El compromiso entre bias y varianza

Vemos un problema de regresión ordinario, donde tenemos N puntos

$$(x_i, y_i)$$

y suponemos que hay una relación causal entre ellos

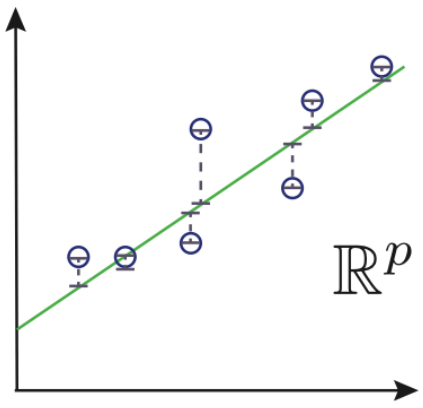
$$y_i = f(x_i)$$

Pero como y_i fue "medido" y no calculado, sabemos que la medición siempre introduce un error

$$y_i = f(x_i) + \gamma_i$$

Donde γ_i es una variable aleatoria, independiente del error de otras mediciones, con distribución gaussiana de media 0 y desviación estandar $\sigma = 1$.

Nuevamente el caso en que podemos inferir la función $f(x)$, como pasa en las ciencias



7 puntos

FIG. 10 Geometric interpretation of least squares regression. The regression function g defines a hyperplane in $\mathbb{R}^p$ (green solid line, here we have $p = 2$) while the residual of data point $\boldsymbol{x}_i$ (hollow circles) is its projection onto this hyperplane (bar-ended dashed line).

En este ejemplo sabemos que

$$f(x) = a + b\,x$$

O sea, tenemos 2 parámetros. Los elegimos de manera que minimicen el error cuadrático medio

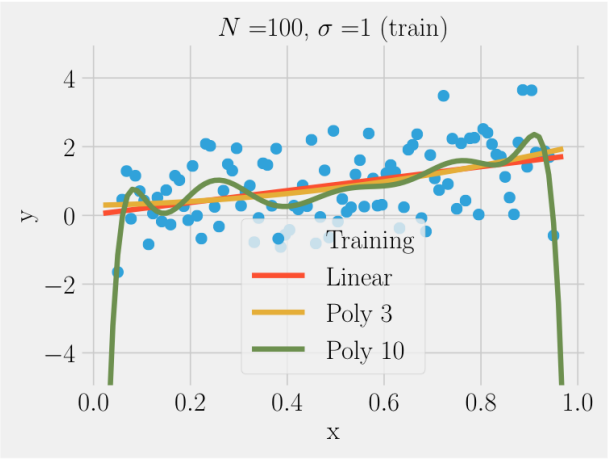
$$ECM = \frac{1}{7}\sum_{i=1}^{7} (\hat{y}_i - y_i)^2$$

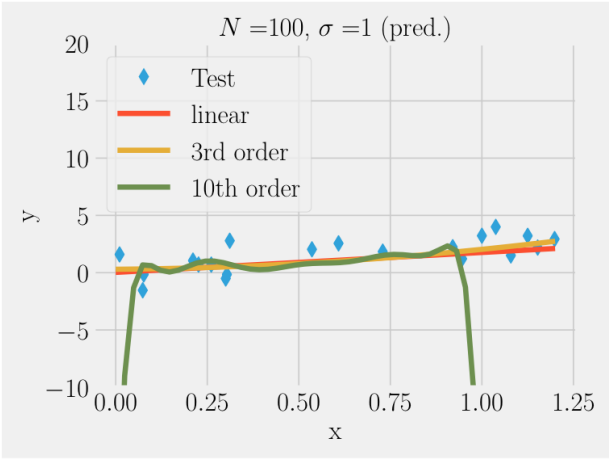
donde $\hat{y}_i$ es el valor medido (con error) e y es el estimado (sin error).

Acá conocemos f y solo necesitamos calcular los mejores parámetros

Pero otros 7 puntos tendremos otros parámetros por culpa del error.

Cuando mayor sea el # de puntos, mayor será el ECM. Pero mejor interpolaré (estimaré puntos no medidos).



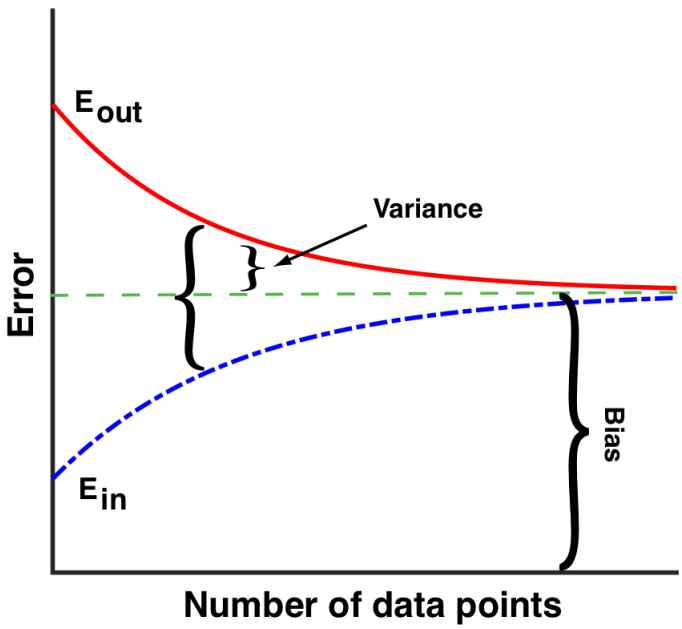


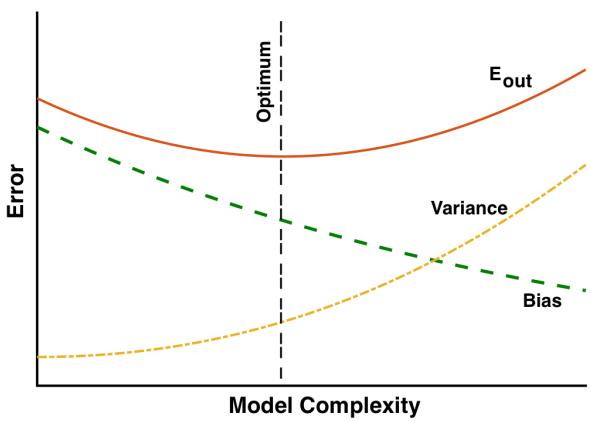
Mirar el ejemplo con 100 puntos. Los bolas azules son datos conocidos usados para ejentar. Los rombos (a la derecha) son datos para verificar (test).

* El polinomio de grado 10 requiere ejentar 11 parámetros
* El " " " 3 " " 4 "
* El " " " 1 " " 2 "

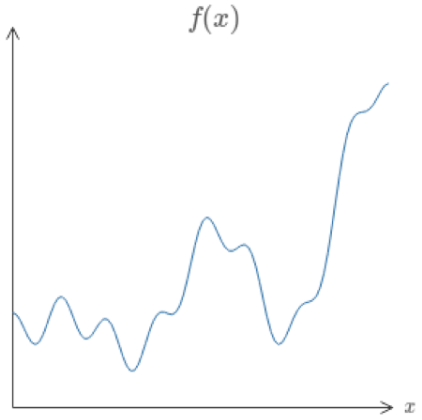
El polinomio de grado 10 ajusta mejor los datos conocidos, pero extrapola mal.

El polinomio de grado 2 ajusta peor los datos conocidos, pero extrapola mejor.



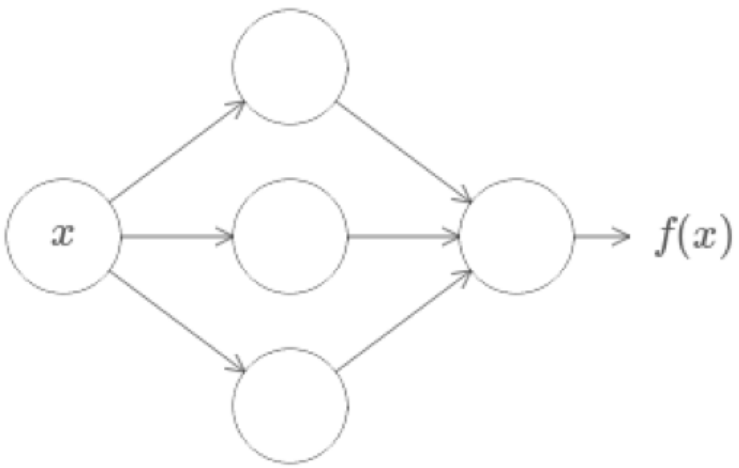


Ahora vuelvo a Machine Learning. Es el mismo problema pero no conocemos $f(x)$, solo los ejes.



función definida a mano

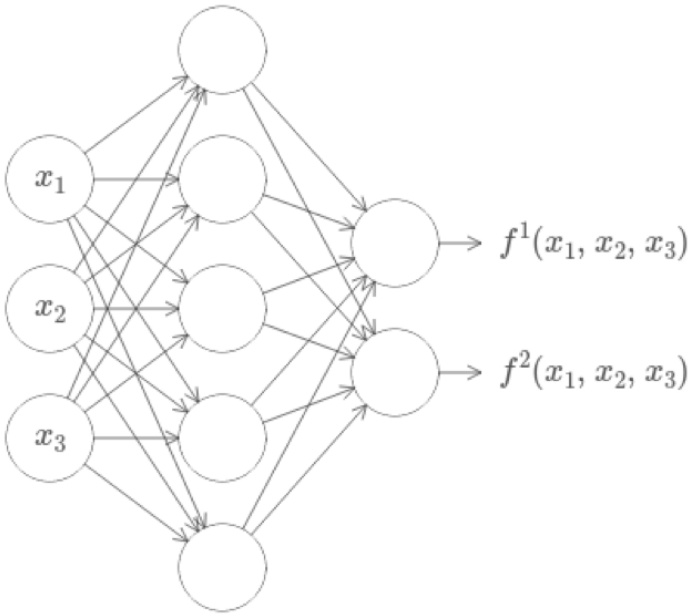
Una de las cosas más curiosas de redes neuronales es que podemos "aproximar" cualquier función por arbitraria que sea.



Esto requiere, por supuesto, conocer "ejemplos".

Esto se puede generalizar a muchas dimensiones

$$f: \mathbb{R}^n \to \mathbb{R}^m$$



Para esto es suficiente una única capa oculta

Teorema de universalidad

"Approximation by superpositions of a sigmoidal function"

G. Cybenko

Mathematics of Control, Signal y Systems 2 (303)

1989

Lo que nos dice el teorema es que, para cierta tolerancia $\varepsilon > 0$ tal que

$$|f(x) - g(x)| < \varepsilon$$

f: función

g: output de la red

existe un número N de neuronas ocultas.

La función debe ser continua

Una red neuronal de una capa oculta se puede usar para aproximar cualquier función continua para cualquier precisión deseada