

Valutazione di un Modello di Regressione

Nei precedenti capitoli abbiamo valutato “a spanne” i modelli di regressione, per valutarli opportunamente dobbiamo avere a disposizione un dataset abbastanza grande da poterci permettere di suddividere i dati in training set e test set appunto. In generale la suddivisione si aggira attorno ad un rapporto 80/20 (80% training set, 20% test set), ma non esiste una regola fissa.

Una volta diviso il dataset, possiamo suddividere il training set in due parti; così da ottenere il validation set. Anche per questa suddivisione non esistono regole fisse.

Hyperparameters

Gli iperparametri sono parametri che non vengono appresi durante il training, ma che influenzano il comportamento del modello. Molti modelli di Machine Learning hanno iperparametri, per quanto riguarda la regressione lineare, di base, ha solo il grado del polinomio. Il grado del polinomio, come abbiamo visto precedentemente determina la capacità del modello. Allo stesso modo λ nella regolarizzazione, che vedremo successivamente è un iperparametro.

Gli iperparametri non vengono appresi durante il training, proprio perché se fosse così il modello non sarebbe in grado di generalizzare bene: ad esempio nella polinomial regression, se il grado del polinomio fosse un parametro appreso, il modello potrebbe avere un grado del polinomio molto alto, per minimizzare l'errore sul training set, ma non generalizzerebbe bene. Questo fenomeno è noto come **overfitting**; e da qui nasce la necessità di un validation set.

Ovviamente questa divisione dipende dalla quantità di dati a disposizione, nel caso del nostro esempio i dati a disposizione sono poco più di un centinaio; in questi casi è utile sfruttare la tecnica del **cross-validation**.

Underfitting e Overfitting

L'Underfitting si verifica quando il modello non ottiene buone prestazioni ne sul training set, ne sul test set.

L'Overfitting si verifica quando il modello ottiene buone prestazioni sul training set ma non sul test set.

The No Free Lunch Theorem Contrariamente a quanto si possa pensare, non esiste un modello che sia il migliore in assoluto per tutti i problemi.

Regulaization / Regolarizzazione

La regolarizzazione è una qualsiasi modifica che apportiamo al modello per ridurre l'errore di generalizzazione (ma non il training error).

Il comportamento dell'algoritmo è influenzato infatti, non solo dalla capacità del modello (spazio delle ipotesi); ma anche dall'identità delle funzioni utilizzate. Per esempio, la regressione lineare ha uno spazio delle ipotesi composto esclusivamente da funzioni lineari e, nel caso non ci sia relazione lineare tra i dati (e.g. $\sin(x)$), non sarà in grado di generalizzare bene.

Potremmo modificare il criterio di ottimizzazione per la regressione lineare includendo un termine regolarizzatore (denotato con $\Omega(w)$) nella funzione di costo.

Nello specifico caso del weight decay il rego è uguale a: $\Omega(w) = w^T w$. Dunque il criterio sarà:

$$J(w) = \text{MSE}_{\text{train}} + \lambda w^T w$$

in questo modo minimizziamo una somma che comprende sia l'errore quadratico medio sul training set, sia il termine di regolarizzazione. In questo caso il termine λ è un iperparametro che regola

l'importanza del termine di regolarizzazione. Con $\lambda = 0$ il modello si comporta come una regressione lineare standard, mentre con $\lambda > 0$ il modello tenderà a preferire pesi più piccoli, da questo il nome weight decay.

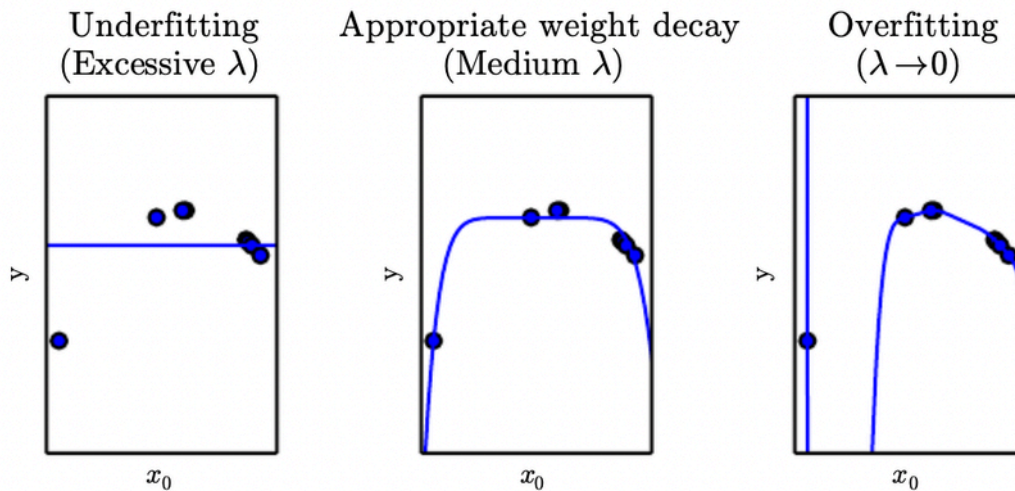


Figure 1: “Il modello utilizzato ha solo funzioni di grado 9, mentre il dataset è generato da una funzione quadratica.”

Nel campo del Machine Learning esistono diverse varianti per quanto riguarda le tecniche di regolarizzazione.

Famiglia delle L^p norme; generalizzata con la formula:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

con n ad indicare le dimensioni e $p \in [1, +\infty)$.

- La norma 1 è banalmente la somma dei valori assoluti dei componenti.
- La norma 2 o Norma Euclidea, è la radice quadrata della somma dei quadrati dei valori:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$

Dataset Augmentation

Il miglior modo per avere un modello che generalizza bene è trainarlo su più dati e, spesso, il dataset a disposizione non è abbastanza grande. Un modo per risolvere questo problema sono le tecniche di dataset augmentation. Questo approccio è molto efficace con le task di classificazione, object recognition e speech recognition. Com'è facile immaginare per quanto concerne l'object recognition, possiamo ruotare, scalare, e traslare le immagini; per lo speech recognition possiamo aggiungere rumore alle registrazioni.

L'iniezione di rumore è alla base di alcuni modelli unsupervised, come il denoising autoencoder. La noise injection può inoltre essere implementata negli hidden layer.