

# Simulazione

Nell'esame ci sarà una penalizzazione per le risposte sbagliate, tipo 20%

All'esame saranno 20 domande

Question 1

Not yet answered

Marked out of 1.00

Flag question

Which of the following is a commonly used cost function for regression tasks?

- ☐ a. Categorical Cross-Entropy
- ☐ b. Mean Squared Error (MSE)
- ☐ c. Hinge Loss
- ☐ d. Binary Cross-Entropy

## b. Mean Squared Error (MSE)

**Mean Squared Error (MSE)** è una funzione di costo comunemente utilizzata per i compiti di regressione. Calcola quanto si discostano le previsioni del modello dai valori reali, penalizzando maggiormente gli errori più grandi poiché eleva al quadrato la differenza.

- **a. Categorical Cross-Entropy:** usata per la classificazione multi-classe, non per la regressione.
- **c. Hinge Loss:** utilizzata per la classificazione binaria, ad esempio con SVM, non per la regressione.
- **d. Binary Cross-Entropy:** usata per la classificazione binaria, non per la regressione

Question 2

Not yet answered

Marked out of 1.00

Flag question

Explicit constraints implemented by re-projection only have an effect when the weights become large and attempt to leave the constraint region.

- ☐ True
- ☐ False

**Vero**

I vincoli espliciti implementati tramite la ri-proiezione hanno effetto quando i pesi diventano grandi e tendono a uscire dalla regione di vincolo. In tali casi, i vincoli vengono applicati riportando i pesi all'interno dei limiti definiti tramite la proiezione.

Quando i pesi del modello vengono aggiornati durante l'ottimizzazione, è possibile che escano da una regione definita o violino i vincoli stabiliti (ad esempio, rimanere all'interno di un intervallo specifico o soddisfare una condizione).

Quando ciò accade, la ri-proiezione "riporta" i valori dei pesi entro la regione consentita, proiettandoli letteralmente indietro nello spazio del vincolo.

Question 3

Not yet answered

Marked out of 1.00

Flag question

What is the main difference between 'model-agnostic' and 'model-specific' methods in the context of eXplainable AI?

- ☐ a. Model-agnostic methods require detailed knowledge of the model architecture, while model-specific methods work independently of the model type.
- ☐ b. Model-agnostic methods rely on neural networks, while model-specific methods are used with decision trees and linear models only.
- ☐ c. Model-agnostic methods can be applied to any type of ML model without modifications, while model-specific methods require a specific family of models
- ☐ d. Model-agnostic methods are more accurate but slower than model-specific methods, which are faster but less reliable

### c. Model-agnostic methods can be applied to any type of ML model without modifications, while model-specific methods require a specific family of models

I **metodi model-agnostic** funzionano con qualsiasi modello di machine learning, poiché non richiedono accesso o modifiche all'architettura interna del modello. Al contrario, i **metodi model-specific** sono progettati per una determinata famiglia di modelli e spesso utilizzano dettagli specifici del modello, come la struttura delle reti neurali o le caratteristiche degli alberi decisionali.

- **a.** I metodi model-agnostic non richiedono la conoscenza dettagliata dell'architettura del modello, mentre i metodi model-specific sì.
- **b.** I metodi model-agnostic non si limitano all'uso delle reti neurali; possono essere utilizzati con qualsiasi tipo di modello, e i metodi model-specific non sono limitati a decision trees o modelli lineari.

- **d.** Non è vero che i metodi model-agnostic siano necessariamente più accurati ma più lenti rispetto ai metodi model-specific; la velocità e l'affidabilità dipendono dall'implementazione e dall'uso specifico.

Question 4

Not yet answered

Marked out of 1.00

Flag question

Which of the following is a property of the cosine function that makes it less ideal for use in deep learning networks

- ☐ a. The function is non-differentiable.
- ☐ b. The function grows exponentially.
- ☐ c. The function oscillates between -1 and 1.
- ☐ d. The function has a range of  $[-1, 1]$

**c. The function oscillates between -1 and 1.**

La funzione coseno è meno ideale per l'uso nelle reti neurali profonde principalmente a causa della sua natura oscillante. Durante l'addestramento, le reti neurali si basano su gradienti stabili per aggiornare i pesi. Tuttavia, la funzione coseno, che oscilla tra -1 e 1, può causare problemi come gradienti che vanno a zero o oscillazioni, rendendo più difficile la convergenza del modello durante l'ottimizzazione.

- **a.** La funzione coseno è differenziabile, quindi non presenta il problema della non-differenziabilità.
- **b.** La funzione coseno non cresce esponenzialmente; oscilla in modo regolare.
- **d.** Il fatto che la funzione coseno abbia un intervallo di valori compreso tra -1 e 1 non è di per sé un problema, poiché molte funzioni di attivazione, come la sigmoide o la tangente iperbolica, hanno intervalli limitati e vengono comunque utilizzate nelle reti neurali. Il problema principale è l'oscillazione.

Question 5

Not yet answered

Marked out of 1.00

Flag question

Bagging is a form of ensemble model

☐ True

☐ False

**Vero**

**Bagging** (Bootstrap Aggregating) è una tecnica di ensemble che combina più modelli base (tipicamente lo stesso tipo di modello, come gli alberi decisionali) per migliorare le prestazioni complessive. Il principio alla base di bagging è quello di creare diversi modelli base utilizzando campioni diversi dei dati (ottenuti tramite il bootstrap, ovvero il campionamento con sostituzione) e combinare i loro risultati per ottenere una previsione più robusta e accurata. Un esempio comune di bagging è l'algoritmo **Random Forest**.

La caratteristica principale di bagging è che tutti i modelli base sono costruiti utilizzando lo stesso algoritmo (ad esempio, alberi decisionali), ma ciascun modello è addestrato su un campione casuale diverso del dataset originale, ottenuto tramite il **bootstrap** (campionamento con sostituzione). Ogni volta che si estrae un dato, questo può essere selezionato più volte, perché non viene rimosso dall'insieme di dati originale.

Question 6

Not yet answered

Marked out of 1.00

Flag question

Regularizing the bias parameters can introduce underfitting

☐ True

☐ False

**Vero**

**Regolarizzare i parametri di bias** (ovvero applicare una penalizzazione anche ai bias) può effettivamente causare **underfitting**. I bias sono essenziali per permettere al modello di adattarsi ai dati, poiché consentono di spostare la funzione di previsione in modo che anche quando le caratteristiche di input sono tutte zero, il modello possa comunque fare previsioni.

Se si applica troppa regolarizzazione ai bias (ovvero se li si penalizza troppo), si potrebbe limitare la capacità del modello di adattarsi correttamente ai dati. Questo

potrebbe ridurre la capacità del modello di apprendere informazioni utili dai dati, portando a un'**underfitting**, ossia una situazione in cui il modello non riesce ad adattarsi adeguatamente ai dati di addestramento.

In sintesi, una regolarizzazione troppo forte sui bias può "penalizzare" troppo il modello, impedendogli di apprendere correttamente e riducendo la sua capacità di fare previsioni accurate.

Question 7

Not yet answered

Marked out of 1.00

Flag question

In a neural network the nonlinearity causes the most interesting loss function to become convex

☐ True

☐ False

## False

In una rete neurale, le funzioni di perdita più interessanti non sono generalmente convesse a causa della non linearità introdotta dai livelli e dalle funzioni di attivazione. La non linearità rende il processo di ottimizzazione complesso, portando spesso a un paesaggio di perdita che contiene molteplici minimi locali, massimi, e sella, rendendo la funzione di perdita non convessa. Una funzione di perdita convessa è una funzione con un solo minimo globale, ma nelle reti neurali, a causa della loro struttura complessa, questo non è il caso nella maggior parte delle applicazioni.

Question 8

Not yet answered

Marked out of 1.00

Flag question

In general, when building machine learning models, the true data generating process is known and used for learning

☐ True

☐ False

## False

In generale, nei modelli di machine learning, il vero processo generativo dei dati non è noto. Gli algoritmi di apprendimento automatico tentano di fare inferenze, predizioni o decisioni basandosi su dati osservabili, ma la vera distribuzione sottostante o il meccanismo che genera i dati non è conosciuto con certezza. I modelli apprendono da campioni di dati e cercano di approssimare o modellare il

processo, ma esiste sempre un grado di incertezza rispetto alla "vera" distribuzione dei dati.

Question 9

Not yet answered

Marked out of 1.00

Flag question

The RMSProp algorithm modifies AdaGrad to perform better when applied to a convex function.

- ☐ True  
☐ False

## False

L'algoritmo RMSProp è stato sviluppato per migliorare AdaGrad, ma il suo scopo principale non è specificamente legato alle funzioni convesse. AdaGrad accumula il quadrato dei gradienti, il che può portare a un apprendimento troppo lento in alcune situazioni, specialmente su problemi non stazionari. RMSProp affronta questo problema mantenendo una media mobile del quadrato dei gradienti, migliorando così la capacità di adattarsi meglio al gradiente in scenari più complessi, spesso non convessi, come le reti neurali profonde. Pertanto, l'attenzione di RMSProp è focalizzata su problemi non stazionari e non esclusivamente sulle funzioni convesse.

Question 10

Not yet answered

Marked out of 1.00

Flag question

What is the purpose of the latent space in an autoencoder?

- ☐ a. To store raw input data  
☐ b. To store backup copies of the input data  
☐ c. To represent data in a lower-dimensional space for feature learning  
☐ d. To provide a copy of the output data

## c. To represent data in a lower-dimensional space for feature learning.

Il "latent space" in un autoencoder è una rappresentazione compressa dei dati di input in uno spazio di dimensioni ridotte. Il suo scopo principale è estrarre caratteristiche o pattern rilevanti dai dati, facilitando il processo di apprendimento delle caratteristiche più importanti. Questa rappresentazione ridotta può essere utilizzata per compiti come la riduzione della dimensionalità, la scoperta di nuove caratteristiche, la compressione dei dati e altre applicazioni di apprendimento non supervisionato.

Question 11

Not yet answered

Marked out of 1.00

Flag question

Early stopping is a form of regularization that prevents overfitting by stopping training as soon as validation loss starts to decrease

- ☐ True
- ☐ False

**False.**

L'early stopping è una tecnica di regolarizzazione che previene l'overfitting interrompendo l'allenamento di un modello quando il **loss di validazione inizia a peggiorare**, non a diminuire. In altre parole, il training si interrompe quando il modello comincia a mostrare segni di sovra-adattamento ai dati di training, ossia quando il loss di validazione smette di migliorare o inizia ad aumentare, indicando che il modello sta perdendo la sua capacità di generalizzare sui dati non visti.

Question 12

Not yet answered

Marked out of 1.00

Flag question

The standard error of the estimated mean of the gradient decreases less than linearly with the number of samples used.

- ☐ True
- ☐ False

**True.**

L'errore standard della stima della media del gradiente diminuisce con l'aumentare del numero di campioni utilizzati, ma questa diminuzione avviene a una velocità **meno che lineare**. Infatti, l'errore standard della media diminuisce proporzionalmente all'inverso della radice quadrata del numero di campioni ( $1/\sqrt{n}$ ). Questo significa che, per ottenere riduzioni significative dell'errore standard, è necessario un incremento molto grande nel numero di campioni, riflettendo una relazione sub-lineare.

Question 13

Not yet answered

Marked out of 1.00

Flag question

Weight adjustments in machine learning are aimed at learning a good separation function

- ☐ True
- ☐ False

**True.**

In molti algoritmi di machine learning, in particolare nei modelli supervisionati come le reti neurali, l'obiettivo dell'aggiornamento dei pesi è quello di apprendere una funzione di separazione o una funzione di decisione che distingua correttamente le diverse classi o realizzi accurate predizioni. Durante il processo di apprendimento, i pesi vengono regolati per minimizzare l'errore tra le predizioni del modello e i valori target, portando il modello a trovare un confine o una funzione di separazione che meglio approssima la relazione tra gli input e le etichette o risultati desiderati.

Question **14**

Not yet answered

Marked out of 1.00

🚩 Flag question

The sigmoid function has a sensitive gradient when  $z$  is close to zero

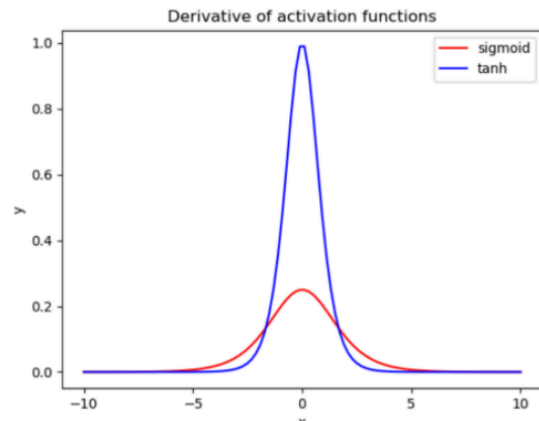
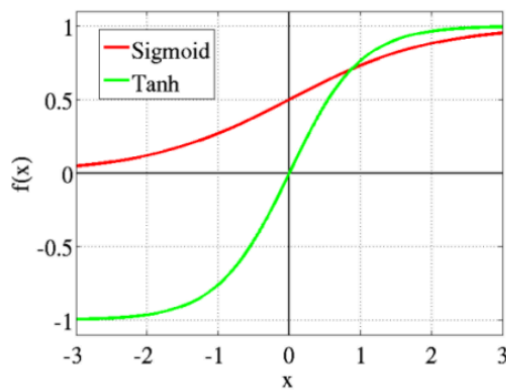
☐ True

☐ False

**True.**

La funzione sigmoide ha una pendenza molto ripida quando il valore di input  $z$  è vicino a zero, il che significa che il gradiente è più sensibile in questa regione. Quando  $z$  si trova vicino a zero, il valore della derivata della funzione sigmoide è massimo, indicando che piccoli cambiamenti in  $z$  comportano cambiamenti significativi nell'uscita della funzione. Questa sensibilità consente di ottenere aggiornamenti significativi dei pesi durante l'apprendimento. Tuttavia, al di fuori di questa regione (quando  $z$  è molto grande o molto piccolo), il gradiente si appiattisce, portando al problema del vanishing gradient.





Question 15

Not yet answered

Marked out of 1.00

Flag question

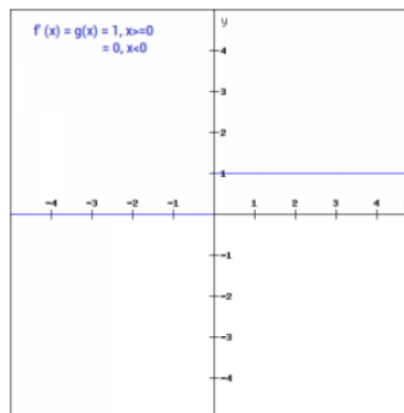
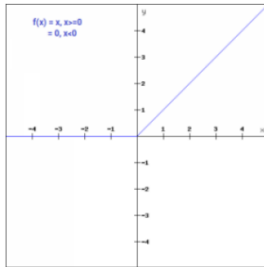
The advantage of the ReLU functions is that its gradient is constant for every value of  $z$

☐ True

☒ False

**False.**

La funzione ReLU (Rectified Linear Unit) non ha un gradiente costante per ogni valore di  $z$ . Il gradiente di ReLU è pari a 1 per valori di input positivi ( $z > 0$ ) e 0 per valori di input negativi ( $z \leq 0$ ). Questo significa che la funzione è lineare (con pendenza unitaria) per valori positivi, mentre per i valori negativi non si attiva (il gradiente è nullo). Questa caratteristica rende ReLU utile per la propagazione del gradiente nei modelli di deep learning, ma non implica un gradiente costante per tutti i valori di  $z$ .



Question **16**

Not yet answered

Marked out of 1.00

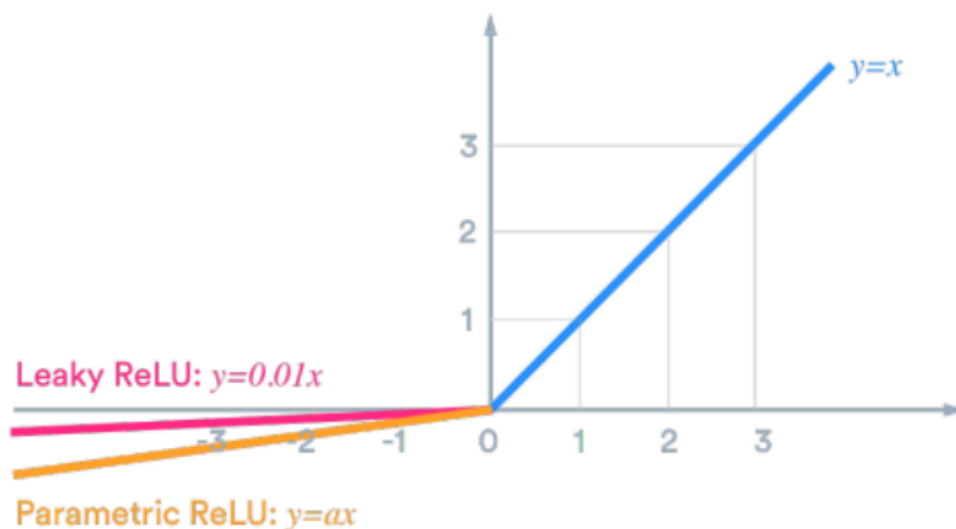
Flag question

Leaky ReLUs saturates when  $z < 0$

- ☐ True
- ☐ False

## False

Le Leaky ReLU non saturano quando  $z < 0$ . A differenza della ReLU standard, che restituisce 0 per valori negativi di  $z$ , la Leaky ReLU consente un piccolo gradiente negativo (tipicamente  $\alpha z$ , dove  $\alpha$  è un piccolo valore positivo) per  $z < 0$ , evitando la saturazione.



Question 17

Not yet answered

Marked out of 1.00

Flag question

Kernel functions implicitly map features into a higher-dimensional space

☐ True

☐ False

## True

Le funzioni kernel mappano implicitamente le caratteristiche in uno spazio di dimensioni superiori senza la necessità di calcolare esplicitamente questa mappatura. Questo è il principio alla base del metodo del **kernel trick** utilizzato in tecniche come le **Support Vector Machines (SVM)**, dove il kernel permette di calcolare i prodotti scalari nello spazio di dimensioni superiori senza dover eseguire la trasformazione diretta dei dati.

Question 18

Not yet answered

Marked out of 1.00

Flag question

The softmax function gives as output a probability distribution that can always be interpreted as a confidence level

☐ True

☐ False

## False

????

Anche se la funzione softmax restituisce una distribuzione di probabilità, non sempre può essere interpretata come un "livello di fiducia". In alcuni casi, specialmente in contesti con classi sbilanciate o con modelli non ben allenati, la probabilità più alta potrebbe non riflettere accuratamente la fiducia del modello. La softmax calcola solo le probabilità relative tra le classi, ma non garantisce che il valore massimo sia sempre un'indicazione affidabile di una previsione corretta. Quindi, sebbene la softmax produca probabilità, non sempre queste possono essere interpretate come livelli di fiducia nel senso stretto del termine.

Question 19

Not yet answered

Marked out of 1.00

Flag question

Regularizing estimators may increase the gap between training error and validation error

☐ True

☐ False

**False**

???

La regolarizzazione riduce il rischio di overfitting, migliorando la capacità di generalizzazione del modello. Questo spesso porta a un aumento dell'errore di addestramento (perché il modello è meno complesso), ma una riduzione dell'errore di validazione. In generale, la regolarizzazione riduce il divario tra errore di addestramento ed errore di validazione, non lo aumenta.

Question 20

Not yet answered

Marked out of 1.00

Flag question

Maxout can be used to approximate a convex function

☐ True

☐ False

**True**

Maxout è una funzione di attivazione che può essere utilizzata nelle reti neurali. Funziona prendendo il massimo tra un insieme di input, il che le permette di approssimare funzioni convesse.

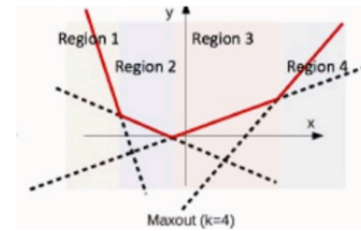
In particolare, Maxout è una funzione **lineare a tratti**. Utilizzando più funzioni lineari e selezionando il massimo tra di esse, può approssimare qualsiasi funzione convessa. Questo avviene perché una funzione convessa può essere espressa come una combinazione di tratti lineari, e Maxout è in grado di creare queste combinazioni lineari.

**Maxout units** : It learns the activation function itself

It divides  $z$  in groups of  $k$  values and each maxout unit outputs the max element of one of these groups

$$g(z)_i = \max_{j \in G^{(i)}} z_j$$

$G^{(i)}$  = Set of indices of the group  $i$



Question **21**

Not yet answered

Marked out of 1.00

Flag question

When penalizing the norm, L1 results in more sparse weights than L2

☐ True

☐ False

**True**

Quando si penalizza la norma dei pesi nei modelli di machine learning, la regolarizzazione **L1** tende a produrre **pesi sparsi**, cioè molti pesi vengono ridotti a zero. Questo accade perché la penalizzazione L1 ha un effetto che "spinge" i pesi verso zero, portando a una selezione delle variabili più significative.

Al contrario, la regolarizzazione **L2** penalizza i pesi in modo che siano piccoli, ma non li porta a zero, quindi tende a ridurre i pesi in modo uniforme piuttosto che renderli esattamente nulli.

Question **22**

Not yet answered

Marked out of 1.00

Flag question

What is the main purpose of an autoencoder?

- ☐ a. To predict future data points based on input
- ☐ b. To map the input directly to an output without transformation
- ☐ c. To classify data into predefined categories
- ☐ d. To attempt to copy its input to its output, focusing on significant data features

**d. "To attempt to copy its input to its output, focusing on significant data features"**

Un **autoencoder** è una rete neurale utilizzata principalmente per **apprendere una rappresentazione compatta** (o codifica) dei dati, cercando di "copiarli" nel miglior modo possibile, ma concentrandosi sulle caratteristiche più significative. È composto da due parti: un **encoder** che comprime i dati e un **decoder** che cerca di ricostruire l'input originale a partire dalla rappresentazione compressa.

- **a.** "To predict future data points based on input": Questo descrive più il **concetto di rete neurale predittiva** (ad esempio, per le serie temporali), ma non è lo scopo di un autoencoder.
- **b.** "To map the input directly to an output without transformation": Un autoencoder trasforma effettivamente l'input (comprimi e ricostruisci) e non lo mappa direttamente senza trasformazione.
- **c.** "To classify data into predefined categories": Questo è lo scopo di una **rete neurale di classificazione**, non di un autoencoder, che non si occupa di classificazione ma di ricostruzione dei dati.

Question **23**

Not yet answered  
Marked out of 1.00

Flag question

Rectified Linear Unit (ReLU) is a suitable choice as a hidden unit because it is continuous

- ☐ True  
☐ False

**False**

La **Rectified Linear Unit (ReLU)** è una funzione di attivazione comunemente utilizzata nelle reti neurali, ma la sua principale caratteristica non è la continuità. In effetti, **ReLU** non è continua in tutti i punti, poiché ha una discontinuità in  $x=0$  (passa improvvisamente da 0 a una pendenza di 1).

La principale ragione per cui ReLU è una scelta popolare come unità nascosta è che:

- È **computazionalmente semplice** (molto più veloce da calcolare rispetto ad altre funzioni come sigmoide o tanh).
- È **non lineare**, permettendo alla rete di apprendere rappresentazioni complesse dei dati.

- È in grado di **ridurre il problema del vanishing gradient** che affligge altre funzioni di attivazione come la sigmoide o tanh.

Quindi, mentre ReLU è utile e ampiamente adottata, il motivo per cui è una scelta comune non riguarda la continuità, ma piuttosto la sua capacità di migliorare l'apprendimento e la sua semplicità computazionale.

Question **24**

Not yet answered

Marked out of 1.00

Flag question

During the learning process, if the Hessian matrix of the cost function is ill-conditioned, even very small steps can lead to an increase in the cost function

☒ True

☐ False

**True**

La **matrice Hessiana** è una matrice che contiene tutte le seconde derivate parziali di una funzione e viene utilizzata per analizzare la curvatura della funzione, in particolare nel contesto dell'ottimizzazione.

Se la matrice Hessiana della funzione di costo è **mal condizionata**, significa che ci sono direzioni in cui la funzione cambia molto più rapidamente rispetto ad altre. In questo caso, anche piccole variazioni nei parametri possono portare a cambiamenti **grandi e imprevedibili** nel valore della funzione di costo, soprattutto in direzioni strette o "appiattite".

- Quando la matrice Hessiana è mal condizionata, i gradienti in alcune direzioni possono essere molto piccoli, mentre in altre direzioni possono essere molto grandi, causando **instabilità** nell'ottimizzazione.
- Anche **piccole modifiche** nei parametri possono quindi portare a **un aumento del costo** anziché una riduzione, perché la direzione in cui il modello sta facendo aggiornamenti potrebbe non essere quella giusta.

Quindi, la risposta è **vera**: una matrice Hessiana mal condizionata può causare che anche **piccole modifiche** portino a un **aumento** della funzione di costo.

Question **25**

Not yet answered

Marked out of 1.00

🚩 Flag question

Momentum update rule accumulates previous values of the gradient

☒ True

☐ False

## True

La regola di aggiornamento con **momentum** è una tecnica utilizzata nell'ottimizzazione per accelerare la convergenza e ridurre le oscillazioni durante l'aggiornamento dei pesi.

Il **momentum** accumula una frazione del gradiente precedente, in modo che l'aggiornamento corrente dipenda non solo dal gradiente attuale, ma anche dalle informazioni sui gradienti precedenti. Questo aiuta a "smussare" l'andamento degli aggiornamenti, facendo in modo che il modello faccia passi più consistenti verso la direzione di discesa del gradiente, specialmente in presenza di variazioni rumorose o forti oscillazioni.

**In a neural network the nonlinearity causes the most interesting loss function to become non convex**

🏆 Results

You answered:

True

False



The correct answer was



True


## Vero



In una rete neurale, le funzioni di attivazione non lineari (come ReLU, sigmoide, tanh, ecc.) sono utilizzate per introdurre complessità e permettere al modello di apprendere relazioni non lineari tra gli input e gli output. Tuttavia, questa non linearità porta la funzione di perdita complessiva a essere tipicamente **non convessa**, il che significa che la superficie della funzione di perdita può avere molteplici minimi locali o globali, rendendo il processo di ottimizzazione più complicato rispetto a un problema convesso.

**The loss function produces a numerical score that also depends on the set of parameters  $\theta$  which characterizes the FFN model**

 Congratulations! 

True 

False

### Vero

La funzione di perdita in un modello di rete neurale feed-forward (FFN) dipende dal set di parametri  $\theta$ , che includono i pesi e i bias del modello. Durante l'addestramento, l'obiettivo è minimizzare la funzione di perdita modificando questi parametri per migliorare le prestazioni del modello.

**Regularization functions are added to the loss functions to reduce their training error**

True

False

### Falso

Le funzioni di regolarizzazione non sono progettate per ridurre l'errore di addestramento, ma piuttosto per **ridurre l'overfitting** del modello, migliorando le sue prestazioni su dati non visti (generalizzazione). Queste funzioni aggiungono un termine di penalità alla funzione di perdita per evitare che il modello diventi troppo complesso o aderisca eccessivamente ai dati di addestramento.

**The gradient can be estimated through an iterative procedure that uses at each iteration only a sample of training examples**

True

False


### Vero

Il gradiente può essere stimato efficacemente utilizzando solo un campione casuale dei dati di training ad ogni iterazione (mini-batch), invece di dover

utilizzare l'intero dataset. Questo approccio, chiamato SGD (Stochastic Gradient Descent), rende l'addestramento più veloce ed efficiente, specialmente con grandi quantità di dati.

---

### Which of the following statements are true ?

 You can select multiple choices

When using SGD with mini-batches the model updates do not depend on the number of training examples

When using SGD with mini-batches the number of updates to reach convergence does not depend on the number of training examples

Once the SGD converges it is still useful to add more training examples

The choice of cost functions is tightly coupled with the choice of the output unit

**1 vera** ma ambigua. in SGD con mini-batch, ogni aggiornamento del modello dipende solo dalla dimensione del mini-batch che abbiamo scelto, non dal numero totale di esempi nel dataset di training. L'ambiguità potrebbe sorgere perché, sebbene il singolo aggiornamento non dipenda dal numero totale di esempi, il processo di training complessivo (numero di epoche, convergenza, etc.) ne è influenzato.

**2 falsa** perchè gli aggiornamenti dipendono dalla dimensione del dataset, poiché influenza il gradiente calcolato.

**3 falsa** perchè se l'SGD ha già raggiunto la convergenza, aggiungere altri dati non porterebbe benefici significativi senza riaddestramento.

**4 vera** perchè la funzione di costo deve essere appropriata al tipo di output che vogliamo ottenere (per esempio, cross-entropy per classificazione binaria, MSE per regressione).