

به نام خدا

هوش مصنوعی و محاسبات زیستی

گزارش پروژه نهایی

پردازش سیگنال های EEG

سجاد آفرتی

96101154

فاز یک:

انتخاب ویژگیها: تعداد 675 ویژگی از سیگنال های هر نمونه در حوزه زمان و 300 ویژگی در حوزه فرکانس استخراج شده اند؛ این ویژگی ها به شرح زیر میباشند:

ویژگی های حوزه زمان:

1- واریانس زمانی داده های هر کانال برای هر نمونه و همچنین ارتباط داده های هر دو کانال برای هر نمونه (کوواریانس داده های کانال های هر نمونه): مشخص است که تعداد ویژگی هایی که از این مورد استخراج میشوند عبارت اند از:

$$n = 30 * 29 * 0.5 + 30 = 465$$

2- فرم فاکتور داده های موجود در هر کانال: تعداد ویژگی های بدست آمده از هر نمونه برابر با 30 میباشد.

3- برای هر کانال در هر نمونه تعداد ویژگی هایی که کمتر از صفر و بیشتر از صفر اند محاسبه میشوند. (60 ویژگی)

4- برای هر کانال در هر نمونه مقادیر ماکسیمم و مینیمم سیگنال در حوزه زمان و همچنین زمان متناظر با آنها محاسبه میشود. (120 ویژگی)

ویژگی های حوزه فرکانس:

برای هر کانال موارد زیر محاسبه میشوند که به طبع آن از هر نمونه 300 ویژگی استخراج میشود:

فرکانس بیشینه، فرکانس میانه، فرکانس میانگین، انرژی نسبی در هر یک از 7 باند فرکانسی معروف.

بعد از استخراج ویژگی های فوق برای هر نمونه معیار ماتریس پخشی تک بعدی برای استخراج ویژگی های موثر بکار گرفته شده است؛ با استفاده از این معیار تعداد 20 ویژگی از حوزه فرکانس و 10 ویژگی از حوزه زمان استخراج شدند که در ادامه آنها را بیان میکنیم:

ویژگی های استخراج شده در حوزه زمان:

کوواریانس بین کانال های 15 و 7، واریانس کانال 15، کوواریانس کانال 15 و 11، کوواریانس کانال 19 و 11، کوواریانس کانال 15 و 19، کوواریانس کانال 20 و 11، کوواریانس کانال 20 و 15، واریانس کانال 20، کوواریانس کانال 21 و 15، دامنه ماکسیمم دیتای نمونه برداری شده ی کانال 19 در هر نمونه

ویژگی های استخراج شده در حوزه فرکانس:

انرژی موجود در باند گاما2 برای کانال های 1، 4، 5، 6، 8، 9، 10، 11، 12، 13، 14، 17، 25

انرژی موجود در باند گاما1 برای کانال های 5، 8، 9، 11، 13، 14، 17

همانطور که بیان شد تعداد 20 ویژگی در حوزه فرکانس و تعداد 10 ویژگی حوزه زمان و در مجموع 30 ویژگی استخراج شده است.

شبکه MLP:

با توجه به ویژگی های استخراج شده، برای طراحی شبکه MLP از دو لایه استفاده شده؛ با تغییر تعداد نورون های هر لایه بین 1 تا 20 در نهایت به این نتیجه رسیدم که بهترین شبکه MLP ممکنه یک شبکه ی MLP با 17 نورون در لایه اول و 14 نورون در لایه دوم میباشد. سپس روی چنین شبکه ای تمامی روش های ممکن برای آموزش را امتحان کردم و در نهایت بهترین روش ممکن روش trainoss محاسبه شد.

یک شبکه MLP با شرایط فوق رو 10 بار با 5_fold_cross_validation آموزش دادم (جمعاً 50 بار) و در نهایت بهترین شبکه را روی داده های تست اعمال کردم. اینکار را بدلیل بستگی accuracy این شبکه به نقاط اولیه انجام دادم تا بهترین شبکه ممکن را بتوانم استخراج کنم. در این شرایط همواره بهترین شبکه ممکنه دارای accuracy بین 73 تا 81 درصد بود.

در اینجا برچسب هایی که برای داده های تست با نام "test_labels_MLP_phase1" ارائه شده است توسط شبکه ای که با نام "best_net_MLP_phase1" در میان فایل های ارائه شده ذخیره شده است برچسب خورده است. این شبکه دارای accuracy = 80.61% بوده است.

شبکه RBF:

با استفاده از ویژگی های استخراج شده برای یافتن بهترین شبکه RBF ممکن شعاع را در بازه 0.5-2 و با گامهای 0.1 و تعداد نورون ها رو نیز در بازه 5-20 تغییر دادم. در نهایت بهترین شبکه RBF ممکن با $r = 1.3$, $n = 3$ آموزش داده شد. دقت شود که این شبکه به نقطه اولیه وابسته نیست و نیازی به تکرار و آموزش چندباره ی شبکه برای انتخاب بهترین شبکه ممکن نیست.

در اینجا نیز برچسب هایی که برای داده های تست با نام "test_labels_RBF_phase1" ارائه شده است توسط شبکه ای که با نام "best_net_RBF_phase1" در میان فایل های ارائه شده ذخیره شده است برچسب خورده است. این شبکه دارای $accuracy = 81.21\%$ بوده است.

همانطور که میبیند و گفته شد دقت شبکه MLP متغیر و بین 74٪ تا 81٪ بوده است؛ در حالی که دقت بهترین شبکه RBF ممکن ثابت و برابر 81.21٪ بوده است. دلیل وجود این موارد را میتوان بستگی داشتن شبکه MLP به نقاط اولیه و عدم وجود این مورد در شبکه RBF دانست.

فاز دو:

در این فاز قرار است دسته ای از ویژگی هارا با در نظر گرفته وجود رابطه میان آنها انتخاب کنیم؛ در اینجا از دو الگوریتم برای انتخاب این ویژگی ها استفاده کردم. این دو الگوریتم در دو تابع با نام های genetic.m, genetic1.m ارائه شده اند.

تابع genetic.m: این تابع که مبتنی بر الگوریتم های تکاملی پیاده سازی شده است مختصرا به شرح زیر است:

1. **تابع سازگاری:** تابع سازگاری در اینجا همان معیار fisher در فضای چند بعدی میباشد که میخواهیم دسته ویژگی های انتخاب شده دارای بیشترین مقدار ممکن در این بعد باشند.
2. **نحوه رمزگذاری:** اگر بخواهیم از تمامی ویژگی های موجود k ویژگی را استخراج کنیم، از یک کروموزوم به طول k استفاده میکنیم که مقدار آلل هر یک میتواند یکی از 975 ویژگی موجود باشد.
3. **نحوه تولید جمعیت اولیه:** اینکار را به صورت جایگشتی و رندوم انجام میدهیم.
4. **Crossover:** برای اینکار از crossover یکنواخت با امکان تصحیح استفاده میکنیم؛ به آن معنا که هنگامی که از دو والد مربوطه، دو فرزند را بدست می آوریم ممکن است مقادیر آلل های موجود در یک فرزند تکراری باشد؛ در این صورت یک احتمال در نظر میگیریم که در صورت وقوع این اتفاق این آلل ها تغییر یابند.
5. **جهش:** برای ایجاد جهش از جهش استاندارد استفاده میکنیم. اینکار ممکن است باعث تکراری شدن بعضی آلل ها در یک فرزند باشد که البته ایرادی ندارد؛ چون میخواهیم حداکثر k ویژگی را استخراج کنیم.

تابع genetic1.m: این تابع نیز مبتنی با الگوریتم های تکاملی میباشد ولی در طراحی آن از محاسبات بیشتری بهره بردم و در نتیجه از نظر زمانی حدودا 3 یا 4 برابر از الگوریتم قبلی ناپهینه تر است؛ هر چند از نظر دقت هر دو الگوریتم تقریبا در یک سطح میباشد:

1. **تابع سازگاری:** تابع سازگاری در اینجا نیز همان معیار fisher در فضای چند بعدی میباشد که میخواهیم دسته ویژگی های انتخاب شده دارای بیشترین مقدار ممکن در این بعد باشند.
2. **نحوه رمزگذاری:** اگر بخواهیم از تمامی ویژگی های موجود k ویژگی را استخراج کنیم، از یک کروموزوم به طول k استفاده میکنیم که مقدار آلل هر یک میتواند یکی از 975 ویژگی موجود باشد.

3. نحوه تولید جمعیت اولیه: اینکار را به صورت جایگشتی و رندوم انجام میدهیم.

4. Crossover: در این الگوریتم crossover نداریم ولی در عوض برای تولید فرزندها در هر گام به صورت زیر عمل میکنیم:

- 1) در هر والد چک میکنیم که بدترین ویژگی ای که در آن دسته قرار دارد و میتوان آن را حذف نمود پیدا میکنیم.
- 2) ویژگی موردنظر را حذف میکنیم.
- 3) حال در بین سایر ویژگی های موجود بررسی میکنیم کدام ویژگی اگر به ویژگی های فعلی افزوده شود، این دسته ویژگی را به دسته ویژگی بهتری تبدیل میکند.
- 4) دلیل عدم وجود crossover نیز همین است که هر والد تولید شده به صورت مستقل به مقدار بهینه خود نزدیک میشود.
- 5) همانطور که میبینید این الگوریتم از هوشمندی بهتری برخوردار است ولی از نظر زمانی و همچنین دقت نتایج تقریباً الگوریتم اول ارجحیت دارد.

5. جهش: برای ایجاد جهش از جهش استاندارد استفاده میکنیم. اینکار ممکن است باعث تکراری شدن بعضی آل ها در یک فرزند باشد که البته ایرادی ندارد؛ چون میخواهیم حداکثر k ویژگی را استخراج کنیم. از آنجا که هر پاسخ به صورت مستقل به مقدار بهینه خود نزدیک میشود ممکن است در طول فرایند در نقاط بهینه محلی الگوریتم متوقف شود. حال این فرایند جهش باعث پیمایش کلیه فضای جستجو میشود.

با استفاده از الگوریتم genetic ارائه شده بهترین شبکه MLP ممکن دارای دقت بین 76 تا 86 درصد و بهترین شبکه RBF ممکن دارای دقت بین 80 تا 87 بوده است.

برچسب هایی که در اینجا به عنوان نتایج نهایی با نام های "test_labels_RBF_phase2" و "test_labels_MLP_phase2" ارائه شده اند توسط شبکه های ارائه شده با نام های "best_net_MLP_phase2" و "best_net_RBF_phase2" برچسب گذاری شده اند؛ این شبکه به ترتیب دارای accuracy های 84.67 و 84.24 درصد بوده اند.