AGID | Agenzia per l'Italia Digitale

Linee guida per l'IA nella pubblica amministrazione

Training, Validazione, Fine-tuning di modelli e utilizzo di RAG

Strumento B

Versione 1.0 del 11.03.2026 – Pubblicazione in consultazione pubblica.

Indice

Finalità dell'allegato

Questo allegato alle Linee Guida Sviluppo ha l'obiettivo di fornire una panoramica sui requisiti minimi e le buone pratiche raccomandate per addestrare modelli di IA, validarne le caratteristiche operative, effettuare attività di fine-tuning per scopi specifici ed infine integrare RAG nei sistemi di IA destinati all'uso interno della Pubblica Amministrazione.

Il Processo di Addestramento (Training) dei Modelli

L'addestramento rappresenta la fase centrale dello sviluppo di un modello di Intelligenza Artificiale. In questa fase, l'algoritmo alla base del modello impara ad identificare pattern e correlazioni all'interno di un dataset precedentemente elaborato. Tecnicamente, questo processo consiste nell'ottimizzazione iterativa dei parametri interni del modello (*pesi e bias*) attraverso una funzione di perdita (*loss function*), che misura lo scostamento tra l'output generato e il valore target desiderato. Questa tecnica è quella maggiormente utilizzata per le famiglie di IA appartenenti al Machine Learning (ML), in particolare alla sottoclasse del Deep Learning (DL).

Tali tecniche, tuttavia, sono applicabili a tutti i modelli basati su *Gradient Descent* (Discesa del Gradiente), includendo dunque:

- Reti Neurali (Deep Learning), famiglie di LLM o SLM;
- Regressioni Lineari/Logistiche, modelli statistici classici;
- Support Vector Machines (SVM).

In questi casi, l'obiettivo della fase di training è trovare il minimo di una funzione di costo, definita di seguito: $J(w,b) = \frac{1}{n}\sum_{i=1}^{n} L(\hat{y}_i, y_i)$

Dove, $L(\hat{y}_i, y)$ (*Loss Function*) rappresenta la perdita calcolata su un singolo esempio di addestramento e misura l'errore tra la previsione del modello e il valore reale, mentre $J(w,b)$ è la funzione di costo e rappresenta la media delle perdite su tutti gli n esempi di dataset, *w* rappresenta il vettore dei pesi (*weights*) e *b* è uno scalare che rappresenta i bias del modello su cui si operano aggiustamenti ricorsivi al fine di minimizzare il valore della funzione di costo.

La complessità dei parametri *w* e *b* varia significativamente in base alla complessità dell'architettura scelta all'interno della famiglia di IA (modelli supervisionati e modelli Generativi). Ad esempio: nel caso di modelli a *regressione lineare* il parametro *w* è rappresentato come un vettore e *b* come uno scalare; nel caso di modelli basati su *reti neurali* il parametro *w* è rappresentato come una matrice di pesi (che collegano gli input ai neuroni dello specifico strato) e *b* come un vettore di valori di bias per ogni singolo neurone del layer; nel caso infine di modelli basati su *GenAI (famiglia dei Transformer)*, il parametro *w* non è più un semplice vettore, ma è rappresentato da un insieme di tensori multidimensionali, dove un tensore rappresenta una estensione matematica (e computazionale) del concetto di matrice e può essere caratterizzato da un numero arbitrario di dimensioni. Il parametro *b*, in questo ultimo caso, è rappresentato come un vettore complesso di dimensioni molto grandi e molto variabile.

Il *training* di un modello di IA appartenente alle famiglie sopra citate, si articola generalmente in tre fasi chiave (con particolari specificazioni tecniche da apportare in base al particolare modello di IA utilizzato):

1. **Feed-forward:** in questa fase i dati di addestramento passano attraverso l'architettura del modello per generare una previsione di output. Essendo l'architettura del modello rappresentabile come un grafo computazionale composto da strati (*layers*) di neuroni artificiali, questa fase consiste in una mappatura matematica non lineare che trasforma il dato di input (vettore o tensore) attraverso una sequenza di operazioni affini seguite da funzioni di attivazione, proiettando l'informazione in spazi di rappresentazione via via più astratti fino alla previsione finale.

2. **Backpropagation:** in questa fase l'errore calcolato viene propagato a ritroso per determinare l'influenza di ogni parametro sul risultato finale. Essenzialmente, questa fase consiste nel calcolo analitico del gradiente della funzione di perdita rispetto ai parametri del modello attraverso l'applicazione sistematica della regola della catena (*chain rule*) lungo il grafo computazionale. Tale processo di *differenziazione automatica in modalità inversa* permette di quantificare l'influenza infinitesimale di ogni singolo peso e bias sull'errore complessivo, fornendo le informazioni necessarie per l'aggiornamento dei parametri del modello stesso.

3. **Optimization:** questa è la fase operativa in cui i gradienti calcolati durante la *Backpropagation* vengono utilizzati per modificare effettivamente i pesi che descrivono la funzione di perdita del modello. L'obiettivo è risolvere un problema di minimizzazione della funzione di perdita sopra descritta, attraverso un processo iterativo di aggiornamento.

Considerazioni sull'Infrastruttura e Scalabilità Computazionale

Una Pubblica Amministrazione con profilo di *Operatore Esperto* che abbia necessità, per i propri fini istituzionali, di addestrare un modello Intelligenza Artificiale, dovrà necessariamente dimensionare correttamente l'infrastruttura da utilizzare, sia essa *on-premises* o in modalità *Infrastructure as a Service* (IaaS), al fine di espletare correttamente le tre fasi sopra descritte (*Feed-forward, Backpropagation, Optimization*).

Per una PA, l'efficienza non risiede solo nella mera capacità di calcolo, ma anche nella corretta integrazione dei processi e delle attività (che consumano risorse computazionali) con il modello di architettura logica di riferimento, descritta nella Linea Guida Sviluppo di IA.

In particolare, una PA deve tenere in considerazione i seguenti vincoli e caratteristiche del modello basato su orchestratore:

- **Estensione Dati (Infrastruttura di Comunicazione e Storage):** Il processo di training richiede un'elevata ampiezza di banda (*throughput*) per il trasferimento dei dati dallo storage alle unità di calcolo. Un'infrastruttura di comunicazione sottodimensionata crea colli di bottiglia durante il caricamento dei *mini-batch*, rendendo inefficiente l'eventuale uso di acceleratori hardware che rimarrebbero in stato di *idle*. In ambito PA, ciò implica la necessità di storage ad alte prestazioni, implementato ad esempio attraverso lo standard aperto *Non-Volatile Memory Express* (NVMe) e interconnessioni veloci (es. utilizzando gli standard aperti InfiniBand o 100GbE).
- **Estensione Modelli (Infrastruttura Computazionale):** Si riferisce alla potenza di calcolo pura necessaria per le operazioni di *Feed-forward* e *Backpropagation*. Per i modelli oggetto di questo Allegato, è indispensabile la disponibilità di cluster di GPU soprattutto in caso di modelli di GenAI. La PA che si configura come *Operatore Esperto*

ha facoltà di realizzare un dimensionamento *on-premises*, scelta che richiede di garantire la sostenibilità energetica dei consumi delle infrastrutture di calcolo, oppure un dimensionamento mediante approccio IaaS, ovvero acquistando capacità computazionale "on-demand" e pagando solo per le ore effettive di training.

- **Estensione Tool e Orchestratore (Controllo):** L'infrastruttura deve prevedere un livello di gestione, attraverso l'orchestratore centralizzato in grado di coordinare il flusso tra i dati di training e i modelli. Questo livello deve essere leggero ma resiliente, capace di gestire il *checkpointing* del modello (salvataggio degli stati intermedi) per evitare la perdita di progressi in caso di guasti hardware.

Validazione e Ottimizzazione degli Iperparametri

La fase di validazione è il processo iterativo volto a valutare le prestazioni del modello su un set di dati indipendente (*Validation Set*), non utilizzato quindi durante la fase di addestramento del modello stesso.

L'obiettivo primario di questa attività è quello che viene denominato *ottimizzazione degli iperparametri* o fase di validazione, ovvero la regolazione dei parametri di configurazione del modello settati durante l'addestramento iniziale. Tra i parametri si possono ad esempio citare il *learning rate*, che rappresenta la dimensione del passo (*step size*) che l'algoritmo di ottimizzazione sopra citato compie a ogni iterazione al fine di minimizzare la funzione di costo sopra descritta; la *profondità della rete*, che rappresenta il parametro riferito alla numerosità degli *strati* sequenziali della rete neurale posti tra l'input e l'output del modello o la *dimensione dei batch*, ovvero l'iperparametro che determina quanti esempi di addestramento vengono analizzati dal modello prima che venga effettuato un singolo aggiornamento dei pesi del modello stesso (i modelli GenAI usano batch di migliaia di campioni per stabilizzare l'apprendimento su miliardi di parametri.).

Affinché la Pubblica Amministrazione con profilo da *Operatore esperto* possa garantire l'affidabilità del sistema sul quale sta effettuando attività di training e validazione, questa deve tener conto delle seguenti attività critiche:

- **Prevenzione dell'Overfitting:** Monitorando la *validation loss* rispetto alla *training loss*, è possibile applicare tecniche di *Early Stopping*, interrompendo l'addestramento nel momento in cui il modello inizia a perdere *capacità di generalizzazione*. Il *training loss* rappresenta il valore della funzione di costo *J(w, b)*, definita precedentemente, e viene calcolato esclusivamente sui dati che il modello sta usando per imparare, indicando quanto bene il modello stia aderendo ai dati di addestramento. Questo parametro, nei casi ottimali, deve diminuire in modo costante durante la fase di training. Se ciò non si verifica, potrebbe essere un segnale di bassa disponibilità di capacità computazionale oppure che il learning rate impostato non è ottimale.

  Il parametro *validation loss,* invece, rappresenta il valore della funzione di costo calcolato su un set di dati "invisibile" al modello durante l'aggiornamento dei pesi. Questo parametro rappresenta la misura della qualità reale del modello, ovvero, indica come il modello si comporterebbe nel mondo reale su dati mai visti prima. Se durante la fase di validazione del modello, il *training loss* scende ma contestualmente il *validation loss* inizia a salire, si può dedurre che il modello sta smettendo di imparare e sta iniziando a memorizzare, causando il cosiddetto fenomeno dell'*Overfitting*.

  La *capacità di generalizzazione,* per un modello addestrato da una Pubblica Amministrazione con profilo di *Operatore esperto*, è una caratteristica molto importante in quanto rileva la capacità di un modello di fornire risposte corrette anche a quesiti o dati leggermente diversi da quelli usati per addestrarlo, garantendo quindi una adeguata affidabilità del servizio pubblico erogato attraverso tale modello.

  Le tecniche di *early stopping*, infine, consistono nella possibilità per la PA che sta operando l'addestramento del modello di interrompere automaticamente il processo di addestramento nel momento esatto in cui la *validation loss* smette di scendere e inizia a risalire. Dal punto di vista del consumo, una corretta applicazione di queste tecniche, conduce ad un risparmio energetico e computazionale non trascurabile.

Quando un modello di IA è addestrato da una PA, la valutazione dell'affidabilità del sistema nel processo di validazione non può basarsi esclusivamente su parametri e caratteristiche di *accuratezza globale* (*Accuracy*). *L'Operatore esperto,* infatti, deve basare le attività di addestramento

su una serie di metriche specifiche per il dominio d'uso, fondamentali per identificare bias sistematici o errori critici:

- **Precision (Precisione):** Misura la capacità del modello di non classificare erroneamente la positività dei campioni esaminati.
- **Recall (Sensibilità):** Valuta la capacità del modello di individuare tutti i casi positivi pertinenti. È vitale in domini di sicurezza o sanità pubblica, dove un "falso negativo" (un caso mancato) rappresenta un rischio elevato.

Nel caso di modelli generativi, la validazione si estende a metriche quali la *Perplexity* (che misura il valore dell'incertezza della scelta della parola successiva in un modello di GenAI), per misurare la coerenza logica del linguaggio e test di *Groundedness*, per verificare che i modelli non generino allucinazioni, ma rimangano ancorati ai dati certificati forniti dalla PA.

Per una PA, questa fase è cruciale. Un modello, infatti, non deve solo essere preciso, ma deve resistere a dati sporchi, tentativi di manipolazione o casi limite che potrebbero portare a decisioni amministrative errate o discriminatorie. Per le applicazioni di GenAI nella PA, la robustezza non è solo un parametro tecnico ma una garanzia di *Accountability*.

Test di Robustezza e Analisi Avversaria

La fase di robustezza costituisce il collaudo di sicurezza del modello. Per una PA con profilo di *Operatore esperto,* un sistema non deve solo essere preciso in condizioni ideali, ma come detto, deve resistere a dati degradati, tentativi di manipolazione o casi limite che potrebbero indurre a decisioni amministrative errate o discriminatorie.

Al fine di condurre al meglio questa fase, una PA dovrebbe osservare le seguenti tecniche di verifica di robustezza:

- **Analisi Avversaria (Adversarial Attacks):** La PA attraverso dei tool facenti parte del modello di architettura, deve testare la vulnerabilità del modello a perturbazioni degli input. Nella GenAI, ciò si traduce nel contrasto alla *Prompt Injection*, ovvero il tentativo di scavalcare le istruzioni di sistema per indurre il modello a generare output non autorizzati o rivelare dati sensibili.

- **Affidabilità e Accountability:** Per le applicazioni di GenAI nella PA, la robustezza è una garanzia di responsabilità istituzionale. Devono essere previsti dei "guardrail" (filtri di contenuto) che analizzino l'output prima della somministrazione all'utente, agendo come una membrana di sicurezza esterna.
- **Mitigazione dei Bias:** La robustezza include la verifica dell'equità (*Fairness*). È necessario accertare che il modello non presenti derive discriminatorie, garantendo che le prestazioni rimangano costanti tra i diversi sottogruppi di popolazione serviti dalla PA.

Specializzazione del Modello: Fine-tuning

Mentre il training iniziale crea un modello con capacità generali, il *Fine-tuning* è il processo di specializzazione di un modello pre-addestrato (come un LLM *open-weights*) su un dominio verticale specifico della Pubblica Amministrazione. Questa tecnica appartiene alla famiglia del *Transfer Learning*.

Per un *Operatore Avanzato o Esperto*, il fine-tuning permette di adattare il comportamento, il tono e il lessico del modello ai procedimenti interni della amministrazione senza dover sostenere gli ingenti costi di un addestramento *ex-novo*.

Requisiti e Buone Pratiche per la PA:

- *Qualità del Dataset Verticale* (braccio *Dati* del modello architetturale di riferimento): Il successo del fine-tuning dipende dalla cura di un dataset piccolo ma di altissima qualità (es. sentenze, pareri legali, modulistica tecnica). La PA deve assicurare che i dati siano privi di pregiudizi e conformi alle norme sulla privacy, poiché in questa fase i parametri del modello vengono effettivamente modificati.
- *Adattamento delle Prestazioni* (braccio *Modelli*): Attraverso tecniche efficienti come LoRA (Low-Rank Adaptation) o QLoRA (Quantized Low-Rank Adaptation), è possibile effettuare il fine-tuning agendo solo su una minima frazione dei parametri del modello. Questo riduce drasticamente il fabbisogno di GPU, rendendo la specializzazione

sostenibile anche su infrastrutture *on-premises* di medie dimensioni, riducendo al contempo anche il consumo energetico derivante dal calcolo computazionale.

Retrieval-Augmented Generation (RAG)

Questa tecnica è, ad oggi, la più rilevante per la PA poiché permette di utilizzare modelli generativi (LLM/SLM) su dati interni riservati senza dover ri-addestrare il modello, garantendo che le risposte siano ancorate a fonti documentali certe.

La **RAG** è un'architettura che estende le capacità del braccio *Modelli* del modello architetturale di riferimento basato sull'orchestratore centralizzato, collegandolo dinamicamente a una base di conoscenza esterna gestita dal braccio *Dati*.

Questa tecnica consente di non fare affidamento esclusivamente sulla conoscenza statica del modello di IA appresa durante la fase di training, bensì il sistema "recupera" (retrieval) documenti e informazioni pertinenti ed utili a formulare una risposta al quesito dell'utente e li fornisce al modello come contesto per generare la risposta.

Per un *Operatore Avanzato o Esperto* della PA, l'implementazione di un sistema RAG deve seguire questi requisiti tecnici:

- **Indicizzazione e Vector Database (Estensione Dati):** I documenti dell'ente (delibere, circolari, regolamenti) devono essere trasformati in vettori numerici (*embeddings*) e memorizzati in un database vettoriale. Questo processo permette all'Orchestratore di effettuare ricerche semantiche, trovando informazioni basate sul significato e non solo sulla corrispondenza esatta delle parole.
- **Prompt Augmentation (Estensione Tool):** L'Orchestratore ha il compito di costruire un "Super-Prompt" che include: 1) La domanda dell'utente, 2) I frammenti di documenti recuperati, 3) L'istruzione di rispondere *esclusivamente* sulla base dei documenti forniti.

- **Citazione delle Fonti e Verificabilità:** A differenza del training puro, la RAG permette al modello di citare i riferimenti normativi esatti da cui ha tratto l'informazione. Questo è un requisito di *trasparenza* essenziale per l'azione amministrativa, riducendo drasticamente il rischio di allucinazioni.

Si possono citare dei vantaggi strategici per la PA che sviluppa sistemi di questo tipo, come ad esempio:

1. **Sicurezza e Privacy:** I dati sensibili rimangono confinati nel braccio Dati dell'architettura della PA e non vengono mai "assorbiti" dai pesi del modello durante un addestramento, riducendo i rischi di *data leakage* (si verifica quando informazioni provenienti dal dataset di test o di validazione "filtrano" nel dataset di addestramento).

2. **Aggiornamento in Tempo Reale:** Per aggiornare la conoscenza del sistema è sufficiente aggiungere un nuovo documento al database vettoriale, senza necessità di costosi cicli di re-training computazionale.

3. **Efficienza dei Costi:** La RAG riduce drasticamente la necessità di calcolo intensivo nel braccio Modelli dell'architettura di riferimento, permettendo l'uso di modelli più piccoli e veloci (SLM) con prestazioni eccellenti su domini verticali.