

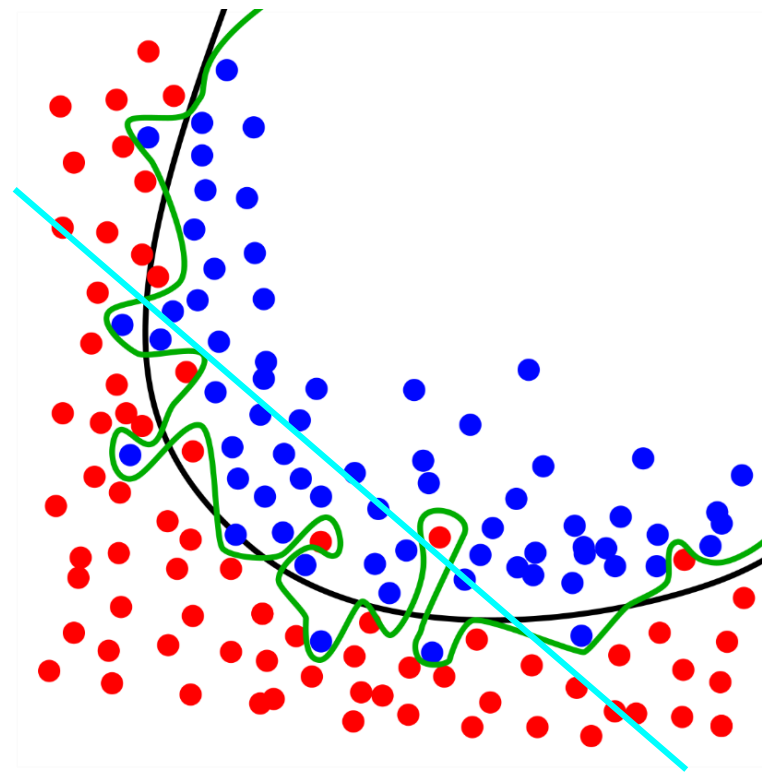
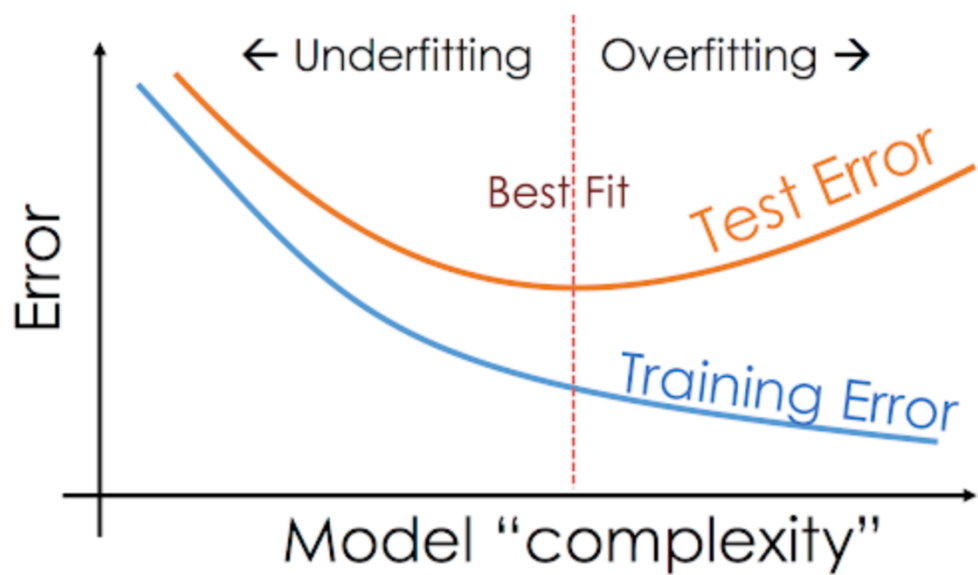
رسالة محمد

منظم سازی

Regularization

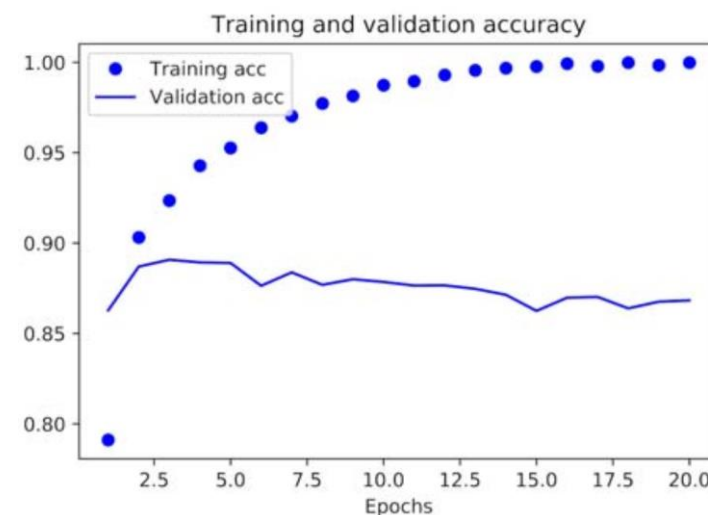
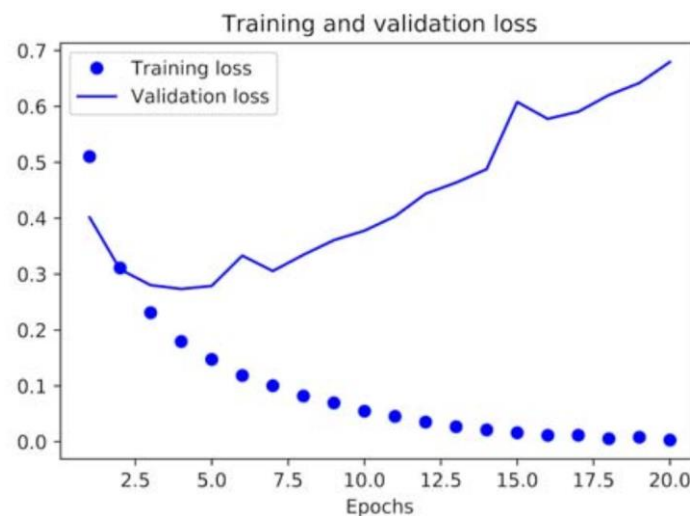
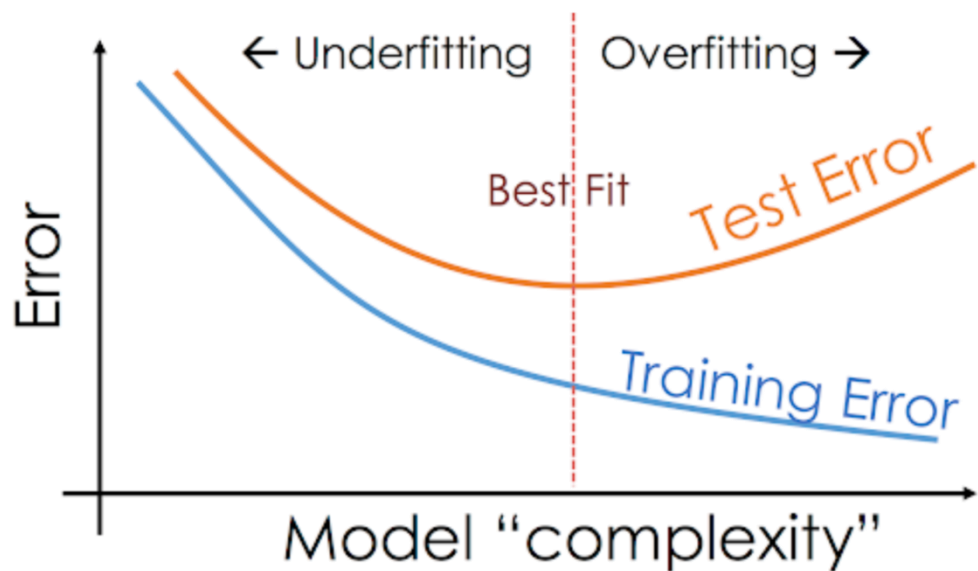
منظم سازی

- یک مسئله اساسی در یادگیری ماشین این است که چگونه الگوریتمی بسازیم که نه تنها بر روی داده‌های آموزشی بلکه برای ورودی‌های جدید نیز به خوبی عمل کند



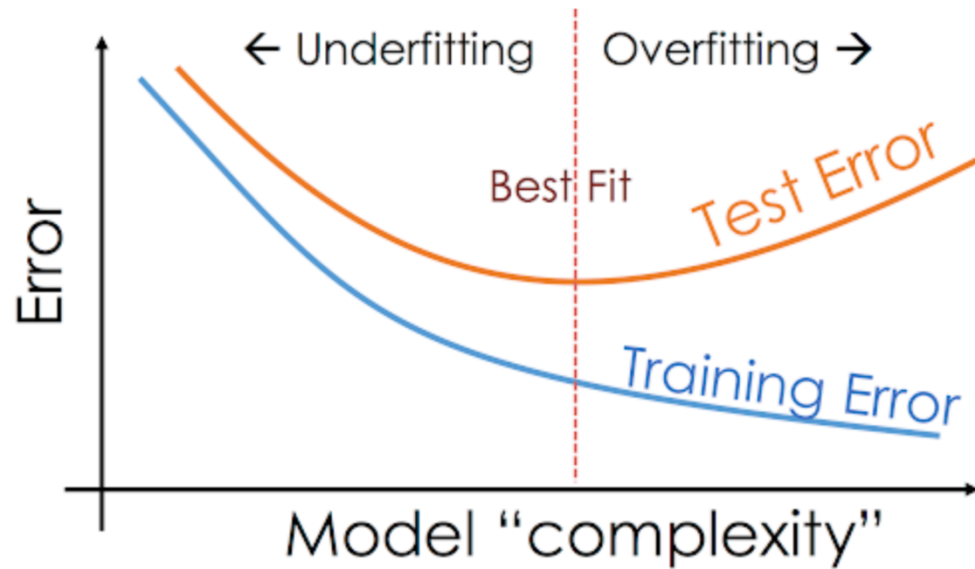
منظم سازی

- عملکرد مدل‌ها بر روی داده‌های اعتبارسنجی که در آموزش مدل استفاده نشده‌اند معمولاً پس از چند دوره به اوج خود می‌رسد و سپس شروع به تنزل می‌کند - مدل به سرعت شروع به overfit شدن بر روی داده‌های آموزشی می‌کند
- یادگیری نحوه مقابله با overfitting یکی از نکات بسیار کلیدی در یادگیری ماشین است



بهینه‌سازی و تعمیم‌دهی

- بهینه‌سازی به تعیین پارامترهای مدل برای به دست آوردن بهترین عملکرد ممکن در داده‌های آموزشی (یادگیری در ML) اشاره دارد
- تعمیم‌دهی به نحوه عملکرد مناسب مدل آموزش دیده بر روی داده‌هایی که تا کنون مشاهده نکرده است اشاره دارد

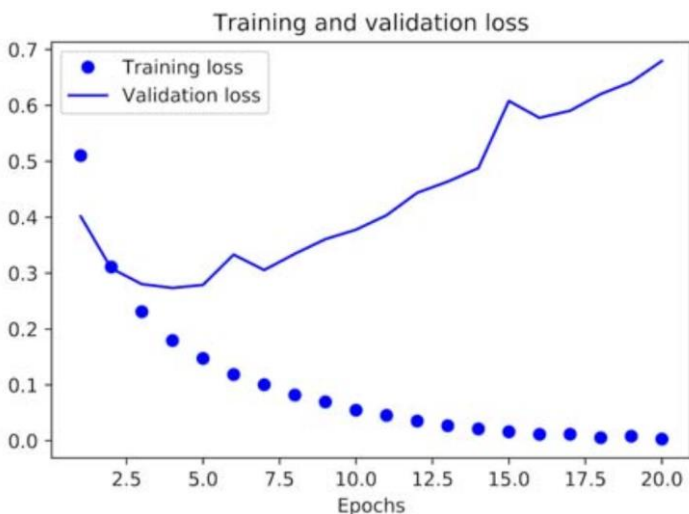


- هدف دستیابی به تعمیم‌دهی مناسب است

- اما کنترلی بر روی تعمیم‌دهی نداریم!
- تنها می‌توانیم بر اساس داده‌های آموزشی پارامترهای مدل را تعیین کنیم

بهینه‌سازی و تعمیم‌دهی

- در ابتدای آموزش، بهینه‌سازی و تعمیم‌دهی با هم کاملاً مرتبط هستند
 - به مدل گفته می‌شود underfit است
 - شبکه هنوز تمام الگوهای مرتبط با مسئله مورد نظر در داده‌های آموزشی را یاد نگرفته است
- پس از چند تکرار، بهبود تعمیم‌دهی متوقف می‌شود و سپس شروع به تنزل می‌کند
 - مدل شروع به overfit شدن می‌کند
 - الگوهایی را می‌آموزد که مخصوص داده‌های آموزشی است اما ارتباط درستی با مسئله مورد نظر ندارد و گمراه‌کننده است



Cat



Dog

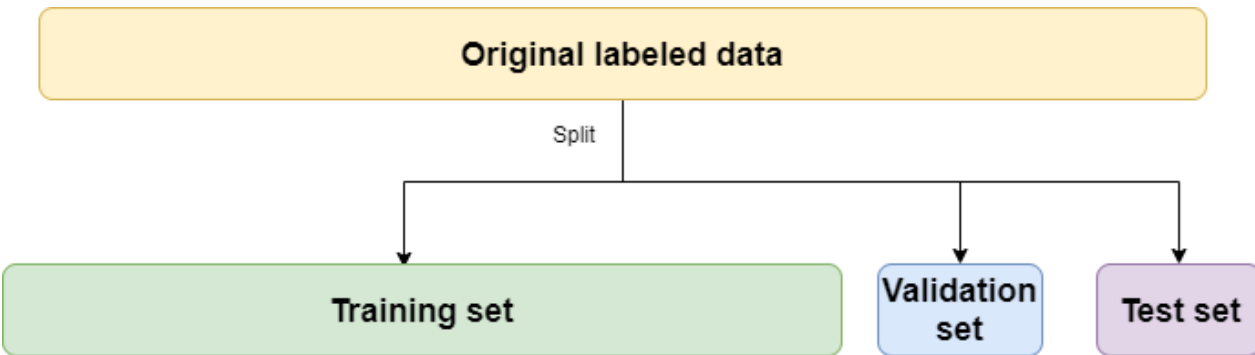


کاهش ابعاد شبکه

- اگر یک شبکه فقط بتواند تعداد کمی از الگوها را حفظ کند، فرآیند بهینه‌سازی آن را مجبور می‌کند تا بر برجسته‌ترین الگوها تمرکز کند، که شانس بیشتری برای تعمیم خوب دارند
- به صورت شهودی، یک مدل با پارامترهای بیشتر دارای ظرفیت حفظ بیشتری است و بنابراین می‌تواند به راحتی یک نگاشت کامل را بین نمونه‌های آموزشی و اهداف آنها بیاموزد بدون آنکه تعمیم‌دهی داشته باشد
- اگر شبکه منابع محدودی برای حفظ داشته باشد، نمی‌تواند به راحتی چنین نگاشتی را یاد بگیرد
 - شبکه مجبور خواهد بود بازنمایی‌های فشرده‌ای را یاد بگیرد که دارای قدرت پیش‌بینی بالایی باشند
- نیاز است تا تعادلی میان ظرفیت بیش از حد زیاد و بیش از حد کم برقرار شود

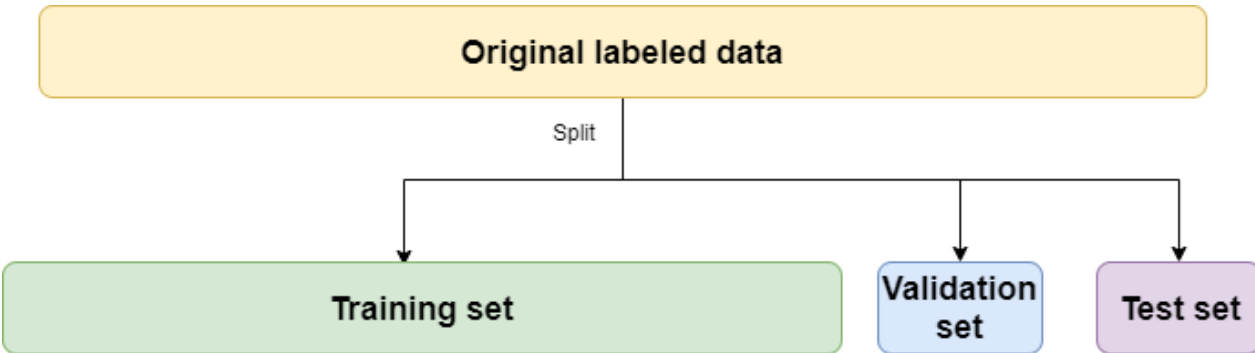
ارزیابی مدل‌های یادگیری ماشین

- در یادگیری ماشین، هدف دستیابی به مدل‌هایی است که بتوانند تعمیم‌دهی خوبی داشته باشند و بر روی داده‌هایی که قبلاً ندیده‌اند به خوبی عمل کنند
- بسیار مهم است که بتوانیم قدرت تعمیم‌دهی مدل را به طور قابل اعتماد اندازه‌گیری کنیم
- ارزیابی یک مدل با تقسیم داده‌های موجود به سه بخش انجام می‌شود:
 - آموزشی (training)
 - اعتبارسنجی (validation)
 - آزمون (test)



ارزیابی مدل‌های یادگیری ماشین

- آموزش بر روی داده‌های آموزشی انجام می‌شود (برای بهینه‌سازی پارامترها)
- مدل بر روی داده‌های اعتبارسنجی ارزیابی می‌شود (برای بهینه‌سازی ابرپارامترها)
 - دلیل این است که توسعه یک مدل همیشه شامل تنظیم پیکربندی آن است
 - به عنوان نمونه، انتخاب تعداد لایه‌ها و ابعاد هر لایه
 - تنظیم ابرپارامترهای شبکه یک نوع یادگیری است
 - با وجود آنکه از داده‌های اعتبارسنجی برای آموزش پارامترهای شبکه هیچ استفاده‌ای نمی‌شود، اما این امکان وجود دارد که مدل به داده‌های اعتبارسنجی overfit شود
- مدل نهایی را فقط یک بار بر روی داده‌های آزمون ارزیابی می‌کنیم



اعتبارسنجی Hold-out

Listing 4.1 Hold-out validation

```
num_validation_samples = 10000
np.random.shuffle(data)
validation_data = data[:num_validation_samples]
data = data[num_validation_samples:]
training_data = data[:]

model = get_model()
model.train(training_data)
validation_score = model.evaluate(validation_data)

# At this point you can tune your model,
# retrain it, evaluate it, tune it again...

model = get_model()
model.train(np.concatenate([training_data,
                             validation_data]))
test_score = model.evaluate(test_data)
```

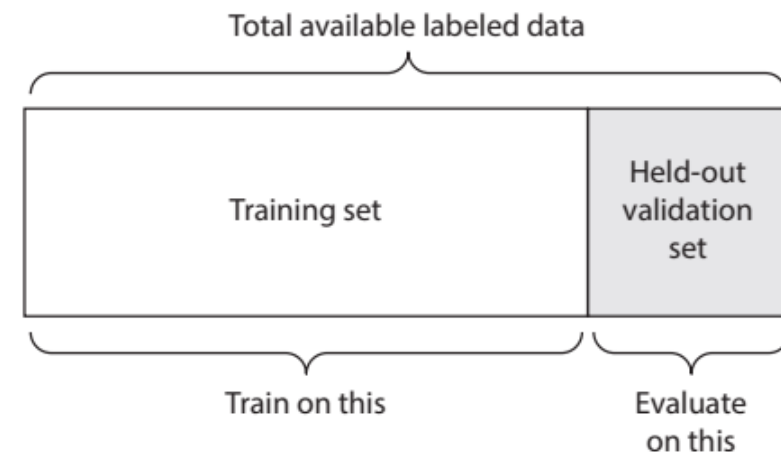
Shuffling the data is usually appropriate.

Defines the validation set

Defines the training set

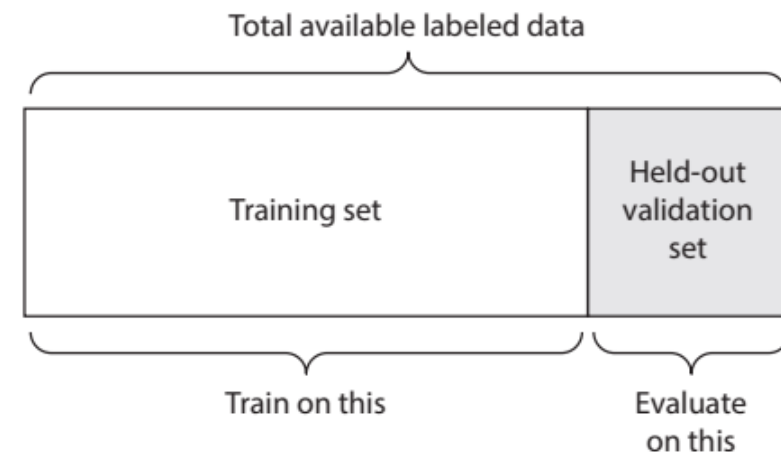
Trains a model on the training data, and evaluates it on the validation data

Once you've tuned your hyperparameters, it's common to train your final model from scratch on all non-test data available.



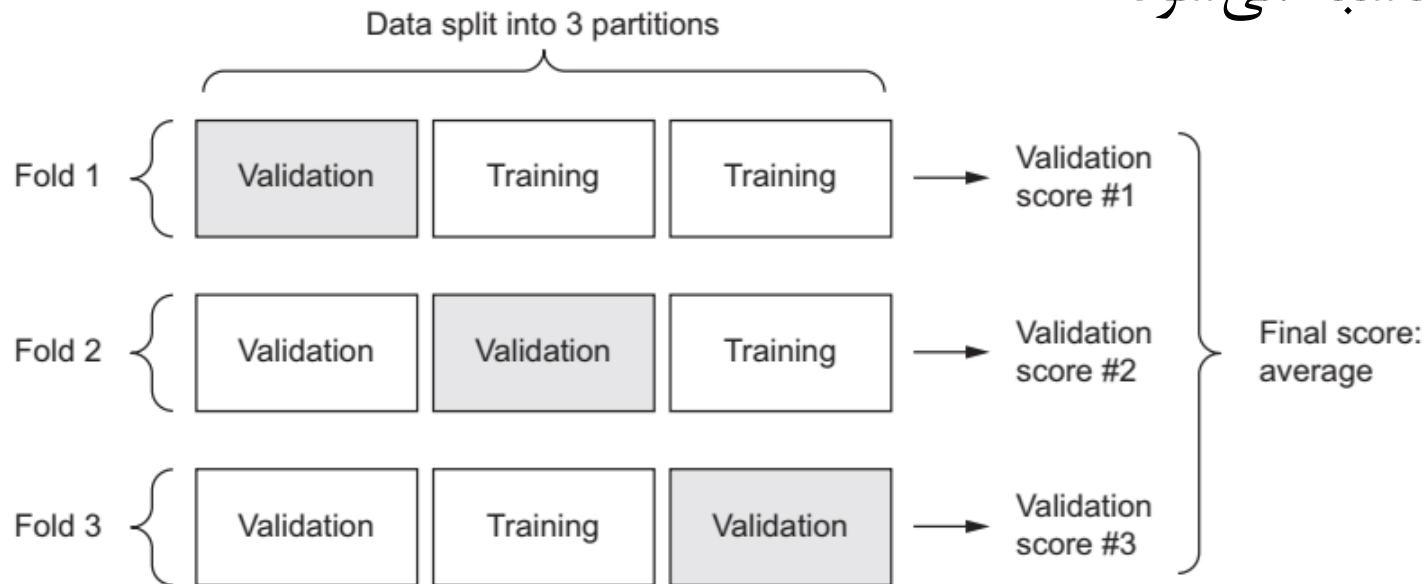
اعتبارسنجی Hold-out

- اگر داده‌های کمی در دسترس باشد، مجموعه‌های اعتبارسنجی و آزمون ممکن است نمونه‌های بسیار کمی داشته باشند و نتوانند از نظر آماری نماینده خوبی برای داده‌های مورد نظر باشند
- تقسیم‌بندی تصادفی داده‌ها باعث خواهد شد نتایج بسیار متفاوتی از عملکرد مدل بدست بیاید



اعتبارسنجی متقابل K-fold

- مجموعه داده را به K بخش با اندازه مشابه تقسیم می‌کنیم
- برای هر بخش i ، یک مدل با استفاده از داده‌های $K - 1$ بخش دیگر آموزش می‌بیند و بر روی داده‌های بخش i ارزیابی می‌شود
- امتیاز نهایی با استفاده از این K آزمایش محاسبه می‌شود



اعتبارسنجی متقابل K-fold

Listing 4.2 K-fold cross-validation

```
k = 4
num_validation_samples = len(data) // k

np.random.shuffle(data)

validation_scores = []
for fold in range(k):
    validation_data = data[num_validation_samples * fold:
                           num_validation_samples * (fold + 1)]
    training_data = data[:num_validation_samples * fold] +
                    data[num_validation_samples * (fold + 1):]

    model = get_model()
    model.train(training_data)
    validation_score = model.evaluate(validation_data)
    validation_scores.append(validation_score)

validation_score = np.average(validation_scores)

model = get_model()
model.train(data)
test_score = model.evaluate(test_data)
```

Selects the validation-data partition

Uses the remainder of the data as training data. Note that the + operator is list concatenation, not summation.

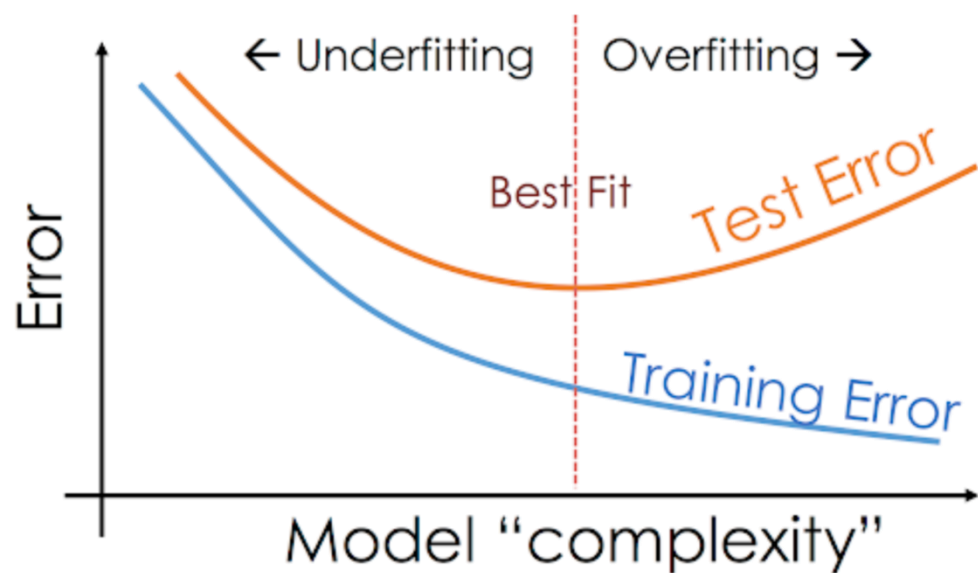
Creates a brand-new instance of the model (untrained)

Validation score: average of the validation scores of the k folds

Trains the final model on all non-test data available

منظم‌سازی

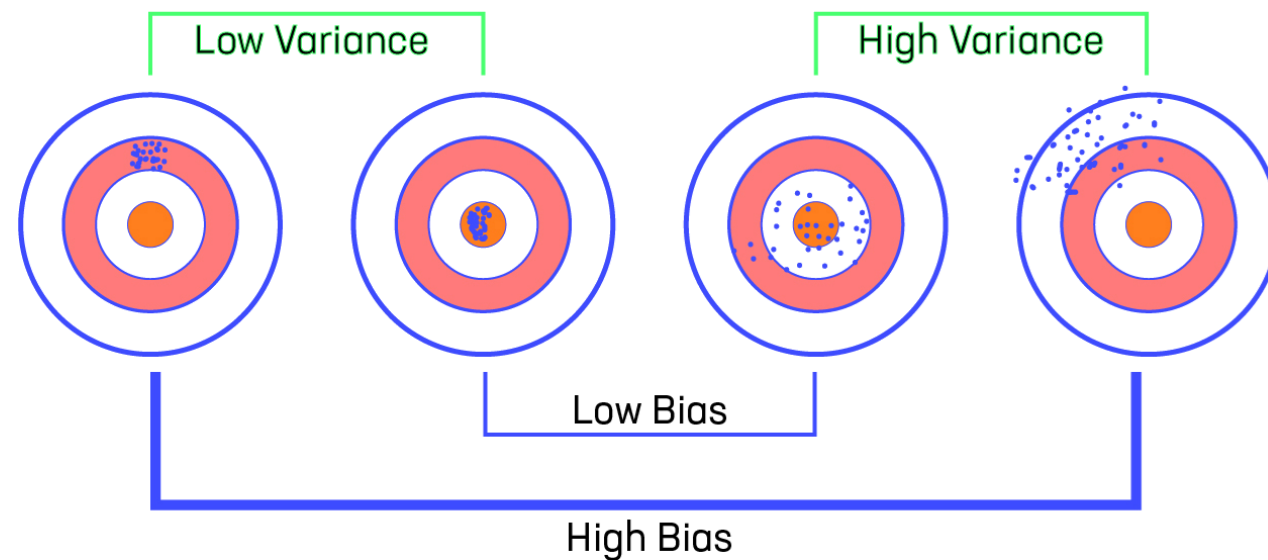
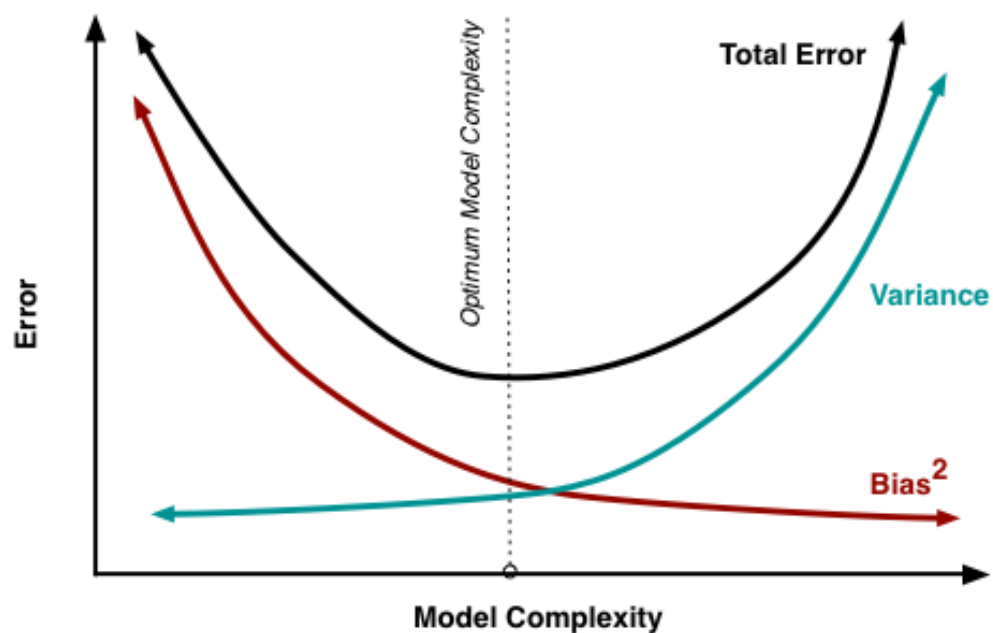
- یک مسئله اساسی در یادگیری ماشین این است که چگونه الگوریتمی بسازیم که نه تنها بر روی داده‌های آموزشی بلکه برای ورودی‌های جدید نیز به خوبی عمل کند
- بسیاری از استراتژی‌های یادگیری ماشین به دنبال کاهش خطای آزمون هستند حتی اگر به افزایش خطای آموزش منجر شود



- به این استراتژی‌ها regularization (منظم‌سازی) گفته می‌شود

بایاس و واریانس

- بایاس اختلاف میان میانگین پیش‌بینی‌های مدل ما نسبت به مقادیر صحیح است
- واریانس میزان تغییرات پیش‌بینی مدل را نشان می‌دهد



بایاس و واریانس

- خطای بایاس خطایی است که از مفروضات اشتباه در الگوریتم یادگیری ناشی می‌شود
- خطای واریانس خطایی است که به دلیل حساسیت بالا نسبت به تغییرات کوچک در مجموعه داده ایجاد می‌شود

