Deep learning

2.3. Bias-variance dilemma

François Fleuret

https://fleuret.org/dlc/

We can visualize over-fitting for our polynomial regression by generating multiple training sets $\mathscr{D}_1, \ldots, \mathscr{D}_M$, training as many models $f_1, \ldots, f_M$, and computing empirically the mean and standard deviation of the prediction at every point.

As we will see, when the capacity increases, or the regularization decreases, the mean of the predicted value gets right on target, but the prediction varies more across runs.

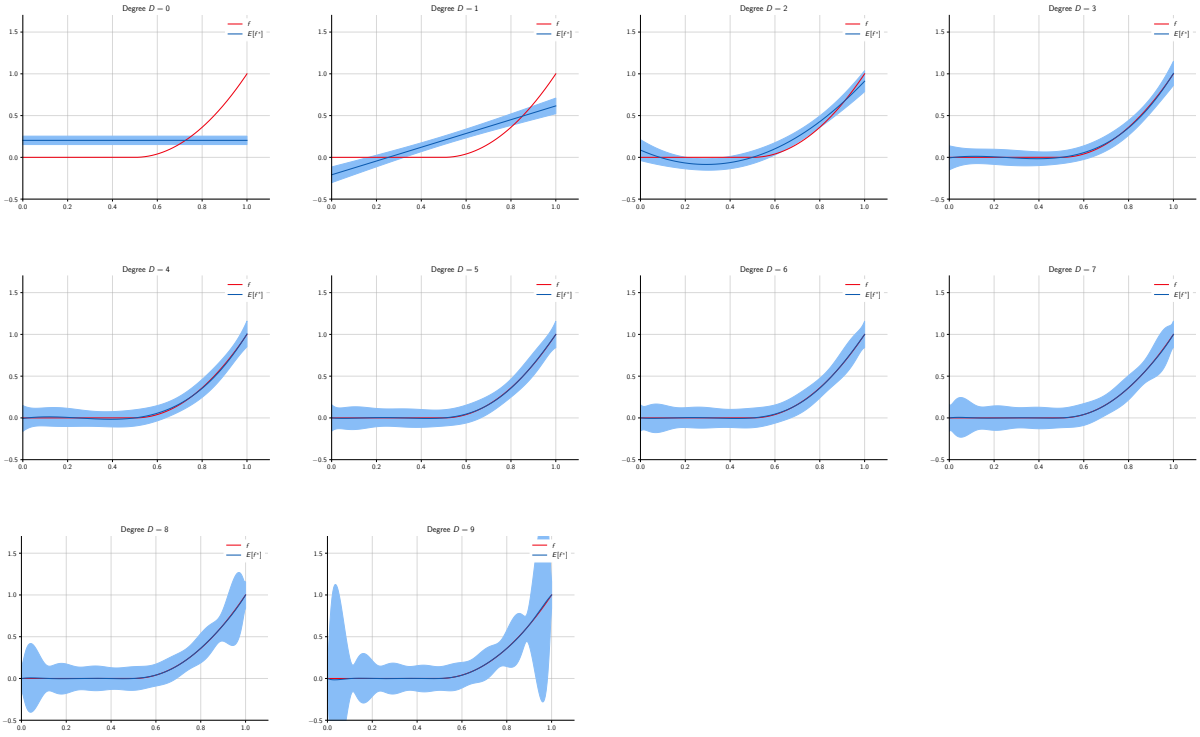
Notes

Given the trained models $f_1, \ldots, f_M$, we can compute an empirical mean prediction at x as

$$\bar{f}(x) = \frac{1}{M} \sum_{m=1}^{M} f_m(x),$$

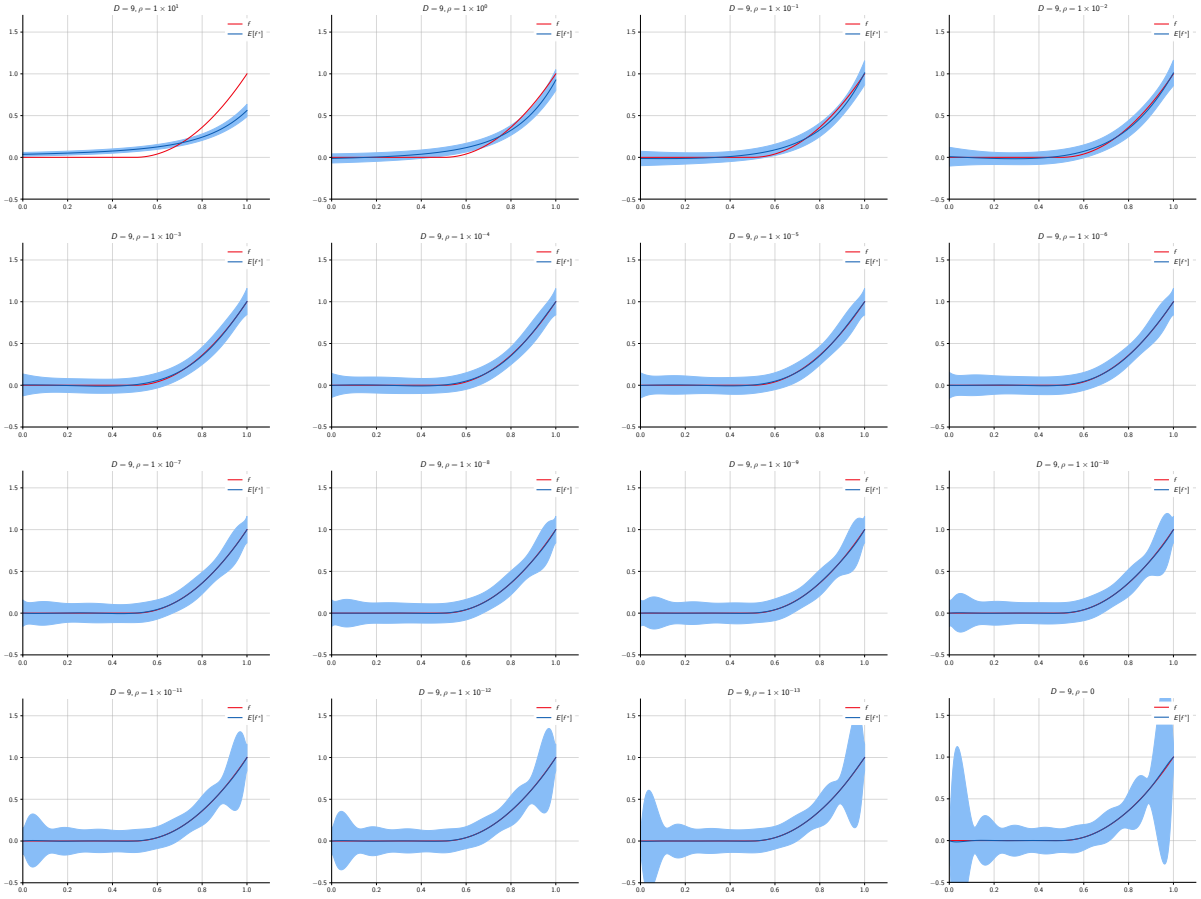
and the empirical variance of the prediction at x with

$$\sigma(x) = \frac{1}{M-1} \sum_{m=1}^{M} \left(\bar{f}(x) - f_m(x)\right)^2.$$



Notes

The red curve correspond to the "true" structure of the data we want to fit, it is constant equal to zero on $[0, 0.5]$ and quadratic on $[0.5, 1]$.
For $m = 1, \ldots, M$, we generate a training set $\mathcal{D}_m$ by taking $x_1^m, \ldots, x_n^m$ regularly spaced in $[0, 1]$, and computing each y_n^m as $f(x_n^m)$ added to a random Gaussian noise. Then we fit a polynomial f_m of degree D.
Then we can compute for every x the mean and standard deviation of $f_1(x), \ldots, f_M(x)$.
The blue curve is the mean prediction average over $M = 10000$ runs, and the blue area shows $\pm$ the standard deviation.
As the degree increases, the prediction (blue curve) gets more and more aligned with the true functional (red curve), but the standard deviation increases more and more. A large variance (or standard deviation) shows the discrepancy between all the trained models.
Note that for large D, even with $M = 10,000$, the variance is so large that the estimation of the standard deviation is noisy, resulting in oscillations on the graph.



Notes

Same observations as in the previous slide, now for the weight ρ of the quadratic penalty decreasing instead of D increasing.

We can formalize these observations as follows:

Let x be fixed, y the "true" value associated to it, f^* the predictor we learned from the data-set $\mathscr{D}$, and $Y = f^*(x)$ be the value we predict at x.

If we consider that the training set $\mathscr{D}$ is a random quantity, then f^* is random, and consequently Y is.

We have

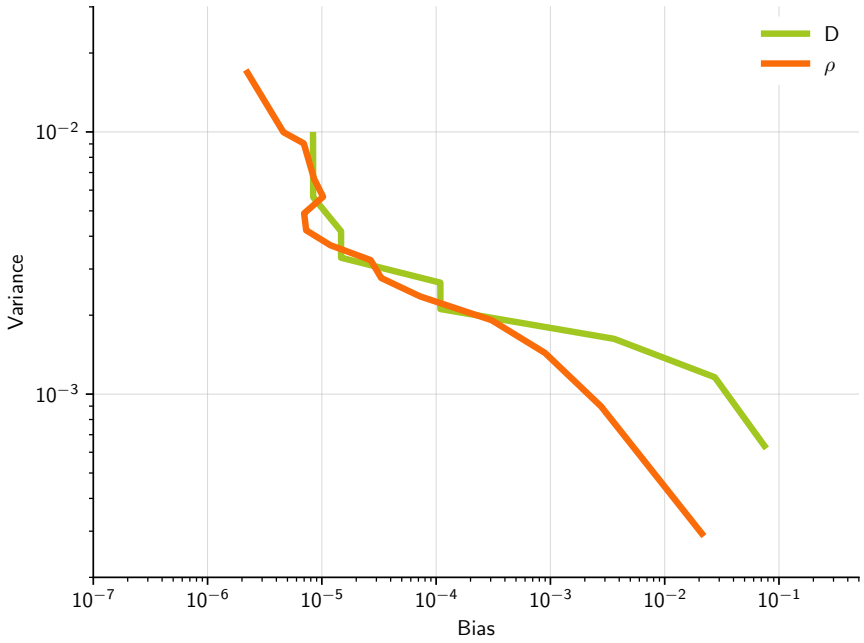
$$\begin{aligned}
\mathbb{E}_{\mathscr{D}}\left((Y-y)^2\right) &= \mathbb{E}_{\mathscr{D}}\left(Y^2-2Yy+y^2\right)\\
&= \mathbb{E}_{\mathscr{D}}\left(Y^2\right)-2\mathbb{E}_{\mathscr{D}}\left(Y\right)y+y^2\\
&= \underbrace{\mathbb{E}_{\mathscr{D}}\left(Y^2\right)-\mathbb{E}_{\mathscr{D}}\left(Y\right)^2}_{\mathbb{V}_{\mathscr{D}}(Y)}+\underbrace{\mathbb{E}_{\mathscr{D}}\left(Y\right)^2-2\mathbb{E}_{\mathscr{D}}\left(Y\right)y+y^2}_{(\mathbb{E}_{\mathscr{D}}(Y)-y)^2}\\
&= \underbrace{\left(\mathbb{E}_{\mathscr{D}}\left(Y\right)-y\right)^2}_{\text{Bias}}+\underbrace{\mathbb{V}_{\mathscr{D}}\left(Y\right)}_{\text{Variance}}
\end{aligned}$$

This is the **bias-variance decomposition:**

- the bias term quantifies how much the model fits the data on average,
- the variance term quantifies how much the model changes across data-sets.

(Geman and Bienenstock, 1992)

From this comes the **bias-variance tradeoff:**



Reducing the capacity makes f^* fit the data less on average, which increases the bias term. Increasing the capacity makes f^* vary a lot with the training data, which increases the variance term.

Notes

The plot shows the variance as a function of the bias in the two previous setup. The green curve when varying D, and the red curve when varying ρ.
We see that when we decrease the bias (i.e. the model does better on average), we increase the variance (i.e. the model fluctuates more between data-sets and generalizes less).

Is all this probabilistic?

Conceptually model-fitting and regularization can be interpreted as Bayesian inference.

This approach consists of **modeling the parameters A of the model themselves as random quantities following a prior distribution μ_A.**

By looking at the data $\mathscr{D}$, we can estimate a posterior distribution for the said parameters,

$$\mu_A(\alpha \mid \mathscr{D} = \mathbf{d}) \propto \mu_{\mathscr{D}}(\mathbf{d} \mid A = \alpha)\, \mu_A(\alpha),$$

and from that their most likely values.

So instead of a penalty term, we define a prior distribution, which is usually more intellectually satisfying.

For instance, consider a polynomial model with Gaussian prior, that is

$$\forall n,\ Y_n = \sum_{d=0}^{D} A_d\, X_n^d + \Delta_n,$$

where

$$\forall d,\ A_d \sim \mathcal{N}(0, \xi),\ \forall n,\ X_n \sim \mu_X,\ \Delta_n \sim \mathcal{N}(0, \sigma)$$

all independent.

For clarity, let $A = (A_0, \ldots, A_D)$ and $\alpha = (\alpha_0, \ldots, \alpha_D)$.

Remember that $\mathscr{D} = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$ is the (random) training set and $\mathbf{d} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is a realization.

$$
\begin{aligned}
&\log \mu_A(\alpha \mid \mathscr{D} = \mathbf{d}) \\
&\qquad = \log \frac{\mu_{\mathscr{D}}(\mathbf{d} \mid A = \alpha)\, \mu_A(\alpha)}{\mu_{\mathscr{D}}(\mathbf{d})} \\
&\qquad = \log \mu_{\mathscr{D}}(\mathbf{d} \mid A = \alpha) + \log \mu_A(\alpha) - \log Z \\
&\qquad = \log \prod_n \mu(x_n, y_n \mid A = \alpha) + \log \mu_A(\alpha) - \log Z \\
&\qquad = \log \prod_n \mu(y_n \mid X_n = x_n, A = \alpha)\, \underbrace{\mu(x_n \mid A = \alpha)}_{= \mu(x_n)} + \log \mu_A(\alpha) - \log Z \\
&\qquad = \log \prod_n \mu(y_n \mid X_n = x_n, A = \alpha) + \log \mu_A(\alpha) - \log Z' \\
&\qquad = \underbrace{-\frac{1}{2\sigma^2} \sum_n \left(y_n - \sum_d \alpha_d x_n^d \right)^2}_{\text{Gaussian noise on } Y} \underbrace{- \frac{1}{2\xi^2} \sum_d \alpha_d^2}_{\text{Gaussian prior on } A} - \log Z''.
\end{aligned}
$$

Taking $\rho = \sigma^2/\xi^2$ gives the penalty term of the previous slides.

Regularization seen through that prism is intuitive: The stronger the prior, the more evidence you need to deviate from it.

References

S. Geman and E. Bienenstock. **Neural networks and the bias/variance dilemma**. Neural Computation, 4:1–58, 1992.