29/05/24 09:42

- Utile per calcolare la media mobile in economia
- Operazione di aggregazione

Convoluzione

- Serve per generare una nuova immagine a partire da un immagine originale
 - Es. filtri instagram, che migiorano i colori, danno una semitrasparenza, effetto blur etc...

Per le reti neurali convoluzionali, metto in cascata diverse trasformazioni di immagini per riuscire a ridurre una immagine grande in un vettore

- Partendo dal dominio delle immagini riusciamo a fare delle classificazioni (es. riconoscere se c'è babbo natale in una immagine oppure no)
- i valori del kernel (della matrice che va sovrapposta all'immagine originale, chiamata anche matrice di convoluzione o finestra) vengono dedotti con l'allenamento

Layers visti fino ad ora

Dense

- Neuroni classici, diverse funzioni di attivazione Sigmoide ->
 - Relu ->
 - Softmax ->
- Reti convoluzionali
 - Dieci layers per l'architettura di base 1. Layer convoluzionale
 - 2. Max pooling layer
 - i. Presa una immagine la dimezza di dimensioni prendendo il pixel più grande della finestra
 - - ii. Ottimo per ridurre le informazioni presenti in una immagine, prendendo solo le info più rilevanti
- Questi due layers vengono ripetuti più volte in sequenza, per effettuare trasformazioni successive essi sono il blocco base delle reti convoluzionali -> si chiamano blocchi base
- DIFFERENZA TRA CLASSIFICAZIONE E REGRESSIONE

Classificazione e regressione sono due tipi di problemi fondamentali nell'apprendimento automatico.

Entrambi coinvolgono la predizione di un output a partire da un insieme di input, ma differiscono nel tipo di output che producono e nei metodi utilizzati per la predizione.

Classificazione **Scopo**: Predire una categoria o etichetta discreta per un dato input. Caratteristiche:

Output: L'output è una variabile categoriale. Può essere binario (due classi) o multi-classe (più di due

classi). Esempio di classificazione binaria: Determinare se un'email è "spam" o "non spam".

- Esempio di classificazione multi-classe: Classificare il tipo di frutto come "mela", "banana" o "ciliegia". Funzioni di Attivazione: Spesso usano la funzione di attivazione sigmoid (per classificazione binaria) o
 - softmax (per classificazione multi-classe). Funzione di Loss: Utilizzano funzioni di perdita come la cross-entropy. Esempi di Algoritmi:
- Alberi di decisione Support Vector Machines (SVM)
- - K-Nearest Neighbors (KNN) Reti neurali (con softmax o sigmoid nell'output layer)

camere, ecc.

Esempi di Algoritmi:

Metodi di Valutazione: Accuratezza, precisione, recall, F1-score, ROC-AUC.

Output: L'output è una variabile continua.

- Regressione
- **Scopo**: Predire un valore continuo per un dato input. Caratteristiche:
 - Funzione di Loss: Utilizzano funzioni di perdita come la mean squared error (MSE) o mean absolute error (MAE).

Funzioni di Attivazione: Spesso utilizzano una funzione di attivazione lineare nell'output layer.

o Esempio: Predire il prezzo di una casa basato su caratteristiche come la dimensione, il numero di

- Regressione lineare
- Regressione polinomiale Support Vector Regression (SVR)
 - Reti neurali (con funzione di attivazione lineare nell'output layer)
 - **Metodi di Valutazione**: Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared.
- Differenze Principali

• Regressione: Output continuo (numerico).

- Tipo di Output:
- *Classificazione*: Output discreto (categorie/etichette).

2. Funzioni di Perdita: o *Classificazione*: Cross-entropy, log-loss.

- Regressione: Mean Squared Error (MSE), Mean Absolute Error (MAE).
- 3. Funzioni di Attivazione nell'Output Layer: o Classificazione: Sigmoid (per classificazione binaria), Softmax (per classificazione multi-classe).
- *Regressione*: Lineare. 4. Metodi di Valutazione:
- Classificazione: Accuratezza, precisione, recall, F1-score, ROC-AUC. Regressione: MSE, MAE, R-squared. In sintesi, la classificazione mira a suddividere gli input in categorie distinte, mentre la regressione si
- concentra sulla previsione di valori numerici continui
 - **ESEMPIO RETE CONVOLUZIONALE**

2. Max pooling layer

- 1. Convolutional layer
- 3. Convolutional layer 4. Max pooling layer 5. Flatten -> vengono stirate e diventano molti float che finiscono come segnali di input per lo strato
- Ad alto livello ho: Estrazione delle features -> flatten -> classificazione o regressione

successivo (come avviene con tutti gli altri layer, l'output diventa l'input del layer successivo)

TRANSFER LEARNING invece di avere i blocchi base, prendo una rete già fatta (già allenata) e la inserisco al posto dei blocchi

TRANNSFORMERS

base

6. Dense layer 7. Dense layer

Punto chiave per la generazione di testo, padre dei LLM

Nel deep learning, i modelli sono costituiti da una serie di layer (strati) che trasformano i dati di input in

output, passando attraverso diverse rappresentazioni intermedie. I layer principali nel deep learning

Input Layer (Strato di Input): o È il primo strato del modello e riceve i dati di input. Non esegue alcuna trasformazione sui dati, ma li passa al successivo strato.

includono:

Hidden Layers (Strati Nascosti): Sono i layer intermedi tra l'input e l'output. Ogni hidden layer esegue calcoli e trasformazioni sui dati ricevuti dal layer precedente. I principali tipi di hidden layers includono:

(Gated Recurrent Unit).

Output Layer (Strato di Output):

di neuroni durante l'addestramento.

distribuzione di probabilità su più classi.

layer successivo. Convolutional Layers: Utilizzati principalmente nelle reti neurali convoluzionali (CNN) per l'elaborazione di immagini, eseguono convoluzioni che aiutano a estrarre caratteristiche locali dai dati. Recurrent Layers: Utilizzati nelle reti neurali ricorrenti (RNN) per elaborare sequenze di dati,

precedente per migliorare la stabilità e la velocità di addestramento.

nell'output layer dipende dal tipo di problema: • Regressione: Utilizza una funzione di attivazione lineare per produrre un valore continuo. • Classificazione binaria: Utilizza una funzione di attivazione sigmoide per produrre una probabilità. • Classificazione multi-classe: Utilizza una funzione di attivazione softmax per produrre una

È l'ultimo strato del modello e produce il risultato finale. Il tipo di funzione di attivazione utilizzata

Dense (o Fully Connected) Layers: Ogni neurone di un layer è connesso a tutti i neuroni del

come testo o serie temporali. Esempi includono LSTM (Long Short-Term Memory) e GRU

Dropout Layers: Utilizzati per prevenire l'overfitting, disattivano casualmente una frazione

Normalization Layers: Come Batch Normalization, che normalizza l'output del layer

Ogni tipo di layer ha un ruolo specifico nella trasformazione dei dati e nella capacità del modello di apprendere rappresentazioni complesse dai dati.

Le funzioni di attivazione sono una componente cruciale delle reti neurali, poiché introducono non linearità nel modello, permettendo di apprendere rappresentazioni complesse dai dati. Ecco le principali funzioni di

attivazione utilizzate nel deep learning, il loro scopo e le loro differenze: Sigmoid (o Logistic) Function:

problemi di classificazione binaria.

Tanh (Tangente Iperbolica) Function:

minore rispetto alla sigmoid.

mantenendo i valori positivi invariati.

FUNZIONI DI ATTIVAZIONE

• **Pro**: Uscita compresa tra o e 1, interpretabile come probabilità. • Contro: Saturazione ai valori estremi (vicino a o o 1), causando il problema del vanishing gradient, rendendo difficile l'addestramento delle reti profonde.

 \circ **Formula**: tanh (x)=ex-e-xex+e-x\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} tanh(x)=ex+e-xex-e-x

o Scopo: Introduce non linearità mantenendo un calcolo semplice, trasformando negativi in zero e

Scopo: Trasforma il valore di input in un output compreso tra o e 1, rendendola adatta per

o **Pro**: Simile alla sigmoid, ma con output centrato intorno a zero, il che può aiutare a rendere l'addestramento più veloce rispetto alla sigmoid. Contro: Anche questa funzione soffre del problema del vanishing gradient, sebbene in misura

Scopo: Trasforma il valore di input in un output compreso tra -1 e 1.

 \circ **Formula**: $\sigma(x)=11+e-x \cdot (x) = \frac{1}{1}e^{-x} \cdot (x)=1+e-x1$

- **ReLU (Rectified Linear Unit) Function:** \circ Formula: ReLU(x)=max (o,x)\text{ReLU}(x) = \max(o, x)ReLU(x)=max(o,x)
 - Pro: Risolve in parte il problema del vanishing gradient, accelerando l'addestramento delle reti profonde. o Contro: Può soffrire del problema dei "neuroni morti", dove i neuroni possono bloccarsi a zero e
- Softmax Function:
 - nell'output layer per problemi di classificazione multi-classe. • **Pro**: Produce una distribuzione di probabilità, facilitando l'interpretazione dei risultati di
- classificazione. • Contro: Non utilizzabile per hidden layers, solo per l'output layer in problemi di classificazione. Ogni funzione di attivazione ha i propri vantaggi e svantaggi, e la scelta della funzione dipende spesso dal tipo
- smettere di apprendere.
 - \circ **Formula**: softmax(xi)=exi \sum jexj\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{ij}} e^{x_j}} softmax(xi)=∑jexjexi Scopo: Trasforma un vettore di valori in una distribuzione di probabilità, usata principalmente
- di problema, dall'architettura della rete e dalle specifiche esigenze del modello.