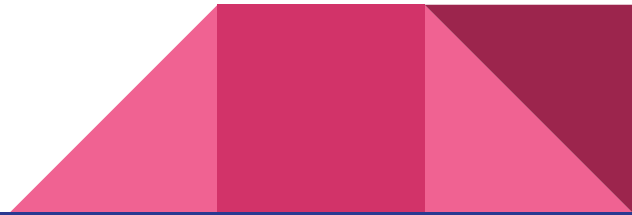
19. Concetti di Machine Learning

Corso di Python per il Calcolo Scientifico

Angelo Cardellicchio
angelo.cardellicchio@stiima.cnr.it

Outline

- Descrizione del dataset
- Generalizzazione
- Iperparametri
- Regolarizzazione



Descrizione del dataset

- Per descrivere un dataset si utilizza la **design matrix**
- Questa contiene tante righe quanti sono i campioni disponibili, e tante colonne quante sono le feature
- Ad esempio, il dataset Iris è definito da una design matrix del tipo:

$$X \in R^{150 \times 4}$$

- L'elemento $X(i, 1)$ descrive la lunghezza del sepalo della pianta i, l'elemento $X(j, 2)$ la larghezza del sepalo della pianta j, e così via.
- In caso di dataset supervisionato è associato anche un vettore di label. Ad esempio, per Iris:

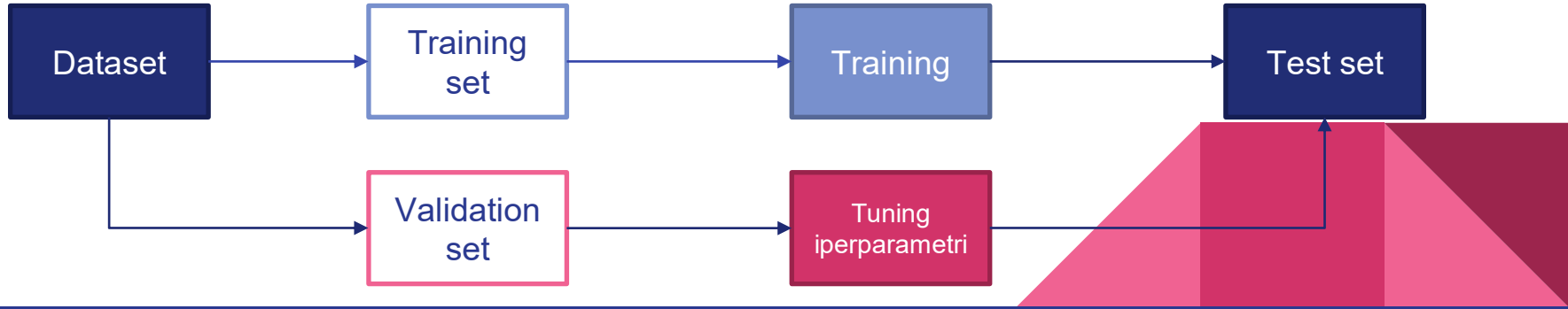
$$y \in R^{150}$$

Generalizzazione

- Le performance di un algoritmo di machine learning non vanno valutate sui dati su cui è stato addestrato (**training set**), ma su un insieme di dati indipendente (**test set**).
- Questo permette di raggiungere l'obiettivo della **generalizzazione** dell'algoritmo.
- I dati di training e test sono generati dallo stesso **meccanismo di generazione**.
- Le ipotesi che facciamo è che i dati siano **indipendenti** ed **identicamente distribuiti**.
- Date queste ipotesi, l'addestramento del modello deve far sì che:
  - si minimizzi l'errore di training (in caso contrario, avremo **underfitting**);
  - si minimizzi il gap tra errore di training ed errore di test (in caso contrario, avremo **overfitting**).
- Per controllare underfitting ed overfitting si modifica la **capacità** del modello.

Iperparametri

- Gli iperparametri sono delle *impostazioni* di un modello che non vengono impostate sul dataset di addestramento.
- Di solito agiscono sulla capacità del modello.
- Vanno impostati tenendo conto di un insieme di dati, che non devono essere né appartenenti al set di training, né al set di test.
- Di conseguenza, abbiamo bisogno di un **set di validazione**, preso dal set di training.



Regolarizzazione

- Il **No Free Lunch theorem** dice che non esiste un algoritmo migliore di altri in assoluto.
- Possiamo solo individuare un algoritmo che funziona meglio sul nostro specifico problema.
- Abbiamo visto che per migliorare le performance possiamo cambiare la capacità del nostro modello; tuttavia, questa non è l'unica soluzione possibile.
- Un altro modo è quello di **selezionare quali tipi di funzione usare nel nostro spazio di ricerca**.
- Di solito, si fa in modo che ci sia più di un tipo di funzione nello spazio di ricerca. In questo caso, le diverse funzioni hanno un diverso peso.

Regolarizzazione

- Ad esempio, la regressione lineare può includere il concetto di **decadimento dei pesi** (**weight decay**).
- Per farlo, modifichiamo la funzione di costo in tal modo:

$$L(w) = MSE_{train} + \lambda w^T w$$

- Il termine w indica l'entità dei diversi pesi w_i, mentre λ controlla il contributo del secondo termine.
- Minimizzare $L(w)$ implica scegliere un compromesso tra errore di training e pesi di dimensioni ridotte. Più forte è λ, maggiore è la nostra volontà di **regolarizzare** i pesi, ottenendo soluzioni con pendenza inferiori, o che danno peso ad un numero limitato di feature.
- Questa procedura è volta a ridurre l'errore di generalizzazione mantenendo costante l'errore sul dataset di training, riducendo quindi l'overfitting.

Domande?

42