

Machine Learning

Esercitazione di laboratorio su Convolutional Neural Network (CNN)

In questa esercitazione eseguiremo l'addestramento di una CNN mediante il framework `Caffe`.

Il problema di classificazione da risolvere è lo stesso già affrontato nell'esercitazione dei classificatori. Si tratta di classificare come "Amianto" e "Non amianto" (2 classi) pattern provenienti da un set di immagini aeree acquisite con un drone professionale. A differenza delle esercitazioni precedenti i pattern di input sono "raw data" nella forma di piccoli blocchi di immagine 31×31 pixel (RGB). La CNN deve apprendere autonomamente le feature significative da estrarre dai dati grezzi (*representation learning*) per poi effettuare la classificazione.

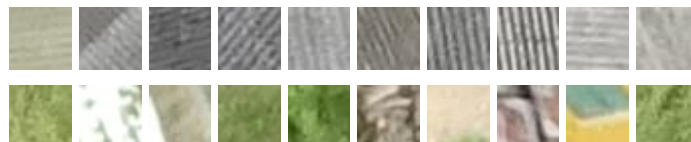


Figura 1 esempi di pattern di classe Amianto (prima riga) e Non-amianto (seconda riga)

1) CONOSCENZE UTILI

Il framework `Caffe` (molto conosciuto per il deploy rapido di CNN allo stato dell'arte) è mantenuto dal "Berkeley Vision and Learning Center" al link <http://caffe.berkeleyvision.org>. Attraverso la seguente documentazione on-line è possibile acquisire competenze minime per utilizzare il framework:

- [Nets, Layers, and Blobs](#): comprendere le basi per la definizione di una rete.
- [Forward/Backward](#): comprendere le computazioni fondamentali di un modello basato su livelli.
- [Loss](#): la definizione della Loss Function per la definizione del task da apprendere.
- [Solver](#): il solver come configurazione di alto livello per la definizione del processo di ottimizzazione.
- [Layer Catalogue](#): catalogo dei livelli disponibili in `Caffe`.
- [Interfaces](#): interfacce mediante le quali è possibile utilizzare `Caffe`.
- [Data](#): come caricare pattern ed etichette.

2) ACCEDERE AD UN SERVER REMOTO PER LA COMPUTAZIONE

- Scaricare ([qui](#)) e installare il software **Xming**, che consente il forwarding grafico X11 da Linux a Windows. Visto che il vostro utente non possiede i diritti di Amministratore, installare **Xming** sul vostro desktop.
- Lanciare l'eseguibile `XLaunch.exe` presente nella cartella in cui si è installato **Xming**. Si aprirà una finestra in cui dovrete procedere con il pulsante "Next" fino alla fine lasciando le impostazioni di default.
- Collegarsi al server Linux via `ssh` mediante l'eseguibile **Putty**. Prima di aprire la sessione `ssh` con le credenziali sotto riportate assicurarsi di aver abilitato nel setting Connection → SSH → X11 il checkbox "Enable X11 forwarding":

IP Server: 137.204.72.37
Username: studente_ml
Password: null_password

- Una volta effettuato l'accesso creare una directory con il proprio nome e cognome oppure il nome del gruppo al quale si appartiene. (Per approfondimenti sull'uso della shell Unix, si veda [qui](#)). In alternativa sarà possibile creare la directory anche tramite l'interfaccia di **WinScp**.

3) MATERIALE NECESSARIO (contenuto nel file .ZIP scaricabile dal sito del corso)

- *cnn_amianto.py* (script python da lanciare per eseguire l'addestramento)
- *cnn_utilities.py* (funzioni di utilità usate dallo script precedente)
- *solver.prototxt* (file che specifica i parametri di training)
- *net.prototxt* (file di configurazione della CNN)

4) ADDESTRARE UNA CNN MEDIANTE CAFFE

- Scaricare il materiale aggiuntivo dell'esercitazione presente sul sito del corso, e utilizzare **WinScp** per copiare tutti i file nella cartella creata sopra.
- Analizziamo ora insieme i 4 file.
- Il server oltre a 1 CPU (12 core), monta una GPU Nvidia Titan X (3072 cores, 12 GB). L'esecuzione può essere lanciata su GPU o CPU semplicemente modificando nel file *cnn_amianto.py* l'impostazione *execution_mode* (GPU/CPU).
- Lanciate lo script attraverso il comando da console (Putty): *python cnn_amianto.py*. Per non sovraccaricare il server, lo script rimane in attesa di risorse libere (abbiamo impostato un massimo di 5 esecuzioni contemporanee su GPU).
- Il minibatch impostato (*batch_size* in *net.prototxt*) è pari a 100. Visto che il training set è composto da 20000 pattern, per eseguire il training di un'epoca occorrono 200 iterazioni. Le 1000+1 iterazioni (*train_iterations*) inizialmente previste in *cnn_amianto.py* coprono 5 epoche.
- Potete "giocare" con i seguenti parametri per cercare di ottenere prestazioni migliori:
 - *train_iterations* (in *cnn_amianto.py*)
 - *base_lr* (in *solver.prototxt*) provare a modificare il learning rate
 - *type* (in *solver.prototxt*) provare a impostare "sgd" o "adam"
 - *num_output* (in *net.prototxt*) solo nei livelli di convoluzione della rete (presenti 3 livelli di convoluzione).

N.B.: Se dopo la prima esecuzione rinominate il file di log *trainLogCur.txt* in *trainLogPre.txt* nel grafico del run successivo (se non avete variato il numero di iterazioni) verrà visualizzato con una linea tratteggiata l'andamento del precedente run (utile per fare confronti). Siccome usando la GPU caffe non è deterministico, lanciando con gli stessi parametri e lo stesso seme possiamo ottenere risultati leggermente diversi.