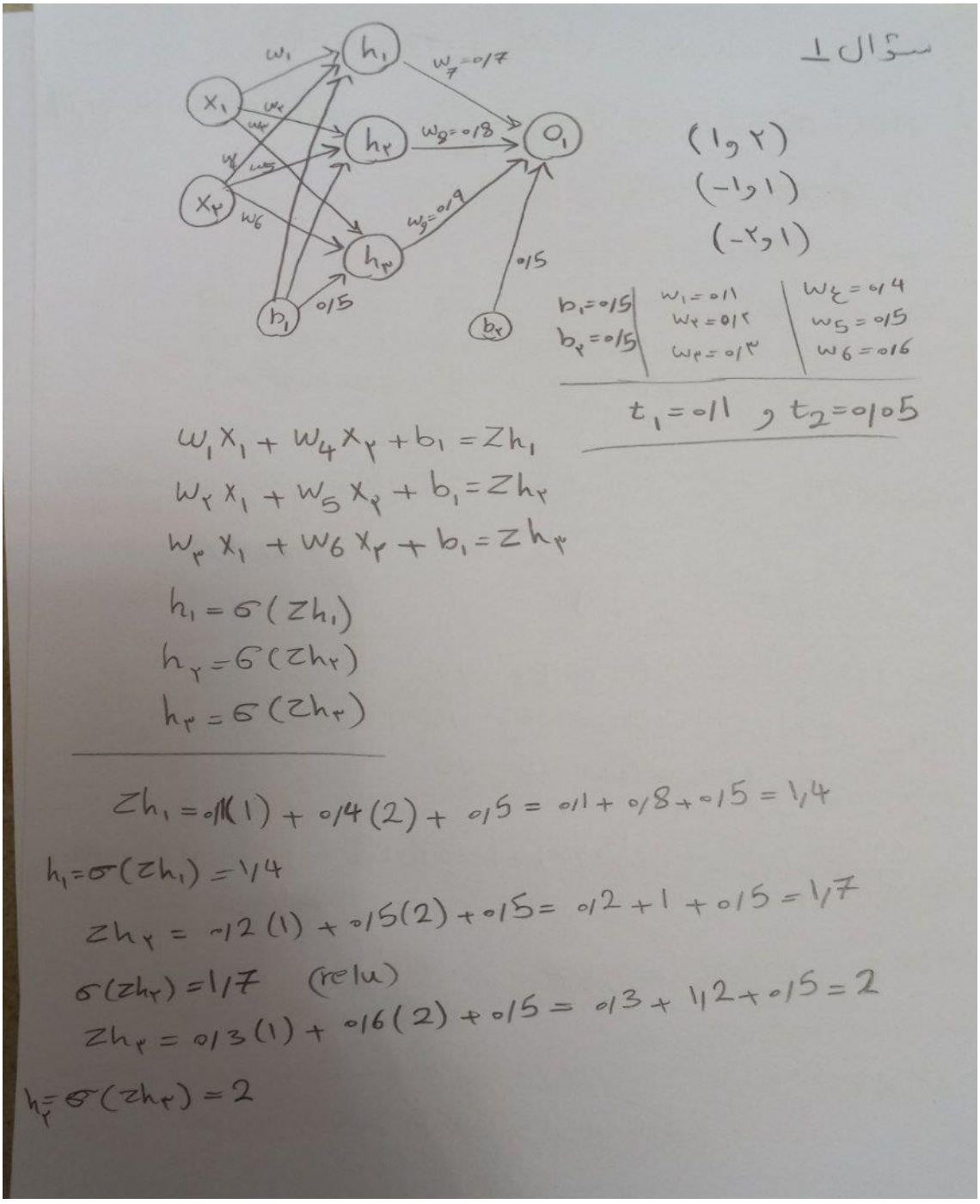
تمرین اول هوش محاسباتی

Yasmin Madani
97532265

فهرست

سوال۱


سوال ۱
w_1
w_2
w_3
w_4
w_5
w_6
$w_7=0/7$
$w_8=0/8$
$w_9=0/9$
x_1
x_2
h_1
h_2
h_3
o_1
b_1
b_2
0/5
0/5
(۱، ۲)
(-۱، ۱)
(-۲، ۱)
$b_1=0/5$
$b_2=0/5$
$w_1=0/1$
$w_2=0/2$
$w_3=0/3$
$w_4=0/4$
$w_5=0/5$
$w_6=0/6$
$t_1=0/1$ و $t_2=0/05$
$w_1x_1+w_4x_2+b_1=zh_1$
$w_2x_1+w_5x_2+b_1=zh_2$
$w_3x_1+w_6x_2+b_1=zh_3$
$h_1=\sigma(zh_1)$
$h_2=\sigma(zh_2)$
$h_3=\sigma(zh_3)$
$zh_1=0/1(1)+0/4(2)+0/5=0/1+0/8+0/5=1/4$
$h_1=\sigma(zh_1)=1/4$
$zh_2=0/2(1)+0/5(2)+0/5=0/2+1+0/5=1/7$
$\sigma(zh_2)=1/7$ (relu)
$zh_3=0/3(1)+0/6(2)+0/5=0/3+1/2+0/5=2$
$h_3=\sigma(zh_3)=2$

$h_1 w_7 + h_2 w_8 + h_3 w_9 + b_2 = z o_1$

$(1.4)(0.17) + (1.7)(0.18) + (2)(0.19) + 0.15 = 0.198 + 1.36 + 1.8 = 4.64$

$O_1 = \sigma(z o_1) = 0.99$ (sigmoid)

$z h_1 = 0.11(-1) + 0.14(1) + 0.15 = 0.18 \quad \sigma(z h_1) = 0.18$

$z h_2 = 0.12(-1) + 0.15(1) + 0.15 = 0.18 \quad \sigma(z h_2) = 0.18$

$z h_3 = 0.13(-1) + 0.16(1) + 0.15 = 0.18 \quad \sigma(z h_3) = 0.18$

$0.15 + (0.18)(0.17) + (0.8)(0.18) + (0.18)(0.19) = 0.156 + 0.164 + 0.172$

$= 2.142$

$O_1 = \sigma(z o_1) = 0.91$

$z h_1 = 0.11(-2) + 0.14(1) + 0.15 = 0.17$

$z h_2 = 0.12(-2) + 0.15(1) + 0.15 = 0.16$

$z h_3 = 0.13(-2) + 0.16(1) + 0.15 = 0.15$

$0.15 + (0.17)(0.17) + (0.16)(0.18) + (0.15)(0.19) = 0.149 + 0.148 + 0.145 + \%$

$= 1.92$

$O_1 = \sigma(z o_1) = 0.87$

//

$$E = (o_1 - t_1)^2$$

$$\frac{dE}{do_1} = 2(o_1 - t_1)$$

$$\sigma = \frac{1}{1+e^{-x}}$$

$$\frac{d\sigma}{dx} = \frac{e^{-x}}{(1+e^{-x})^2}$$

$$\frac{d\sigma}{dx} = \sigma(x)(1-\sigma(x))$$

$$w_7 h_1 + w_8 h_2 + w_9 h_3 + b_2 = zo_1$$

$$\frac{dzo_1}{dw_7} = h_1 \qquad \frac{dzo_1}{dw_8} = h_2 \qquad \frac{dzo_1}{dw_9} = h_3$$

$$\frac{dzo_1}{db_2} = 1$$

$$\frac{dE}{dw_7} = \frac{dE}{do_1} \times \frac{do_1}{dzo_1} \times \frac{dzo_1}{dw_7} = 2(o_1 - t_1)(o_1(1-o_1))h_1$$

$$= 2(0/99 - 0/1)(0/99(1-0/99))1/4 = 0/0024$$

$$\frac{dE}{dw_7} = 0/0024$$

$$w_7 = w_7 - \alpha \frac{dE}{dw_7} = 0/7 - (0/01)(0/0024) = 0/6997$$

به طریق مشابه برای سایر مولفه ها نیز عمل می کنیم .

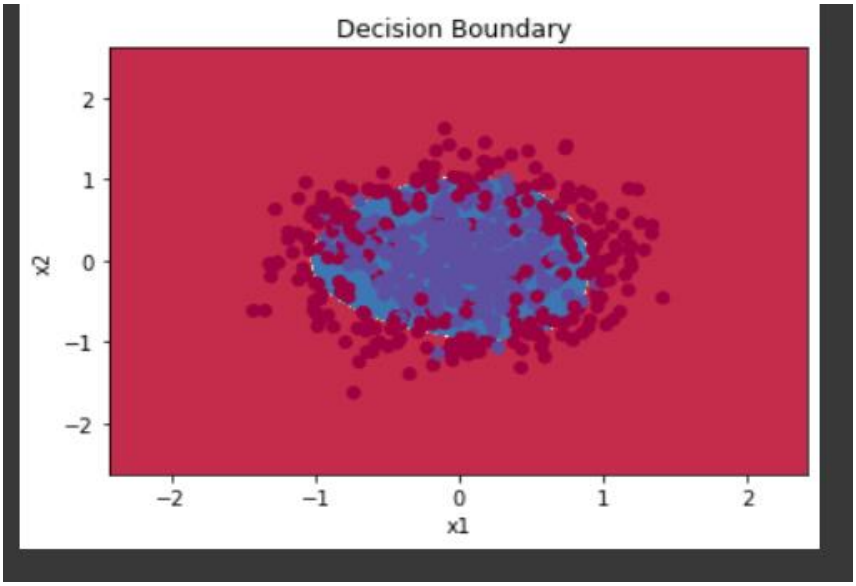
بله با توجه به آن که این مسئله را با پرسپترون هم می توان حل کرد و مقابل دیتا خطی است با آرا لاین نیز می توانیم دو کلاس را از هم جدا کنیم .

سوال۲

برای پیاده سازی یک پرسپترون چندلایه نیاز داریم تا توابع اکتیواسیون مورد نظر را به کمک روابط ریاضی پیاده سازی کنیم.

سپس ویژگی های یک شبکه عصبی چندلایه مانند تعدادلایه های میانی و نورون های هر کدام نورون های خروجی و ورودی و مقادیر اولیه وزن‌ها وبایاس‌ها را تعیین کنیم.

از آن پس مطابق الگوریتم شبکه MLP با صدا زدن forwardpassو backward عملیات تعیین باند را انجام میدهیم و دقت را محاسبه می نماییم.



توجه داریم اگر تعداد زیادی نورون برای لایه میانی انتخاب کنیم شبکه اورفیت می شود یا به عبارتی داده هارا حفظ می کند.

سوال۳

این مجموعه داده از ۵۰۰۰۰ تصویر آموزشی رنگی ۳۲*۳۲ و ۱۰۰۰۰ تصویر آزمایشی است که در۱۰ دسته برچسب گذاری شده اند.

این ده دسته در جدول زیر آمده اند.

Label	Description
0	airplane
1	automobile
2	bird
3	cat
4	deer
5	dog
6	frog
7	horse
8	ship
9	truck

برای لود کردن این دیتا کافی است تابع لود را از برای این دیتاست از کتابخانه کراس در تنسورفلو صدا بزنیم و خروجی و خروجی را به صورت داده ی تست و ترین ذخیره کنیم.

```
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```

سپس داده ها را برای ورود به شبکه آماده و نرمال می کنیم.

یک شبکه با سه لایه به کمک کتابخانه کراس می سازیم. و آن را کامپایل و فیت میکنیم در این بخش ۰.۱۵ از داده های ترین را به ولیدیشن اختصاص داده ایم.

نتایج زیر به دست آمده از تغییرات مدل و تحلیل آن هاست.

شبکه با اپتیمایزر SGD

epoch	learning rate	Batch size	Activation function	Hidden unit	Validation split	Train acc	Val acc	Test acc
100	0.0001	512	relu	32	0.15	0.4730	0.4509	0.4481
100	0.001	512	relu	32	0.15	0.3677	0.3559	0.3564
100	0.001	512	relu	16	0.15	0.3529	0.3457	0.3456
200	0.0001	512	tanh	32	0.15	0.3114	0.3037	0.3122
50	0.01	512	sigmoid	32	0.15	0.3586	0.3471	0.3509
50	0.01	512	relu	512	0.1	0.4813	0.4014	0.4561
50	0.01	512	relu	128	0.1	0.4690	0.4510	0.4464
20	0.01	512	relu	128	0.1	0.4180	0.4120	0.4101

شبکه با اپتیمایزر Adam

epoch	learning rate	Batch size	Activation function	Hidden unit	Validation split	Train acc	Val acc	Test acc
100	0.0001	512	relu	32	0.15	0.4728	0.4511	0.4497
100	0.0001	512	relu	32	0.1	0.5056	0.4776	0.4762
100	0.0001	512	relu	16	0.1	0.4564	0.4334	**0.4373**
100	0.0001	512	relu	512	0.1	0.6841	0.5404	**0.5373**
100	0.0001	512	relu	400	0.1	0.6661	0.5302	**0.5292**
100	0.0001	512	relu	200	0.1	0.6168	0.5262	**0.5155**
100	0.0001	512	relu	50	0.1	0.5222	0.4854	**0.4797**
50	0.0001	256	relu	256	0.15	0.5712	0.5075	0.4985
50	0.0001	512	relu	32	0.15	0.4427	0.4224	0.4308
50	0.001	512	sigmoid	128	0.15	0.4881	0.4617	0.4520

50	0.0001	256	relu	128	0.15	0.5354	0.4857	0.4861
20	0.0001	512	relu	128	0.1	0.4676	0.4508	0.4516
50	0.0001	32	relu	512	0.1	0.7216	0.5534	0.5385
100	0.0001	512	relu	128	0.1	0.5802	0.5158	0.5052
50	0.0001	32	relu,relu	256,256	0.1	0.7454	0.5384	0.5302
100	0.0001	512	relu,relu	64,32	0.15	0.5251	0.4884	0.4801
100	0.001	512	relu,relu	64,32	0.15	0.4825	0.4497	**0.4587**
50	0.0001	512	relu,relu	1204,256	0.1	0.7070	0.5422	**0.5417**

آزمایشهای فوق صورت گرفته و طبیعتا اپتیمایزر Adamعملکرد بهتری دارد اما از آنجا که SGD راه مطرح شده در کلاس درسی است بهترین نتیجه آن را درنظر میگیریم.

همان طور که از جداول بالا مشاهده می شود انتخاب پارامتر ها در نتیجه تاثیر متفاوتی دارد. برای مثال با پیچیده کردن شبکه، شبکه دچار اورفیت میشود و به عبارتی شبکه داده‌ها را حفظ می کند مانند آنچه در سطر پایانی جدول Adam مشاهده می کنیم. پیچیده شدن شبکه می تواند دلایل مختلفی داشته باشد از جمله می توان به تعداد نورون های لایه میانی و تعداد لایه ها اشاره کرد.

از دیگر عواملی که دقت را کاهش می دهد می توان به این اشاره کرد که بخش زیادی از داده ها را برای ولیدیشن در نظر بگیریم از این رو تعداد داده‌های بخش آموزش کم می شود و شبکه آموزش کافی نمی‌بیند.

مومنتوم

یک مشکل با روش گرادیان این است که می تواند در مورد مسائل بهینه سازی که دارای مقادیر زیادی انحنا یا گرادیان نویز دارهستند، به اطراف فضای جستجو بپرد، و می تواند در نقاط مسطح در فضای جستجو که گرادیان ندارند گیر کند. یکی از روش‌های حل این مشکل اضافه کردن تاریخچه به معادله به‌روزرسانی پارامتر بر اساس گرادیانی است که در به‌روزرسانی‌های قبلی با آن مواجه شده‌ایم.

مومنتوم به جستجو اجازه می‌دهد در یک جهت در فضای جستجو اینرسی ایجاد کند و بر نوسانات گرادیان‌های نویزدار و گیرکردن در نقاط مسطح فضای جستجو غلبه کند.

مزیت مومنتوم این است که تغییرات بسیار کوچکی در SGD ایجاد می‌کند، اما سرعت یادگیری را افزایش می‌دهد.

با فعال سازی ممنتوم دقت در هرسه بخش اموزش ،ولیدیشن و تست افزایش می یابد.

epoch	learning rate	Batch size	Activation function	Hidden unit	Validation split	Train acc	Val acc	Test acc
50	0.01	512	relu	512	0.1	0.4890	0.4650	0.4614

روش Weight Decay

با اضافه کردن یک عبارت جریمه به تابع هزینه یک شبکه عصبی کار می کند که اثر کوچک شدن وزن ها را در طول انتشار پس زمینه دارد. این کمک می کند تا شبکه از برازش بیش از حد داده های آموزشی و همچنین مشکل گرادیان انفجاری جلوگیری کند. در شبکه های عصبی دو پارامتر قابل تنظیم هستند. این وزن ها و سوگیری ها هستند. وزن ها مستقیماً بر رابطه بین ورودی ها و خروجی های آموخته شده توسط شبکه عصبی تأثیر می گذارند زیرا در ورودی ها ضرب می شوند. از نظر ریاضی، سوگیری ها فقط رابطه را از رهگیری خنثی می کنند. بنابراین ما معمولا فقط وزنه ها را منظم می کنیم.

با فعال سازی این مورد درشبکه دقت هر سه فاز افزایش پیدا کرد.

epoch	learning rate	Batch size	Activation function	Hidden unit	Validation split	Train acc	Val acc	Test acc
50	0.01	512	relu	512	0.1	0.4894	0.4656	0.4620

سوال ۴

به نظر می رسد شبکه MLPقابلیت جنرال شدن بیشتری نسبت به سایر و پرسپترون کمترین قابلیت را داشته باشد چرا که شبکه عصبی چند لایه میتواند داده ها با انواع تقسیم بندی و مرزها را تفکیک کرده اما سایرین به مسائل تفکیک پذیر خطی محدود اند.

overfit

هنگامی که الگوریتم های یادگیری ماشین ساخته می شوند، از مجموعه داده های نمونه برای آموزش مدل استفاده می کنند. با این حال، زمانی که مدل برای مدت طولانی روی داده‌های نمونه تمرین می‌کند یا زمانی که مدل بسیار پیچیده است، می‌تواند شروع به یادگیری «نویز» یا اطلاعات نامربوط در مجموعه داده‌ها کند. وقتی مدل نویز را به خاطر می‌سپارد و خیلی نزدیک به مجموعه آموزشی منطبق می‌شود، مدل «فرابرازش » می‌شود و نمی‌تواند به خوبی به داده‌های جدید تعمیم دهد. اگر یک مدل نتواند به خوبی به داده های جدید تعمیم دهد، آنگاه نمی تواند وظایف طبقه بندی یا پیش بینی را که برای آن در نظر گرفته شده است انجام دهد.

آموزش یا ترین بیش اندازه شبکه می تواند از دلایل رخ داد این پدیده باشد. از دیگر دلایل دیگر می تواند پیچیده کردن شبکه با افزایش لایه های مخفی یا تعداد نورون‌های هر لایه به طوری که نامناسب با تعداد داده آموزش و ویژگی‌های آن باشد ،اشاره کرد.

نرخ خطای پایین و واریانس بالا شاخص های خوبی برای تشخیص این پدیده هستند.

از دیگر نشانه ها می‌توان به تفاوت چشمگیر در دقت فاز آموزش و ولیدیشن و تست اشاره کرد.

اگر داده های آموزشی دارای نرخ خطای کم و داده های آزمون دارای نرخ خطای بالایی باشند، سیگنال بیش از حد برازش می دهد.

در حالی که استفاده از یک مدل خطی به ما کمک می کند تا از برازش بیش از حد جلوگیری کنیم، بسیاری از مسائل دنیای واقعی، غیرخطی هستند. علاوه بر درک نحوه تشخیص بیش از حد برازش، مهم است که بدانیم چگونه به طور کلی از بیش برازش جلوگیری کنیم.

روش های جلوگیری از overfit

- **توقف زودهنگام**

این روش به دنبال توقف آموزش قبل از شروع یادگیری نویز درون مدل توسط مدل است. این رویکرد باعث می شود روند آموزش خیلی زود متوقف شود و منجر به مشکل معکوس کمبود تناسب شود. در اینجا یافتن "sweet spot" بین underfitting و overfitting هدف نهایی است.

- **آموزش با داده‌های بیشتر**

گسترش مجموعه آموزشی برای گنجاندن داده‌های بیشتر می‌تواند دقت مدل را با فراهم کردن فرصت‌های بیشتر برای تجزیه و تحلیل رابطه غالب بین متغیرهای ورودی و خروجی افزایش دهد. با این حال، زمانی که داده های تمیز و مرتبط به مدل تزریق می شود، این روش موثرتر است. در غیر این صورت، می توانید به پیچیدگی بیشتر مدل ادامه دهد و باعث جابجایی بیش از حد آن شود.

- **افزایش داده ها**

در حالی که بهتر است داده های تمیز و مرتبط را به داده های آموزشی خود تزریق کنید، گاهی اوقات داده های نویز برای پایدارتر کردن مدل اضافه می شود. با این حال، این روش باید با صرفه جویی انجام شود.

- **انتخاب ویژگی**

وقتی یک مدل می‌سازید، تعدادی پارامتر یا ویژگی خواهید داشت که برای پیش‌بینی یک نتیجه معین استفاده می‌شوند، اما بسیاری از اوقات، این ویژگی‌ها ممکن است برای دیگران زائد باشد. انتخاب ویژگی فرآیند شناسایی مهمترین آنها در داده های آموزشی و سپس حذف موارد نامربوط یا اضافی است. این معمولا با کاهش ابعاد اشتباه گرفته می شود، اما متفاوت است. با این حال، هر دو روش به ساده سازی مدل شما برای ایجاد روند غالب در داده ها کمک می کند.

- **منظم سازی**

اگر بیش از حد برازش زمانی اتفاق بیفتد که یک مدل خیلی پیچیده است، منطقی است که تعداد ویژگی ها را کاهش دهیم. اما اگر ندانیم کدام ورودی ها را در طول فرآیند انتخاب ویژگی حذف کنیم روش‌های منظم‌سازی می‌تواند بسیار مفید باشد. منظم‌سازی یک "جریمه" برای پارامترهای ورودی با ضرایب بزرگ‌تر اعمال می‌کند، که متعاقباً مقدار واریانس در مدل را محدود می‌کند. این روش به دنبال شناسایی و کاهش نویز درون داده‌ها هستند.

underfit

وقتی یک مدل داده نمی تواند رابطه بین متغیرهای ورودی و خروجی را به طور دقیق ضبط کند، و نرخ خطای بالایی را هم در مجموعه آموزشی و هم در داده های دیده نشده ایجاد می کند.

این پدیده زمانی اتفاق می‌افتد که یک مدل خیلی ساده باشد، که می‌تواند نتیجه نیاز یک مدل به زمان آموزش بیشتر، ویژگی‌های ورودی بیشتر یا منظم‌سازی کمتر باشد. مانند برازش بیش از حد، زمانی که یک مدل کمتر برازش می‌کند، نمی‌تواند روند غالب را در داده‌ها ایجاد کند، که منجر به خطاهای آموزشی و عملکرد ضعیف مدل می‌شود. اگر یک مدل نتواند به خوبی به داده های جدید تعمیم دهد، نمی توان از آن برای کارهای طبقه بندی یا پیش بینی استفاده کرد. تعمیم یک مدل به داده های جدید در نهایت چیزی است که به ما امکان می دهد هر روز از الگوریتم های یادگیری ماشین برای پیش بینی و طبقه بندی داده ها استفاده کنیم.

بایاس بالا و واریانس کم شاخص های خوبی برای عدم تناسب هستند. از آنجایی که این رفتار را می توان در حین استفاده از مجموعه داده آموزشی مشاهده کرد، مدل های کم برازش معمولا آسان تر از مدل های بیش برازش شده شناسایی می شوند.