



دانشگاه خوارزمی

دانشکده فنی و مهندسی

گزارش تمرین 5

نام و نام خانوادگی: آرمین صبورمقدم

شماره دانشجویی: 4003904509

نام استاد: دکتر پدرام

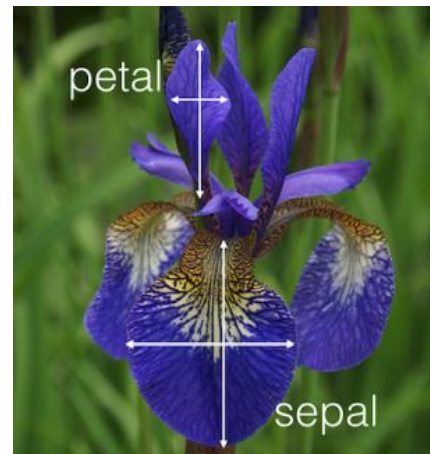
تاریخ: 6 دی ماه 1400

## راهنمای سلول های کد:

کد نوشته شده دارای 12 سلول بترتیب: 1. خواندن داده های ورودی Iris 2. تابع بردار ساز 3. تابع ماتریس ساز رندم 4. ورودی های کاربر 5. تابع محاسبه کننده مجموع ضرایب وزنها در ورودی ها 6. توابع فعال ساز 7. کد گذاری توابع فعال ساز 8. محاسبه مشتق توابع فعال ساز و کد گذاری آنها 9. کلاس بندی داده های iris و جدا کردن داده های train و test 10. اصلی ترین سلول که سلول شبکه عصبی است 11. محاسبه دقت روی داده های آموزشی 12. محاسبه دقت روی داده های تست

## توضیحات کامل هر سلول:

سلول 1: ابتدا داده های ورودی را از کتابخانه ی sklearn خوانده و دیتاست آن را به دو قسمت داده های train و test تقسیم میکنیم. تقسیم بندی صورت گرفته بصورت 80 درصد داده های آموزشی (120 تا) و 20 درصد داده های تست (30 تا) است. این دیتا ها مجموعا 150 تا رکورد بدین گونه که هر رکورد حاوی 4 فیچر اصلی به ترتیب طول سپال، عرض سپال، طول پتال و عرض پتال است (همانند شکل)



سلول 2: در این سلول 4 تا متغیر گرفته و بصورت برداری مقادیرشان را ارجاع میدهیم

سلول 3: این سلول ورودی بترتیب n و m را میگیرد و ماتریس رندم آن را میسازد با n سطر و m ستون از این ماتریس در سلول اصلی برای مقدار دهی اولیه ماتریس وزن ها استفاده میکنیم.

سلول 4: مقادیر اولیه از جمله تعداد لایه های مخفی، تعداد نورون های هر لایه، نوع تابع فعال ساز در هر لایه و نرخ یادگیری را از کاربر میگیریم.

سلول 5: این سلول تابع ای است که مجموع ضرایب وزن ها در ورودی ها (مانند grandmother cell) را محاسبه میکند.

سلول 6: این سلول حاوی تمامی توابع فعالیت است که بترتیب شامل تابع فعالیت خطی، سیگموئید، تانژانت هایپربولیک، رلو و در نهایت سافت مکس است. که متاسفانه برای سیگموئید بالاتر از 100 و یا کمتر از -100 خطای overflow میدهد.

سلول 7: این تابع را برای کدگذاری توابع سلول 6 استفاده میکنیم تا در سلول اصلی به راحتی آنرا فراخوانی کنیم.

سلول 8: همانند سلول 7 این تابع نیز برای کدگذاری و همچنین محاسبه ی مشتق توابع فعالیت تعریف شده است.

سلول 9: کلاس بندی در این سلول اتفاق می افتد و سه کلاس iris که به نام های setosa ، versicolor و virginica به ترتیب به کلاس های 0 ، 1 و 2 برچسب گذاری بصورت label می شوند. 50 تا رکورد اول دیتاست Iris برای کلاس 0، 50 تا رکورد دوم برای کلاس 1 و 50 تای آخر برای کلاس 2 است.

سلول 10 ( سلول اصلی): ابتدا ماتریس اصلی وزنها را میسازیم که ماتریس بزرگ 3 بعدی است که هر عنصر از ماتریس بزرگ شامل زیر ماتریس وزنهاست. که در ابتدا همگی بصورت رندم مقدار دهی میشوند. سپس وارد حلقه ی epoch میشویم برای محاسبه ی تعداد iteration ها. و روی داده های آموزشی برای آموزش شبکه عصبی و اپدیت وزنها دلتا را حساب میکنیم و شبکه عصبی را بهبود میدهیم.

سلول 11: در این سلول دقت شبکه عصبی را روی داده های آموزشی که 120 عدد هستند میسنجیم و تعداد خطا ها را نیز محاسبه میکنیم.

سلول 12: : در این سلول دقت شبکه عصبی را روی داده های تست که 30 عدد هستند میسنجیم و تعداد خطا ها را نیز محاسبه میکنیم.

### اجرای شبکه عصبی روی داده های تست:

در جدول تهیه شده شامل حالت های مختلف برای اجرای شبکه عصبی روی داده های تست با تغییر هایپر پارامتر هایی چون نرخ یادگیری، تابع فعالیت لایه ورودی، توابع فعالیت لایه های مخفی، تابع فعالیت لایه خروجی، تعداد نورون ها در هر لایه شبکه عصبی را ران نموده و تعداد خطا ها از 30 تا داده تست و همچنین دقت شبکه عصبی محاسبه گردیده است.

**نکته مهم:** لایه ی ورودی همواره دارای 4 نورون و لایه ی خروجی همواره دارای 3 نورون در

همه ی حالت های مقدار دهی هایپر پارامتر ها میباشد.

**آغاز مقدار دهی هایپر پارامتر ها و محاسبه خطای آنها:**

همیشه برای حل چنین مسائلی بصورت آزمون و خطا از ساده ترین راه حل ها شروع کرده و به ندرت شبکه پیچیده میسازیم چرا که اگر روی شبکه ساده دقت بالایی بگیریم از لحاظ زمان اجرا بصرفه تر میباشد. ابتدا با هایپر پارامتر هایی که تعیین آنها ساده تر هستند شروع میکنیم. تعداد iteration ها یا همان epoch که ساده ترین شان است را همواره برابر 100 در نظر میگیریم چرا که اگر به دقت نسبتا خوبی ( بالای 40 درصد برسیم ) به سادگی میتوانیم با افزایش تعداد epoch ها یادگیری نظارت شده را روی شبکه عصبی و آپدیت وزنهای آن تقویت کنیم.

مرحله بعدی تعیین مقدار نرخ یادگیری است که برای ساده ترین شبکه عصبی ( grandmother cell ) با تک نورون لایه مخفی (طبق **نکته مهم** مجموعا 8 نورون) تست گردید. و به ازای مقادیر 0.1 و 0.5 و 0.9 هیچ تغییری در دقت شبکه رخ نداد تنها به ازای مقدار 0.01 در شبکه با افزایش 26.69 دقت مواجه شدیم. پس برای سادگی کار تمامی نرخ یادگیری را 0.1 در نظر گرفته و چنان چه دقت نسبتا خوبی رسیدیم آن را 0.01 در نظر میگیریم تا تغییرات را ببینیم. اما در ادامه مشاهده شد هنگام افزایش تعداد نورون ها در لایه مخفی یادگیری به خوبی صورت نمیپذیرد و به همان مقدار اولیه 0.1 برمیگردیم.

حال تعداد لایه های مخفی را 1 در نظر گرفته و شبکه را برای تنها 1 نورون و حالت های مختلف توابع فعالیت روی لایه ورودی و مخفی و همچنین خروجی تست میکنیم. اگر تمامی توابع فعالیت تمامی لایه ها را یکسان در نظر بگیریم. بدترین تابع فعالیت tanh و sigmoid هستند با دقت 16.60 در مقابل linear و relu با دقت 23.30 درصد.

حال ترکیب مختلفی از توابع فعالیت را در نظر میگیریم در این حالت بهترین ترکیب برای تک لایه مخفی و تک نورون حالت تابع فعالیت relu یا linear برای لایه ورودی و همچنین linear برای لایه مخفی و در آخر sigmoid برای لایه خروجی با دقت 40 درصد میباشد.

تعداد نورون ها را در تک لایه مخفی به 2 افزایش داده در این حالت بهترین دقت برای تمامی لایه ها sigmoid میباشد با دقت 60 درصد. مقدار epoch را از 100 به 200 افزایش داده و دقت بالاتر 70 درصد را دریافت میکنیم. اما اگر باز هم epoch را افزایش دهیم دقت به 66.70 درصد کاهش میابد.

حال به مقادیر اولیه خود باز گردیم همان epoch صد و نرخ یادگیری 0.1 و تک لایه مخفی با 2 نورون اگر در این حالت ترکیب توابع مختلف فعالیت را اعمال کنیم بهترین دقت برای ترکیب relu و sigmoid و relu میباشد با دقت 60.04 درصد که epoch را به 300 افزایش دهیم مقدار دقت به 66.66 درصد نیز میرسد.

حال برای تک لایه مخفی و اینبار 3 نورون درون آن در نظر بگیرید به بهترین ترکیبی که دست پیدا میکنیم ترکیب relu و sigmoid و relu است که به دقت خیلی خوب 86.66 درصد میرسیم و با افزایش epoch به 300 به مقدار عالی 90 درصد تنها با 3 خطا میرسیم.

در این مرحله بیاید 2 لایه مخفی را در نظر بگیریم بهترین حالتی که دست پیدا میکنیم بترتیب با تعداد 4 و 3 نورون در هر لایه مخفی با ترکیب بترتیب relu و sigmoid و relu و linear میرسیم با دقت 90 درصد.

در جدول زیر تمامی تحلیل های ذکر شده در بالا را میتوانید در یک نگاه مشاهده کنید:

epoch	Learinnig rate	Input layer Activation_F	Hidden layers Activation_F	Output Activation_F	Number of neurons in each layer	Number of errors out of 30	accuracy
100	0.9	linear	linear	linear	1	23	23.30%
100	0.5	linear	linear	linear	1	23	23.30%
100	0.1	linear	linear	linear	1	23	23.30%
100	0.1	relu	relu	relu	1	23	23.30%
100	0.1	tanh	tanh	tanh	1	25	16.60%
100	0.1	sigmoid	sigmoid	sigmoid	1	25	16.60%
100	0.1	linear	linear	relu	1	23	23.30%
100	0.1	relu	linear	sigmoid	1	18	40%
100	0.1	linear	linear	sigmoid	1	18	40.00%
100	0.1	linear	linear	linear	2	23	23.30%
100	0.1	relu	relu	relu	2	23	23.30%
100	0.1	tanh	tanh	tanh	2	25	16.60%
100	0.1	sigmoid	sigmoid	sigmoid	2	17	43.30%
100	0.01	sigmoid	sigmoid	sigmoid	2	25	16.66%
200	0.1	sigmoid	sigmoid	sigmoid	2	12	60.00%
500	0.1	sigmoid	sigmoid	sigmoid	2	9	70.00%
1000	0.1	sigmoid	sigmoid	sigmoid	2	10	66.70%
100	0.1	relu	sigmoid	relu	2	12	60.04%
300	0.1	relu	sigmoid	relu	2	10	66.66%
100	0.1	linear	sigmoid	relu	2	11	63.33%
100	0.1	relu	sigmoid	relu	3	4	86.66%
500	0.1	relu	sigmoid	relu	3	3	90%
100	0.1	relu	sigmoid, sigmoid	relu	3,2	5	83.33%
500	0.1	relu	sigmoid, tanh	relu	2,3	4	86.66%
100	0.1	relu	sigmoid, tanh	relu	4,2	4	86.66%
500	0.1	relu	sigmoid, relu	linear	4,3	3	90%