

## ۱ شرح مسئله

در این قسمت به بررسی عوامل موثر بر یادگیری مدل می‌پردازیم که هرکدام تاثیر متفاوتی بر روی مدل دارند. هدف ما در این بخش کم کردن خطا از طریق کاهش بایاس و واریانس است که با استفاده از داده *validation* می‌توان بین حالات مختلف برای مدل بهترین را انتخاب کرد. انتخاب پارامترها از این طریق در قسمت‌های مختلفی مانند *Regularization* و *Optimization* انجام می‌شود. همین‌طور به بررسی داده‌هایی که برای آموزش مدل استفاده می‌شود می‌پردازیم که چگونه باعث بهبود عملکرد و دقت مدل شده و رابطه آن با پیچیدگی مدل به چه صورت است.

## ۲ *Regularization*

وقتی فضای فرضیه پیچیده‌ای داریم و با استفاده از *regularization* باعث محدود کردن زیرفضای جست و جو در حین فرآیند یادگیری می‌شود که از این طریق از *overfit* شدن جلوگیری می‌کند. برای استفاده از این روش انواع مختلفی وجود دارد که در آن‌ها به اندازه وزن‌ها یا توان دو آن‌ها یک ضریب اختصاص می‌دهیم. هر چه این ضریب بیشتر باشد اثر قسمت مربوط به *regularization* نیز بیشتر می‌شود و فضای جست و جوی ما محدودتر می‌گردد.

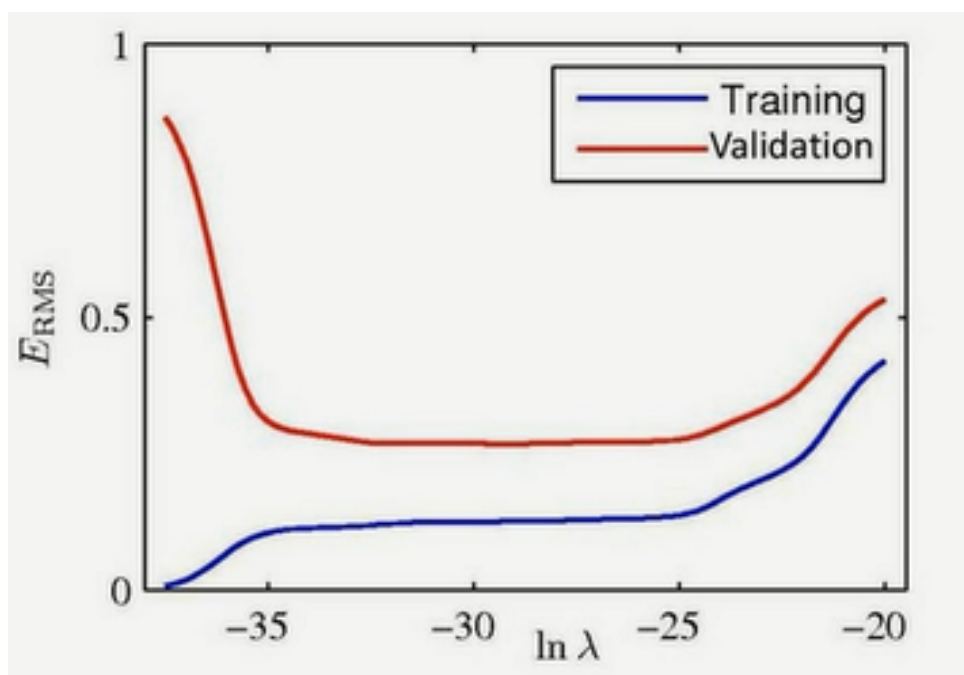
## ۳ *Optimization*

در فرآیند بهینه‌سازی پارامترهای متفاوتی دخیل هستند، مانند نقطه شروع، نرخ یادگیری و ... ممکن است در طی فرآیند یادگیری، مدل ما در یک *local optimum* قرار بگیرد که در نتیجه با ادامه دادن فرآیند مدل ما یادگیری بیشتری نخواهد داشت و این در حالی است که یک نقطه *global optimum* داریم که دقت بهتری به ما داده و در *generalization* بهتر عمل می‌کند.

از عوامل موثر در رسیدن به *global optimum* می‌توان به انتخاب نقطه شروع مناسب اشاره کرد. نقطه شروعی که مقدار *loss* آن کمتر باشد به احتمال بیشتری به *global optimum* می‌رسد. راهکار موجود دیگر تغییر میزان نرخ یادگیری است که با جلو رفتن فرآیند یادگیری میزان آن کم شده تا با دقت بهتری به وضعیت مناسب برای مدل از نظر *generalization* برسیم.

## ۴ Hyperparameters

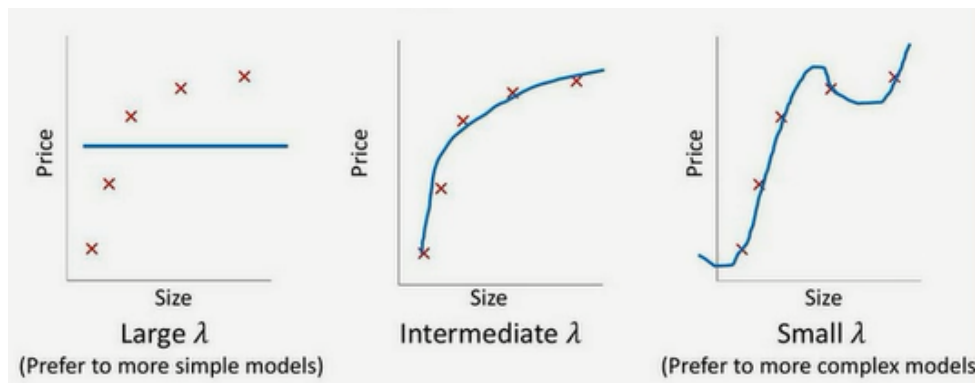
یک سری دیگر از پارامتر هایی که در فرآیند یادگیری موثر هستند، *hyperparameter* ها هستند که مقادیر گسسته ای به خود گرفته که تنظیم آن ها به یادگیری بهتر کمک می‌کند. این پارامتر ها می‌توانند در دامنه های متفاوت مورد بررسی قرار بگیرند که بفهمیم در کدام دامنه از مقادیر یادگیری مدل ما بهینه خواهد بود. از راه هایی که برای پیدا کردن بهترین تنظیمات برای *hyperparameter* ها استفاده می‌شود، *crossvalidation* است. این روش به این صورت است که داده را به یک سری قسمت تقسیم کرده و همه قسمت ها بجز یک قسمت از داده های *train* را جدا کرده و آن قسمت از داده را *validation* می‌نامیم. سپس مدل را با استفاده از داده *train* آموزش داده و بر روی داده *validation* تست می‌کنیم. در ادامه این فرآیند را روی دیگر قسمت های ایجاد شده انجام داده تا هر قسمت یک بار به عنوان *validation* استفاده شود، سپس از دقت مدل روی این حالات مافاوت میانگین میگیریم. با ثابت گرفتن نقطه شروع و انجام *validation* برای یک پارامتر و تست مقادیر متفاوت آن، میتوان مسئله *biasvariancetradeoff* را مورد بررسی قرار داد که با زیاد کردن پارامتر مورد نظر واریانس به سرعت کم می‌شود.



## Example ۵

در این بخش به بررسی چند مثال در زمینه بهبود عملکرد مدل در زمینه یادگیری می‌پردازیم:

### Regularization Parameters

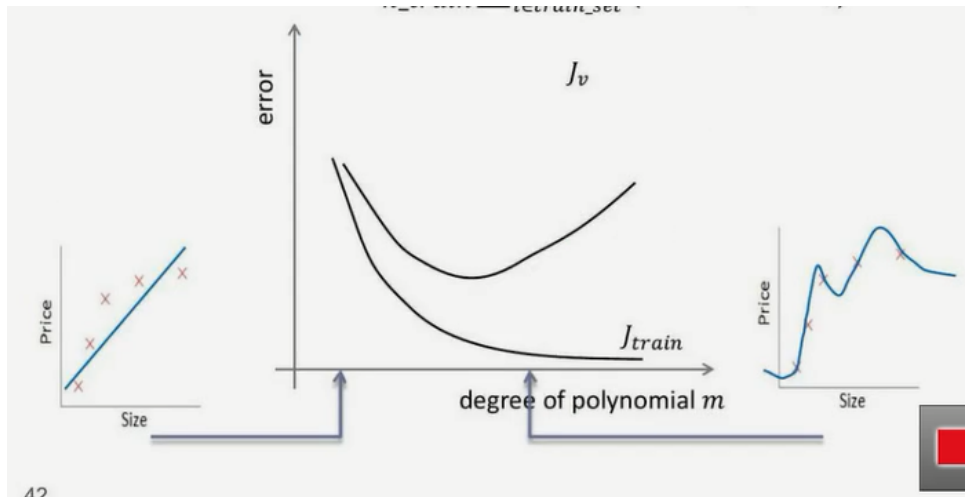


در تصویر دوم تاثیر پارامتر  $\lambda$  را بر روی عملکرد مدل مشاهده میکنید.  $\lambda$  در واقع ضریب اندازه وزن مدل در اضافه کردن *generalization* به مدل است که اگر بسیار بزرگ باشد باعث می‌شود که وزن‌ها نتوانند زیاد شوند و دامنه انتخاب وزن‌ها محدود شده و مدل ما *underfit* می‌شود. اگر مقدار  $\lambda$  را خیلی کوچک در نظر بگیریم باعث می‌شود که وزن‌ها بتوانند مقادیر وزن‌ها بتوانند در دامنه خیلی بزرگ‌تری انتخاب شوند که باعث این می‌شود که مدل بر روی داده ورودی ما *overfit* بشود. در شکل میانی تاثیر انتخاب یک  $\lambda$  مناسب دیده میشود که در آن روند کلی داده در مدل دنبال شده و دچار *overfit* نشده.

### Model Complexity

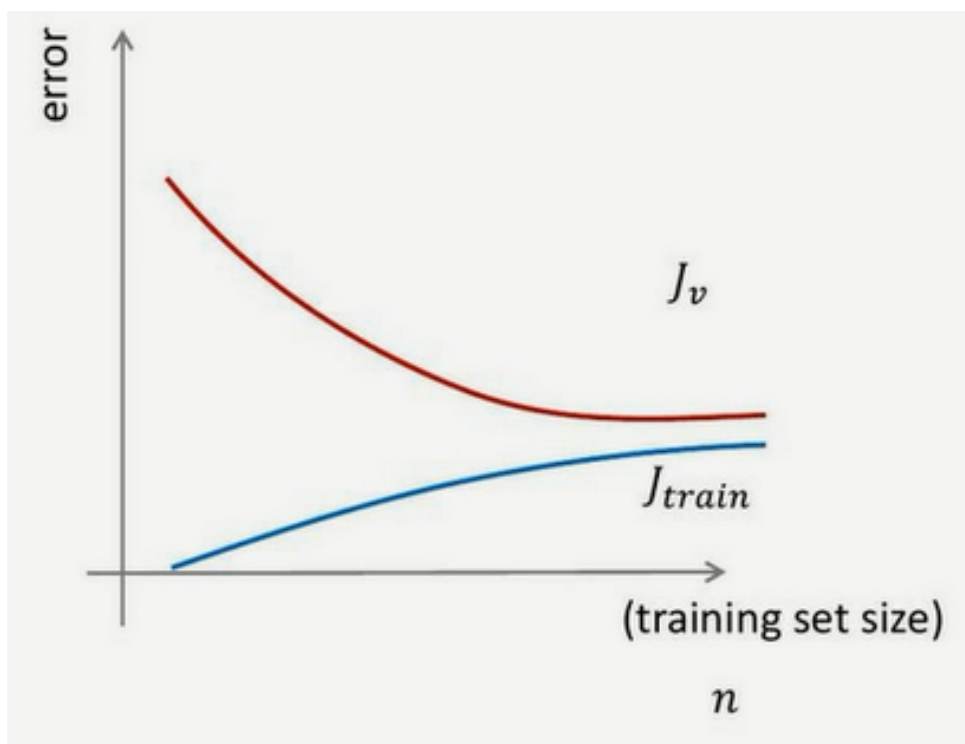
در ادامه پارامتر دیگری که در یادگیری مدل موثر است را بررسی می‌کنیم که آن میزان پیچیدگی مدل است. در این مثال میزان پیچیدگی را با درجه چند جمله ای که می‌خواهیم آموزش دهیم مدل می‌کنیم زیرا هرچه درجه چند جمله ای بیشتر باشد مدل نیز پیچیده تر خواهد بود.

در این جا هرچه درجه مدل را بالاتر می‌بریم، مدل روی داده های آموزشی عملکرد بهتری خواهد داشت و بایاس مدل کم میشود که در نتیجه عملکرد روی داده *validation* نیز بهتر می‌شود. از یک نقطه به بعد واریانس زیاد شده و مدل شروع به *overfit* کردن می‌کند. در این جا هرچه پیچیدگی مدل زیاد شود باعث می‌شود که عملکرد مدل روی داده های تست کم شده و *generalization* به خوبی انجام نشود.



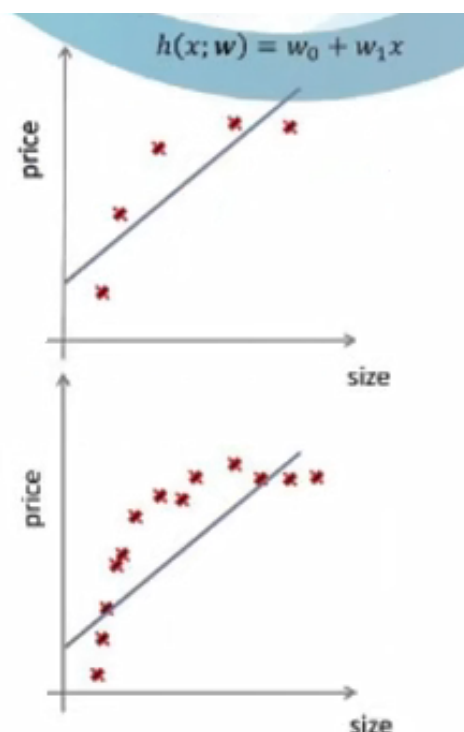
## TrainingData

از عوامل موثر دیگر داده ای است که برای آموزش مدل استفاده میکنیم. در زیر نشان داده شده که با افزایش تعداد نمونه ها در داده آموزشی عملکرد مدل بهبود می یابد که به این دلیل است که بایاس ثابت می ماند زیرا فضای داده ها با افزایش نمونه های ثابت است و از طرفی میزان واریان کاهش می یابد که دقت را برای *validation* بهتر می کند.

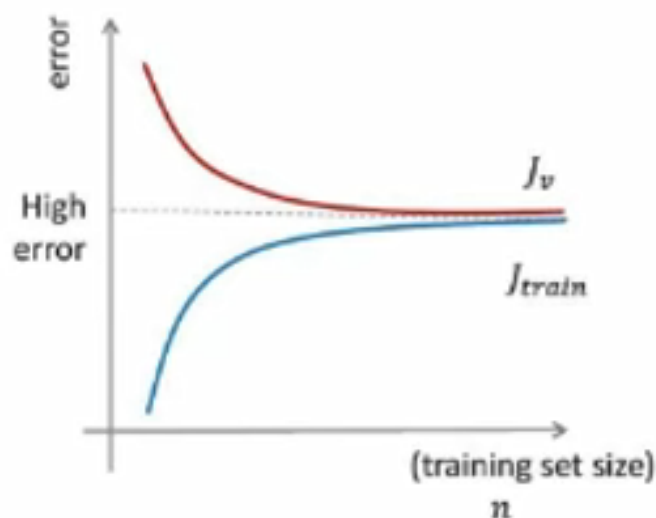


این دو نمودار با زیاد شدن داده آموزش به هم نزدیک می شوند که اگر تعداد نمونه ها در داده آموزشی زیاد شود، دقت مدل بر روی داده آموزش می تواند تخمینی از عملکرد آن روی داده های *validation* باشد.

البته زیاد شدن تعداد نمونه ها به شرطی می تواند تاثیر گذار باشد که مدل نیز پیچیدگی لازم برای تحلیل داده ها را داشته باشد و اگر مدل ما برای تحلیل فضا ساده باشد، زیاد کردن نمونه های داده برای آموزش مدل تاثیری نخواهد داشت:



در تصویر زیر می بینیم که مدل توانایی پردازش داده های بیشتر را ندارد و نمیتواند عملکرد بهتری نسبت به حالتی که داده کمتری در اختیار دارد داشته باشد. در واقع می توان گفت که زیاد شدن تعداد داده ها برای آموزش مدل تا یک حد می تواند به بهبود مدل کمک کند:



در تصویر بالا دیده می‌شود که بعد از این که تعداد نمونه‌ها را زیاد کردیم، دقت روی داده آموزش و *validation* همگرا می‌شوند و بعد از آن زیاد کردن تعداد نمونه‌های تأثیری در عملکرد مدل نخواهد داشت. همچنین می‌توان اثر تعداد نمونه‌ها را بر روی مدل‌های مختلف دید، هرچه مدل پیچیده‌تر باشد و فضای پیچیده‌تری را یادبگیرد، اضافه کردن یک نمونه باعث تغییر در عملکرد مدل می‌شود. این مورد را می‌توان در مثال بعدی دید که در آن گپ بین نمودارها و شیب متفاوت نشان دهنده پیچیده بودن فضا دارد که به زیاد کردن داده خیلی زود به همگرایی نمیرسیم و تغییرات عملکرد با اضافه کردن داده جدید بیشتر است زیرا با فضای پیچیده‌تر مدل‌های متفاوت بیشتری می‌توان استفاده کرد که به همین دلیل اضافه کردن نمونه جدید باعث می‌شود مدل جدید در فضای حالت، عملکرد بهتری داشته باشد و *generalization* بهتر انجام شود.