.4393
```

# کاهش رگولاریشن LSTM

ساختار معماری آخر شبکه LSTM را در شکل روبه رو میبینید:



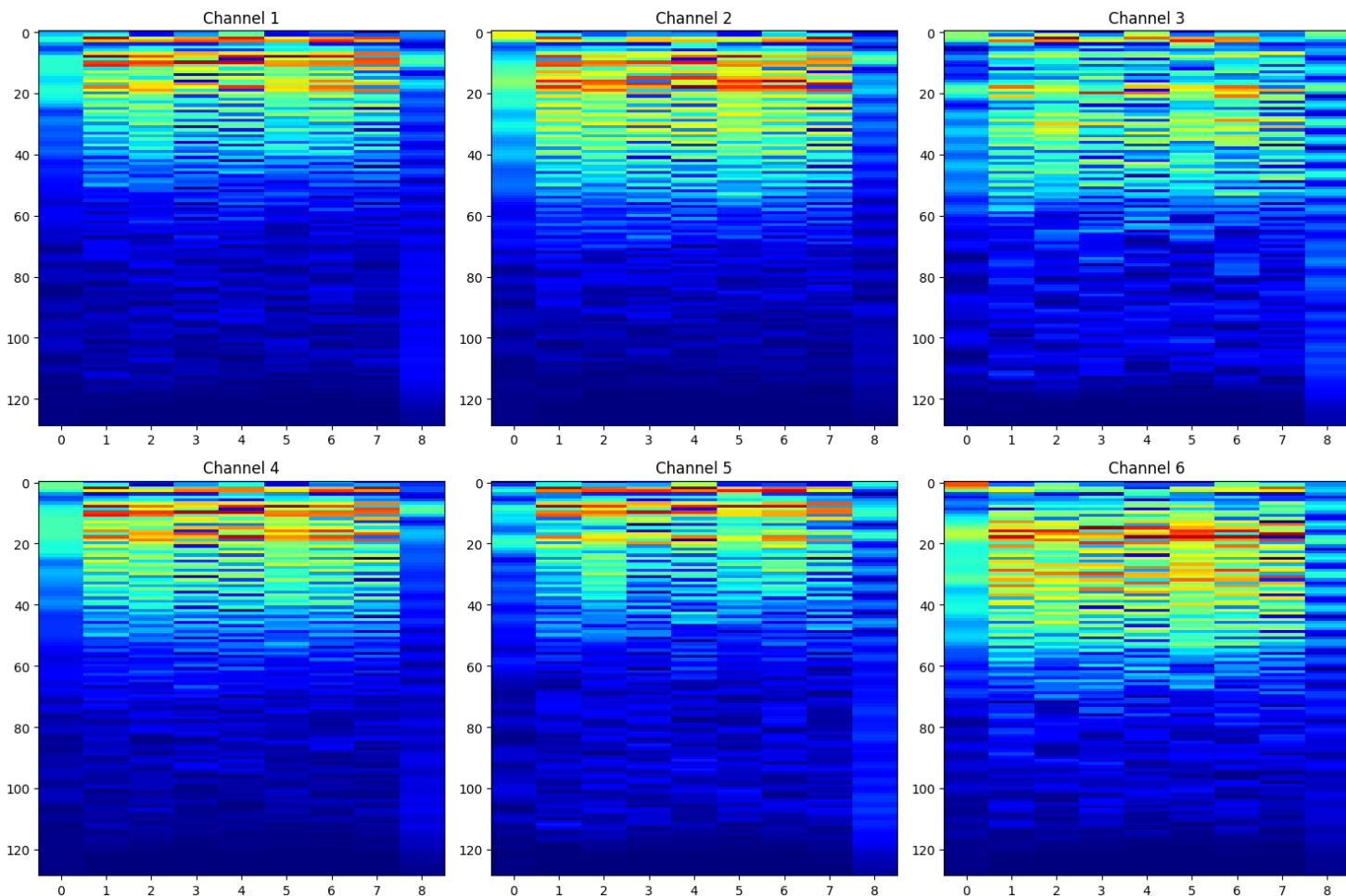
در اینجا مدل را ساده کردیم و با تغییر هایپرپارامتر های شبکه در بهینه ساز شبکه به مدلی با نتایج بهتر دست پیدا کردیم که نتیجه آن را در پایین میبینید که به دقت ۵۹ درصد رسیده است. هرچند این دقت کافی نیست و به دنبال معماری و متد بهتری می‌رویم.

```
Epoch 14/40
100/100 [=====] - 17s 171ms/step - loss: 0.5072 - accuracy: 0.5500 - val_loss: 0.5088 - val_accuracy: 0.5850
Epoch 15/40
100/100 [=====] - 17s 172ms/step - loss: 0.5097 - accuracy: 0.5506 - val_loss: 0.5035 - val_accuracy: 0.5925
Epoch 16/40
100/100 [=====] - 17s 171ms/step - loss: 0.5097 - accuracy: 0.5494 - val_loss: 0.5055 - val_accuracy: 0.5850
Epoch 17/40
100/100 [=====] - 17s 172ms/step - loss: 0.5071 - accuracy: 0.5487 - val_loss: 0.5055 - val_accuracy: 0.5925
Epoch 18/40
100/100 [=====] - 17s 172ms/step - loss: 0.5081 - accuracy: 0.5500 - val_loss: 0.4987 - val_accuracy: 0.5925
```

# پیاده سازی معماری های برپایه CNN در حوزه فرکانس

این روش از هر ۶ کلاس اول از ۱۲ کلاس داده هر سابجکت (که تعیین بیماری افراد نقش اساسی تری دارند) تبدیل شورت تایم فوریه یا ویولت میگیریم.

خروجی تبدیل شورت تایم فوریه روی ۶ کانال سابجکت اول:



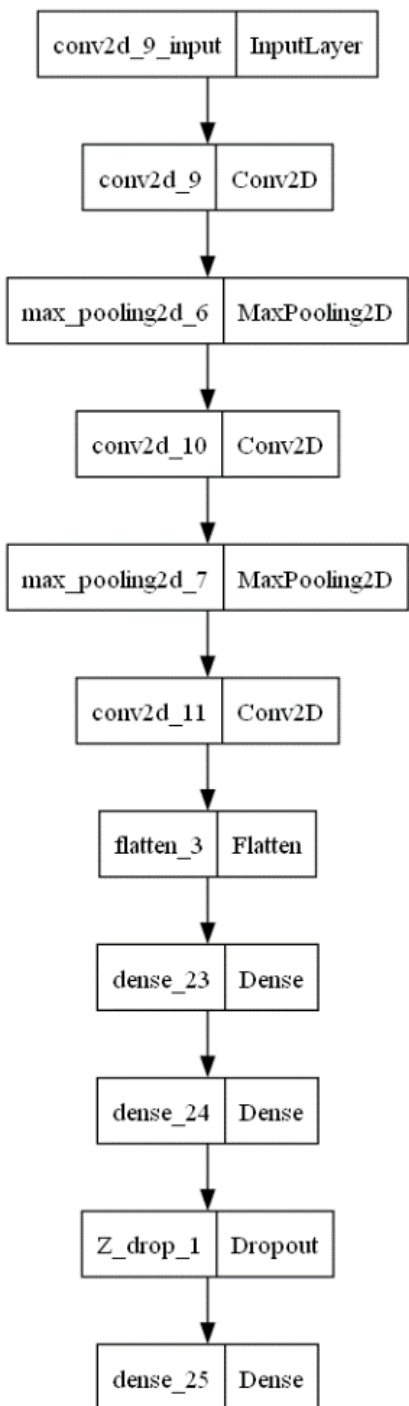
# پیاده سازی معماری های برپایه CNN در حوزه فرکانس

حال این ۶ تصویر را در بعد سوم به هم متصل کرده و به شبکه عصبی CNN ساده ای با یک لایه کاولوشن دادیم. در ابتدا مدل از افزایش گرادیان فاحش رنج می برد و با طول زمان دقت مدل به شدت کاهش پیدا می کرد.

```
Epoch 5/10
13/13 [=====] - 1s 61ms/step - loss: 52.0195 - accuracy: 0.3331 - val_loss: 78.2931 - val_accuracy: 0.4225
Epoch 6/10
13/13 [=====] - 1s 62ms/step - loss: 313.6123 - accuracy: 0.3388 - val_loss: 343.1458 - val_accuracy: 0.5500
Epoch 7/10
13/13 [=====] - 1s 61ms/step - loss: 1392.1661 - accuracy: 0.3100 - val_loss: 1641.2991 - val_accuracy: 0.5500
Epoch 8/10
13/13 [=====] - 1s 67ms/step - loss: 5442.3721 - accuracy: 0.3613 - val_loss: 4791.3901 - val_accuracy: 0.5500
Epoch 9/10
13/13 [=====] - 1s 60ms/step - loss: 19718.0391 - accuracy: 0.3000 - val_loss: 16921.5586 - val_accuracy: 0.5500
Epoch 10/10
13/13 [=====] - 1s 63ms/step - loss: 53637.8984 - accuracy: 0.3113 - val_loss: 35996.9414 - val_accuracy: 0.5500
13/13 [=====] - 0s 10ms/step - loss: 35996.9375 - accuracy: 0.5500
Validation loss: 35996.9375
Validation accuracy: 0.550000011920929
```

# پیاده سازی معماری های برپایه CNN در حوزه فرکانس

سپس با کاهش learning rate و افزایش رگولاریزیشن در ساختار شبکه روبه رو این مشکل حل شد. با تغییر هایپرپارامتر های شبکه در بهینه ساز شبکه به مدلی با نتایج بهتر دست پیدا کردیم که نتیجه آن را در پایین میبینید که به دقت ۵۹ درصد رسیده است. هرچند این دقت کافی نیست و به دنبال معماری و متد بهتری می رویم.



```
Epoch 14/40
100/100 [=====] - 17s 171ms/step - loss: 0.5072 - accuracy: 0.5500 - val_loss: 0.5088 - val_accuracy: 0.5850
Epoch 15/40
100/100 [=====] - 17s 172ms/step - loss: 0.5097 - accuracy: 0.5506 - val_loss: 0.5035 - val_accuracy: 0.5925
Epoch 16/40
100/100 [=====] - 17s 171ms/step - loss: 0.5097 - accuracy: 0.5494 - val_loss: 0.5055 - val_accuracy: 0.5850
Epoch 17/40
100/100 [=====] - 17s 172ms/step - loss: 0.5071 - accuracy: 0.5487 - val_loss: 0.5055 - val_accuracy: 0.5925
Epoch 18/40
100/100 [=====] - 17s 172ms/step - loss: 0.5081 - accuracy: 0.5500 - val_loss: 0.4987 - val_accuracy: 0.5925
```



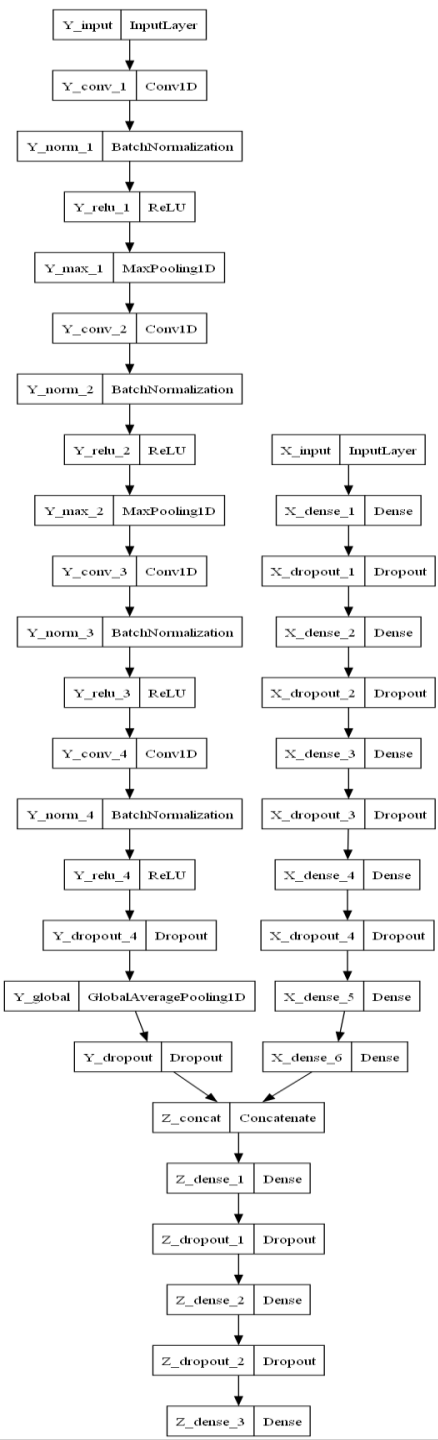
# CNN یک بعدی در حوزه زمانی (مدل موفق نهایی)

شاخه سمت راست اطلاع جدولی هر سابجکت را می‌گیرد و شاخه سمت چپ سیگنال قلبی فرد در ۱۲

کانال را بوسیله شبکه CNN می‌گیرد و بررسی می‌کند. این دوشاخه در نهایت به هم می‌رسند جایی که

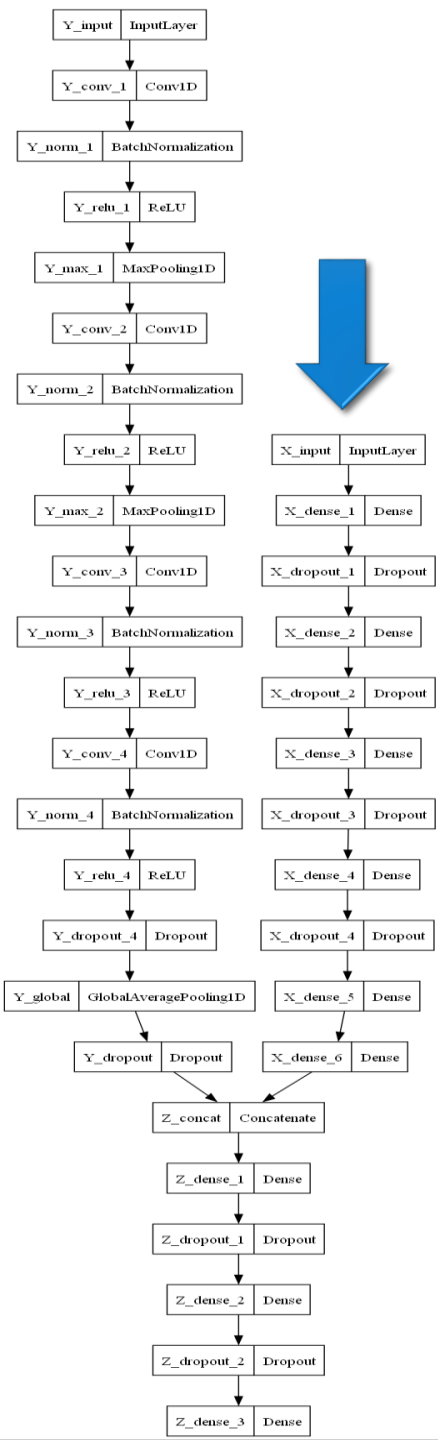
توسط سه لایه Fully connected تصمیم‌گیری نهایی خروجی مدل، هم براساس سیگنال که در شاخه

سمت راست پردازش شده و هم براساس دیتا جدولی که از شاخه سمت چپ بررسی شده، انجام می‌شود.



# CNN یک بعدی در حوزه زمانی

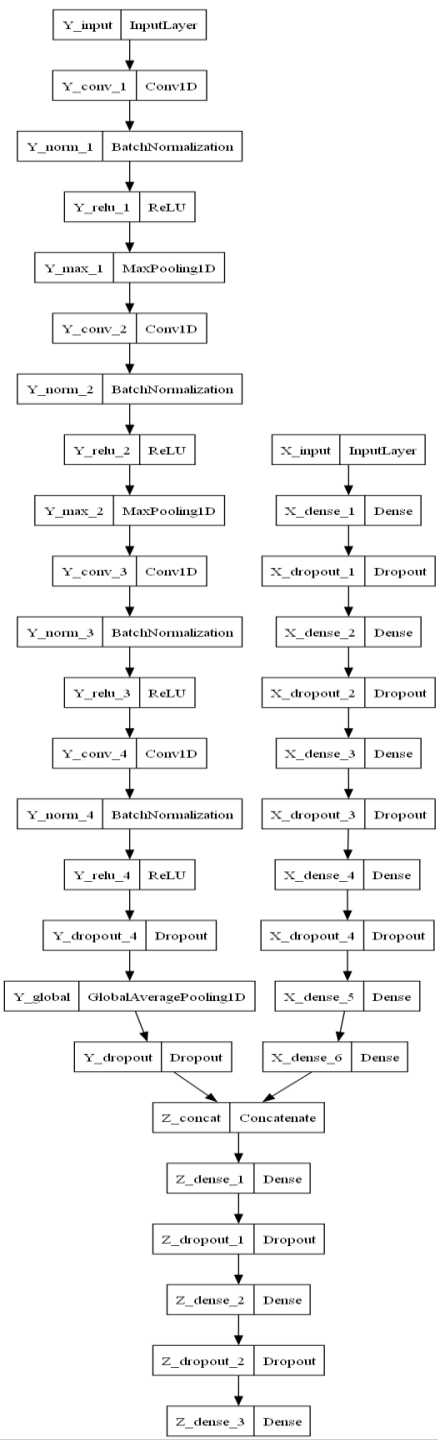
ابتدا شاخه سمت راست را آموزش می دهیم و همانطور که مشاهده می کنید دقت ۷۹.۵۷ برای این شاخه که با دیتای جدولی هر سابجکت کار می کند، بدست می آید:



Epoch 109/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4049 - binary\_accuracy: 0.8256 - precision: 0.7277 - recall: 0.4891 - val\_loss: 0.4895 - val\_binary\_accuracy: 0.7905 - val\_precision: 0.6436 - val\_recall: 0.4156  
Epoch 110/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4071 - binary\_accuracy: 0.8242 - precision: 0.7219 - recall: 0.4889 - val\_loss: 0.4862 - val\_binary\_accuracy: 0.7937 - val\_precision: 0.6585 - val\_recall: 0.4118  
Epoch 111/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4054 - binary\_accuracy: 0.8248 - precision: 0.7235 - recall: 0.4904 - val\_loss: 0.4872 - val\_binary\_accuracy: 0.7944 - val\_precision: 0.6612 - val\_recall: 0.4118  
Epoch 112/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4060 - binary\_accuracy: 0.8247 - precision: 0.7300 - recall: 0.4805 - val\_loss: 0.4877 - val\_binary\_accuracy: 0.7944 - val\_precision: 0.6612 - val\_recall: 0.4118  
Epoch 113/120  
236/236 [=====] - 1s 2ms/step - loss: 0.4051 - binary\_accuracy: 0.8257 - precision: 0.7254 - recall: 0.4936 - val\_loss: 0.4838 - val\_binary\_accuracy: 0.7944 - val\_precision: 0.6612 - val\_recall: 0.4118  
Epoch 114/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4056 - binary\_accuracy: 0.8238 - precision: 0.7212 - recall: 0.4872 - val\_loss: 0.4859 - val\_binary\_accuracy: 0.7957 - val\_precision: 0.6633 - val\_recall: 0.4182  
Epoch 115/120  
236/236 [=====] - 1s 2ms/step - loss: 0.4032 - binary\_accuracy: 0.8249 - precision: 0.7299 - recall: 0.4815 - val\_loss: 0.4905 - val\_binary\_accuracy: 0.7957 - val\_precision: 0.6646 - val\_recall: 0.4156  
Epoch 116/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4051 - binary\_accuracy: 0.8240 - precision: 0.7191 - recall: 0.4919 - val\_loss: 0.4840 - val\_binary\_accuracy: 0.7944 - val\_precision: 0.6718 - val\_recall: 0.3926  
Epoch 117/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4052 - binary\_accuracy: 0.8253 - precision: 0.7329 - recall: 0.4796 - val\_loss: 0.4915 - val\_binary\_accuracy: 0.7918 - val\_precision: 0.6475 - val\_recall: 0.4182  
Epoch 118/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4027 - binary\_accuracy: 0.8254 - precision: 0.7299 - recall: 0.4849 - val\_loss: 0.4901 - val\_binary\_accuracy: 0.7911 - val\_precision: 0.6455 - val\_recall: 0.4169  
Epoch 119/120  
236/236 [=====] - 1s 3ms/step - loss: 0.4050 - binary\_accuracy: 0.8242 - precision: 0.7281 - recall: 0.4798 - val\_loss: 0.4894 - val\_binary\_accuracy: 0.7905 - val\_precision: 0.6441 - val\_recall: 0.4143

# CNN یک بعدی در حوزه زمانی

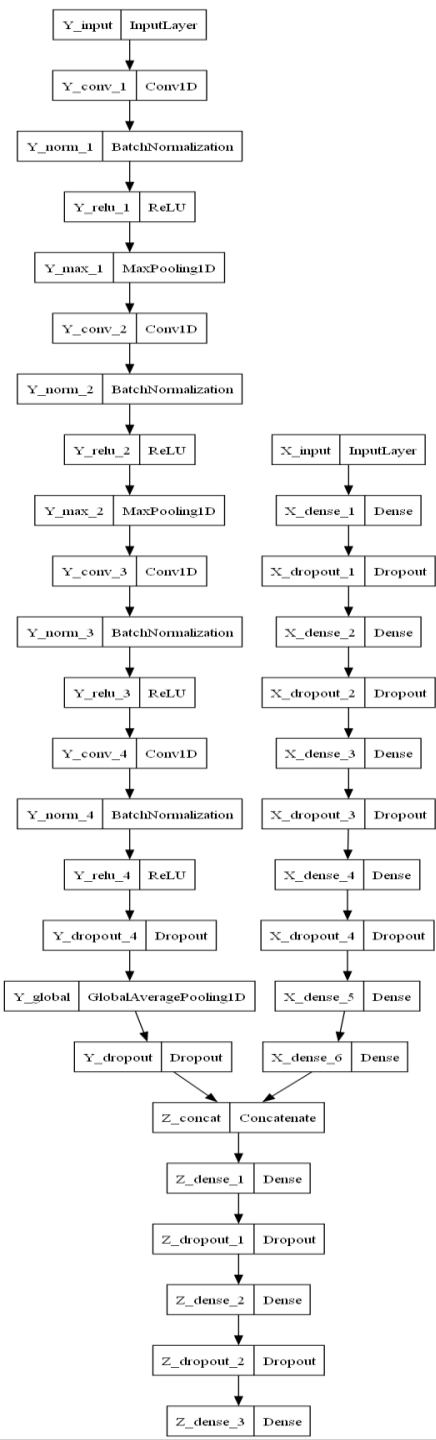
در آخر، مدل کلی را فیت کردیم که ساختار شبکه آن را در عکس روبه رو و پارامترهای مهم شبکه را در زیر مشاهده می کنید:



```
1 adam = optimizers.legacy.Adam(learning_rate=0.01, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.001)
2 modelZ.compile( loss= 'binary_crossentropy', optimizer= adam, metrics=['binary_accuracy', 'Precision', 'Recall'])
3
4 callback = [
5     callbacks.ModelCheckpoint( monitor= 'val_binary_accuracy', save_best_only= True, filepath= root_path+'model'),
6     callbacks.EarlyStopping( monitor= 'val_binary_accuracy', restore_best_weights= True, patience=10, min_delta= 0.001),
7     callbacks.TensorBoard(log_dir= root_path+'/logs')
8 ]
9
10 fitting = modelZ.fit( [X_train_scaled, Y_train_scaled], Z_train, epochs=20, batch_size=128,
11                       callbacks= callback, validation_data=([X_val_scaled, Y_val_scaled] , Z_val))
```

# CNN یک بعدی در حوزه زمانی

در اینجا فرآیند آموزش کل شبکه عصبی را مشاهده می کنید که مدل بر داده تست به دقت ۷۹.۲۴ درصد رسیده است:



```
Epoch 10/20
30/30 [=====] - ETA: 0s - loss: 0.4460 - binary_accuracy: 0.8102 - precision: 0.6694 - recall: 0.4843
INFO:tensorflow:Assets written to: ./ptb-xl-dataset-1.0.3\model\assets
```

```
INFO:tensorflow:Assets written to: ./ptb-xl-dataset-1.0.3\model\assets
```

```
30/30 [=====] - 8s 267ms/step - loss: 0.4460 - binary_accuracy: 0.8102 - precision: 0.6694 - recall: 0.4843 - val_loss: 0.4843 - val_binary_accuracy: 0.7911 - val_precision: 0.6554 - val_recall: 0.3964
```

```
Epoch 11/20
30/30 [=====] - 5s 181ms/step - loss: 0.4438 - binary_accuracy: 0.8120 - precision: 0.6783 - recall: 0.4794 - val_loss: 0.4867 - val_binary_accuracy: 0.7888 - val_precision: 0.6417 - val_recall: 0.4054
```

```
Epoch 12/20
30/30 [=====] - 6s 183ms/step - loss: 0.4429 - binary_accuracy: 0.8118 - precision: 0.6774 - recall: 0.4801 - val_loss: 0.4941 - val_binary_accuracy: 0.7832 - val_precision: 0.6194 - val_recall: 0.4079
```

```
Epoch 13/20
30/30 [=====] - 5s 180ms/step - loss: 0.4467 - binary_accuracy: 0.8120 - precision: 0.6804 - recall: 0.4754 - val_loss: 0.4838 - val_binary_accuracy: 0.7901 - val_precision: 0.6545 - val_recall: 0.3900
```

```
Epoch 14/20
30/30 [=====] - 5s 182ms/step - loss: 0.4411 - binary_accuracy: 0.8125 - precision: 0.6779 - recall: 0.4839 - val_loss: 0.4892 - val_binary_accuracy: 0.7878 - val_precision: 0.6367 - val_recall: 0.4079
```

```
Epoch 15/20
30/30 [=====] - 5s 180ms/step - loss: 0.4428 - binary_accuracy: 0.8133 - precision: 0.6860 - recall: 0.4746 - val_loss: 0.4922 - val_binary_accuracy: 0.7773 - val_precision: 0.5949 - val_recall: 0.4207
```

```
Epoch 16/20
30/30 [=====] - 5s 180ms/step - loss: 0.4443 - binary_accuracy: 0.8109 - precision: 0.6724 - recall: 0.4832 - val_loss: 0.4890 - val_binary_accuracy: 0.7832 - val_precision: 0.6194 - val_recall: 0.4079
```

```
Epoch 17/20
30/30 [=====] - ETA: 0s - loss: 0.4397 - binary_accuracy: 0.8127 - precision: 0.6860 - recall: 0.4699
INFO:tensorflow:Assets written to: ./ptb-xl-dataset-1.0.3\model\assets
```

```
INFO:tensorflow:Assets written to: ./ptb-xl-dataset-1.0.3\model\assets
```

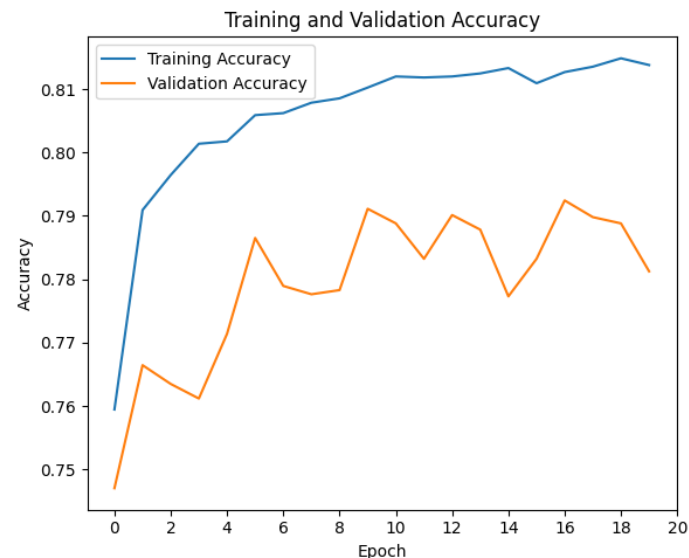
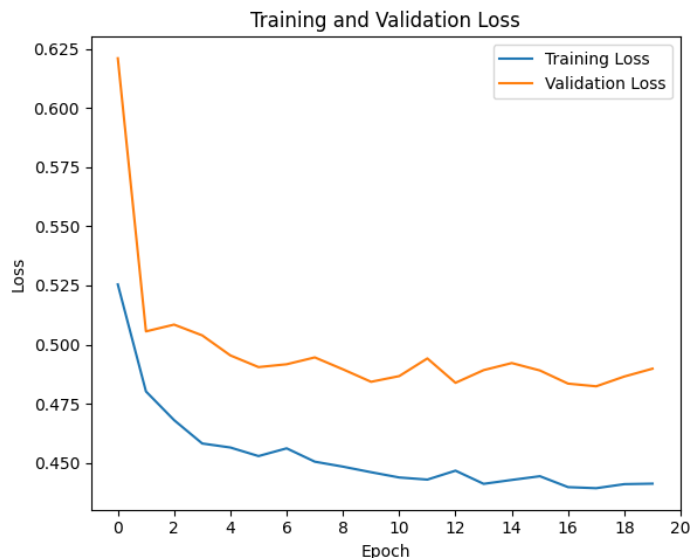
```
30/30 [=====] - 8s 277ms/step - loss: 0.4397 - binary_accuracy: 0.8127 - precision: 0.6860 - recall: 0.4699 - val_loss: 0.4835 - val_binary_accuracy: 0.7924 - val_precision: 0.6736 - val_recall: 0.3747
```

```
Epoch 18/20
30/30 [=====] - 5s 181ms/step - loss: 0.4393 - binary_accuracy: 0.8135 - precision: 0.6936 - recall: 0.4625 - val_loss: 0.4823 - val_binary_accuracy: 0.7898 - val_precision: 0.6462 - val_recall: 0.4041
```

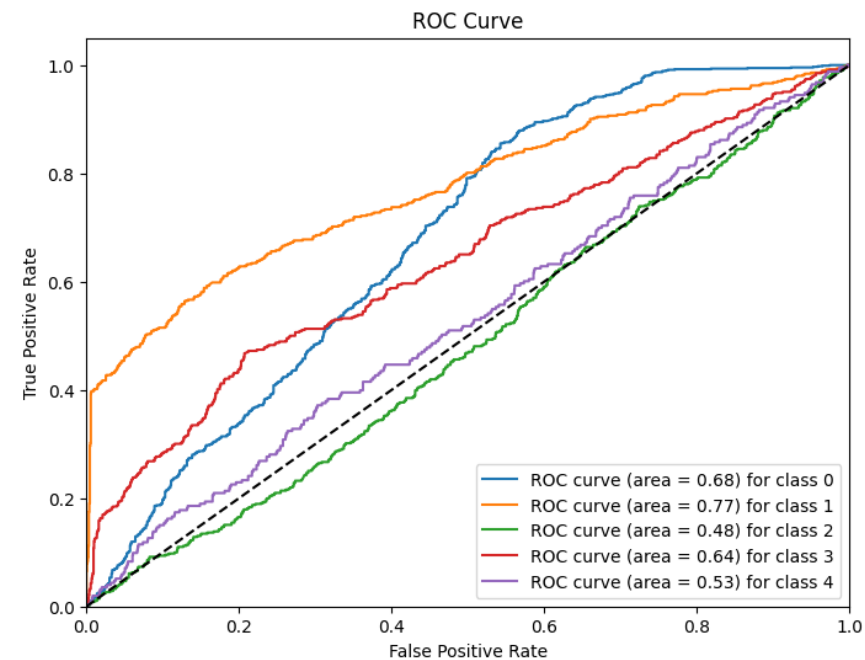
```
Epoch 19/20
30/30 [=====] - 5s 179ms/step - loss: 0.4410 - binary_accuracy: 0.8149 - precision: 0.6944 - recall: 0.4706 - val_loss: 0.4865 - val_binary_accuracy: 0.7888 - val_precision: 0.6452 - val_recall: 0.3977
```

# CNN یک بعدی در حوزه زمانی

در اینجا میتوانید نمودار دقت و هزینه مدل در طی روند آموزش، هم بر داده آموزش و هم بر داده تست را مشاهده کنید:



همچنین در شکل زیر نمودار ROC در کلاس های مختلف را مشاهده می کنید:



# جمع بندی و نتیجه گیری

# تحقیقات بیشتر در آینده

- بهبود نتایج شبکه‌های عصبی عمیق
- یادگیری ماشین تقویت شده
- ترکیب داده‌های چند منبع
- استفاده از شبکه‌های مختلط
- حل مسائل امنیتی و حریم خصوصی

# مقایسه نتایج و نتیجه‌گیری نهایی

- در این پژوهش به بررسی مقدمه و مفاهیم پردازش سیگنال ECG و سپس دیتاست و مدل سازی شبکه های عصبی با معماری های مختلف پرداختیم و با تغییر معماری مدل ها و هایپرپارامترهای شبکه به بهترین مدل خود با معماری CNN یک بعدی رسیدیم.
- همانطور که مشاهده کردیم با افزایش پیچیدگی مدل ها یا افزایش regularization مدل ها لزوما نمی‌توان به مدل بهتری دست پیدا کرد و با توجه به مشکل بوجود آمده در خروجی یک مدل باید از این روش ها استفاده کرد تا به دقت بالاتر دست یافت.
- با تست و بررسی مدل هایی بر پایه شبکه های LSTM و CNN بر روی دیتاست PTB-XL و مقایسه دقت نهایی این مدل ها به مدلی بر پایه CNN یک بعدی، با دقت مناسب دست پیدا کردیم که با دقت ۷۹.۲۴ درصد به پیش‌بینی بیماری ساجکت در ۵ کلاس می‌پردازد.



# مقایسه نتایج و نتیجه‌گیری نهایی

- با توجه به بررسی‌های انجام شده در قسمت پیشینه پژوهش که بهترین دقت مقاله اول در بین هر سه مدل بررسی شده‌ی آنها ۷۶ درصد و بهترین دقت مقاله دوم در بین همه مدل‌های بررسی شده‌ی آنها ۷۴.۹-۷۹.۱ درصد بود، دقت بدست آمده در این پژوهش (۷۹.۲۴) با اختلاف اندکی بهتر از این مدل‌ها عمل می‌کند و به بهترین دقت مقاله سوم در بین همه مدل‌های بررسی شده‌ی آنها که ۷۷.۹-۸۰.۲ درصد بود، بسیار نزدیک است در میان بازه آن قرار دارد، در حالی که با روشی کاملاً متفاوت نسبت به این مقاله بر پایه شبکه‌های عصبی به این دقت رسیده است.
- در نتیجه می‌توان گفت مدل‌های شبکه عصبی بخصوص CNN ها می‌توانند به دقت قابل قبولی برای تشخیص و کلاس بندی ها از سیگنال نوار قلب دست پیدا کنند. در آخر امیدواریم که نتایج حاصله بتواند به بهبود دقت و صحت روش های تشخیص بیماری‌های قلبی کمک نماید.

**با سپاس از توجه شما**