Artificial Neural Networks
Lecture Notes

Prof. Dr. Sen Cheng

Winter Semester 2019/20

4 Model selection/ Regularization

4.1 Generalization

The goal of curve fitting is to minimze the loss function. Most worry about *underfitting*, i.e., the model is not able to describe the data very well (Fig. 1).

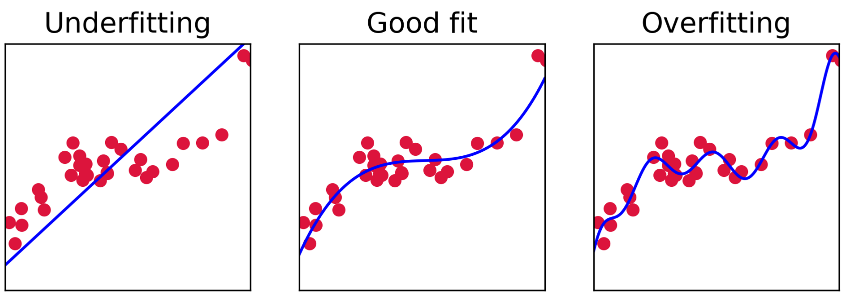

Figure 1: Example of underfitting and overfitting.

However, describing the data well that we already have is only a means to an end. The fit can be too good (*overfitting*) and start capturing the noise. The actual goal is to use the model to make predictions about future data. So what we actually want is that the model generalizes to data that the model was not trained on. If the model captures the noise in the *training data*, it will not generalize well to new data, also called *test data* (Fig. 2).

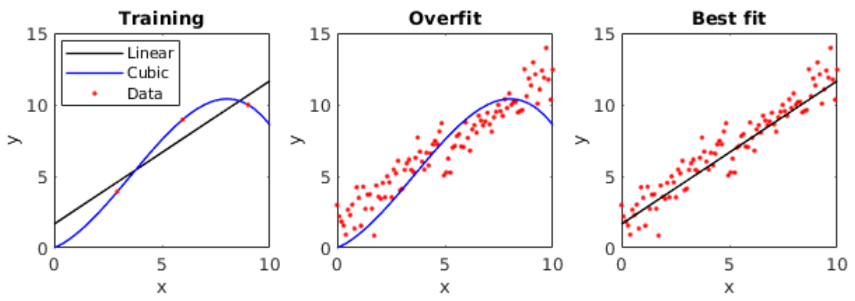

Figure 2: Overfitting leads to bad predicitons. A cubic model (blue) perfectly fits the training data (left), but it fails to generalize to new data (middle), which is better captured by the linear fit (right). Note that a cubic model has more parameters than the data points available for training in this case.

It is, therefore, a good practice to split the available dataset into *training data* and *test data*. The former is used to determine the parameters of the model in an optimization procedure, the latter is used to evaluate how well the model is doing.

4.2 Bias-variance tradeoff

When evaluating a fitted function on test data, there are three sources of error. First, the *bias* is the systematic error that the estimator $\hat{f}$ makes in describing the data, i.e., $\text{bias}[\hat{f}] = f - \mathrm{E}\left[\hat{f}\right]$. It is the result of a mismatch between the fitted function and the function underlying the data. Second, the *variance* is the fluctuation of the estimator from one dataset to another, i.e., $\text{var}[\hat{f}] = \mathrm{E}\left[(\mathrm{E}\left[\hat{f}\right] - \hat{f})^2\right]$. It is the result of the sensitivity of the estimator to small fluctuations in the data. Third, the *irreducible error* is the error inherent in the data, i.e., $\text{var}[y] = \sigma^2$. It turns out that the error of any estimator on any dataset is simply a sum of these three errors.

$$\text{error} = \mathrm{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \text{bias}[\hat{f}]^2 + \text{var}[y] + \text{var}[\hat{f}] \tag{1}$$

Proof:

Since $y = f(x) + \varepsilon$, the error can be expanded

$$\mathrm{E}\left[\left(f + \varepsilon - \hat{f}\right)^2\right] \tag{2a}$$
$$= \mathrm{E}\left[\left(f - \mathrm{E}\left[\hat{f}\right] + \varepsilon + \mathrm{E}\left[\hat{f}\right] - \hat{f}\right)^2\right] \tag{2b}$$
$$= \mathrm{E}\left[(f - \mathrm{E}\left[\hat{f}\right])^2 + \varepsilon^2 + (\mathrm{E}\left[\hat{f}\right] - \hat{f})^2 + 2(f - \mathrm{E}\left[\hat{f}\right])\varepsilon + 2(f - \mathrm{E}\left[\hat{f}\right])(\mathrm{E}\left[\hat{f}\right] - \hat{f}) + 2\varepsilon(\mathrm{E}\left[\hat{f}\right] - \hat{f})\right] \tag{2c}$$

The function f and the expectation value of a random variable $\mathrm{E}\left[\cdot\right]$ are deterministic, ε is independent of $\hat{f}$.

$$= (f - \mathrm{E}\left[\hat{f}\right])^2 + \mathrm{E}\left[\varepsilon^2\right] + \mathrm{E}\left[(\mathrm{E}\left[\hat{f}\right] - \hat{f})^2\right] + 2(f - \mathrm{E}\left[\hat{f}\right])\underbrace{\mathrm{E}\left[\varepsilon\right]}_{=0} + 2(f - \mathrm{E}\left[\hat{f}\right])\underbrace{\mathrm{E}\left[\mathrm{E}\left[\hat{f}\right] - \hat{f}\right]}_{\mathrm{E}[\hat{f}] - \mathrm{E}[\hat{f}] = 0} + 2\underbrace{\mathrm{E}\left[\varepsilon\right]}_{=0}\mathrm{E}\left[\mathrm{E}\left[\hat{f}\right] - \hat{f}\right] \tag{2d}$$
$$= (f - \mathrm{E}\left[\hat{f}\right])^2 + \mathrm{E}\left[\varepsilon^2\right] + \mathrm{E}\left[(\mathrm{E}\left[\hat{f}\right] - \hat{f})^2\right] \tag{2e}$$
$$= \text{bias}[\hat{f}]^2 + \sigma^2 + \text{var}[\hat{f}] \tag{2f}$$

Models of low complexity (which often, but by far not always, correlates with the number of parameters) tend to underfit, and model of high complexity tend to overfit (Fig. 3). Since the total error is the sum of bias and variance, the two can be traded-off. The optimum would be where the total in minimal.

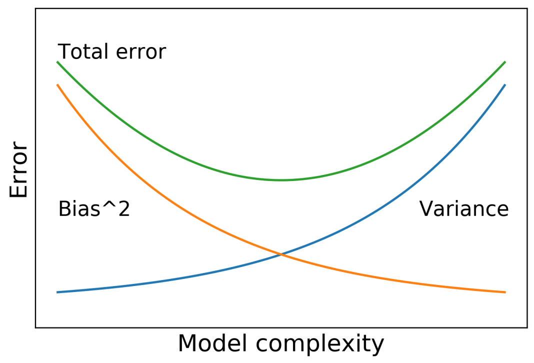

Figure 3: Bias-variance tradeoff.

4.3 Model selection

The selection of the model class should be based on the data. Fig. 3 tells us what model complexity would be optimal, but unfortunately it is a theoretical result assuming we knew the correct function and noise. In pratice, both are unknown. So we can only estimate the bias by using the training error and the variance using the generalization error.

In addition, most model classes have hyperparameters that need to be tuned. In the end, we choose the model class and hyperparameter that yield the lowest generalization error. However, now the generalization error cannot be used to gauge the performance of the model anymore, since we optimized the model and its hyperparameters to get as low a generalization error as possible. Therefore, we need to split the dataset into three subsets:

$$\begin{array}{c|c|c} \text{training data} & \text{validation data} & \text{test data} \\ \downarrow & \downarrow & \downarrow \\ \text{fitting} & \text{model selection} & \text{performance} \end{array}$$

The *validation data* is used for model selection and the test data for assessing the performance of the chosen model. Do not tweak the model or its hyperparameters to reduce the test error.

4.4 Cross validation

Often we don't have very large datasets available, so splitting the data into three subsets might not leave enough data. One way to solve this problem is to use *n-fold cross-validation* (Fig. 4). In this method, the data is partitioned into n blocks and the model is trained and tested n times. Each time or 'epoch', one of the blocks is used as the test set and the remaining blocks as the training set. The mean error on the test set across epochs is then used as a performance measure of the model, which can be used to compare it to alternatives.

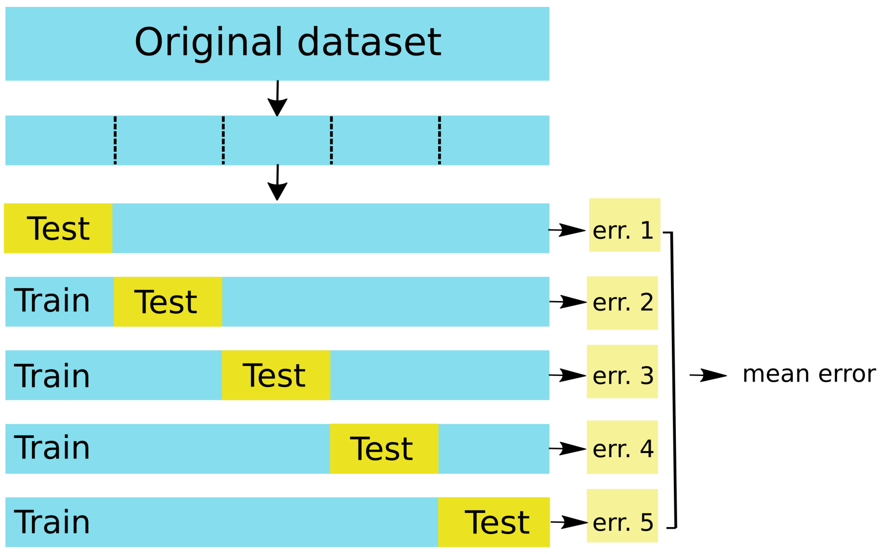

Figure 4: Example of 5-fold cross-validation

4.5 Regularization

Model selection is difficult. It would be nice to learn from the data, which parameters are needed and which are not. This can be accomplished by constraining the parameters during optimization and is called *regularization*. During normal fitting, the optimization is very likely to assign large values to the parameters that are unimportant for describing the data. This counterintuitive result comes about because changing an unimportant parameter doesn't hurt model performance, but can account for some variance caused by noise. We can reduce this behavior, by including the values of the parameters themselves as an additional term in the loss function.

$$\hat{\theta} = \arg\min_{\theta} [L(\theta, X, Y) + \lambda R(\theta)] \tag{3}$$

The l-norm is frequently used for normalization, i.e., $R(\theta) = l_k(\theta)$. The regularizing term in the loss function forces the optimization to make the parameter values as small as possible, unless that would drive up the error in the first term. The hyperparameter λ controls the importance of the regularizing term. The larger λ, the more model complexity is penalized.

When using gradient descent, *early stopping* can also be used for regularization.

4.6 Ridge regression

Ridge regression is linear regression with l_2-norm regularization.

Model class: $Y = X\theta + \varepsilon$
Loss function:

$$L(\theta, X, Y) = (Y - X\theta)^T (Y - X\theta) + \lambda\theta^T\theta \tag{4}$$

Optimization:

$$\begin{aligned} 0 &= \nabla_\theta L(\theta, X, Y) && \text{(5a)} \\ &= \nabla_\theta \left(Y^TY - 2\theta^T X^T Y + \theta^T X^T X\theta + \lambda\theta^T\theta\right) && \text{(5b)} \\ &= -2X^TY + 2X^TX\theta + 2\lambda\theta && \text{(5c)} \\ &= -X^TY + (X^TX + \lambda I)\theta && \text{(5d)} \end{aligned}$$

$$\hat{\theta} = (X^TX + \lambda I)^{-1} X^T Y \tag{6}$$

The predictors need to be standardized (same scale) before performing ridge regression.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{N}\sum_{i=1}^N (x_{ij} - \bar{x}_{\cdot j})^2}} \tag{7}$$

Otherwise the parameters have very different scales and they would contribute very differently to the regularizing term.

Regularization can also help in cases, in which the matrix X^TX is singlular or close to singular, i.e., not invertible or inversion is unstable. This occurs in particular when the colums of X, i.e., the observations, are linearly dependent.

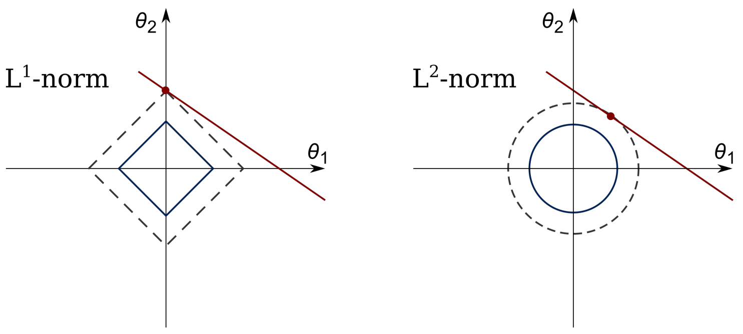

Figure 5: Constraints in ridge and lasso regression. Source (Adapted from): `https://commons.wikimedia.org/wiki/File:L1_and_L2_balls.svg`, User:Nicoguaro. Licensed under Creative Commons BY 4.0.

4.7 Lasso regression

Lasso regression is linear regression with l_1-norm regularization. The acronym "LASSO" stands for Least Absolute Shrinkage and Selection Operator.

Model class: $Y = X\theta + \varepsilon$
Loss function:

$$L(\theta, X, Y) = (Y - X\theta)^T (Y - X\theta) + \lambda \sum_{j=1}^{m} |\theta_j| \tag{8}$$

Optimization: There is no analytical solution for lasso regression. So, numerical methods have to be used to estimate the parameters.

Lasso is more likely than ridge regression to yield parameters that are zero. The resulting models are sparser (Fig. 5).