

## به نام خدا

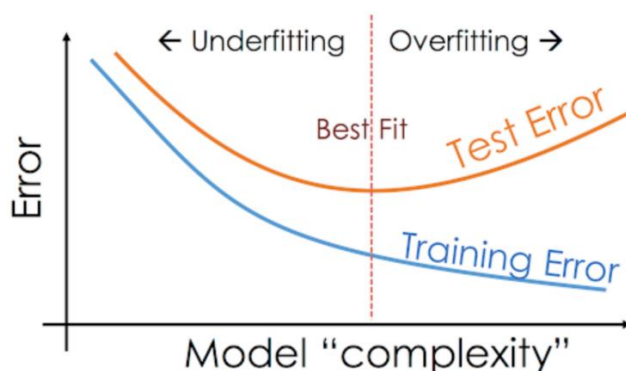
استاد: دکتر محمدرضا محمدی  
درس مبانی بینایی کامپیوتر

نام: فاطمه زهرا بخشنده  
شماره دانشجویی: 98522157

## گزارش تمرین 12:

### سوال اول:

یک مسئله اساسی در یادگیری ماشین این است که چگونه الگوریتمی بسازیم که نه تنها بر روی داده های آموزشی بلکه برای ورودی های جدید نیز به خوبی عمل کند. **Underfitting**: اگر مدل خیلی ضعیف باشد، و complexity آن پایین باشد ممکن است نتواند داده های training را هم به خوبی یاد بگیرد. در این زمان دقت مدل هم روی داده train و هم روی داده test کم است. به این حالت underfitting گفته می شود. یعنی مدل هنوز نتوانسته مشخصه متمایز کلاس ها را یاد بگیرد.



underfit شدن یک شبکه می تواند یک سری دلیل داشته باشد. مثلاً:

- مدل برای تسک مورد نظر بیش از حد ساده است.
- ساختار مدل انتخابی برای تسک مورد نظر مناسب نیست. برای مثال برای تسک های مربوط به تصویر مدل های convolution ای مناسب هستند و عملکرد بسیار بهتری دارند. یا برای تسک های مربوط به دیتاهای با طول متغیر، مدل های RNN مناسب هستند.
- تعداد epoch ها کم است.
- در مواردی هم ممکن است دیتاست ما، دیتاست مناسبی نبوده و بسیار بی کیفیت است.

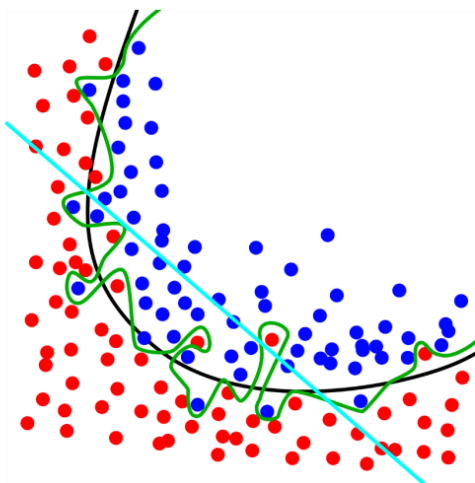
**Overfitting:** هرچه مدل پیچیده تر می شود، ظرفیت یادگیری آن، افزایش یافته و خطای training کاهش پیدا می کند. اما test error تا یک زمانی همراه training error کم شده و پس از یک جایی به بعد اتفاقا بدتر می شود. به این حالت overfitting گفته می شود.

Overfitting زمانی اتفاق می افتد که یک مدل دیتای آموزشی را به خوبی یاد گرفته و در واقع حفظ می کند، اما دقت آن روی داده validation و test خیلی خوب نیست و با دقت آن روی داده train فاصله زیادی دارد (واریانس بالایی داریم). در واقع مدل جزئیات داده آموزشی را حفظ می کند و سعی کرده روندی را در داده ها پیش بینی کند که بیش از حد نویزی است و مختص داده آموزشی است.

اما این قضیه روی عملکرد مدل تاثیر منفی دارد چون واقعیت موجود در همه داده ها را منعکس نمی کند و عمومیت آن کم است. overfit شدن یک شبکه می تواند یک سری دلیل داشته باشد. مثلا:

- سایز دیتاست کم است.
- دیتا های آموزشی cleaned نیستند و نویزی هستند.
- تعداد لایه ها و نورون ها زیاد هستند و شبکه برای مسئله ما بیش از حد پیچیده است.

برای مثال شکل زیر را در نظر بگیرید:



در منحنی آبی کمرنگ، underfit اتفاق افتاده است و مدل بایاس زیادی دارد.

مرز تصمیم مشکی رنگ، منحنی مناسبی است و فیچر های خوبی را یاد گرفته و تعمیم دهی مناسبی دارد.

در منحنی سبز رنگ، overfit روی داده های آموزشی اتفاق افتاده است و مدل واریانس بالایی دارد. در این مدل اگر یکی از داده های نویزی قرمز را در نظر بگیریم، این داده دارای الگویی است که مخصوص داده آموزشی بوده و ارتباطی با مسئله اصلی ندارد و گمراه کننده است.

اگر این داده وجود نداشت، مرز تصمیم سبز رنگ عوض میشد، پس در واقع مدل ما پیچیده بوده و خیلی حساس به داده های training شده است، و با عوض شدن تنها چند داده، شکل مرز ممکن است خیلی عوض شود. پس واریانس بالاست.

حالا اگر در داده validation یا test یک داده آبی در کنار این دایره نویری قرمز وجود داشته باشد، منحنی سبز رنگ، بخاطر مرز تصمیم حساس به داده آموزشی، این داده را به عنوان داده قرمز معرفی می کند که اشتباه است، در صورتی که اگر مرز تصمیم مشکلی را در نظر بگیریم چون قابلیت تعمیم آن خوب است، این داده را درست پیش بینی خواهد کرد.

### راه حل رفع این موارد:

- ❖ اگر شبکه ما دچار underfit شود می توانیم از روش های زیر استفاده کنیم:
  - ممکن است مدل ما برای تسک مورد نظر بیش از حد ساده است، پس مدل را پیچیده تر می کنیم. برای مثال تعداد لایه ها و تعداد نرون های هر لایه را افزایش می دهیم.
  - ممکن است مدل انتخابی برای تسک مورد نظر مناسب نباشد، پس ساختار مدل را عوض می کنیم. برای مثال برای تسک های مربوط به تصویر مدل های convolution ای مناسب هستند و عملکرد بسیار بهتری دارند. یا برای تسک های مربوط به دیتاهای با طول متغیر، مدل های RNN مناسب هستند.
  - تعداد epoch ها را بیشتر می کنیم تا مدل زمان بیشتری برای یادگیری داشته باشد.
  - در مواردی هم ممکن است دیتاست ما، دیتاست مناسبی نبوده و بسیار بی کیفیت بوده است. در این موارد دیتاست خود را تغییر داده و بهبود می دهیم.

- ❖ اگر شبکه ما دچار overfit شود می توانیم از روش های زیر استفاده کنیم:
  - ساده تر کردن مدل
  - استفاده از دیتای بیشتر
  - Regularization، که خود، روش های متفاوتی را شامل می شود:
    - استفاده از روش جریمه L1 یا L2 که قبل از انجام به روزرسانی معمولی مبتنی بر گرادیان، بردار وزن را در هر گام با یک ضریب ثابت کاهش می دهند. و منجر به اضافه کردن هزینه به Loss function برای آپدیت وزن ها می شود و وزن ها کوچکتر می شوند. (Weight decay).
    - استفاده از Drop out که در هر iteration نود های مختلفی صفر می شوند و در واقع در هر iteration شبکه ما به یک صورت متفاوت کوچکتر می شود. (در موقع test، Drop out نداریم).

- Data augmentation، یعنی ایجاد تغییراتی در دیتا مثلاً چرخاندن یا زاویه دادن تصاویر، flip، افزودن نویز و ...، و اضافه کردن این دیتاها به دیتاست.
- Early stopping: یک جایی زودتر تا  $w$  ها خیلی بزرگتر نشده اند و شبکه overfit نشده است learning را متوقف کنیم. اما مشکل این کار این است که دو هدف optimize کردن cost function و overfit نشدن را باهم mix کرده ایم و دیگر به صورت مستقل آن ها را انجام نمی دهیم.

**منابع:** اسلاید ها، کورس deep learning Andrew Ng

## سوال پنجم:

برای محاسبه دقت متوسط (AP) مراحل زیر را انجام می دهیم:

با توجه به مقادیر داده شده، در تصویر 8 پروپوزال داشتیم، که الگوریتم detection ما هر پروپوزال را با احتمالی که در ستون score مشخص شده است، positive تشخیص می دهد. همچنین با IOU، به ازای هر prediction، سنجیده شده که آیا prediction درستی است یا خیر. مثلاً ستون correct\_25 حالتی را نظر گرفته که اگر prediction ما مقدار IOU بیشتر از 0.25 با ground truth داشت، True و اگر کمتر بود False است. به همین ترتیب ستون های correct\_50 و correct\_75 به ترتیب مربوط به IOU = 0.50 و IOU = 0.75 می باشند.

- $AP_{25}$  مقدار AP به ازای IOU = 0.25 است.

- $AP_{50}$  مقدار AP به ازای IOU = 0.50 است.

- $AP_{75}$  مقدار AP به ازای IOU = 0.75 است.

1- ابتدا تشخیص ها را بر اساس امتیاز آنها مرتب میکنیم.

2- سپس threshold را بین همه مقادیر دو به دو امتحان می کنیم. و برای هر حالت precision و recall را حساب می کنیم.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

3- در آخر طبق نقاط مختلفی که داریم، منحنی precision بر حسب recall را رسم کرده، و سطح زیر این منحنی Average Precision در نظر گرفته می شود.

مقادیر داده شده به صورت زیر است:

score	correct_25	correct_50	correct_75
0.84	TRUE	TRUE	TRUE
0.79	FALSE	FALSE	FALSE
0.89	FALSE	FALSE	FALSE
0.96	TRUE	TRUE	FALSE
0.47	FALSE	FALSE	FALSE
0.39	TRUE	FALSE	FALSE
0.74	TRUE	TRUE	TRUE
0.29	FALSE	FALSE	FALSE

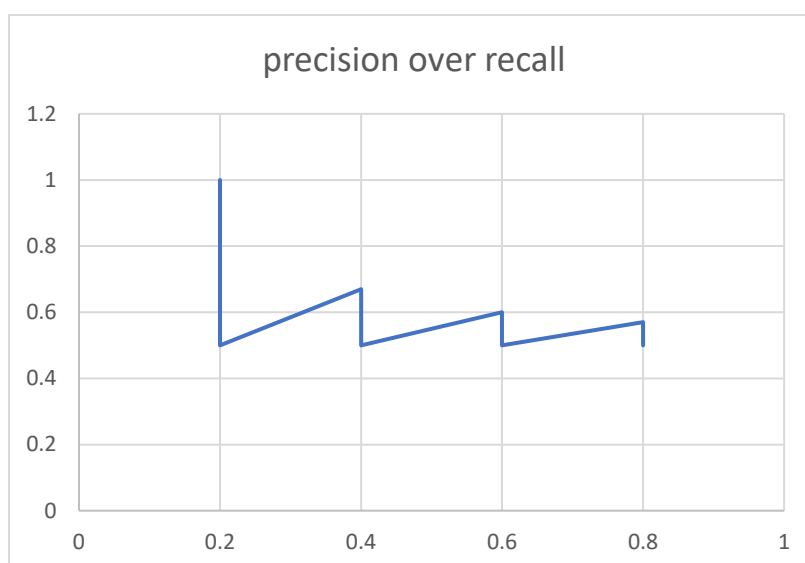
برای محاسبه  $AP_{25}$  از مقادیر ستون correct\_25 استفاده می کنیم، و مرحله‌ای که در بالا ذکر شد را برای آن انجام می دهیم. سپس به طور مشابه  $AP_{50}$  و  $AP_{75}$  را نیز محاسبه می کنیم.

### محاسبه $AP_{25}$ :

- 1- تشخیص ها را بر اساس امتیاز آنها مرتب میکنیم.
- 2- threshold را بین همه مقادیر دو به دو امتحان می کنیم. و برای هر حالت precision و recall را حساب می کنیم. (داخل تصویر 5 آبجکت وجود داشته است)

Rank	score	correct_25	precision	recall
1	0.96	TRUE	1.00	0.20
2	0.89	FALSE	0.50	0.20
3	0.84	TRUE	0.67	0.40
4	0.79	FALSE	0.50	0.40
5	0.74	TRUE	0.60	0.60
6	0.47	FALSE	0.50	0.60
7	0.39	TRUE	0.57	0.80
8	0.29	FALSE	0.50	0.80

- 3- طبق نقاط مختلفی که داریم، منحنی precision بر حسب recall را رسم کرده، و سطح زیر این منحنی Average Precision در نظر گرفته می شود.



$$AP_{25} = \left( \frac{(0.5 + 0.67) \times 0.2}{2} \right) + \left( \frac{(0.5 + 0.6) \times 0.2}{2} \right) + \left( \frac{(0.5 + 0.57) \times 0.2}{2} \right) = \left( \frac{3.34 \times 0.2}{2} \right) = 0.334$$

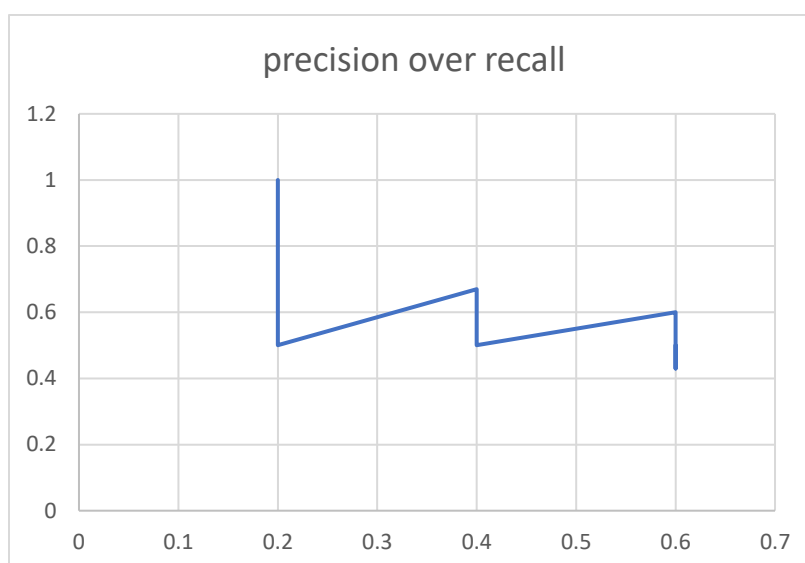
$$AP_{25} = 0.334$$

### محاسبه $AP_{50}$ :

- 1- تشخیص ها را بر اساس امتیاز آنها مرتب میکنیم.
- 2- threshold را بین همه مقادیر دو به دو امتحان می کنیم. و برای هر حالت precision و recall را حساب می کنیم. (داخل تصویر 5 آبجکت وجود داشته است)

Rank	score	correct_50	precision	recall
1	0.96	TRUE	1.00	0.20
2	0.89	FALSE	0.50	0.20
3	0.84	TRUE	0.67	0.40
4	0.79	FALSE	0.50	0.40
5	0.74	TRUE	0.60	0.60
6	0.47	FALSE	0.50	0.60
7	0.39	FALSE	0.43	0.60
8	0.29	FALSE	0.38	0.60

- 3- طبق نقاط مختلفی که داریم، منحنی precision بر حسب recall را رسم کرده، و سطح زیر این منحنی Average Precision در نظر گرفته می شود.



$$AP_{50} = \left( \frac{(0.5 + 0.67) \times 0.2}{2} \right) + \left( \frac{(0.5 + 0.6) \times 0.2}{2} \right) = \left( \frac{2.27 \times 0.2}{2} \right) = 0.227$$

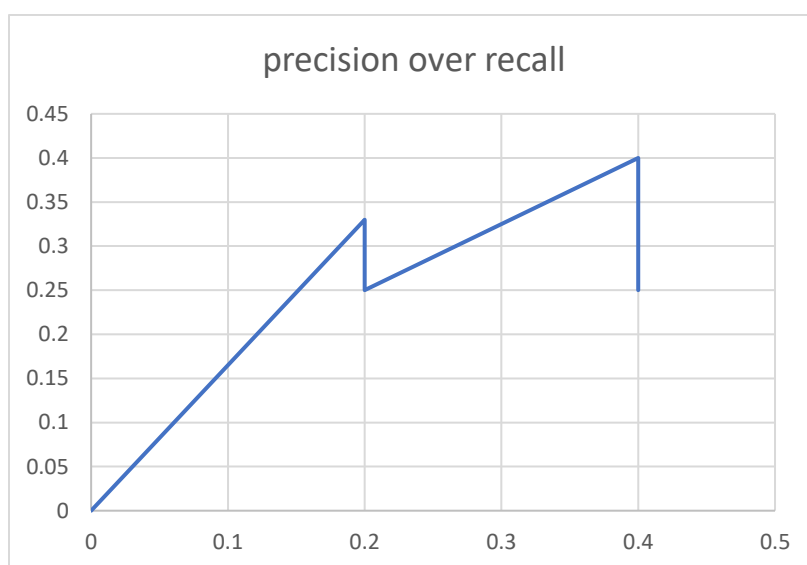
$$AP_{50} = 0.227$$

### محاسبه $AP_{75}$ :

- 1- تشخیص ها را بر اساس امتیاز آنها مرتب میکنیم.
- 2- threshold را بین همه مقادیر دو به دو امتحان می کنیم. و برای هر حالت precision و recall را حساب می کنیم. (داخل تصویر 5 آبجکت وجود داشته است)

Rank	score	correct_75	precision	recall
1	0.96	FALSE	0.00	0.00
2	0.89	FALSE	0.00	0.00
3	0.84	TRUE	0.33	0.20
4	0.79	FALSE	0.25	0.20
5	0.74	TRUE	0.40	0.40
6	0.47	FALSE	0.33	0.40
7	0.39	FALSE	0.29	0.40
8	0.29	FALSE	0.25	0.40

- 3- طبق نقاط مختلفی که داریم، منحنی precision بر حسب recall را رسم کرده، و سطح زیر این منحنی Average Precision در نظر گرفته می شود.



$$AP_{75} = \left( \frac{0.33 \times 0.2}{2} \right) + \left( \frac{(0.25 + 0.4) \times 0.2}{2} \right) = \left( \frac{0.98 \times 0.2}{2} \right) = 0.098$$

$$AP_{75} = 0.098$$