

باسمه تعالی



دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس هوش محاسباتی

فاز دوم پروژه

علی محرابیان 96102331

استاد: دکتر رضایی

بهار 1399



بخش اول: مقدمه

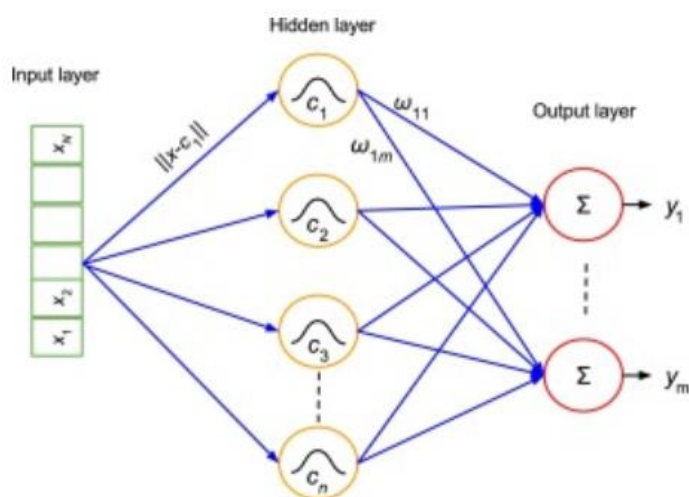
در فاز قبلی به کمک سیستم های فازی، تا حدودی توانستیم 8 نوع بیماری قلبی را به کمک سیگنال های ECG طبقه بندی کنیم. در این فاز قصد داریم که این کار را به کمک شبکه عصبی های مختلف انجام دهیم. در این مرحله تماماً از دست آوردهای فاز قبل بهره گرفتیم که نحوه حصول آن ها به تفصیل در فاز قبل شرح داده شد. در این فاز هم از روش one hot-encoding برای طبقه بندی کلاس های خروجی استفاده کردیم. تنها کار جدیدی که در این فاز تا قبل از مرحله آموزش داده ها انجام دادیم، استفاده از سن افراد به عنوان یک ویژگی جدید و کارآمد و اضافه کردن آن به ماتریس ویژگی ها است.

بخش دوم: معرفی شبکه های عصبی و طبقه بندی و اعتبارسنجی داده ها

شبکه های عصبی از یک دیدگاه به دو دسته shallow neural net و deep neural net تقسیم بندی می شوند. شبکه های عصبی کم عمق به طور کلی به شبکه هایی اطلاق می شود که hidden layer های کمی داشته باشند. این شبکه برای یادگیری داده هایی که حجم کمی دارند، مناسب بوده و بهتر به داده ها fit می شوند. شبکه های عصبی عمیق هم شبکه هایی هستند که hidden layer بیشتری داشته و با پارامترهای زیادی که دارند، امکان یادگیری موارد پیچیده تر را در اختیار ما می گذارند.

(1)

اولین شبکه که برای آموزش داده ها استفاده می کنیم، شبکه عصبی RBF است. شبکه RBF، یک شبکه عصبی با 3 لایه که دارای یک hidden layer است، می باشد. تفاوت اصلی این شبکه با دیگر شبکه ها، این است که از تابع فعالسازی radial basis نظیر تابع گاوسی استفاده می کند. خروجی، ترکیب خطی این توابع را با وزن های مختلف برای تخمین تابع نهایی استفاده می کند. از مزیت های آن می توان به سرعت سریع آن در آموزش اشاره کرد.



در این قسمت از تابع newgrnn برای آموزش داده ها استفاده کردیم. از 40 ویژگی بهینه اول که در فاز قبل نحوه رسیدن به آن ها را بیان کردیم، برای آموزش استفاده می کنیم. یکی از پارامترهای ورودی این تابع متغیر spread است که میزان نرم بودن تابع تخمینی خروجی را مشخص می کند. با سعی و خطا های مختلف، در نهایت مقدار 1 را برای این متغیر اتخاذ کردیم. برای مشخص کردن کلاس خروجی، مشابه با فاز قبل، از یک مقدار Threshold برای طبقه بندی خروجی استفاده کردیم. خروجی های کمتر از 0.3 را برابر با 0 و خروجی های بیشتر از 0.3 را برابر با 1 در نظر می گیریم. لازم به ذکر است که از 5 fold cross validation برای آموزش 85% داده ها استفاده کردیم. 15% داده ها نیز در نهایت به عنوان تست برای اعتبارسنجی داده ها استفاده شدند. خروجی نتایج برای داده های تست به صورت زیر است.



Accuracy Disease	1st try	2st try	3st try	4st try	5st try	Average	Best
Normal	81.25	76.25	88.75	77.5	83.75	81.5	88.75
AF	77.5	77.5	77.5	76.25	75	75.55	77.5
I-AVB	83.75	90	83.75	83.75	86.25	85.5	90
LBBB	97.5	100	97.5	96.25	98.75	98	100
RBBB	78.75	75	73.75	75	77.5	76	78.75
PAC	82.5	87.5	82.5	88.75	78.75	84	88.75
PVC	83.75	86.25	83.75	88.75	87.5	86	88.75
STD	78.75	92.5	87.5	87.5	70	83.25	92.5
STE	93.75	96.25	96.25	95	96.25	95.5	96.25

همان طور که می بینیم، به کمک شبکه های عصبی RBF توانستیم خروجی مطلوبی بگیریم. همان طور که به یاد داریم در فاز قبلی، با سعی تلاش های زیاد و مواجه شدن با چالش های بسیار، به درصد صحت بهتری بیش از 45% نتوانستیم برسیم. می بینیم که در این قسمت به کمک یک شبکه عصبی کم عمق ساده، خروجی های بسیار مطلوب تری نسبت به فاز قبل داریم. حال در هر قسمت بهترین شبکه را انتخاب کرده و از نتایج آن برای رسم Confusion matrix استفاده می کنیم.



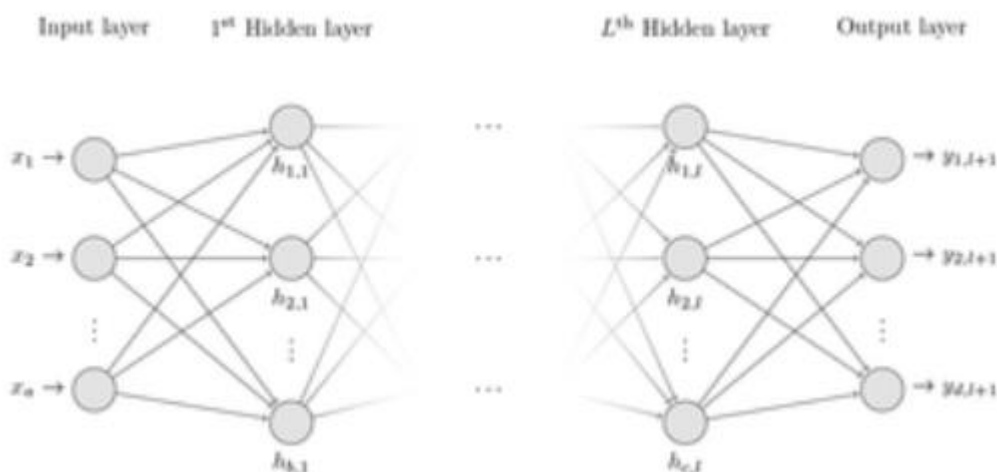
Confusion Matrix										
Output Class	1	2	3	4	5	6	7	8	9	
	7 8.8%	0 0.0%	1 1.3%	0 0.0%	8 10.0%	0 0.0%	2 2.5%	1 1.3%	1 1.3%	35.0% 65.0%
	0 0.0%	12 15.0%	0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	92.3% 7.7%
	0 0.0%	0 0.0%	3 3.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	10 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 8.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 5.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 17.5%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 10.0%	100% 0.0%
Target Class										

می توان دید که برای حالت نرمال، نسبت کسانی که واقعا بیمار بوده اند به کل کسانی که بیمار تشخیص داده شده اند، کم است. برای بیماری RBBB هم، درصد صحت بین لیبیل هایی که درست تشخیص داده شده اند، کم است. در این قسمت به درصد صحت کلی 82% رسیدیم.



(2)

دومین شبکه ای که برای آموزش داده ها استفاده می کنیم، شبکه عصبی MLP است. شبکه پرسپترون چند لایه، از جمله شبکه های ساده است که برای طبقه بندی و آموزش داده ها استفاده می شود. می توان از تابع فعالسازی های گوناگون و همچنین hidden layer با سائزهای گوناگون بهره برد. یکی از چالش های موجود در این شبکه ، سائز hidden layer است. اگر این مقدار زیاد باشد، می توان مدل های پیچیده تری را یادگیری کرد ولی از طرف دیگر، امکان overfitting با توجه به حفظ کردن الگوی ورودی برای شبکه ایجاد می شود. کم بودن این مقدار نیز باعث عدم توانایی یادگیری مناسب می شود.



در این قسمت از 70 ویژگی موثر اول برای آموزش شبکه با سائز 12 استفاده می شود. 85% داده ها برای آموزش و 15% برای تست استفاده می شود. از threshold برابر با 0.2 برای طبقه بندی کلاس های خروجی استفاده کردیم. نکته دیگری که لازم به ذکر است این است که هر شبکه را 3 بار پیاپی آموزش می دهیم و از آن برای اعتبار سنجی خروجی استفاده می شود. 5 fold cross validation را نیز در این قسمت به کار بردیم.

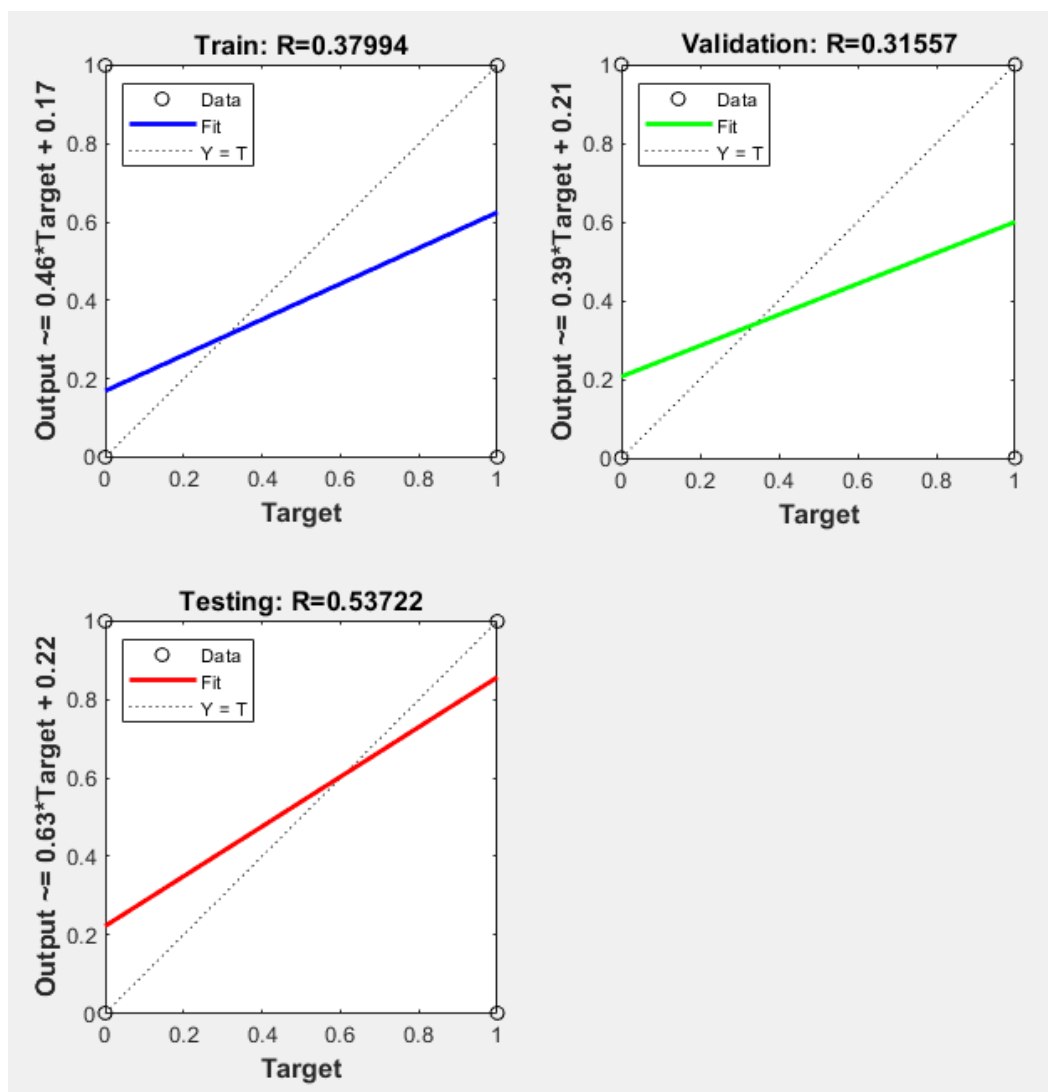


Accuracy Disease	1st try	2st try	3st try	4st try	5st try	Average	Best
Normal	82.5	76.25	85	78.75	81.25	80.75	85
AF	76.25	85	82.5	75	80	75.55	85
I-AVB	80	85	82.5	75	85	81.5	85
LBBB	100	97.5	98.75	100	100	99.25	100
RBBB	83.75	70	85	76.25	88.75	80.75	88.75
PAC	87.5	86.25	88.75	85	93.75	88.25	93.75
PVC	88.75	87.5	86.25	87.5	93.75	88.75	93.75
STD	77.5	77.5	81.25	87.5	77.5	80.25	81.25
STE	93.75	92.5	95	91.25	96.25	93.75	96.25

همان طور که ملاحظه می شود، در این قسمت اندکی درصد صحت ها نسبت به حالت قبل کاهش می یابد. این مورد می تواند با توجه به این که ساختار دو شبکه متفاوت است، دلایل متعددی داشته باشد. در این قسمت هم بهترین شبکه را برای هر کلاس جداگانه انتخاب می کنیم. در اینجا می خواهیم از اطلاعاتی که هنگام آموزش شبکه های عصبی ذخیره کردیم، استفاده کنیم. در ادامه، نمودارهای مختلفی را رسم کرده و درباره اطلاعاتی که آن ها در اختیار ما می گذارند، توضیح می دهیم.



نمودارهای بالا، performance، های شبکه های آموزش دیده را نمایش می دهد. در مواردی که نمودار test به مقدار قابل توجهی نسبت به validation، افزایش یافته و فاصله گرفته، احتمال overfitting داریم. در مواردی که این پدیده رخ داده، شبکه ها را دوباره با تعداد epoch ها و learning rate های مختلف آموزش می دهیم.



منحنی رگرسیون بالا از کلاس اول به طور نمونه استخراج شده است. هرچه خط حاصله به شیب 45 درجه در شکل نزدیک تر باشد، خروجی های ما به خروجی های اصلی نزدیک تر بوده اند. در مواردی که این خطوط انحراف زیادی از حد مطلوب داشتند، با تغییر پارامترهای موجود، شبکه ها را دوباره آموزش دادیم. برای جلوگیری از شلوغ شدن، به رسم همین یک مورد بسنده می کنیم. لازم به ذکر است که خلاصه آموزش تمامی شبکه در فایل ارسال شده ذخیره شده است.



در نهایت، confusion matrix به صورت زیر است.

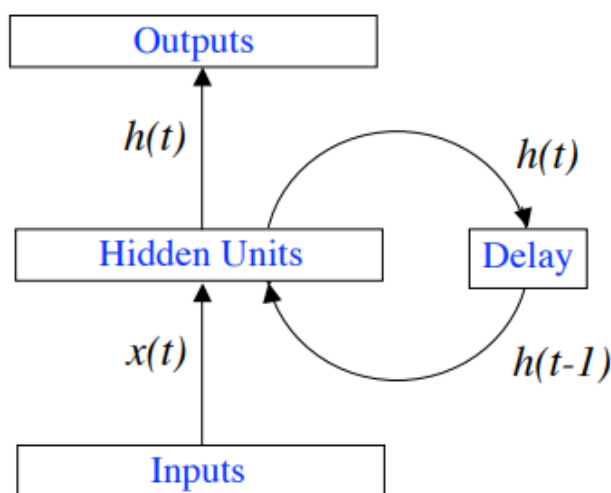
Confusion Matrix										
Output Class	1	2	3	4	5	6	7	8	9	
	2 2.5%	0 0.0%	1 1.3%	2 2.5%	3 3.8%	0 0.0%	2 2.5%	3 3.8%	6 7.5%	10.5% 89.5%
	0 0.0%	4 5.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	80.0% 20.0%
	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	50.0% 50.0%
	0 0.0%	0 0.0%	0 0.0%	4 5.0%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
	0 0.0%	1 1.3%	1 1.3%	0 0.0%	22 27.5%	0 0.0%	2 2.5%	0 0.0%	0 0.0%	84.6% 15.4%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 3.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	2 2.5%	0 0.0%	0 0.0%	66.7% 33.3%
	0 0.0%	2 2.5%	0 0.0%	0 0.0%	2 2.5%	0 0.0%	1 1.3%	5 6.3%	0 0.0%	50.0% 50.0%
	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.5%	85.7% 14.3%
										Target Class
										1 2 3 4 5 6 7 8 9

همان طور که حدس زدیم، در این حالت درصد صحت خروجی نسبت به حالت قبل کمتر شد. در این حالت به درصد 62% رسیدیم.



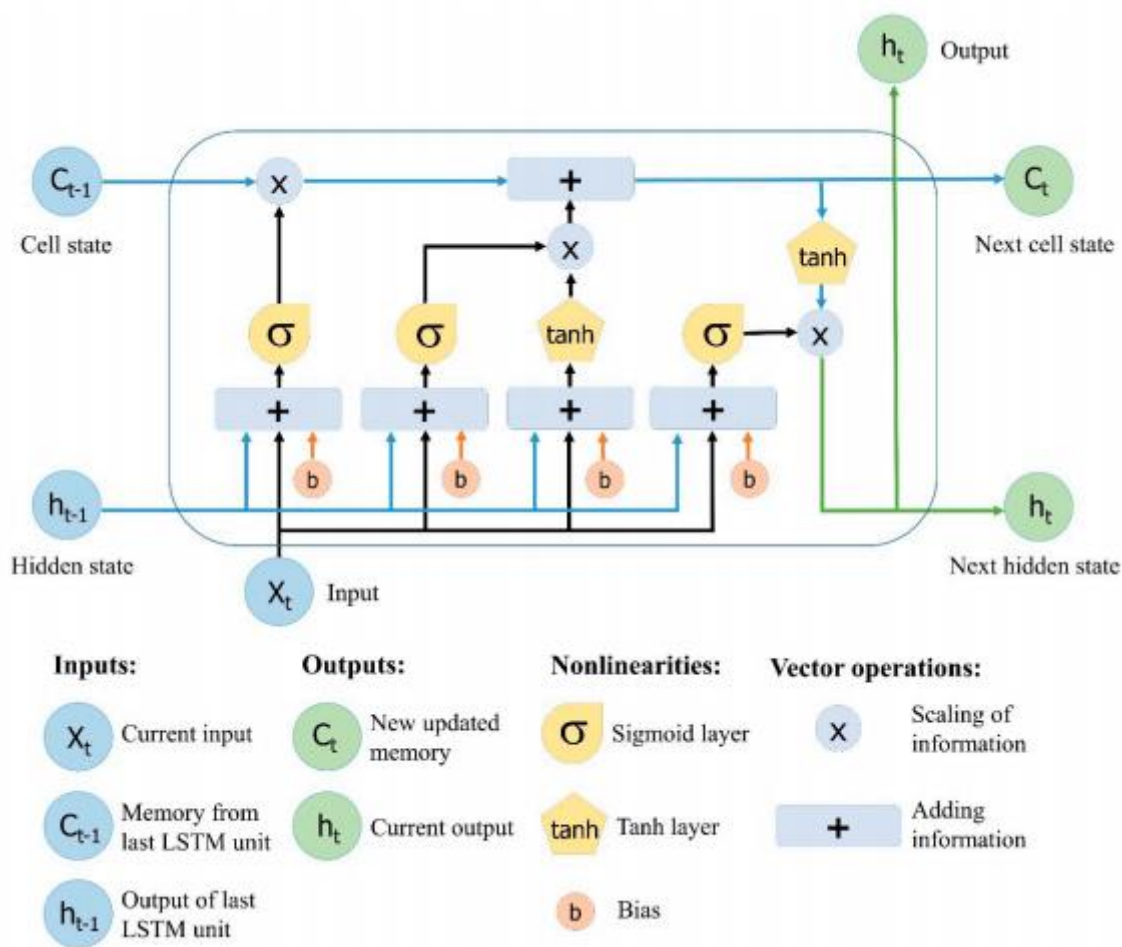
(3)

یکی دیگر از طبقه بندی هایی که برای شبکه های عصبی وجود دارد، شبکه های عصبی Feed forward و Recurrent هستند. در شبکه های Feed forward، ورودی ها به طور مستقیم از هر لایه به لایه بعدی منتقل می شوند و ورودی های هر لایه، خروجی های لایه قبلی هستند. در این شبکه، هیچ حلقه فیدبکی یا دوری وجود ندارد. شبکه هایی که تا به این جا بررسی کردیم، از این نوع بودند. در شبکه های Recurrent که می توانند انواع بسیار مختلفی داشته باشند، ورودی های یک لایه، لزوماً از لایه قبلی نیستند و می توانند از طریق یک حلقه فیدبکی از همان لایه یا حتی لایه های جلوتر باشند. پس به نوعی این شبکه ها دارای حافظه می باشند.



همان طور که ملاحظه می شود، خروجی های یک لایه دوباره می توانند به عنوان ورودی استفاده شوند. از این شبکه ها بیشتر در یادگیری رشته داده ها و پردازش گفتار استفاده می شود.

یکی از شبکه های بازگشتی معروف که برای طبقه بندی داده ها استفاده می شود، شبکه های LSTM هستند. این شبکه یک ساختار زنجیروار تکرار شونده دارد که به شکل زیر است.



شبکه LSTM از بلاک های حافظه ای به نام سلول تشکیل می شود. در هر بلاک طبق شکل مقادیر cell state و Hidden state به بلاک بعدی منتقل می شود.



در این شبکه از gate ها برای اضافه یا حذف کردن داده ها استفاده می شود. در مرحله اول به کمک تابع سیگموید مقادیری که از خروجی قبلی زاید بوده را حذف می کنیم.

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

در مرحله بعدی، تصمیم گیری برای ذخیره کردن ورودی انجام می پذیرد. در قسمت اول به کمک تابع سیگموید، اطلاعات جدید ذخیره شده یا در نظر گرفته نمی شوند. در قسمت بعد به کمک تابع \tanh ، به مقادیر ورودی وزن مشخصی با توجه به اهمیت آن ها، بین $(-1, 1)$ ، داده می شود.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$N_t = \tanh(w_n[h_{t-1}, x_t] + b_n)$$

$$C_t = C_{t-1}f_t + N_ti_t$$

در نتیجه مقدار cell state بروز رسانی می شود. در نهایت مقدار hidden state خروجی نیز، ضربی از مقدار فیلتر شده cell state خواهد بود.

$$O_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$$h_t = O_t \tanh(C_t)$$



در این قسمت از دو شیوه مختلف برای رسیدن خروجی استفاده کردیم. اول از همان راه سابق و طبقه بندی داده ها استفاده کردیم. در این شیوه برای این قسمت یک مشکل اساسی وجود دارد. مشکل این است که چون شبکه حافظه دار است و در کلاس جداگانه، برچسب های 0 زیادی وجود دارد، پس ویژگی های این برچسب، بیشتر در شبکه تاثیر می گذارد و توانایی تشخیص بیماران که برچسب 1 دارند را ندارد. به همین جهت، برای حل این مشکل، از شیوه دیگری استفاده کردیم. در ابتدا در هر قسمت جداگانه، برچسب های 1 را پیدا کرده و برحسب تعداد، چندین بار به طور مجدد آن ها را به داده ها اضافه می کنیم تا برچسب های 1 بیشتری برای آموزش داشته باشیم. سپس به جای classification، از regression برای حل مسئله استفاده کردیم. کلاس 1 را به خروجی 1 تا 20 و کلاس 0 را به خروجی 1- تا 20- نگاشت می کنیم. پس از آموزش، میانگین نقاط خروجی را محاسبه کرده و آن هایی که مقدار بیشتر از 0 دارند را برابر با برچسب 1 در نظر می گیریم. این شیوه کمک می کند تا خروجی ها، حداکثر فاصله ممکن را داشته باشند و جدا کردن آن ها ساده باشد. این حالت همان طور که ادامه می بینیم، خروجی خیلی بهتری نسبت به حالتی که از classification استفاده کردیم دارد. در صفحه بعد، درصد صحت ها گزارش شده اند.

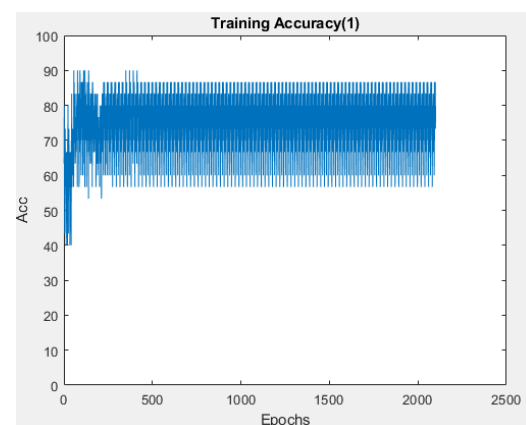
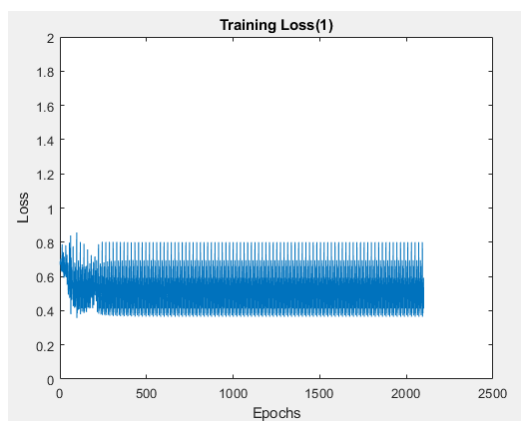
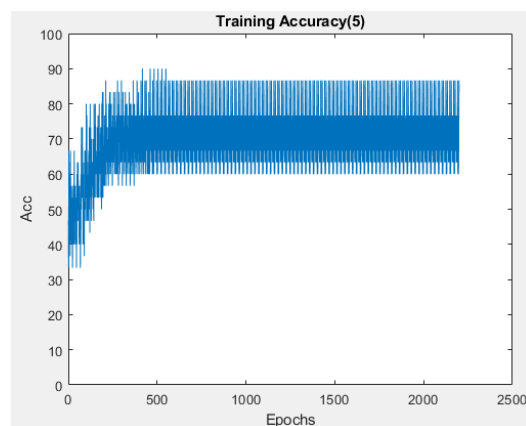
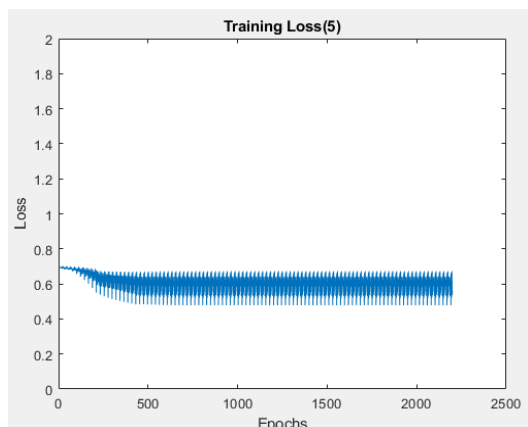


Accuracy Disease	1st try	2st try	3st try	4st try	5st try	Average	Best
Normal	72.06	70.59	77.94	75	72.06	73.53	77.94
AF	67.61	70.42	66.20	71.83	73.24	69.86	73.24
I-AVB	70.31	89.06	73.44	73.44	67.19	74.69	89.06
LBBB	84.62	64.62	84.62	84.62	73.85	78.47	84.62
RBBB	56.96	41.77	53.16	54.43	53.16	51.9	56.96
PAC	84.13	74.60	74.6	63.49	74.6	74.28	84.13
PVC	79.03	82.26	77.42	85.48	80.65	80.97	85.48
STD	67.69	66.15	76.92	70.77	78.46	72	78.46
STE	75.76	80.3	83.33	84.85	71.21	79.09	83.33

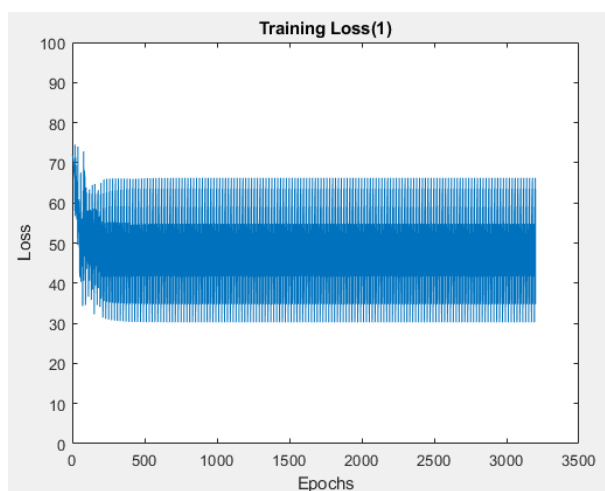
در این قسمت می بینیم که نسبت به قسمت های قبلی، درصد های صحت برای هر کلاس، در مواردی افت و در مواردی افزایش داشته اند. ابتدا شاید این مورد به ذهن برسد که یادگیری خوبی انجام نشده است اما نباید از یاد برد که هدف نهایی ما، تشخیص 8 بیماری مختلف بود و درصد صحت هر کلاس به طور جداگانه، معیار اصلی ما برای سنجش نیست. ابتدا نمودار های زیر، دقت و خطای آموزش را برای حالتی که از classification استفاده کردیم، نشان می دهند. خروجی ها در ظاهر مطلوب هستند و پس از محاسبه دقت نهایی، به دقتی بهتر از 20% دست پیدا نکردیم.



خروجی های زیر برای حالت classification هستند.



نمودار های زیر، مقدار خطا را برای حالت regression نشان می دهد.





خروجی نهایی به صورت زیر است.

Confusion Matrix										
Output Class	1	2	3	4	5	6	7	8	9	
	7 8.8%	1 1.3%	3 3.8%	1 1.3%	2 2.5%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	46.7% 53.3%
	0 0.0%	8 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	4 5.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	2 2.5%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	66.7% 33.3%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 11.3%	0 0.0%	1 1.3%	2 2.5%	0 0.0%	75.0% 25.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 2.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	1 1.3%	1 1.3%	1 1.3%	0 0.0%	2 2.5%	0 0.0%	13 16.3%	0 0.0%	72.2% 27.8%
	2 2.5%	1 1.3%	0 0.0%	1 1.3%	2 2.5%	0 0.0%	0 0.0%	0 0.0%	11 13.8%	64.7% 35.3%
Target Class										

همان طور که ملاحظه می شود، نتیجه نسبت به حالت classification، خیلی بهتر است. در این قسمت از حالت MLP نیز خروجی بهتری گرفتیم اما همچنان شبکه عصبی RBF، بهترین خروجی را برای ما داشت. به طور کلی دیدیم که شبکه های عصبی، انعطاف و سهولت بیشتری نسبت به شبکه های فازی داشته و نتایج آن ها بهتر و قابل اطمینان تر است.