

Wooclaps

L5 Regolarizzazione

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

The loss function produces a numerical score that also depends on the set of parameters θ which characterizes the FFN model



Congratulations!



True



False

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

True

False

Il training error è solo sugli esempi di training, se aggiungiamo questo termine è per ottimizzare l'errore sugli esempi non visti nel training set, quindi per non fare

overfitting.

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

True

False

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

risposta: 1 e 4. 1: Quando abbiamo tanti esempi, se usiamo mini batch potremmo convergere prima di usare tutti gli esempi, l'update dipende solo dal mini batch e non rispetto a tutto il training set perchè l'update accade rispetto al mini batch.

2: la risposta non è chiara, dipende dal tipo di dati di training che abbiamo. Non ho capito il resto della spiegazione

3: se il gradiente ha raggiunto la convergenza non abbiamo bisogno di altri esempi

Simulazione parte 1

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

4

- 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

<

4

>

- 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.

<

4

>

- 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**

peggiорare, 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.

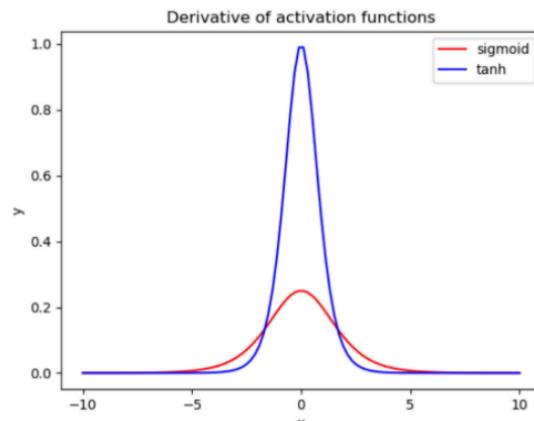
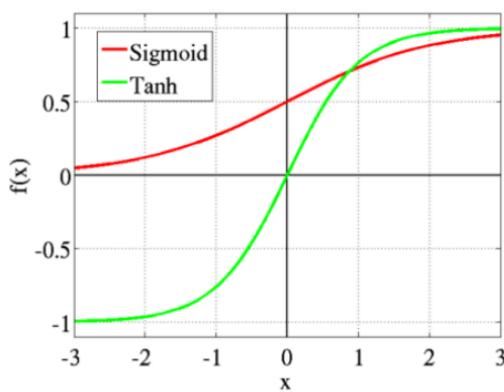
Question 14

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.

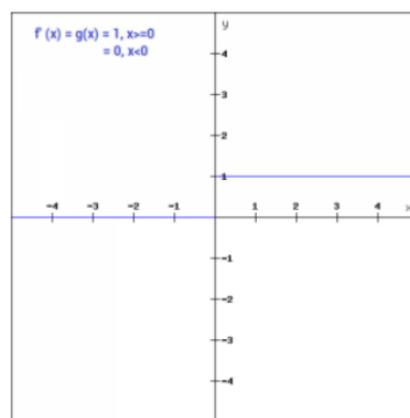
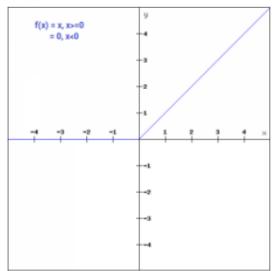
**Question 15**

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

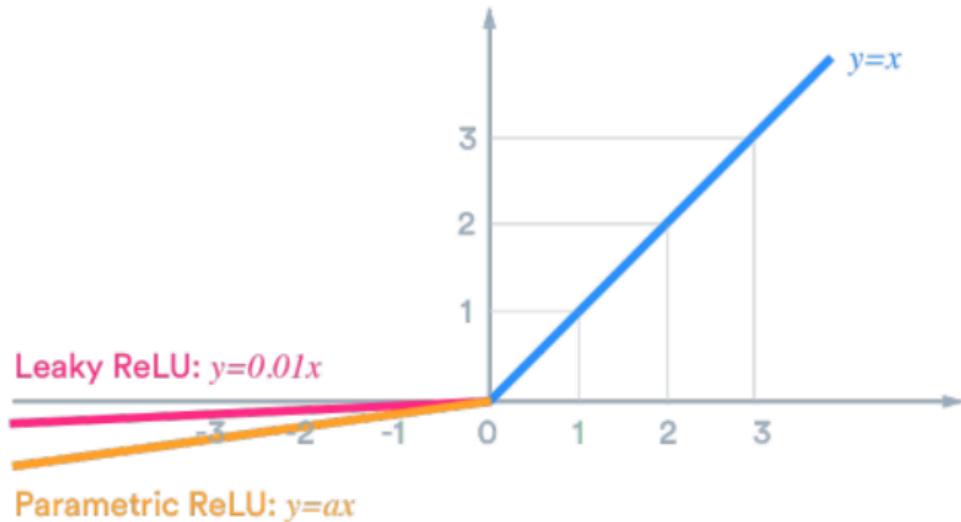
Leaky ReLUs saturates when $z < 0$



- 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 αz , dove α 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

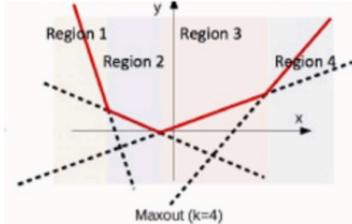
convessa. Questo avviene perché una funzione convessa può essere espressa come una combinazione di tratti lineari, e Maxout è in grado di creare queste combinazioni lineari.

Maxout units : It learns the activation function itself

It divides \mathbf{z} in groups of k values and each maxout unit outputs the max element of one of these groups

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

$G^{(i)}$ = 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 $x=0$ ($y=0$ per $x < 0$ e $y=x$ per $x \geq 0$).

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.

Altre domande viste a lezione

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 θ 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 θ , 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 gli updates dipendono dagli esempi nel mini batch

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. A meno che gli esempi che avanzano sono molto diversi da quelli visti, quindi se per esempio non ho mischiato i dati randomizzando.

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).

Altro da telegram sulla prima parte

Algorithms for weight adjustment in a classification network are made to linearly separate the points belonging to the two classes

Results

You answered

True X

False

The image shows a Wooclap poll interface. At the top, a statement is displayed in bold blue text: "Algorithms for weight adjustment in a classification network are made to linearly separate the points belonging to the two classes". Below this, the word "Results" is followed by a small icon. A horizontal line separates this from the "You answered" section. In this section, there are two options: "True" and "False". The "True" option is highlighted with a green background and has a red circle with a white "X" to its right, indicating it was selected but is incorrect. The "False" option is in a standard grey box. The entire interface is set against a light grey background.

Mentre alcuni algoritmi di classificazione, come il percepitrone lineare, cercano di trovare una separazione lineare, molti altri, specialmente quelli utilizzati in contesti di deep learning, sono capaci di apprendere separazioni non lineari, rendendo l'affermazione iniziale falsa nel caso generale.

The image shows a Wooclap poll interface. At the top, it says "In initialization". Below that, there is a note: "You can select multiple choices". There are four options listed in separate boxes:

- Larger weights break symmetry more
- Smaller weights propagate information more efficiently
- Large weights make the model more likely to reach solutions with good generalization property
- Small weights make the model more robust

At the bottom right is a blue "Submit" button with a white arrow icon.

Io metterei la prima vera perchè pesi più grandi, anche se con piccole differenze, fanno in modo che poi i valori calcolati possano variare di più (però con pesi non troppo grandi, altrimenti hai exploding gradient)

La seconda vera perchè pesi più piccoli sono più stabili, non hai exploding gradient, però anche qua se sono troppo piccoli hai vanishing gradient

La terza non so, la metterei falsa, perchè avere pesi grandi causa instabilità, però questa instabilità potrebbe essere vista come "generalizzazione" visto che il modello non impara bene dai dati, però andrei più sul falso

La quarta vera perchè appunto pesi piccoli sono più stabili, però cazzo significa "robusto", troppo generale

"Larger weights break symmetry more" is **true** - this helps avoid the problem of neurons learning the same features

Pesi troppo piccoli possono causare il problema del vanishing gradient (gradient che svanisce), dove i gradienti diventano estremamente piccoli durante la retropropagazione, rallentando o bloccando l'apprendimento. Tuttavia, pesi

eccessivamente grandi possono causare il problema dell'*exploding gradient* (gradiente che esplode), rendendo instabile il training. Quindi, è importante trovare un giusto mezzo. Inizializzare con pesi piccoli ma non troppo è generalmente una buona pratica, ma "più efficiente" è un'affermazione troppo generica.

The last two statements are actually incorrect:

- Large weights typically harm generalization, not help it
- Small weights don't necessarily make the model more robust by themselves In un certo senso, sì. Pesi piccoli possono contribuire a una maggiore stabilità del training, riducendo la probabilità di oscillazioni e di convergenza a minimi locali non ottimali. Inoltre, come accennato prima, aiutano a prevenire l'*exploding gradient*. Tuttavia, "più robusto" è un concetto ampio. In questo contesto, si riferisce principalmente alla stabilità del training e alla prevenzione di problemi di ottimizzazione.

A screenshot of a Wooclap poll interface. The top bar is blue with the Wooclap logo on the left and a settings gear icon on the right. The main area has a white background with a title and three options in rounded rectangular boxes. At the bottom is a blue 'Submit' button with a white arrow icon.

A good initialization procedure

assignes large weights

assignes extremely small weights

none of the two

Submit

none of the two



Adam algorithm

ⓘ You can select multiple choices

it also takes into account the curvature of the cost function through the second order derivates

it uses the momentum strategy

it is based on RMSProp

Submit

- **It uses the momentum strategy.**
- **It is based on RMSProp.**

RMSProp

ⓘ You can select multiple choices

it takes into account the square gradients

it is a modification of the AdaGrad algorithm

it does not require hyperparameters

Submit

- "**it takes into account the square gradients**" (**prende in considerazione i gradienti al quadrato**): **VERO**. RMSProp utilizza una media mobile esponenziale dei *quadrati* dei gradienti per scalare il learning rate per ogni parametro. Questo è il meccanismo chiave di RMSProp che lo differenzia da altri ottimizzatori.
- "**it is a modification of the AdaGrad algorithm**" (**è una modifica dell'algoritmo AdaGrad**): **VERO**. RMSProp è stato proposto come una soluzione al problema del learning rate in continua diminuzione di AdaGrad. AdaGrad accumula la somma dei quadrati di tutti i gradienti passati, il che può portare il learning rate a diventare troppo piccolo troppo velocemente, bloccando l'apprendimento. RMSProp introduce una media mobile esponenziale per i quadrati dei gradienti, dando più peso ai gradienti recenti e mitigando questo problema.
- "**it does not require hyperparameters**" (**non richiede iperparametri**): **FALSO**. RMSProp richiede almeno un iperparametro, solitamente indicato con β (beta) o ρ (rho), che controlla il tasso di decadimento della media mobile

esponenziale. Questo iperparametro determina quanto peso viene dato ai gradienti recenti rispetto a quelli passati. Solitamente, un valore comune per β è 0.9. Inoltre, come tutti gli ottimizzatori basati su gradienti, RMSProp richiede un learning rate.

The image shows a Wooclap poll interface. At the top, there's a blue header bar with the Wooclap logo on the left and a settings icon on the right. Below the header, the title "AdaGrad algorithm" is centered. A note below the title says "You can select multiple choices". There are three options listed in separate boxes: "takes into account previous squared gradients", "decreases the learning rate too much in the early stages", and "it uses the same learning rate for all parameters". At the bottom of the poll is a blue "Submit" button with a white arrow icon.

- "**takes into account previous squared gradients**" (**prende in considerazione i gradienti al quadrato precedenti**): **VERO**. AdaGrad accumula la somma dei quadrati di *tutti* i gradienti passati per ogni parametro. Questa somma viene poi utilizzata per scalare il learning rate di quel parametro.
- "**decreases the learning rate too much in the early stages**" (**diminuisce troppo il learning rate nelle fasi iniziali**): **FALSO (con una precisazione)**. AdaGrad *non* diminuisce eccessivamente il learning rate *nelle fasi iniziali*. Il problema principale di AdaGrad è che *accumulando* i quadrati dei gradienti nel tempo, il learning rate continua a *diminuire* e può diventare *troppo piccolo troppo velocemente*, specialmente nelle fasi successive del training, bloccando di fatto l'apprendimento. Quindi, è più corretto dire che diminuisce

troppo il learning rate *nel corso del tempo*, non specificamente nelle fasi iniziali.

- "**it uses the same learning rate for all parameters**" (**usa lo stesso learning rate per tutti i parametri**): **FALSO**. Una delle caratteristiche principali di AdaGrad è che adatta il learning rate *individualmente per ogni parametro*. I parametri che hanno ricevuto gradienti più grandi nel passato avranno un learning rate più piccolo, mentre quelli con gradienti più piccoli avranno un learning rate più grande.

The image shows a Wooclap poll interface. At the top, it says "wooclap". Below that is a blue header bar with a gear icon. The main area has a title "Momentum update rule" and a note "You can select multiple choices". There are three options in boxes:

- accumulates previous values of the cost function
- can be incorporated in SGD
- its step size is larger if previous gradients point in the same direction

A "Submit" button is at the bottom.

- "**accumulates previous values of the cost function**" (**accumula i valori precedenti della funzione di costo**): **FALSO**. Il momentum non accumula i valori della funzione di costo. Accumula una media mobile dei *gradienti* precedenti. Questa media mobile determina la "velocità" con cui ci si muove nello spazio dei parametri.
- "**can be incorporated in SGD**" (**può essere incorporato in SGD**): **VERO**. Il momentum è spesso usato in combinazione con la Stochastic Gradient

Descent (SGD). L'SGD di base può essere soggetto a oscillazioni, specialmente in presenza di rumore o di una superficie di errore complessa. L'aggiunta del momentum aiuta a smussare queste oscillazioni e ad accelerare la convergenza.

- "**its step size is larger if previous gradients point in the same direction**" (**il suo passo è più grande se i gradienti precedenti puntano nella stessa direzione**): **VERO**. Questa è l'idea chiave del momentum. Se i gradienti in iterazioni successive puntano approssimativamente nella stessa direzione, il momentum accumula "velocità" in quella direzione, aumentando la dimensione del passo e accelerando la convergenza. Al contrario, se i gradienti cambiano direzione frequentemente, il momentum smorza le oscillazioni.

The image shows a Wooclap poll interface. At the top, it says "wooclap". Below that is a blue header bar with a gear icon. The main area has a title "Neural networks with many layers". A note says "You can select multiple choices". There are three options in boxes:

- often have extremely steep regions (cliffs)
- often have flat regions
- often have many local minima of similar cost

A "Submit" button is at the bottom.

- "**often have extremely steep regions (cliffs)**" (**spesso hanno regioni estremamente ripide (precipizi)**): **VERO**. Le reti neurali profonde, a causa della loro complessità e del gran numero di parametri, possono presentare regioni nel loro spazio dei parametri in cui la funzione di costo cambia drasticamente anche per piccole variazioni dei parametri. Queste regioni sono

chiamate "precipizi" (cliffs) e possono causare problemi durante l'ottimizzazione, come l'esplosione del gradiente (exploding gradient) o difficoltà nella convergenza.

- **"often have flat regions" (spesso hanno regioni piatte): VERO.** Oltre ai precipizi, le reti neurali profonde possono anche presentare regioni relativamente piatte nella funzione di costo, dove il gradiente è molto piccolo. Queste regioni possono rallentare significativamente l'apprendimento, poiché l'ottimizzatore fa fatica a trovare una direzione di discesa. Questo è legato al problema del vanishing gradient (gradiente che svanisce).
- **"often have many local minima of similar cost" (spesso hanno molti minimi locali di costo simile): VERO.** A causa della loro elevata dimensionalità e non-convessità, le funzioni di costo delle reti neurali profonde tendono ad avere molti minimi locali. Fortunatamente, è stato osservato che molti di questi minimi locali hanno valori di costo simili, quindi anche se l'ottimizzatore non trova il minimo globale, può comunque convergere a una soluzione che generalizza bene.

The image shows a Wooclap poll interface. At the top, there's a blue header bar with the Wooclap logo on the left and a settings icon on the right. Below the header, the title of the poll is "Local minima in deep learning problems". A note says "You can select multiple choices". There are three options listed in boxes:

- are rare
- are more common than saddle points
- are much more likely to have a low cost than a high cost

A "Submit" button with a blue arrow icon is at the bottom. The background of the poll area is white.

- "are rare" (sono rari): vero
- "are more common than saddle points" (sono più comuni dei punti di sella): FALSO c'era nella teoria se ricordo bene, ad alta dimensionalità ci sono più saddle points
- "are much more likely to have a low cost than a high cost" (hanno molta più probabilità di avere un costo basso che un costo alto): VERO. Questa è un'osservazione importante. Anche se ci sono molti minimi locali, la ricerca ha mostrato che molti di questi minimi locali hanno valori di costo simili e relativamente bassi. Questo significa che anche se l'algoritmo di ottimizzazione non trova il minimo globale, è probabile che converga a un minimo locale che fornisce comunque una buona performance.

The image shows a Wooclap poll interface. At the top, it says "wooclap". Below that is a blue header bar with a gear icon. The main question is "ILL conditioning of the Hessians matrix of the cost function". A note below the question says "You can select multiple choices". There are three options in boxes:

- can be partly overcome by using the momentum strategy
- can prevent the gradient to arrive to a critical point
- can imply the very small steps are needed to decrease the cost function

 At the bottom is a blue "Submit" button with a white arrow icon.

- "can be partly overcome by using the momentum strategy" (può essere parzialmente superato usando la strategia del momentum): VERO. Il mal condizionamento dell'Hessiana significa che la curvatura della funzione di costo varia molto in direzioni diverse. Questo può portare l'ottimizzazione a

oscillare e a convergere lentamente. Il momentum aiuta a mitigare questo problema accumulando "velocità" nella direzione del gradiente, smorzando le oscillazioni e consentendo di attraversare regioni con curvatura diversa in modo più efficiente.

- "**can prevent the gradient to arrive to a critical point**" (può impedire al gradiente di arrivare a un punto critico): **FALSO**. Il mal condizionamento dell'Hessiana *non impedisce* al gradiente di arrivare a un punto critico (minimo, massimo o punto di sella). Un punto critico è definito come un punto in cui il gradiente è zero. Il mal condizionamento influenza la *velocità* e la *direzione* con cui l'ottimizzazione si avvicina a un punto critico, ma non impedisce di raggiungerlo.
- "**can imply the very small steps are needed to decrease the cost function**" (può implicare che siano necessari passi molto piccoli per diminuire la funzione di costo): **VERO o falso??**. Un Hessiano mal condizionato può avere autovalori con ordini di grandezza molto diversi. Questo significa che la funzione di costo varia molto rapidamente in alcune direzioni e molto lentamente in altre. Per garantire la stabilità dell'ottimizzazione e per evitare di "saltare" oltre il minimo, è spesso necessario utilizzare learning rate molto piccoli, che corrispondono a passi molto piccoli nella discesa del gradiente.



The accuracy of the estimated mean of the gradient

ⓘ You can select multiple choices

it does not depend on the number of samples used

has a standard error which decreases linearly with the number of samples used

it also depend on redundancies in sample data

Submit

- "**it does not depend on the number of samples used**" (**non dipende dal numero di campioni utilizzati**): **FALSO**. L'accuratezza della stima del gradiente *dipende* fortemente dal numero di campioni utilizzati. In generale, più campioni si usano per stimare il gradiente, più precisa sarà la stima.
- "**has a standard error which decreases linearly with the number of samples used**" (**ha un errore standard che diminuisce linearmente con il numero di campioni utilizzati**): **FALSO**. L'errore standard della media del gradiente diminuisce con la *radice quadrata* del numero di campioni, non linearmente. Questo significa che per dimezzare l'errore standard, è necessario quadruplicare il numero di campioni. Più precisamente, l'errore standard è proporzionale a $1/\sqrt{n}$, dove n è il numero di campioni.
- "**it also depend on redundancies in sample data**" (**dipende anche dalle ridondanze nei dati campione**): **VERO**. Se i dati campione contengono molte ridondanze (cioè, campioni molto simili tra loro), l'aggiunta di ulteriori campioni non porterà a un significativo miglioramento dell'accuratezza della

stima del gradiente. In altre parole, se i dati sono altamente correlati, l'informazione aggiuntiva fornita da nuovi campioni è limitata.

The image shows a Wooclap poll interface. At the top, the Wooclap logo is visible. Below it, the title "Early stopping halt criterion" is centered. Two options are presented in separate boxes: "is typically based on the performance obtained on a validation set" and "is typically based on the performance obtained on the training set". At the bottom right is a blue "Submit" button with a white arrow icon.

- "**is typically based on the performance obtained on a validation set**" (**si basa tipicamente sulla performance ottenuta su un set di validazione**): **VERO**. L'early stopping è una tecnica di regolarizzazione che mira a prevenire l'overfitting. Consiste nel monitorare la performance del modello su un set di validazione durante l'addestramento. L'addestramento viene interrotto quando la performance sul set di validazione smette di migliorare (o inizia a peggiorare), anche se la performance sul set di training continua a migliorare.
- "**is typically based on the performance obtained on the training set**" (**si basa tipicamente sulla performance ottenuta sul set di training**): **FALSO**. Basare l'early stopping sulla performance del set di training sarebbe controproducente. Infatti, la performance sul set di training tende a migliorare costantemente durante l'addestramento, anche quando il modello sta iniziando a sovradattarsi ai dati di training e quindi a generalizzare male su dati nuovi.



A surrogate loss function

acts as a proxy to the true risk being "nice" enough to be optimized efficiently

acts as a proxy to empirical risk being "nice" enough to be optimized efficiently

Submit

The second option is correct: a surrogate loss function "acts as a proxy to empirical risk being 'nice' enough to be optimized efficiently."

A surrogate loss function is used in machine learning as a substitute for the original (empirical) risk function when that original function is difficult to optimize directly. For example, in classification, we often can't directly optimize accuracy (0-1 loss) because it's not differentiable, so we use surrogate losses like cross-entropy or hinge loss that are easier to work with mathematically while still approximating what we really want to optimize.

The first option mentions "true risk" rather than "empirical risk" - this is incorrect because surrogate losses actually approximate the empirical risk (calculated on our training data), not the true risk (which involves the unknown true data distribution).



The final aim of a Machine Learning Model is

the minimization of the true risk function

the minimization of the empirical risk using a surrogate loss function

Submit

The correct answer is:

- **The minimization of the true risk function**

This is the ultimate goal of a machine learning model. However, since the true risk is generally unknown, we approximate it by minimizing the **empirical risk using a surrogate loss function**, which is a practical step toward achieving the minimization of the true risk.



In Machine Learning the Cost function to minimize during the training process is the performance measure P representing the number of correct classifications on the test set

True

False

Submit

La risposta è **Falso**.

La funzione di costo (o funzione di perdita) viene minimizzata durante il *processo di addestramento* utilizzando il *set di training*. L'obiettivo è che il modello impari a generalizzare bene, ovvero a fare predizioni accurate su dati nuovi, che non ha visto durante l'addestramento.

Il set di *test* viene utilizzato *solo alla fine* dell'addestramento, per valutare le performance finali del modello addestrato e stimare quanto bene generalizza a dati nuovi. Non viene utilizzato durante l'addestramento per minimizzare la funzione di costo.

Which of the following sentences are true ?

ⓘ You can select multiple choices

- Parameter Tying impose to a subset of parameters to be equal
- Early stopping is a form of regularization
- Bagging is a form of ensemble model
- Bagging is more effective if the output of models learned are correlated
- Dropout is generally coupled with minibatch-based learning algorithm

 Submit

- "**Parameter Tying impose to a subset of parameters to be equal**" (**Il legame dei parametri impone a un sottoinsieme di parametri di essere uguali**): **VERO.** Il "parameter tying" (legame dei parametri) è una tecnica in cui si costringono alcuni parametri di un modello ad avere lo stesso valore. Questo riduce il numero di parametri indipendenti e può aiutare a prevenire l'overfitting, oltre a poter avere altre motivazioni a seconda del contesto (es. riduzione della complessità computazionale, forzare simmetrie). Un esempio classico è nelle reti Siamese, dove le stesse reti (con gli stessi pesi) vengono usate per processare input diversi.
- "**Early stopping is a form of regularization**" (**L'early stopping è una forma di regolarizzazione**): **VERO.** L'early stopping è una tecnica di regolarizzazione che interrompe l'addestramento di un modello quando la sua performance su un set di validazione smette di migliorare. Questo previene l'overfitting, ovvero la tendenza del modello ad adattarsi troppo ai dati di training, peggiorando la sua capacità di generalizzare a dati nuovi.

- "**Bagging is a form of ensemble model**" (**Il bagging è una forma di modello ensemble**): **VERO.** Il "bagging" (Bootstrap Aggregating) è una tecnica di ensemble learning che consiste nell'addestrare più modelli (tipicamente dello stesso tipo) su sottoinsiemi diversi del dataset di training, ottenuti tramite campionamento con reinserimento (bootstrap). Le predizioni dei singoli modelli vengono poi aggregate (ad esempio tramite media o voto a maggioranza) per ottenere la predizione finale.
- "**Bagging is more effective if the output of models learned are correlated**" (**Il bagging è più efficace se l'output dei modelli appresi è correlato**): **FALSO.** Il bagging è *meno* efficace se l'output dei modelli è correlato. L'efficacia del bagging deriva proprio dalla diversità tra i modelli addestrati. Se i modelli producono predizioni molto simili, l'aggregazione non porta a un significativo miglioramento delle performance. L'obiettivo del bagging è creare modelli il più possibile indipendenti tra loro, in modo che gli errori commessi da un modello possano essere compensati dagli altri.
- "**Dropout is generally coupled with minibatch-based learning algorithm**" (**Il dropout è generalmente accoppiato con algoritmi di apprendimento basati su minibatches**): **VERO.** Il "dropout" è una tecnica di regolarizzazione specifica per le reti neurali. Durante l'addestramento, il dropout disattiva casualmente alcuni neuroni (e quindi i loro collegamenti) durante ogni iterazione. Questo impedisce ai neuroni di co-adattarsi eccessivamente e rende la rete più robusta. Il dropout è particolarmente efficace quando utilizzato con algoritmi di apprendimento basati su minibatches, poiché la casualità introdotta dal dropout varia da minibatch a minibatch, fornendo una maggiore diversità durante l'addestramento.

The image shows a Wooclap poll interface. At the top, the word "wooclap" is visible next to a settings gear icon. Below the header, the question "Which of these sentences is false ?" is displayed. A note below the question says "You can select multiple choices". Two options are listed in separate boxes: "Multitask forces to share a set of parameters across different tasks" and "Multitask improves generalization when tasks are very different". At the bottom of the poll area is a blue "Submit" button with a white arrow icon.

- "**Multitask forces to share a set of parameters across different tasks**" (**Il Multitask Learning forza la condivisione di un insieme di parametri tra diverse task**): **VERO**. Una delle caratteristiche principali del Multitask Learning è proprio la condivisione di rappresentazioni (e quindi di parametri) tra diverse task. L'idea è che imparare task correlate simultaneamente possa migliorare la generalizzazione e l'efficienza rispetto all'apprendimento di ogni task singolarmente. Questa condivisione può avvenire a diversi livelli, ad esempio condividendo i primi strati di una rete neurale e avendo poi strati specifici per ogni task.
- "**Multitask improves generalization when tasks are very different**" (**Il Multitask Learning migliora la generalizzazione quando le task sono molto diverse**): **FALSO**. Il Multitask Learning tende a migliorare la generalizzazione quando le task sono correlate. Se le task sono troppo diverse o addirittura contrastanti, la condivisione di parametri può portare a interferenze negative e peggiorare le performance rispetto all'apprendimento di ogni task separatamente. L'intuizione è che se le task condividono alcune informazioni o rappresentazioni di basso livello, l'apprendimento simultaneo può aiutare a

imparare queste rappresentazioni in modo più efficace, beneficiando tutte le task. Se le task non hanno nulla in comune, la condivisione forzata di parametri può confondere il modello e ostacolare l'apprendimento.

The image shows a Wooclap poll interface. At the top, there's a blue header with the Wooclap logo and a settings icon. Below the header, the question is displayed: "Which of these sentences is true ?". A note below the question says "You can select multiple choices". There are three options listed in separate boxes:

- label smoothing is used for solving regression tasks
- label smoothing makes models robust to possible errors in the training set
- label smoothing can help convergence of Maximum likelihood learning with a softmax classifier and hard targets

At the bottom of the poll interface is a blue "Submit" button with a white arrow icon.

La domanda riguarda il "label smoothing" (arrotondamento delle etichette). Analizziamo le affermazioni:

- "**label smoothing is used for solving regression tasks**" (**il label smoothing è usato per risolvere task di regressione**): **FALSO**. Il label smoothing è una tecnica specifica per problemi di *classificazione*, non di regressione. Nella regressione, l'obiettivo è predire un valore continuo, mentre nella classificazione si predice una classe discreta. Il label smoothing interviene proprio sulla rappresentazione delle etichette di classe.
- "**label smoothing makes models robust to possible errors in the training set**" (**il label smoothing rende i modelli robusti a possibili errori nel set di training**): **VERO**. Uno dei vantaggi principali del label smoothing è la sua capacità di rendere i modelli più robusti al rumore presente nelle etichette del

training set. Invece di usare etichette "hard" (ad esempio, [0, 0, 1] per la classe 2 in un problema a 3 classi), si usano etichette "soft" (ad esempio, [0.05, 0.05, 0.9]). Questo impedisce al modello di diventare eccessivamente sicuro delle sue predizioni e di sovradattarsi a etichette potenzialmente errate.

- **"label smoothing can help convergence of Maximum likelihood learning with a softmax classifier and hard targets"** (**il label smoothing può aiutare la convergenza dell'apprendimento di Massima Verosimiglianza con un classificatore softmax e target hard**): **FALSO**. L'ultima parte dell'affermazione è contraddittoria. Il label smoothing è *proprio un'alternativa* all'uso di "hard targets" (etichette nette). Con target hard e un classificatore softmax, il modello cerca di massimizzare la probabilità della classe corretta fino a 1, il che può portare a problemi di overconfidence e difficoltà di convergenza. Il label smoothing, introducendo una piccola probabilità anche per le altre classi, "ammorbidisce" questo comportamento e facilita la convergenza. Quindi, l'affermazione corretta sarebbe che il label smoothing aiuta la convergenza *invece che* con target hard.

The image shows a Wooclap poll interface titled "Dataset Augmentation". The poll asks: "You can select multiple choices". Three options are listed:

- creates fake data and adds it to the training set
- it is very effective for non supervised tasks
- Injecting noise in the input to a neural network can also be seen as a form of data augmentation

A blue "Submit" button is at the bottom right.

- "**creates fake data and adds it to the training set**" (**crea dati falsi e li aggiunge al set di training**): VERO (detto dalla prof)
- "**it is very effective for non supervised tasks**" (**è molto efficace per task non supervisionate**): Questa affermazione è FALSA (la prof ha detto che è perchè se aumentiamo i dati senza sapere le labels, questo è pericoloso, si rischia di inserire un bias). L'aumento del dataset è una tecnica utilizzata principalmente per task di *apprendimento supervisionato*, dove si hanno etichette associate ai dati. L'obiettivo è aumentare la variabilità dei dati di training per migliorare la generalizzazione del modello. Nelle task non supervisionate, dove non ci sono etichette, l'aumento del dataset non ha lo stesso significato né la stessa utilità.
- "**Injecting noise in the input to a neural network can also be seen as a form of data augmentation**" (**Iniettare rumore nell'input di una rete neurale può essere visto anche come una forma di data augmentation**): Questa affermazione è vera. Aggiungere rumore (ad esempio, rumore gaussiano) agli input è una forma di aumento del dataset. Questo rende il modello più robusto alle variazioni e al rumore presente nei dati reali. È una tecnica particolarmente utilizzata nelle reti neurali.

Which of these sentences are true ?

 You can select multiple choices

Regularizing operators can be seen as soft constraints of the learning optimization problem

Regularization can be done by optimizing with respect to the loss function and then re-projecting the solution on the feasible region $(\mathcal{L}(\Omega(\theta)) < 0)$

Explicit constraints implemented by re-projection do not necessarily encourage the weights to approach the origin.

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

 Submit

- "**Regularizing operators can be seen as soft constraints of the learning optimization problem**" (**Gli operatori di regolarizzazione possono essere visti come vincoli soft del problema di ottimizzazione dell'apprendimento**): **VERO**. Gli operatori di regolarizzazione, come la regolarizzazione L1 e L2, aggiungono un termine alla funzione di perdita che penalizza la complessità del modello (ad esempio, la grandezza dei pesi). Questo può essere interpretato come un vincolo "soft" perché non impone un valore esatto ai pesi, ma li "spinge" verso valori più piccoli, preferibilmente verso lo zero. A differenza dei vincoli "hard" che definiscono un insieme ammissibile rigido, la regolarizzazione influenza l'ottimizzazione in modo più graduale.
- "**Regularization can be done by optimizing with respect to the loss function and then re-projecting the solution on the feasible region $((6)) < 0$** " (**La regolarizzazione può essere fatta ottimizzando rispetto alla funzione di perdita e poi riproiettando la soluzione sulla regione ammissibile $((6)) < 0$**): **VERO**. Questa affermazione descrive un metodo di regolarizzazione basato su *vincoli esplicativi*. Si ottimizza la funzione di perdita senza vincoli, e poi, se la soluzione ottenuta non soddisfa il vincolo (rappresentato da $(6) < 0$, che

presumibilmente definisce la regione ammissibile), si "riproietta" la soluzione sulla regione ammissibile più vicina. Questo è un modo per imporre vincoli "hard".

- "**Explicit constraints implemented by re-projection do not necessarily encourage the weights to approach the origin**" (I vincoli esplicativi implementati tramite riproiezione non necessariamente incoraggiano i pesi ad avvicinarsi all'origine): VERO. (Io ha confermato la prof, dice che è difficile questa, basta che arrivi intorno all'origine, non proprio all'origine) La riproiezione su una regione ammissibile arbitraria non garantisce che i pesi si avvicinino all'origine. Dipende dalla forma della regione ammissibile. Ad esempio, se la regione ammissibile è un cerchio centrato in un punto diverso dall'origine, la riproiezione spingerà i pesi verso il bordo di quel cerchio, non necessariamente verso l'origine. Solo se la regione ammissibile è centrata nell'origine (e.g., $\|w\| < c$ per qualche costante c), la riproiezione tenderà a ridurre la norma dei pesi e quindi avvicinarli all'origine.
- "**Explicit constraints implemented by re-projection only have an effect when the weights become large and attempt to leave the constraint region**" 1 (I vincoli esplicativi implementati tramite riproiezione hanno effetto solo quando i pesi diventano grandi e tentano di uscire dalla regione di vincolo): VERO. La riproiezione ha effetto solo quando la soluzione ottenuta dall'ottimizzazione senza vincoli si trova al di fuori della regione ammissibile. Se la soluzione si trova già all'interno della regione ammissibile, non è necessaria alcuna riproiezione e quindi il vincolo non ha alcun effetto pratico.

wooclap

Consider norm penalizations

ⓘ You can select multiple choices

Sum of absolute weights penalizes small weights more

Squared weights penalizes large values more

L2 results in more sparse weights than L1

the addition of the L2 term modifies the learning rule by shrinking the weight vector by a constant factor on each parameter update

L2 rescales the weights along the axes defined by the eigenvectors of the Hessian matrix

 Submit

- "**Sum of absolute weights penalizes small weights more**" (**La somma dei pesi assoluti penalizza maggiormente i pesi piccoli**): **VERO** (Io ha detto la prof)
- "**Squared weights penalizes large values more**" (**I pesi al quadrato penalizzano maggiormente i valori grandi**): **VERO**. I pesi al quadrato (penalizzazione L2) penalizzano i valori grandi *in modo quadratico*, cioè l'effetto della penalizzazione cresce molto più rapidamente per valori grandi rispetto a valori piccoli. Questo perché la derivata della funzione quadrato ($2w$) è proporzionale al peso stesso. Quindi, un peso doppio subisce una penalizzazione quadrupla.
- "**L2 results in more sparse weights than L1**" (**L2 risulta in pesi più sparsi di L1**): **FALSO**. La penalizzazione L1 tende a produrre soluzioni *più sparse* (con molti pesi esattamente a zero) rispetto alla L2. La L2 tende a ridurre i pesi verso zero, ma raramente li azzera completamente. Questa è una delle principali differenze tra le due tecniche.

- "the addition of the L2 term modifies the learning rule by shrinking the weight vector by a constant factor on each parameter update" (l'aggiunta del termine L2 modifica la regola di apprendimento riducendo il vettore dei pesi di un fattore costante ad ogni aggiornamento del parametro): FALSO (lo ha detto la prof)
- "L2 rescales the weights along the axes defined by the eigenvectors of the Hessian matrix" (L2 riscalà i pesi lungo gli assi definiti dagli autovettori della matrice Hessiana): VERO. Questa è una descrizione più avanzata dell'effetto della regolarizzazione L2. L'Hessiana della funzione di perdita descrive la curvatura della funzione. La L2 riscalà i pesi in base alla curvatura, con una contrazione maggiore lungo le direzioni di maggiore curvatura (autovettori con autovalori maggiori).

The image shows a Wooclap poll interface. At the top, it says "wooclap". Below that is a blue header bar with a gear icon. The main area has a white background with a light gray border. The title "Parameter norm penalties" is centered at the top. A note below it says "You can select multiple choices". There are three options listed in separate boxes:

- "make the network more stable"
- "minor variation or statistical noise on the inputs will result in large differences in the output"
- "encourage the network toward using small weights"

 At the bottom right is a blue "Submit" button with a white arrow icon.

- "make the network more stable" (rendono la rete più stabile): VERO. Le penalizzazioni delle norme, come L1 e L2, contribuiscono a rendere la rete più stabile. Questo significa che piccole variazioni nell'input non dovrebbero causare grandi variazioni nell'output. Una rete con pesi molto grandi può

essere estremamente sensibile a piccole perturbazioni, mentre una rete con pesi più contenuti è generalmente più robusta.

- "**minor variation or statistical noise on the inputs will result in large differences in the output**" (**minime variazioni o rumore statistico negli input risulteranno in grandi differenze nell'output**): **FALSO**. Questa affermazione descrive l'opposto di ciò che si ottiene con le penalizzazioni delle norme. Come detto prima, l'obiettivo è proprio *ridurre* la sensibilità a piccole variazioni nell'input.
- "**encourage the network toward using small weights**" (**incoraggiano la rete ad usare pesi piccoli**): **VERO**. Questo è il meccanismo principale attraverso il quale le penalizzazioni delle norme funzionano. Aggiungendo un termine alla funzione di costo che penalizza la grandezza dei pesi (somma dei valori assoluti per L1, somma dei quadrati per L2), si "spinge" l'ottimizzazione a trovare soluzioni con pesi più piccoli. Questo previene l'overfitting e migliora la generalizzazione.

Which of these sentences are false ?

ⓘ You can select multiple choices

If the weight of the regularization term in the loss function is too high it may imply underfitting

If the weight of the penalization term in the loss function is too high it may imply overfitting

Regularizing the bias parameters can introduce a significant amount of underfitting

Regularizing the bias parameters can introduce a significant amount of overfitting

Usually the bias parameters are not constrained by regularizing constraints



- "**If the weight of the regularization term in the loss function is too high it may imply underfitting**" (Se il peso del termine di regolarizzazione nella funzione di perdita è troppo alto, può implicare underfitting): VERO (quindi falso). Un peso di regolarizzazione molto alto penalizza eccessivamente la complessità del modello, "costringendolo" a essere troppo semplice. Questo può portare a underfitting, ovvero il modello non è in grado di catturare la complessità dei dati di training.
- "**If the weight of the penalization term in the loss function is too high it may imply overfitting**" (Se il peso del termine di penalizzazione nella funzione di perdita è troppo alto, può implicare overfitting): FALSO (quindi vero). Un peso di penalizzazione *troppo alto* causa *underfitting*, come spiegato nel punto precedente. La regolarizzazione serve a *prevenire* l'overfitting, non a causarlo.
- "**Regularizing the bias parameters can introduce a significant amount of underfitting**" (Regolarizzare i parametri di bias può introdurre una quantità significativa di underfitting): VERO (quindi falso). Sebbene sia meno comune regolarizzare i bias rispetto ai pesi, una regolarizzazione eccessiva dei bias può limitare la capacità del modello di adattare l'offset dei dati, portando a underfitting.
- "**Regularizing the bias parameters can introduce a significant amount of overfitting**" (Regolarizzare i parametri di bias può introdurre una quantità significativa di overfitting): FALSO (quindi vero). La regolarizzazione, in generale, *riduce* il rischio di overfitting. Anche nel caso dei bias, una regolarizzazione (se non eccessiva) non causa overfitting.
- "**Usually the bias parameters are not constrained by regularizing constraints**" (Solitamente i parametri di bias non sono vincolati da vincoli di regolarizzazione): VERO (quindi falso). Nella pratica, è *meno frequente* regolarizzare i bias. I bias hanno un ruolo meno diretto nella complessità del modello rispetto ai pesi, e la loro regolarizzazione di solito ha un impatto minore sulle performance. Per questo, spesso vengono lasciati "liberi" dalla regolarizzazione.

wooclap

Regularazing estimators

ⓘ You can select multiple choices

- Reduce bias
- Reduce the gap between training error and validation error
- Reduce underfitting problems
- Can reduce the complexity of large models

Submit

The screenshot shows a Wooclap poll titled "Regularazing estimators". It displays four multiple-choice options: "Reduce bias", "Reduce the gap between training error and validation error", "Reduce underfitting problems", and "Can reduce the complexity of large models". A note indicates that multiple choices can be selected. At the bottom is a blue "Submit" button.

- "**Reduce bias**" (**Riduce il bias**): **FALSO**. aumentano il bias
- "**Reduce the gap between training error and validation error**" (**Riduce il divario tra l'errore di training e l'errore di validazione**): **VERO**. Questo è uno degli obiettivi principali della regolarizzazione. Un grande divario tra l'errore di training e l'errore di validazione è un segno di overfitting. La regolarizzazione, riducendo la complessità del modello, aiuta a ridurre questo divario, migliorando la generalizzazione.
- "**Reduce underfitting problems**" (**Riduce i problemi di underfitting**): **FALSO**. La regolarizzazione non è pensata per risolvere problemi di underfitting. L'underfitting si verifica quando il modello è troppo semplice per catturare la complessità dei dati. In questo caso, è necessario aumentare la complessità del modello (ad esempio, usando un modello più complesso o aggiungendo feature), non ridurla con la regolarizzazione. Anzi, come detto prima, una regolarizzazione eccessiva può *causare* underfitting.
- "**Can reduce the complexity of large models**" (**Può ridurre la complessità di modelli grandi**): **VERO** (**la prof ha detto perchè spingiamo i pesi a 0**). Questo

è il meccanismo principale attraverso il quale la regolarizzazione funziona. Aggiungendo un termine di penalizzazione alla funzione di costo, si scoraggia il modello dall'avere pesi troppo grandi, riducendone la complessità. Questo previene l'overfitting e migliora la generalizzazione.

The image shows a Wooclap poll interface. At the top, it says "wooclap". Below that is a blue header bar with a gear icon. The main question is "Which of these sentences are true ?". A note below the question says "You can select multiple choices". There are four options in boxes:

- Simpler models generalize better
- Multiple hypothesis (ensemble) models generalize better
- More complex models can represent the true data generating process
- In general, when building machine learning models, the data generating process is not known

At the bottom is a blue "Submit" button with a white arrow icon.

- "**"Simpler models generalize better"** (**Modelli più semplici generalizzano meglio**): PARZIALMENTE VERO (la prof ha detto VERO, ma appunto non devono essere troppo piccole altrimenti hai underfitting). Questa affermazione è legata al principio di "parsimonia" o "rasoio di Occam", che preferisce spiegazioni più semplici a spiegazioni più complesse, a parità di capacità esplicativa. In machine learning, questo si traduce nel preferire modelli meno complessi, che tendono a generalizzare meglio a dati non visti durante l'addestramento, *a condizione che il modello sia sufficientemente complesso per catturare la struttura dei dati*. Un modello troppo semplice (alto bias) soffrirà di underfitting e non generalizzerà bene. Quindi, la generalizzazione migliore si ottiene con un giusto equilibrio tra semplicità e capacità di rappresentazione.

- "**Multiple hypothesis (ensemble) models generalize better**" (**Modelli a ipotesi multiple (ensemble) generalizzano meglio**): **VERO**. I modelli ensemble combinano le predizioni di più modelli (spesso più semplici) per ottenere una predizione finale più accurata e robusta. Questa combinazione riduce la varianza e spesso anche il bias, portando a una migliore generalizzazione rispetto ai singoli modelli che compongono l'ensemble. Esempi di modelli ensemble sono Random Forest, Gradient Boosting e bagging.
- "**More complex models can represent the true data generating process**" (**Modelli più complessi possono rappresentare il vero processo di generazione dei dati**): **VERO ma da specificare**. In teoria, un modello sufficientemente complesso ha la capacità di approssimare arbitrariamente bene qualsiasi funzione, incluso il vero processo di generazione dei dati. Tuttavia, in pratica, non conosciamo mai il vero processo di generazione dei dati, e l'utilizzo di modelli eccessivamente complessi porta spesso a overfitting, ovvero il modello si adatta troppo ai dati di training, inclusi il rumore e le fluttuazioni casuali, e generalizza male a dati nuovi.
- "**In general, when building machine learning models, the data generating process is not known**" (**In generale, quando si costruiscono modelli di machine learning, il processo di generazione dei dati non è noto**): **VERO**. Questa è una delle assunzioni fondamentali del machine learning. Se conoscessimo il vero processo di generazione dei dati, non avremmo bisogno di addestrare un modello: potremmo semplicemente utilizzare il processo stesso per fare predizioni. L'obiettivo del machine learning è proprio quello di *apprendere* una buona approssimazione di questo processo a partire dai dati disponibili.

The image shows a Wooclap poll interface. At the top, it says "wooclap". On the right, there is a gear icon. Below the header, the title of the poll is "Regularization". A note below the title says "You can select multiple choices". There are three options listed in boxes:

- it reduces the validation/test error at the expenses of (acceptable) training error
- it enables the model to reach a point that does minimize the loss function
- it enables the model to reduce the variability of data

At the bottom, there is a blue "Submit" button with a white arrow icon.

- "**it reduces the validation/test error at the expenses of (acceptable) training error**" (**riduce l'errore di validazione/test a scapito di un errore di training accettabile**): **VERO**. Questo è uno degli scopi principali della regolarizzazione. L'obiettivo non è minimizzare perfettamente l'errore sui dati di training, ma trovare un compromesso che permetta di generalizzare bene a dati nuovi (dati di validazione/test). La regolarizzazione, quindi, può portare a un leggero aumento dell'errore di training, ma riduce significativamente l'errore sui dati non visti.
- "**it enables the model to reach a point that does minimize the loss function**" (**permette al modello di raggiungere un punto che minimizza la funzione di perdita**): **PARZIALMENTE VERO**. La regolarizzazione *modifica* la funzione di perdita, aggiungendo un termine di penalizzazione. Quindi, il modello non minimizza più la funzione di perdita *originale*, ma una funzione di perdita *regolarizzata*. Il punto raggiunto dal modello minimizza questa nuova funzione, che include la penalizzazione per la complessità. In questo senso, si può dire che "minimizza la funzione di perdita regolarizzata", ma non necessariamente la funzione di perdita originale.

- "it enables the model to reduce the variability of data" (permette al modello di ridurre la variabilità dei dati): **FALSO**. La regolarizzazione *non riduce la variabilità dei dati*. La variabilità dei dati è una proprietà intrinseca dei dati stessi. La regolarizzazione agisce sul *modello*, riducendone la complessità e la sua capacità di adattarsi al rumore presente nei dati. In questo modo, il modello diventa meno sensibile alla variabilità dei dati, ma non la riduce direttamente. Piuttosto, riduce la varianza delle predizioni del modello.

The image shows a Wooclap poll interface. At the top, it says "wooclap". Below that, the title of the poll is "A Gaussian mixture output function". A note indicates "You can select multiple choices". There are three options listed in boxes:

- "can represent multimodal functions"
- "the weight associated to a gaussian in the mixture represents the probability of the output"
- "Mixtures are particularly suitable output for generative models for speech or for movements of objects"

 At the bottom is a blue "Submit" button with a white arrow icon.

- "can represent multimodal functions" (possono rappresentare funzioni multimodali): **VERO**. Le miscele gaussiane sono particolarmente adatte a rappresentare funzioni multimodali, ovvero funzioni con più "picchi" o "modi". Ogni gaussiana nella miscela cattura un modo della distribuzione, e la combinazione di più gaussiane permette di approssimare distribuzioni complesse con più picchi.
- "the weight associated to a gaussian in the mixture represents the probability of the output" (il peso associato a una gaussiana nella miscela rappresenta la probabilità dell'output): **PARZIALMENTE VERO**. Il peso

associato a una gaussiana nella miscela rappresenta la *probabilità che un dato provenga da quella specifica gaussiana*. In termini più precisi, rappresenta la probabilità *a priori* di appartenenza a quella componente della miscela. Non è direttamente la probabilità dell'output finale, ma contribuisce a calcolarla. L'output finale è una combinazione pesata di tutte le gaussiane, dove i pesi sono appunto queste probabilità di appartenenza. Quindi parzialmente vero o falso.

- **"Mixtures are particularly suitable output for generative models for speech or for movements of objects" (Le miscele sono output particolarmente adatti per modelli generativi per il parlato o per i movimenti di oggetti):**
VERO. Le miscele gaussiane sono molto utilizzate in modelli generativi per dati sequenziali come il parlato o i movimenti di oggetti. Questo perché questi tipi di dati spesso presentano una struttura complessa e multimodale, che può essere ben approssimata da una miscela di gaussiane. Ad esempio, nel riconoscimento vocale, diverse pronunce di una stessa parola possono essere rappresentate da diverse gaussiane nella miscela.

The image shows a Wooclap poll interface. At the top, there's a blue header bar with the Wooclap logo on the left and a settings icon on the right. Below the header, the title "Softmax function" is centered. A note says "You can select multiple choices". There are four options listed in boxes:

- is a good choice for representing discrete probability distributions with n possible values
- it is a good output function because it is continuous and differentiable
- since its output is a probability distribution it can always be interpreted as a confidence level
- if the prediction is correct its penalty is always 0

At the bottom, there's a blue "Submit" button with a white arrow icon.

- "**is a good choice for representing discrete probability distributions with n possible values**" (è una buona scelta per rappresentare distribuzioni di probabilità discrete con n valori possibili): VERO. La funzione Softmax trasforma un vettore di numeri reali in una distribuzione di probabilità su n possibili classi. L'output è un vettore di n valori compresi tra 0 e 1, la cui somma è pari a 1. Questo la rende ideale per problemi di classificazione multi-classe.
- "**it is a good output function because it is continuous and differentiable**" (è una buona funzione di output perché è continua e differenziabile): VERO. La continuità e la differenziabilità sono proprietà importanti per l'ottimizzazione tramite gradient descent, che è l'algoritmo di ottimizzazione più comunemente utilizzato per addestrare reti neurali. La differenziabilità permette di calcolare il gradiente della funzione di perdita rispetto ai parametri del modello, che è essenziale per l'aggiornamento dei pesi durante l'addestramento.
- La terza è falsa, perchè potremmo avere dataset sbilanciati (penso)
- "**if the prediction is correct its penalty is always 0**" (se la predizione è corretta, la sua penalità è sempre 0): FALSO. La "penalità" si riferisce alla funzione di perdita (loss function) utilizzata durante l'addestramento. Una loss function comune per problemi di classificazione multi-classe con Softmax è la cross-entropy loss. La cross-entropy loss non è necessariamente 0 anche se la predizione è corretta, a meno che la probabilità predetta per la classe corretta sia esattamente 1. In generale, la cross-entropy loss tende a 0 quando la probabilità predetta per la classe corretta si avvicina a 1, ma non è sempre esattamente 0.

wooclap

The function $\max\{0, \min\{1, Wh+b\}\}$ is a good choice as an output function for classifications problems

You can select multiple choices

No, because it does not return a value between 0 and 1

Yes, because it returns a value between 0 and 1

Yes, because it is linear

No, because it is not good for training

Submit

La prof ha detto che è giusta l'ultima, perchè ha il problema del vanishing gradient.

La funzione $\max\{0, \min\{1, Wh+b\}\}$ è una funzione di attivazione che combina due operazioni:

1. **Wh+b:** Questa è una trasformazione lineare, dove W sono i pesi, h è l'input e b è il bias.
2. **min{1, Wh+b}:** Questa operazione "clippa" il valore di Wh+b a 1. Se Wh+b è maggiore di 1, il risultato sarà 1. Altrimenti, il risultato sarà Wh+b.
3. **max{0, min{1, Wh+b}}**: Questa operazione "clippa" ulteriormente il valore a 0. Se il risultato del passo precedente è minore di 0, il risultato finale sarà 0. Altrimenti, il risultato sarà il valore del passo precedente.

In altre parole, questa funzione produce un output compreso tra 0 e 1. Questo tipo di funzione è nota come "clipping" o "saturazione". In alcuni contesti è anche nota come "unità lineare rettificata limitata" (Clipped Rectified Linear Unit o CReLU).

Analizziamo ora le affermazioni:

- "No, because it does not return a value between 0 and 1" (No, perché non restituisce un valore tra 0 e 1): **FALSO**. Come spiegato sopra, la funzione *restituisce sempre* un valore tra 0 e 1 inclusi.
- "Yes, because it returns a value between 0 and 1" (Sì, perché restituisce un valore tra 0 e 1): **VERO**. Questa è una conseguenza diretta della definizione della funzione.
- "Yes, because it is linear" (Sì, perché è lineare): **FALSO**. La funzione *non è lineare* su tutto il suo dominio. È lineare solo nella regione in cui $0 \leq Wh+b \leq 1$. Al di fuori di questo intervallo, la funzione è costante (0 o 1). La presenza delle funzioni *max* e *min* introduce non linearità.
- "No, because it is not good for training" (No, perché non è buona per l'addestramento): **PARZIALMENTE VERO**. Sebbene restituisca valori tra 0 e 1, e quindi *potenzialmente* utilizzabile per problemi di classificazione, questa funzione presenta dei problemi per l'addestramento, specialmente con algoritmi basati sul gradiente come la backpropagation.
 - **Saturazione:** Quando $Wh+b$ è molto grande o molto piccolo, la derivata della funzione è 0. Questo significa che durante l'addestramento, il gradiente si annulla e l'apprendimento si blocca (problema del "vanishing gradient"). Questo rende difficile l'ottimizzazione.
 - **Non è una vera distribuzione di probabilità:** Per problemi di classificazione multi-classe, tipicamente si usa la funzione Softmax, che produce una vera distribuzione di probabilità (la somma degli output è 1). Questa funzione non garantisce che la somma degli output sia 1, rendendola meno adatta per questo tipo di problemi.

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 sampled randomly

For FFN it is important to initialize all weights to small random values

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

 Submit

- "**When using SGD with mini-batches the model updates do not depend on the number of training examples**" (**Quando si usa SGD con mini-batch gli aggiornamenti del modello non dipendono dal numero di esempi di addestramento**): **VERO**. Con SGD e mini-batch, l'aggiornamento dei pesi del modello viene calcolato solo su un piccolo sottoinsieme di dati (il mini-batch). La dimensione del mini-batch è un iperparametro che viene scelto indipendentemente dalla dimensione del dataset di training. Pertanto, l'aggiornamento *non dipende direttamente* dal numero totale di esempi di training, ma solo dagli esempi presenti nel mini-batch corrente.
- "**When using SGD with mini-batches the number of updates to reach convergence does not depend on the number of training examples**" (**Quando si usa SGD con mini-batch il numero di aggiornamenti per raggiungere la convergenza non dipende dal numero di esempi di addestramento**): **FALSO**. Il numero di aggiornamenti necessari per la convergenza *dipende* dal numero di esempi di addestramento. Anche se ogni singolo aggiornamento si basa su un mini-batch, per "vedere" tutti i dati e convergere ad una buona soluzione, il modello deve passare attraverso più

mini-batch. Più dati ci sono, più mini-batch saranno necessari (e quindi più aggiornamenti) per completare un'epoca (un passaggio completo attraverso l'intero dataset).

- **"Once the SGD converges it is still useful to add more training examples sampled randomly"** (Una volta che SGD converge è ancora utile aggiungere più esempi di addestramento campionati casualmente): falso
- **"For FFN it is important to initialize all weights to small random values"** (Per FFN è importante inizializzare tutti i pesi a piccoli valori casuali): VERO.
Inizializzare i pesi a zero o a valori costanti porterebbe tutti i neuroni dello stesso layer ad apprendere la stessa cosa, rendendo la rete ridondante. Inizializzare i pesi a piccoli valori casuali rompe questa simmetria e permette ai diversi neuroni di apprendere feature diverse. "Piccoli" significa valori vicini allo zero, spesso campionati da una distribuzione normale con media zero e una piccola deviazione standard o usando inizializzazioni specifiche come Xavier/Glorot o He.
- **"The choice of cost functions is tightly coupled with the choice of the output unit"** (La scelta delle funzioni di costo è strettamente legata alla scelta dell'unità di output): VERO. La funzione di costo (loss function) deve essere compatibile con la funzione di output. Ad esempio:
 - Per problemi di regressione con output continuo, si usa spesso l'errore quadratico medio (MSE).
 - Per problemi di classificazione binaria con output sigmoide, si usa la binary cross-entropy.
 - Per problemi di classificazione multi-classe con output softmax, si usa la categorical cross-entropy.

Weights in a network must be initialized

- at zero
- by maintaining symmetry
- with zero variance
- randomly
- it is indifferent

 Submit

- "**at zero**" (**a zero**): **SBAGLIATO**. Inizializzare tutti i pesi a zero è un grave errore. Se tutti i pesi sono zero, tutti i neuroni in ogni strato calcoleranno lo stesso output e quindi avranno gli stessi gradienti durante la backpropagation. Questo significa che tutti i neuroni apprenderanno le stesse feature, rendendo la rete inutile. Si crea una simmetria che impedisce alla rete di apprendere rappresentazioni diverse.
- "**by maintaining symmetry**" (**mantenendo la simmetria**): **SBAGLIATO**. Come spiegato sopra, mantenere la simmetria (ad esempio, inizializzando tutti i pesi allo stesso valore, incluso zero) è esattamente ciò che si vuole evitare. La simmetria impedisce ai neuroni di specializzarsi e apprendere feature distinte.
- "**with zero variance**" (**con varianza zero**): **SBAGLIATO**. Inizializzare i pesi con varianza zero significa che tutti i pesi avrebbero lo stesso valore (che potrebbe essere zero o un altro valore costante). Questo ricade nel problema della simmetria descritto sopra.
- "**randomly**" (**casualmente**): **CORRETTO**. L'inizializzazione casuale dei pesi è fondamentale per rompere la simmetria e permettere ai diversi neuroni di

apprendere feature diverse. I pesi vengono tipicamente inizializzati con piccoli valori casuali, campionati da una distribuzione di probabilità come una distribuzione normale (gaussiana) con media zero e una piccola deviazione standard, oppure usando distribuzioni specifiche come la distribuzione di Xavier/Glorot o la distribuzione di He.

- **"it is indifferent" (è indifferente): SBAGLIATO.** L'inizializzazione dei pesi *non è affatto indifferente*. Una cattiva inizializzazione può portare a problemi di convergenza, gradienti che esplodono (exploding gradients) o gradienti che svaniscono (vanishing gradients), rendendo l'addestramento molto difficile o impossibile.

The image shows a Wooclap poll interface. At the top, it says "wooclap". Below that is a blue header bar with a gear icon. The main title of the poll is "Maxout". A note below the title says "You can select multiple choices". There are four options listed in boxes:

- has as special cases ReLU and Leaky ReLU
- requires less parameters to be learned
- does not have the problem of saturation
- can approximate any convex function

A "Submit" button is at the bottom.

- **"has as special cases ReLU and Leaky ReLU" (ha come casi speciali ReLU e Leaky ReLU): VERO.** Maxout è una generalizzazione di ReLU e Leaky ReLU. Se configurata opportunamente, una unità Maxout può comportarsi esattamente come una ReLU o una Leaky ReLU.

- "**requires less parameters to be learned**" (**richiede meno parametri da apprendere**): **FALSO**. Al contrario, Maxout *richiede più parametri* rispetto a ReLU o Leaky ReLU. Ogni unità Maxout è composta da k input lineari, e per ogni input lineare ci sono pesi e bias da apprendere. Questo aumenta il numero totale di parametri nel modello.
- "**does not have the problem of saturation**" (**non ha il problema della saturazione**): **VERO**. Una delle principali motivazioni per l'introduzione di Maxout è stata proprio evitare il problema della saturazione, che affligge funzioni di attivazione come la sigmoide e la tanh, e in parte anche ReLU (per input negativi). Maxout, essendo basata su un'operazione di massimo tra più input lineari, non presenta regioni di saturazione.
- "**can approximate any convex function**" (**può approssimare qualsiasi funzione convessa**): **VERO**. Questa è una proprietà teorica importante di Maxout. Si dimostra che una rete con unità Maxout può approssimare arbitrariamente bene qualsiasi funzione convessa continua.

Leaky ReLUs

ⓘ You can select multiple choices

saturate when the input is less than 0

need to perform exponential operations

tend to blow up activation with the output range of [0,inf]

Submit

La domanda riguarda le Leaky ReLU (Unità Lineari Rettificate "Perdenti").

Analizziamo le affermazioni:

- **"saturate when the input is less than 0" (saturano quando l'input è minore di 0): FALSO.** La principale motivazione per l'introduzione delle Leaky ReLU è stata proprio superare il problema della saturazione delle ReLU "standard" per input negativi. Le ReLU standard "muoiono" per input negativi, ovvero l'output è sempre 0 e quindi il gradiente è 0, bloccando l'apprendimento. Le Leaky ReLU, invece, per input negativi hanno una piccola pendenza, quindi il gradiente non è mai esattamente zero.
- **"need to perform exponential operations" (necessitano di eseguire operazioni esponenziali): FALSO.** Le Leaky ReLU sono computazionalmente molto efficienti perché non richiedono operazioni esponenziali o complesse.
- **"tend to blow up activation with the output range of [0,inf]" (tendono a far esplodere l'attivazione con un range di output di [0,inf]): VERO,** come la RELU

The image shows a Wooclap poll interface. At the top, there is a blue header bar with the Wooclap logo on the left and a settings icon on the right. Below the header, the question is displayed: "Advantages of the ReLU functions are". A note below the question says "You can select multiple choices". There are four options listed in separate boxes: "ReLUs are much simpler computationally", "Reduced likelihood of the gradient to vanish", "The gradient is constant for $z>0$ ", and "Differentiability". At the bottom of the poll is a blue "Submit" button with a white arrow icon.

- "**ReLUs are much simpler computationally**" (**Le ReLU sono computazionalmente molto più semplici**): **VERO**. La ReLU è definita come $\text{ReLU}(x) = \max(0, x)$. Si tratta di un'operazione estremamente semplice: confrontare un numero con zero e restituire il maggiore dei due. Questo la rende molto più veloce da calcolare rispetto a funzioni di attivazione come la sigmoide o la tanh, che richiedono operazioni esponenziali.
- "**Reduced likelihood of the gradient to vanish**" (**Ridotta probabilità che il gradiente svanisca**): **VERO**. Il problema del vanishing gradient si verifica quando il gradiente durante la backpropagation diventa molto piccolo, rallentando o bloccando l'apprendimento. Questo è un problema comune con le funzioni sigmoide e tanh, che saturano per valori di input molto grandi o molto piccoli, con derivata prossima a zero. Le ReLU, per input positivi, hanno gradiente costante pari a 1, il che aiuta a mitigare il problema del vanishing gradient.
- "**The gradient is constant for $z>0$** " (**Il gradiente è costante per $z>0$**): **VERO**. Come accennato sopra, per input positivi ($z > 0$), la derivata della ReLU è 1, quindi il gradiente è costante. Questo è un vantaggio per l'addestramento.
- "**Differentiability**" (**Differenziabilità**): **PARZIALMENTE VERO**. La ReLU non è *differenziabile* nel punto $x = 0$. La derivata in quel punto non è definita. Tuttavia, nella pratica, questo non è un problema significativo. Si usa convenzionalmente assegnare un valore di 0 o 1 come derivata in $x = 0$. Per il resto del dominio ($x \neq 0$), la ReLU è differenziabile. Quindi, si può dire che è "quasi ovunque differenziabile". (LA PROF HA DETTO FALSO, ma si può scrivere che non è differenziabile solo in 0 e questo non è un grande problema, e non accade spesso che si va sullo 0)



Rectified linear unit (ReLU) is proposed to speed up the learning convergence

Yes

No

Submit

La risposta è **Sì**.

The sigmoid function

 You can select multiple choices

saturates for large argument values

has a sensitive gradient when z is close to zero

has a zero gradient when the argument is close to zero

has a large gradient when it reaches saturation

is 0 for negative argument values

in some cases it can produce a sparse network (many zero weights) that may be useful

 Submit

- "**saturates for large argument values**" (**satura per valori di argomento grandi**): **VERO**. La sigmoide è definita come $\sigma(z) = 1 / (1 + \exp(-z))$. Per valori di z molto grandi (positivi), $\sigma(z)$ si avvicina a 1; per valori di z molto grandi in modulo ma negativi, $\sigma(z)$ si avvicina a 0. In queste regioni, la funzione "satura", ovvero la sua derivata si avvicina a zero.
- "**has a sensitive gradient when z is close to zero**" (**ha un gradiente sensibile quando z è vicino a zero**): **VERO**. Quando z è vicino a zero, la derivata della sigmoide è massima (0.25) e quindi il gradiente è "sensibile", ovvero piccole variazioni di z portano a variazioni significative dell'output.
- "**has a zero gradient when the argument is close to zero**" (**ha un gradiente zero quando l'argomento è vicino a zero**): **FALSO**. Come detto nel punto precedente, il gradiente è *massimo* (0.25) quando z è vicino a zero, non zero.
- "**has a large gradient when it reaches saturation**" (**ha un grande gradiente quando raggiunge la saturazione**): **FALSO**. Quando la sigmoide raggiunge la saturazione (per valori di z molto grandi in modulo), il gradiente si avvicina a

zero, non è grande. Questo è il problema del "vanishing gradient" (gradiente che svanisce).

- "**is 0 for negative argument values**" (**è 0 per valori di argomento negativi**): **FALSO**. La sigmoide assume valori tra 0 e 1, ma non è mai esattamente 0 per nessun valore finito di z . Per valori di z molto negativi, si avvicina a 0, ma non lo raggiunge mai.
- "**in some cases it can produce a sparse network (many zero weights) that may be useful**" (**in alcuni casi può produrre una rete sparsa (molti pesi zero) che può essere utile**): **FALSO**. La sigmoide, a differenza della ReLU, non induce sparsità nei pesi. La sparsità si riferisce alla presenza di molti zeri nei pesi della rete, il che può semplificare il modello e migliorarne la generalizzazione. La sigmoide, avendo output sempre compresi tra 0 e 1, non forza i pesi a diventare zero.

The image shows a Wooclap poll interface. At the top, there is a blue header bar with the Wooclap logo on the left and a gear icon on the right. Below the header, there is a horizontal progress bar. In the center of the page, there is a question: "Regularization functions are added to the loss functions to reduce their training error". Below the question, there are two large, rounded rectangular buttons. The top button is labeled "True" and the bottom button is labeled "False". At the bottom of the page, there is a blue "Submit" button with a white arrow icon.

La risposta è **Falso**.

Le funzioni di regolarizzazione non vengono aggiunte per ridurre *direttamente* l'errore di training. Il loro scopo principale è ridurre l'*overfitting* (sovradattamento) e migliorare la *generalizzazione* del modello, ovvero la sua capacità di funzionare bene su dati nuovi, non visti durante l'addestramento.

The screenshot shows a Wooclap poll interface. At the top, there is a blue header bar with the Wooclap logo on the left and a settings icon on the right. Below the header, a question is displayed: "The gradient can be estimated using a sample of training examples because is an expectation". Underneath the question, there are two large rectangular buttons: a blue one labeled "True" and a white one labeled "False". A status message "Waiting for next clap" is visible above the buttons. The background of the poll is light gray.

l'utilizzo di mini-batch e il calcolo del gradiente su di essi si basa proprio sul concetto che il gradiente calcolato sul mini-batch è una stima (un'aspettativa) del gradiente "vero" sull'intero dataset.

In conclusione, l'affermazione è **Vera**.

The loss function produces a numerical score that also depends on the set of parameters θ which characterizes the FFN model

Waiting for next clap

True

False

La risposta è **Vero**.

Una funzione di perdita (o funzione di costo) è una funzione matematica che quantifica quanto bene un modello di machine learning si adatta ai dati di addestramento. In altre parole, misura la "discrepanza" tra le previsioni del modello e i valori reali.

Nel contesto delle reti neurali feedforward (FFN), la funzione di perdita dipende da diversi fattori, tra cui:

- **I dati di input (x):** Gli esempi di addestramento che vengono forniti al modello.
- **I target (y):** I valori reali o le etichette corrispondenti agli input.
- **I parametri del modello (θ):** I pesi e i bias delle connessioni tra i neuroni della rete.

La funzione di perdita calcola un punteggio numerico che riflette la performance del modello *per un dato insieme di parametri θ* . Cambiando i parametri θ , l'output del modello cambia, e di conseguenza anche il valore della funzione di perdita.

L'obiettivo dell'addestramento di una rete neurale è proprio quello di trovare i valori ottimali dei parametri θ che minimizzano la funzione di perdita.

The image shows a Wooclap poll interface. At the top, there is a blue header bar with the 'wooclap' logo on the left and a settings icon on the right. Below the header, the question is displayed: "In a neural network the nonlinearity causes the most interesting loss function to become non convex". Two options are presented in white rounded rectangular boxes: "True" and "False". At the bottom of the poll area is a blue "Submit" button with a white airplane icon and the word "Submit".

Vero

Altre dalle registrazioni

Multilayer networks need to specify the kernel function

True

0% 0

False

100% 13

No, è imparata in automatico

In general, Kernel machines suffer from high computational cost of training when the dataset is large

1

Yes, because their complexity is linear in the number of training examples

55% 6

2

No, because their complexity is linear in the number of training examples

45% 5

3

No, because they do not depend on the number of training examples

0% 0

1. Quando applichiamo kernel function, questa dipende linearmente dal numero di esempi di training, che se è molto grande, lo rende costoso computazionalmente

To apply an iterative numerical optimization procedure for learning the weight of a FFN

1 The cost function may be a function that we cannot evaluate analytically

2 We need to know the analytical form of the gradient

Incorrect answer

3 It is enough to have some way of approximating the gradient

3. Non dobbiamo sapere la forma analitica della cost function, basta avere un modo di approssimare il gradiente.

For training multilayer NN

1 We must adjust the weight of all layers in one go 0% 0 people

2 We can train one layer at time using the error made by each layer in predicting the final result 0% 0 people

3 We can train one layer at time using the error made in reproducing its own input 0% 0 people

3. Possiamo trainare un layer alla volta usando l'errore nel riprodurre il suo stesso input. La seconda non è corretta perchè non possiamo trainare un layer alla volta, usando l'errore che c'è alla fine della rete.

It is always possible to train a neural network by solving a system of equations

1 true

0% 0

2 false

0% 0

Non è sempre possibile dal punto di vista computazionale. (a livello teorico si può formulare il problema)

A SVM can be trained by solving a system of equations while a neural network...
can be always trained by using convex optimization

1 Partially true

0% 0

2 True

0% 0

3 False

0% 0

Falso. è anche parzialmente vero perchè la seconda parte della frase è sbagliata, mentre la prima parte è vera. Però la frase nel complesso è falsa.

Which of these sentences is true ?

- 1 The gradient descent algorithm may not converge if the learning rate is too big 0% 0 people
- 2 The gradient descent algorithm may not converge if the learning rate is too small 0% 0 people
- 3 The gradient descent algorithm always converges to a global optimum 0% 0 people
- 4 The gradient descent algorithm can be also applied for solving maximization problems 0% 0 people
- 5 The gradient descent algorithm may converge to a local optimum 0% 0 people

1 4 5 sono giuste

When using Binary Cross-Entropy Loss function with Softmax Output Function, the derivative of the loss w.r.t. the inputs to the Softmax function is "prediction - true label"

When using Binary Cross-Entropy Loss function with Softmax Output Function, the derivative of the loss w.r.t. the inputs to the Softmax function is "predicti...

- 1 Yes 58% 7 people
- 2 No 42% 5 people

Combinando queste due, l'output è la prediction meno il true label, è vero.

CNNs

Convolution is:

Local in space, local in depth

Local in space, full in depth



Full in space, local in depth

Full in space, full in depth

Given the input volume with size $H \times W \times K$ and a filter bank with size $h \times w \times K$ we want to convolve them. The size of the output volume will be:

$H \times W \times K$

$h \times w \times K$

$(H-h+1) \times (W-w+1) \times K$

$(H-h+1) \times (W-w+1) \times K$

$(H-h+1) \times (W-w+1) \times 1$

$(H+h+1) \times (W+w+1) \times K$

La penultima è quella giusta, perchè stiamo considerando un filtro solo.

How many channels (i.e. depth size) will have the output volume resulting from the convolution of an input volume with 16 channels (i.e. depth=16) with a filter bank of 16 filters?

1

16

32

16

Compute the output after the application of max-pooling to the following input volume IN with a neighborhood of size=2 and stride=2.

$$\text{IN} = [12 \ 23 \ 40 \ 31; \\ 11 \ 15 \ 42 \ 52]$$

[40; 52]

[23 42 52]

[23 52]

[23 52]

Which are common techniques to reduce overfitting?



Congratulations!



Weight decay



Local response normalization

Data augmentation



Data normalization

Dropout



Weight decay (regolarizzazione L2) penalizza i pesi grandi. I pesi più piccoli rendono il modello meno complesso e più robusto, riducendo così il rischio di overfitting.

In data augmentation we have seen different policies (e.g. cropping, rotation, color cast, vignetting, etc.):

All policies are safe to be used to any problem

Only a subset of policies is safe to be used for each problem

We can diagnose the training and understand if we are overfitting:



Congratulations!



By plotting the loss on the training set across epochs

By plotting the loss on the validation set across epochs

By plotting the loss on the test set across epochs

By plotting the accuracy on the training and validation sets across epochs



We can simply continue adding layers to a NN and we will continue to obtain better results



Congratulations!



Yes

No



GoogLeNet (i.e. Inception-v1) introduced the use of auxiliary classifiers:

To mitigate the problem of vanishing gradients

To perform multi-task classification

To reduce overfitting

To mitigate the problem of vanishing gradients

ResNets were able to train a model with 150+ layers by:

Using just one fully-connected layer

Introducing the residual connections

Using just 3x3 convolutional filters

1 si nelle resnet, per non fare esplodere il numero di parametri

3 si è vera nelle resent, anche qui per mantenere il numero di parametri bassi
però l'unica che è vera nel contesto della domanda è la 2

If we have few data:



Congratulations!



We cannot use Deep Learning

We still can use Deep Learning



Model compression

Model compression is only used to allow models to run on mobile devices:

 Waiting for next clap

True

False

Abbiamo visto che è anche usato per poter fare l'inferenza di modelli molto grandi su pc normali.

Which is not a model compression technique?

 Waiting for next clap

Weight sharing

Network pruning

Low rank matrix decomposition

Dropout

Knowledge distillation

Quantization

Magnitude-based pruning removes weights having

The lowest value

The lowest absolute value

The highest absolute value

The highest value

Global MBP tends to outperform layer-wise MBP:

True

False

True, perché a quello globale non interessa dove siano i collegamenti con i pesi più bassi, toglie quella percentuale a prescindere da dove siano, anche se sono tutti sullo stesso layer. Mentre invece quello layer based vuole togliere lo stesso numero da ciascun layer.

Structured pruning

 Waiting for next clap

Aims to preserve network density for computational efficiency

Aims to increase network sparsity for computational efficiency

Quando facciamo pruning strutturato stiamo togliendo neuroni interi, e questo significa che potremmo ridurre un'intera colonna o riga nella matrice, quindi la matrice rimane densa.

RNNs

Recurrent neural networks or RNNs are a family of neural networks for processing sequential data

True

False

**The computation in most RNNs can be decomposed
into 3 blocks of parameters and associated
transformations:**



Congratulations!



From the input to the hidden state



From the hidden state to the input

From the previous hidden state to the next hidden state



From the next hidden state to the previous hidden state

From the hidden state to the output



From the output to the hidden state

What is the name of the algorithm used to train RNNs?



Congratulations!



Backpropagation

Backpropagation through recurrence

Backpropagation throught time



Vanishing gradients are more easy to identify than exploding gradients



Congratulations!



True

False



Exploding gradients are more difficult to handle than vanishing gradients



Congratulations!



True

False



Per l'exploding basta mettere un cap

Vanishing gradients



Congratulations!



Bias parameters to capture short-term dependencies



Bias parameters to capture long-term dependencies

Stiamo prendendo conoscenza solo per i time steps che sono molto vicini nel tempo

Gated RNNs are based on the idea of creating paths through time that have derivatives that neither vanish nor explode



Congratulations!



True



False

LSTMs have the following gates:



Congratulations!



Input gate



Remember gate

Forget gate



Output gate



Hidden gate

Recurrent gate

GRUs have



Congratulations!



significantly less parameters than LSTMs



significantly more parameters than LSTMs

GAN

GANs

learn to sample from the training set

learn to generate data from the training set

learn to interpolate the data in the training set

learn to sample from the training set. Impara a come associare ad ogni random vector, un'immagine che è della stessa distribuzione del training set.

Non sta generando data dal training set, ma sta generando dati che NON sono nel training set, altrimenti staremmo ricreando il training set.

GANs are composed of two models called



Congratulations!



Interpolator

Classifier

Discriminator



Creator

Generator



Inventor

The key layer in the generator is



Congratulations!



Canonical convolution

Inverted convolution

Interpolated convolution

Transposed convolution



Dilated convolution

During the update of the Discriminator, the gradients flow through the Generator



Congratulations!



True

False



Il discriminatore vuole distinguere i samples reali da quelli generali, quindi prende come dati i samples reali e quelli generati. Quindi i gradienti si fermano, quando raggiungono l'input, non vanno nel generatore.

During the update of the Generator, the gradients flow through the Discriminator

Congratulations!

True

False

Federated learning

Federated Learning aims to:

Congratulations!

Collaboratively train a ML model

Independently train a ML model

In Federated Learning, the data



Congratulations!



is shared across parties/sever

is kept private



In Federated Learning

We control how data is distributed across parties/workers

Data on each party/worker is not independent and identically distributed

era l'altra quella giusta (e l'avevo fatta giusta, ma non ho fatto in tempo a fare lo screenshot)

In Federated Learning there is always a server to orchestrate training

True

False

False

In the FedAVG algorithm the central model is updated



Congratulations!



Taking the minimum values of the parameters in the corresponding layers across the models sent by the different workers

Taking the mean values of the parameters in the corresponding layers across the models sent by the different workers



Taking the median values of the parameters in the corresponding layers across the models sent by the different workers

Taking the maximum values of the parameters in the corresponding layers across the models sent by the different workers

Transformers

Transformers can process sequences of arbitrary length



Congratulations!



True



False

Transformers process each input independently

True

False

The only processing module in a transformer layer is self attention



Congratulations!



True

False



Self attention, being a composition of two linear transformations is linear

True

False

The elements that are needed for the computation of self attention are:



Congratulations!



Outputs

Inputs



Values



Variables

Queries



Questions

Keys



Chains

A transformer head is completely defined by 3 weight matrices and 3 biases



Congratulations!



True



False

To obtain the best performance usually transformers use one single head



Congratulations!



True



False

Self supervised learning

**Self-supervised Learning refers to learning methods
in which the models are trained:**



Complimenti!



with supervisory signals that are generated from the data itself by leveraging its structure



with supervisory signals provided by human annotated labels

without the need of any supervisory signal

What is called the task designed for networks to solve with pseudo labels automatically generated based on data attributes?

Pretext task

Self-supervised task

Primary task

Context task

Image generation cannot be used as a pretext task



Complimenti!



True

False



When we use image generation as the pretext task, after training we are interested to use

The generator

The discriminator

Both

Quando vogliamo imparare a generare le immagini vogliamo il generator, in self supervised learning ci interessa il discriminator.

The performance of SSL models are usually measured comparing their accuracy on the pretext task



Complimenti!

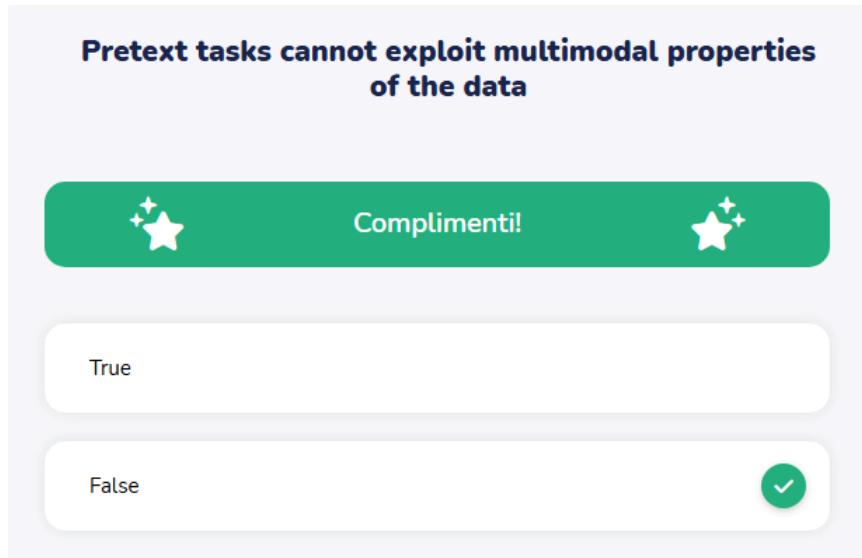


True

False



Usiamo una downstream task, anche perchè spesso non sappiamo neanche che dati sono stati usati per il training della pretext task.



Si può sia per multistream che per multimodal, come per esempio in un video si può usare il video e l'audio.