# LLM

16:10 ••• <del>5</del>1

Ø

**ChatGPT** 

## Settimana 1 → Fondamenta (Python + ML classico)

of Obiettivo: rispolverare Python e assicurarti che i concetti base di ML siano solidi.

- · Giorni 1-2 (Python + librerie essenziali)
  - · Python: funzioni, classi, gestione file.
  - NumPy: array, operazioni vettoriali, broadcasting.
  - Pandas: caricamento CSV, manipolazione tabelle.
  - Matplotlib: plot base.
- Giorni 3–7 (ML classico con freeCodeCamp)
  - Regressione lineare/logistica.
  - · Classificazione.
  - · Clustering (K-means).
  - Overfitting/underfitting.
  - Train/validation/test split.
  - Mini-progetto: Iris dataset o MNIST → allenare un classificatore semplice.



16:10



ChatGPT



# ◆ Settimana 2 → Deep Learning conPyTorch

Obiettivo: padroneggiare le basi delle reti neurali moderne con PyTorch.

- Capitoli video FreeCodeCamp PyTorch (solo essenziali)
  - Tensori e autograd.
  - · Dataset + DataLoader.
  - nn.Module e training loop.
  - · Loss e ottimizzatori.
  - · CNN su CIFAR-10.
- · Esercizi pratici
  - MLP su MNIST (classificazione numeri).
  - CNN su CIFAR-10 (immagini a colori).
  - Documenta ogni esperimento (accuracy, loss curve).
- Jay Alammar (lettura rapida)
  - Illustrated Word2Vec.
  - Illustrated RNN/LSTM.
  - — Ti dà contesto storico su NLP.



+

Fai una domanda





16:10 ••• <del>5</del>1

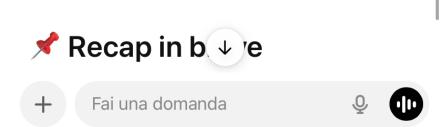
ChatGPT



# Settimana 3 → Oltre i prerequisiti (Transformer + HuggingFace)

of Obiettivo: andare oltre i prerequisiti, così arrivi con un piccolo vantaggio.

- Transformer (Jay Alammar)
  - Illustrated Transformer → capire attenzione, positional encoding.
- · HuggingFace Course (sezioni essenziali)
  - Intro + Tokenizers.
  - Usare pipeline pre-addestrate (es. sentiment analysis, traduzione).
  - · Fine-tuning (anche solo leggendo i concetti).
- Mini-progetto finale (1–2 giorni)
  - Carica un modello HuggingFace preaddestrato (es. BERT, DistilBERT).
  - Fai sentiment analysis o classificazione di testo.
  - Documenta il workflow.



### ı əctumana*j*

- Jay Alammar (Word2Vec, RNN/LSTM, Transformer).
- HuggingFace Course → solo: intro, tokenizer, pipeline, fine-tuning overview.
- Mini-progetto: carica un modello preaddestrato (es. DistilBERT) e fai sentiment analysis o text classification.

#### VIDEO PYTORCH:

- ★ Capitoli da seguire (essenziali)
- √ Totale: ~8–10h effettive di studio + pratica.
  - 1. Introduzione a PyTorch e Tensori
    - Creazione e manipolazione di tensori
    - Operazioni matematiche base
    - Autograd (gradiente automatico)
  - Fondamentale per capire come funziona il training.
  - 2. Dataset e DataLoader
    - Creazione dataset personalizzati
    - DataLoader per batch e shuffle
  - Serve per capire come gestire dati reali.
  - 3. Costruire modelli con nn.Module
    - Definire layer sequenziali
    - Funzioni di attivazione
      - Uso di forward()

  - 4. Training loop
    - Forward → Loss → Backward → Optimizer step
    - Funzioni di loss (MSE, CrossEntropy)
  - Devi saper leggere/scrivere un training loop semplice.
  - 5. Classificazione immagini (MNIST, CIFAR-10)
    - Costruire un MLP su MNIST
    - Costruire una CNN su CIFAR-10
  - Ottimo per consolidare tutto: dataset, modello, training.
  - 6. Introduzione agli Embedding
    - Uso di torch.nn.Embedding su sequenze testuali
  - Ti dà l'infarinatura NLP classico che cercavi.

### 📌 Capitoli da saltare o guardare solo se avanza tempo

• Deployment, salvataggio e caricamento modelli → non ti serve come prerequisito.

- Visione dettagliata su API poco usate (es. funzioni matematiche avanzate di PyTorch).
- Progetti molto lunghi su dataset complessi → rischiano di portarti via tempo senza valore per i prerequisiti.