REDES NEURONALES 2024

Clase 24 parte 1

Jueves 14 de noviembre 2024

FAMAF, UNIVERSIDAD NACIONAL DE CÓRDOBA

INSTITUTO DE FÍSICA ENRIQUE GAVIOLA (UNC-CONICET)

LA RELACIÓN ENTRE E_{in} Y E_{out}

Supongamos que tenemos un conjunto de entrenamiento y construimos un modelo neuronal de aprendizaje supervisado, como por ejemplo, una red feed-forward. Ya vimos qué hay muchos hiperparámetros por determinar, como por ejemplo la razón de aprendizaje η, el número de capas ocultas, el número de neuronas en cada capa oculta, y veremos que aún resta definir varios hiperparámetro más. Sin embargo veremos qué hay dos elementos DETERMINANTES cuya determinación previa al entrenamiento del modelo será muy delicada y difícil de realizar:

- El tamaño del conjunto de entrenamiento.

Es importante que analicemos esto en detalle para apuntar a tener un buen modelo y no caer en confusiones usuales.

Cuanto más simple es nuestro modelo neuronal, o sea, cuanto menos cantidad de parámetros (sinapsis y umbrales) tengamos para ajustar (determinar), mayor será el error de testeo E_{out}. Por otro lado, con pocos parámetros dependerá menos de la realización particular del conjunto de entrenamiento que nos haya tocado en suerte.
Aún con un número muy grande de elementos en el conjunto de entrenamiento, la capacidad de predecir más allá del conjunto de entrenamiento se puede degradar rápidamente si elegimos un modelo con un número inadecuado de parámetros, ya sea porque son muy pocos o porque son demasiados.

Esto nos habla de lo difícil que es el problema que queremos abordar a partir de un conjunto de entrenamiento. No sabemos a priori cuál es el buen modelo (cuantas neuronas en cuantas capas y cuántas sinapsis). Y es costoso explorar el conjunto de todas las arquitecturas posibles.

Recordemos siempre que podemos ser muy buenos en minimizar el error de entrenamiento pero muy malos para tener un buen resultado del error con el conjunto de testeo. Solo la comprensión de estos conceptos, sumado a la búsqueda cuidadosa de arquitecturas adecuadas y a la experiencia adquirida nos llevarán a buen puerto.

¡AJUSTAR BIEN NO IMPLICA PREDECIR BIEN!

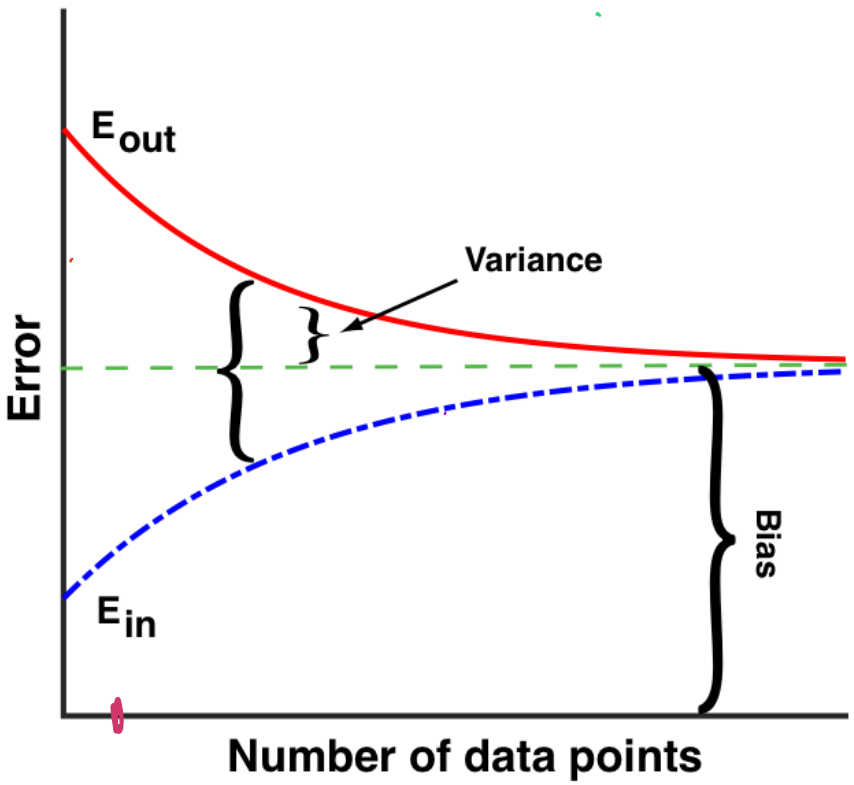
Si nuestro modelo tiene demasiados parámetros para ajustar, el error de testeo será muy grande. Esto, en la jerga del aprendizaje automático se denomina SOBREAJUSTE (OVERFITTING en inglés). En otras palabras, aprenderemos bien el conjunto de entrenamiento pero perdemos la capacidad de captar la tendencia. Tendremos un bajo valor de de error de entrenamiento E_{IN} pero un alto valor de error de testeo E_{OUT}.

EL BALANCE ENTRE EL SESGO Y EL BIAS

¿Qué podemos decir sobre la relación entre el error de entrenemiento y el error de testeo? Este análisis es central para entender cómo construir una buena red neuronal.

El efecto del tamaño del conjunto de entrenamiento

Supongamos que ya hemos elegido la arquitectura de nuestra red neuronal y por ende el número de parámetros. Analicemos en detalle el gráfico esquemático y simple que sigue, el cual representa un problema cualquiera de aprendizaje supervisado que queremos resolver.



De este gráfico sacamos varias conclusiones.

- Una vez que fijamos el número de parámetros, al cual caracterizaremos con el nombre complejidad, si conociéramos la respuesta correcta para infinitos elementos de entrada, el error de entrenamiento *Ein* y *Eout* serían iguales. Uno podría pensar que este valor es una forma de caracterizar a nuestra red, por eso le damos un nombre: sesgo. No hay forma, para este modelo, de tener menor error de testeo *Eout* que sesgo. Para mejorar el modelo solo nos resta cambiar la complejidad. Claro que el sesgo es un concepto teórico. A lo sumo podemos estimarlo, ya que no disponemos de un conjunto de entrenamiento con infinitos elementos etiquetados.

- Para un caso realista de un número finito de elementos en el conjunto de entrenamiento tendremos naturalmente que

$$Ein < bias < Eout$$

- Parece paradójico, pero no lo es, que muchas veces tengamos que aceptar que para tener un buen *Eout* tengamos que tener un mayor *Ein*. Esto es fundamental. Tratemos de tener muchos datos etiquetados para que *Ein* se acerque a *bias*.

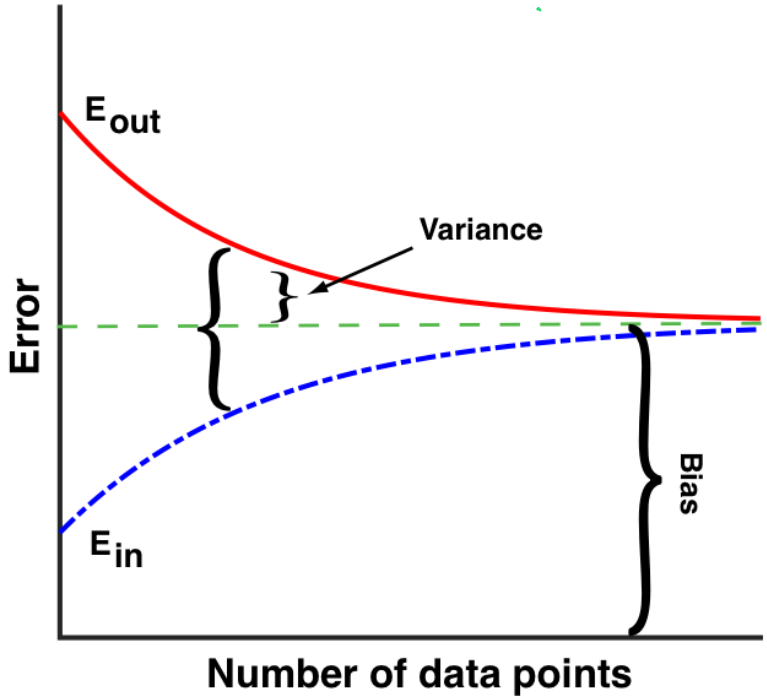
- Hagamos una analogía con el problema de ajustar un conjunto de mediciones *(xi,yi)* con polinomios. Si tenemos pocos puntos será fácil ajustarlos con poco error pero será difícil captar la tendencia. O sea, tendremos un error de aprendizaje *Ein* bajo pero *Eout* alto. Si en cambio tenemos muchos puntos medidos para hacer el ajuste, con el mismo modelo nos costará más tener un error pequeño de aprendizaje *Ein*, pero captaremos mejor la tendencia. Claro que los valores finales de *Ein*, *Eout* y *bias* dependerán de la complejidad del modelo.

- El error de testeo se puede medir desde el bias:

$$Eout = bias + varianza$$

$$varianza = Eout - bias$$

La varianza es una medida del error introducido en el modelo debido al ruido de la muestra (del conjunto de entrenamiento).



- Otra cantidad de interés es la diferencia entre ambos errores, al cual llamaremos error:

$$Error = E_{out} - E_{in}$$

Error mide cuán bien nuestro error de entrenamiento E_{in} refleja el error de testeo. Por eso debemos recordar que nosotros queremos el mejor E_{out} y no el menor E_{in}. *Error* puede ser nulo sólo en el límite de un conjunto de entrenamiento de infinitas componentes .

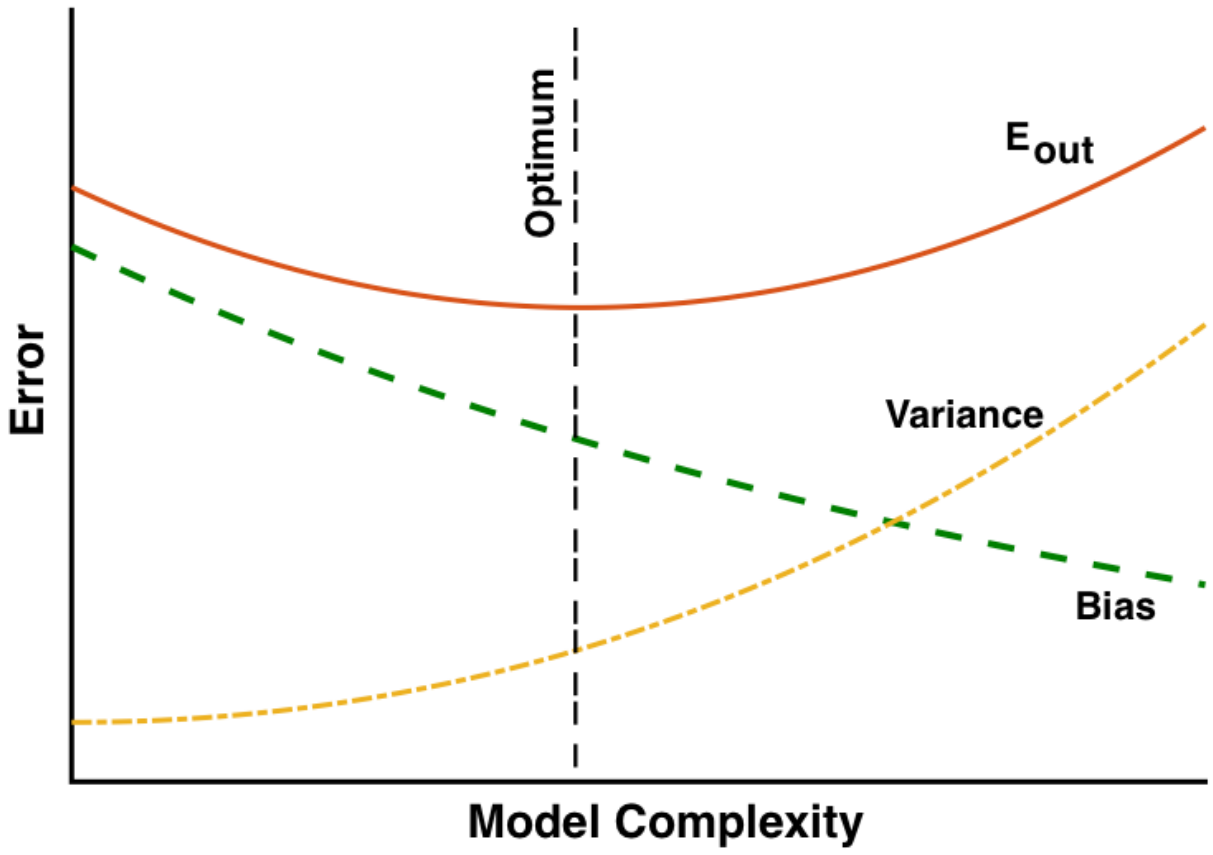
Podemos decir que para una arquitectura dada, nuestro modelo queda definido por tres cantidades:

- E_{in}
- E_{out}
- *bias*

Supongamos que ya hemos elegido el tamaño del conjunto de entrenamiento y ahora queremos ver cómo mejorar el resultado cambiando la complejidad. La complejidad es una mezcla de cosas: Una red es más compleja cuánto más parámetros tiene (sinapsis y umbtprales) pero también cuantas más capas tiene, o sea, cuanto más profunda es.

El efecto de la complejidad

La figura que sigue es un esquema de lo que sucede cuando variamos la complejidad, y vemos que es muy diferente a lo que sucede con el efecto del tamaño del conjunto de entrenamiento.



- Tendemos a creer que cuanto más complejo es el modelo, o sea cuanto, más parámetros tengamos para ajustar, mejor será el resultado y por ende, que menor será menor E_{out}. Sin embargo veremos que esto no es necesariamente cierto.
- Pensemos en el caso análogo de la regresión polinomial por cuadrados mínimos a partir de mediciones (x_i, y_i). Si tengo pocos datos y uso un polinomio de bajo orden, podemos tener en primer lugar un error de aprendizaje pequeño, pero diferentes conjuntos de datos darán poca varianza y por ende un valor alto de E_{in} y bias. Esto significa que el error de testeo será alto y el modelo no captará la tendencia por la simplicidad del modelo.

- A medida que aumentamos la complejidad, disminuyen los errores (E_{in}, E_{out} y *bias*) pero aumenta la varianza. Esto dice que el resultado del modelo dependerá cada vez más de la realización particular del conjunto de entrenamiento.

- En algún momento, cuando aumentamos la complejidad, la varianza crece tanto que a pesar de que Ein y bias disminuyen mucho, el error de testeo comienza a crecer monótonamente. O sea, podemos creer que tenemos un buen modelo pero solo sirve para entrenar pero no para captar el comportamiento que pretende modelar.

- Si nuestro modelo produce un pequeño error de entrenamiento E_{in} y un gran error de testeo E_{out} decimos que estamos en un régimen de sobreajuste (overfitting).

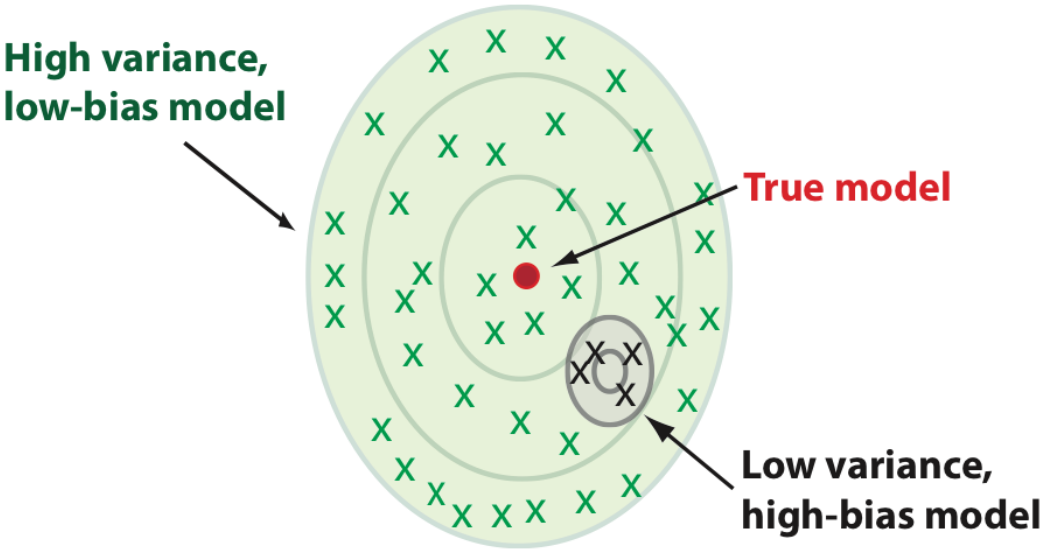
RESUMEN DE LA RELACIÓN ENTRE E_{IN} Y E_{OUT}

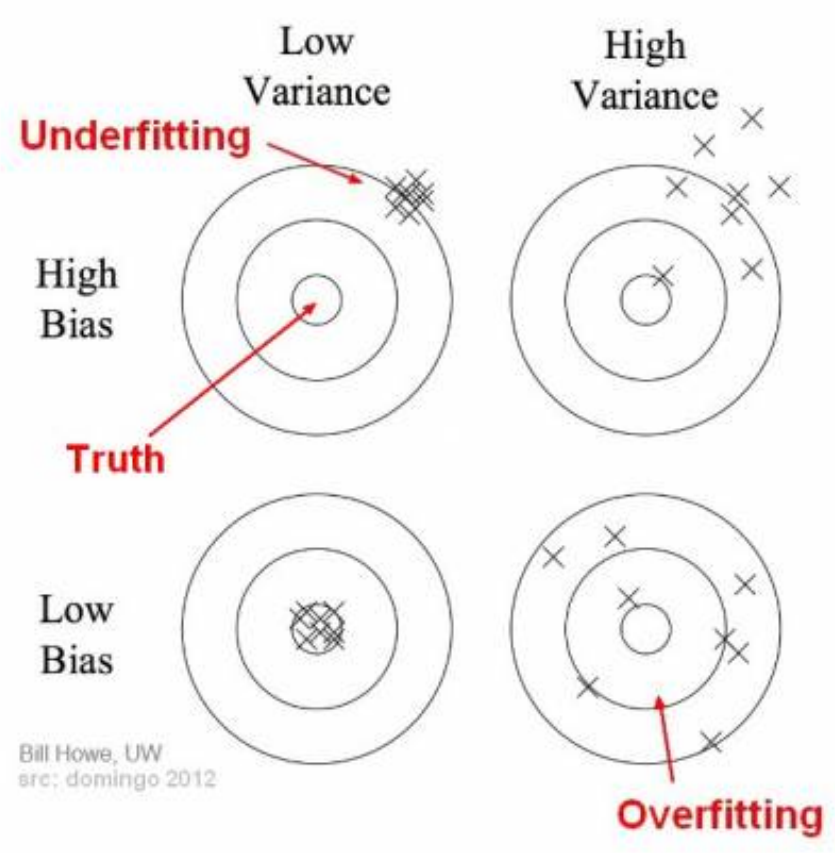
Hemos visto que el tamaño del conjunto de entrenamiento y la complejidad son dos elementos fundamentales allá hora de diseñar el problema de aprendizaje supervisado.

Si nos preguntan cuál es el tamaño del conjunto de entrenamiento, la respuesta es simple: tan grande como podamos y nos permita entrenarlo con la capacidad de cálculo que tengamos.

Si nos preguntan cuál es el grado de complejidad, debemos ser mucho más prudentes. Dado el problema y el tamaño del conjunto de entrenamiento, debemos ajustar muy bien la complejidad para que no sea muy baja para producir subajuste (underfitting) ni muy alta para producir sobreajuste (overfitting).

UNA COSA ES APRENDER Y OTRA COSA ES GENERALIZAR


High variance,
low-bias model
True model
Low variance,
high-bias model


Low
Variance
High
Variance
Underfitting
High
Bias
Truth
Low
Bias
Overfitting
Bill Howe, UW
src: domingo 2012

La tensión entre la varianza y el sesgo refleja la tensión que existe en el aprendizaje automático supervisado entre la complejidad del modelo (el número de sinapsis, umbrales y capas) y la cantidad de elementos del conjunto de datos que usamos para entrenar.

Dado que el número de datos es usualmente limitado y caro, muchas veces es mejor utilizar modelos más simples que puedan captar la tendencia de los datos a pesar de que tengan mucho sesgo. En otras palabras, hay que evitar que el modelo sea tan complejo como para estar en el régimen en el cual el error de generalización aumenta con la complejidad.

-