

TensorBoard

TensorBoard

Premessa

TensorBoard è uno strumento di visualizzazione interattiva fornito da *TensorFlow*.



Le principali caratteristiche della TensorBoard includono:

- ▶ *Visualizzazione di grafi computazionali.*
- ▶ *Monitoraggio delle metriche.*
- ▶ *Visualizzazione di immagini.*
- ▶ *Esplorazione dei dati di input.*
- ▶ *Visualizzazione delle distribuzioni dei dati.*
- ▶ *Visualizzazione di grafici*
- ▶ *Visualizzazione di cluster.*
- ▶ *Analisi di esperimenti.*

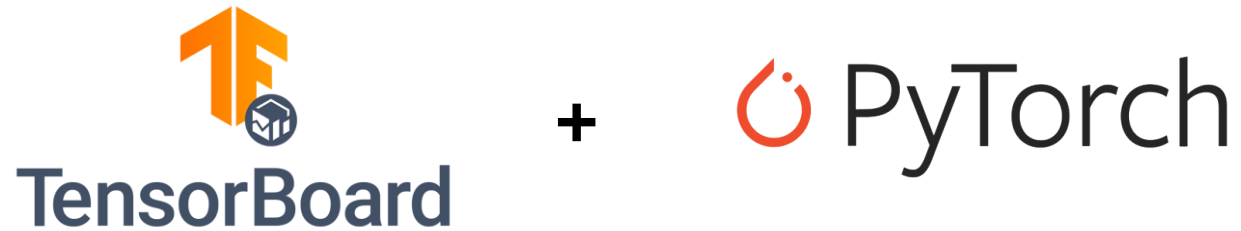




TensorBoard

Installazione

La **TensorBoard**, pur essendo fornita da *TensorFlow*, può venire sfruttata anche da altri framework, fra cui PyTorch.



Dato un ambiente virtuale Python con il framework *PyTorch* installato, è possibile accedere alla **TensorBoard** ed utilizzarla solamente dopo l'installazione del pacchetto dedicato.

L'installazione può essere fatta tramite i gestori **conda** o **pip**:

- *pip install tensorboard*
- *conda install tensorboard*



TensorBoard

Avvio da terminale

Una volta installato il pacchetto Python dedicato alla **TensorBoard**, sarà possibile avviarne una sessione in diversi modi:

Avvio sessione da terminale:

1. Aprire un terminale e attivarvi l'ambiente virtuale.
2. Eseguire il comando: ***tensorboard --logdir runs***
3. Accedere da browser all'indirizzo: ***http://localhost:6006***

```
(CorsoAI2) tensorboard --logdir runs
TensorFlow installation not found - running with reduced feature set.
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.10.0 at http://localhost:6006/ (Press CTRL+C to quit)
█
```

Con tali comandi, si avvierà una sessione **TensorBoard** dedicata a mostrare le informazioni di log salvate nella cartella ***./runs/***.



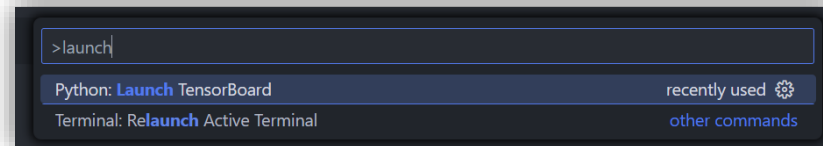
TensorBoard

Avvio da Visual Studio Code

Una volta installato il pacchetto Python dedicato alla **TensorBoard**, sarà possibile avviarne una sessione in diversi modi:

Avvio sessione da vs-code:

1. Aprire il pannello dei comandi di vs-code tramite: ***Ctrl+Shift+P***
2. Digitare e selezionare l'opzione: **>python: Launch TensorBoard**
3. Selezionare una delle opzioni:
 - Usare directory di lavoro corrente.
 - Selezionare un'altra cartella.
 - Immettere l'URL del repository remoto.



Allo stesso indirizzo locale, indicato in precedenza, sarà possibile trovare la sessione **TensorBoard** legata alla directory selezionata.



TensorBoard

Utilizzo nel codice

Utilizzare la **TensorBoard** si traduce nell'inserire parti di codice dedicate a loggare informazioni di vario genere. Nell'integrazione PyTorch, questo oggetto logger è SummaryWriter:

```
from torch.utils.tensorboard import SummaryWriter
```

Eseguito l'import della libreria contenente il logger, è possibile poi creare un oggetto ed utilizzarlo nel codice:

```
writer = SummaryWriter(...)
```

Rif: ([SummaryWriter](#))



TensorBoard

Scrivere sulla tensorboard

L'oggetto `SummaryWriter` possiede una lista di metodi esposti e dedicati al log delle informazioni nella **TensorBoard**:

I principali:

- ▶ `writer.add_scalar()` / `writer.add_scalars()`
- ▶ `writer.add_image()` / `writer.add_images()`
- ▶ `writer.add_graph()` / `writer.add_figure()`
- ▶ `writer.add_embedding()`
- ▶ `writer.add_hparams()`

Rif: ([SummaryWriter](#))



TensorBoard

Loggare metriche

I metodi `writer.add_scalar()` e `writer.add_scalars()` permettono di monitorare l'andamento di una o più metriche sulla **TensorBoard**.

Per metriche si intende qualsiasi valore numerico del quale andamento si vuole tenere traccia nel tempo. Sono metriche l'*accuracy*, la *loss*, la *precision*...

`add_scalar()`

Permette di loggare ed inviare alla **TensorBoard** un nuovo valore associato ad una singola grandezza.

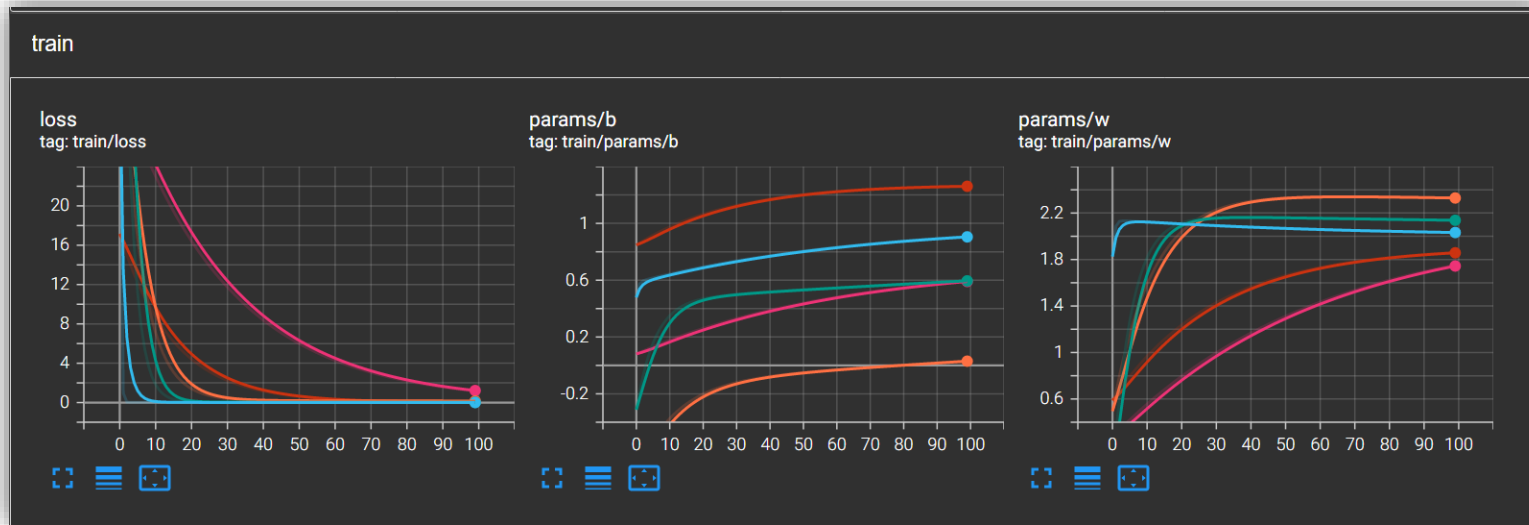
`add_scalars()`

Permette di loggare e di inviare alla **TensorBoard** nuovi valori associati a molteplici grandezze contemporaneamente.



TensorBoard

Loggare metriche



Dopo aver loggato le metriche, saranno visibili nella sessione attiva della **TensorBoard**.

Di seguito un esempio nel quale è mostrato il trend di tre metriche: loss, bias e weight. Le metriche sono state loggate separatamente e per diversi esperimenti. Ogni esperimento è una serie di colore diverso nel grafico.



TensorBoard

Loggare immagini

I metodi `writer.add_image()` e `writer.add_images()` permettono di monitorare una o più immagini nella **TensorBoard**.

Le immagini possono essere di qualsiasi genere: dati di training, test, artefatti, grafici...

`add_image()`

Permette di loggare ed inviare alla **TensorBoard** una sola immagine per step.

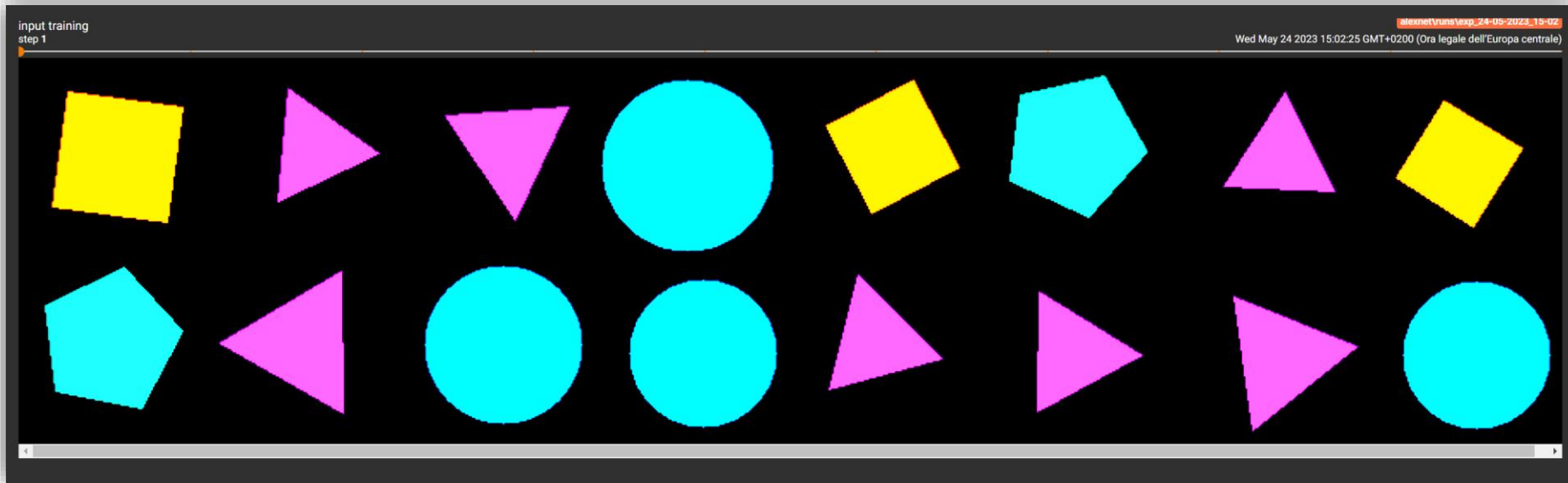
`add_images()`

Permette di loggare e di inviare alla **TensorBoard** gruppi di immagini: un batch di test o training...poi visualizzate a griglia.



TensorBoard

Loggare immagini



Nell'esempio riportato è stato usato il metodo `add_images()` per monitorare sulla **TensorBoard** un intero batch di dati. La **TensorBoard** si occupa di visualizzarli autonomamente a griglia.

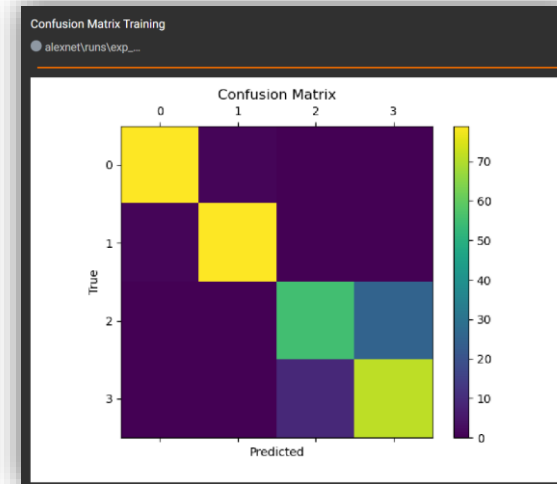
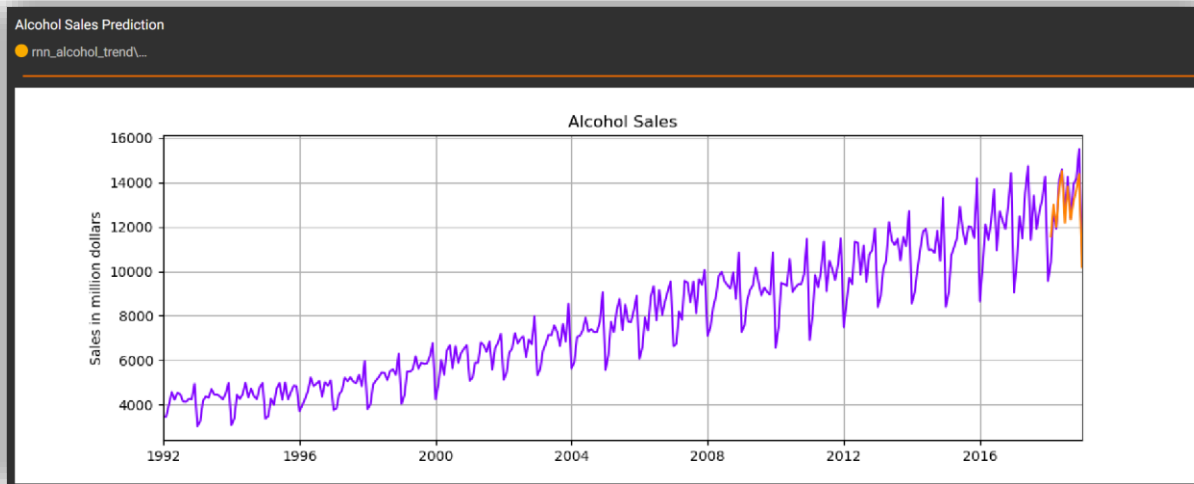
Sarà presente un riferimento all'esperimento, uno slider fra i vari step...



TensorBoard

Loggare figure

Il metodo `writer.add_figure()` permette di monitorare e visualizzare figure create con la libreria *matplotlib* nella TensorBoard.



Aggiungere figure *matplotlib* è utile in quanto, nel caso di altre librerie, sarebbe invece necessario prima convertire le figure e grafici in immagini e poi salvarle.

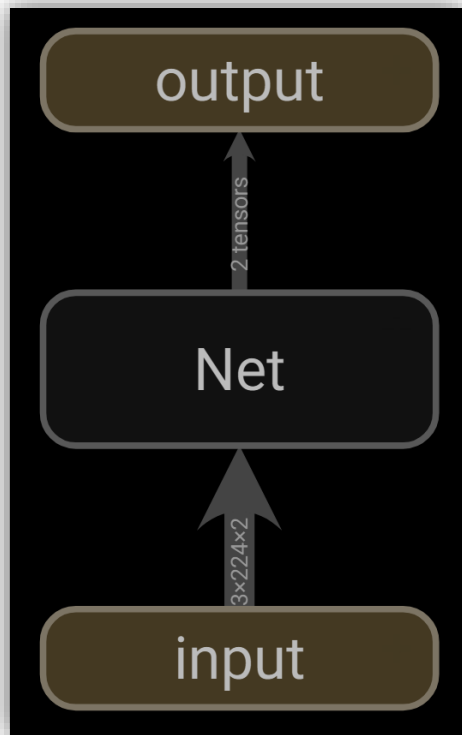
Con le figure sarà possibile monitorare plot, confusion matrix...



TensorBoard

Loggare modelli

Il metodo `writer.add_graph()` permette di monitorare e visualizzare in maniera interattiva modelli di rete neurale nella **TensorBoard**.



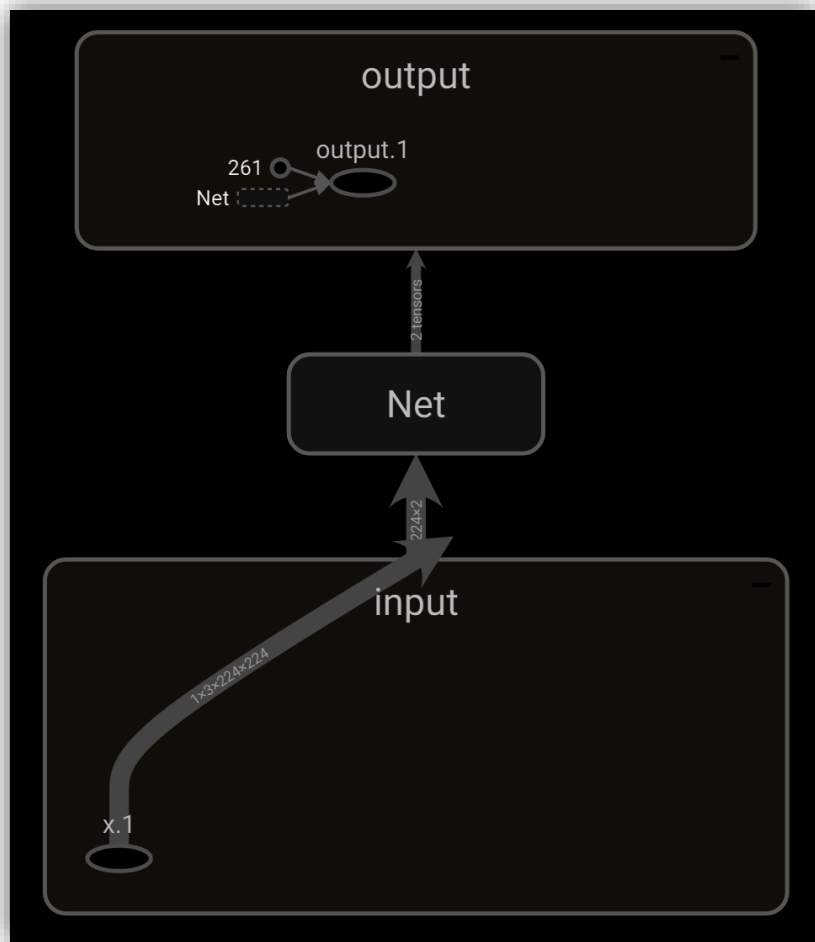
I modelli saranno visualizzabili nella sezione **Graph**.

Del modello saranno mostrati inizialmente: l'input, l'output ed un macro-blocco contenente ulteriori sotto-elementi del modello.

- ▶ Le dimensioni dei tensori che viaggiano nel modello sono visualizzabili.
- ▶ Ogni blocco/modulo è cliccabile ed espandibile.

TensorBoard

Loggare modelli

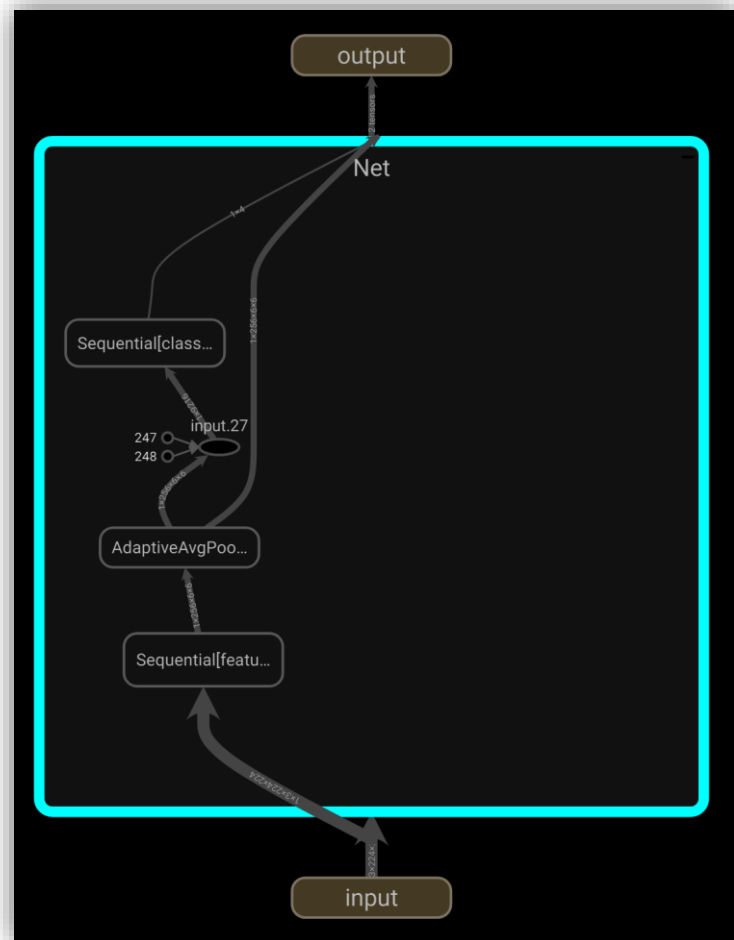


Possono essere espanse le sezioni dedicate all'input e all'output.



TensorBoard

Loggare modelli



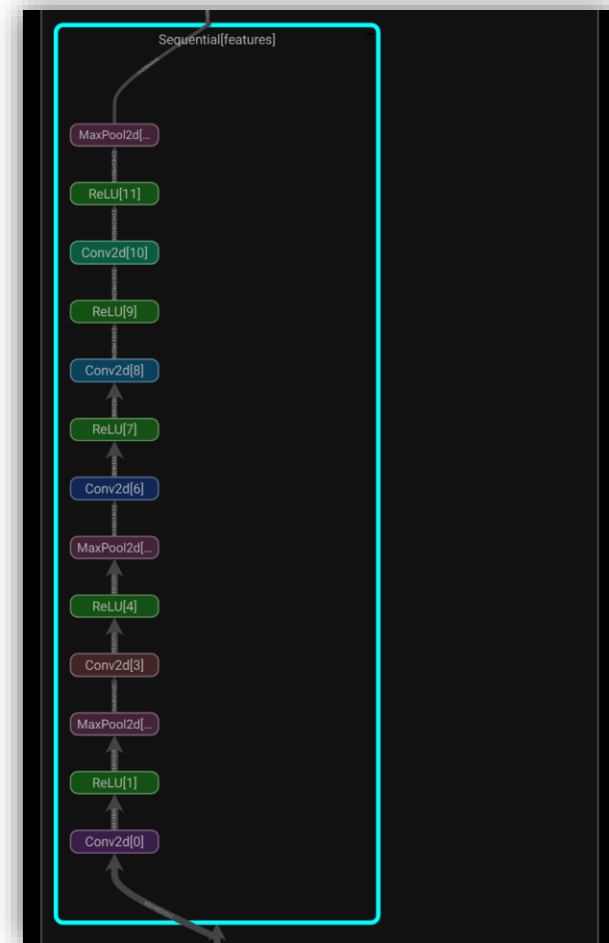
È possibile espandere la sezione **Net** visualizzando informazioni quali:

- ▶ Sotto-blocchi che compongono il modello.
- ▶ Dimensioni dei tensori di input di ogni blocco.
- ▶ Dimensioni dei tensori di output di ogni blocco.



TensorBoard

Loggare modelli



Ogni singolo blocco è a sua volta espandibile. Se ne vedranno nel dettaglio:

- ▶ Strati che costituiscono il blocco.
- ▶ Dimensioni dei tensori di input di ogni strato.
- ▶ Dimensione dei tensori di output di ogni strato.

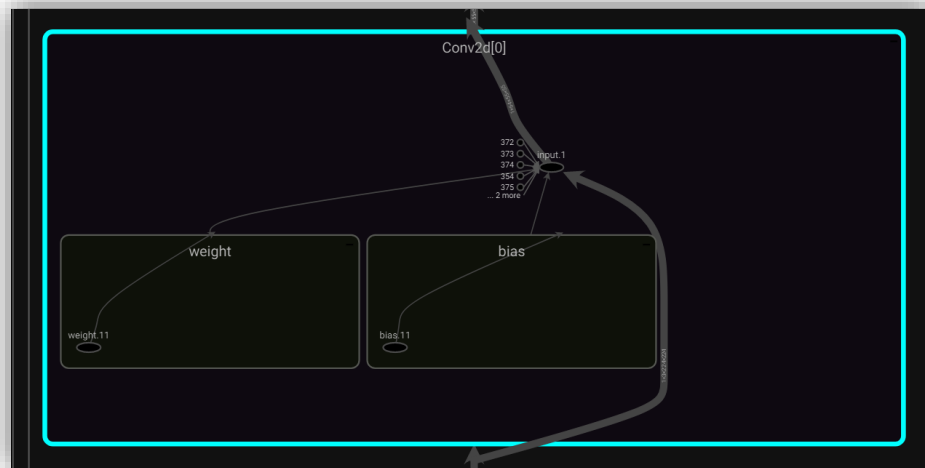




TensorBoard

Loggare modelli

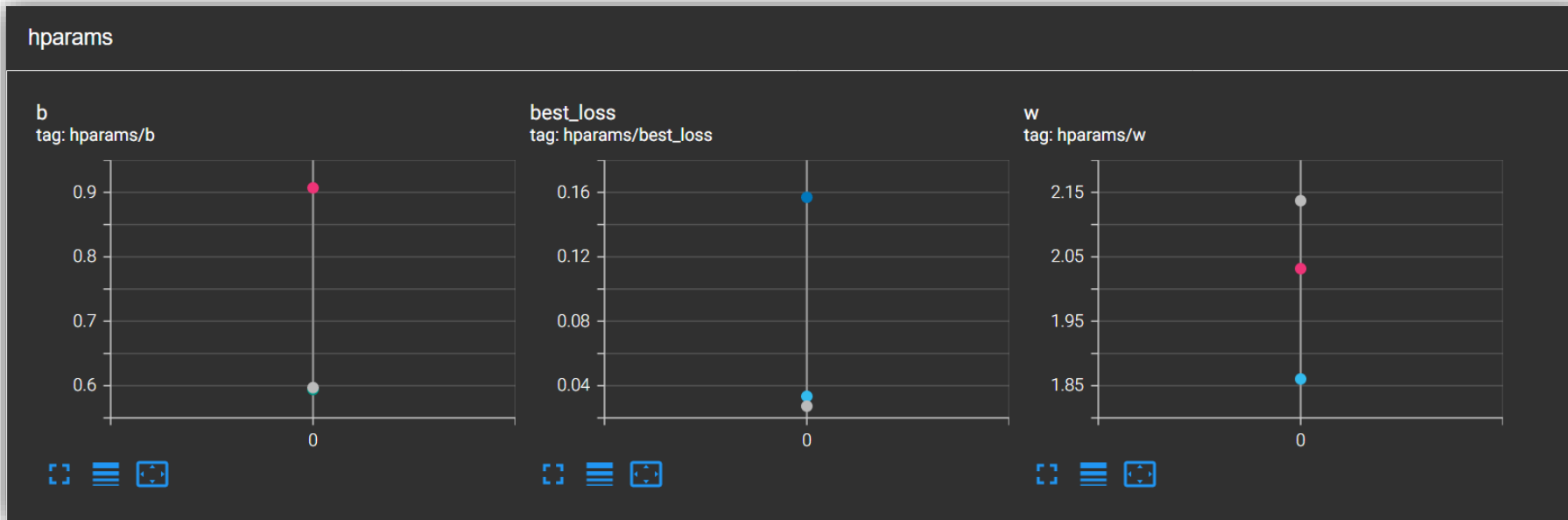
Ogni singolo strato è a sua volta espandibile, fino a visualizzare nel dettaglio i parametri dello strato...





TensorBoard

Loggare iper-parametri



Con `writer.add_hparams()` è permesso monitorare e visualizzare iper-parametri nella **TensorBoard**. Questi ultimi saranno consultabili nella sezione **Scalars**, **Hparams**...

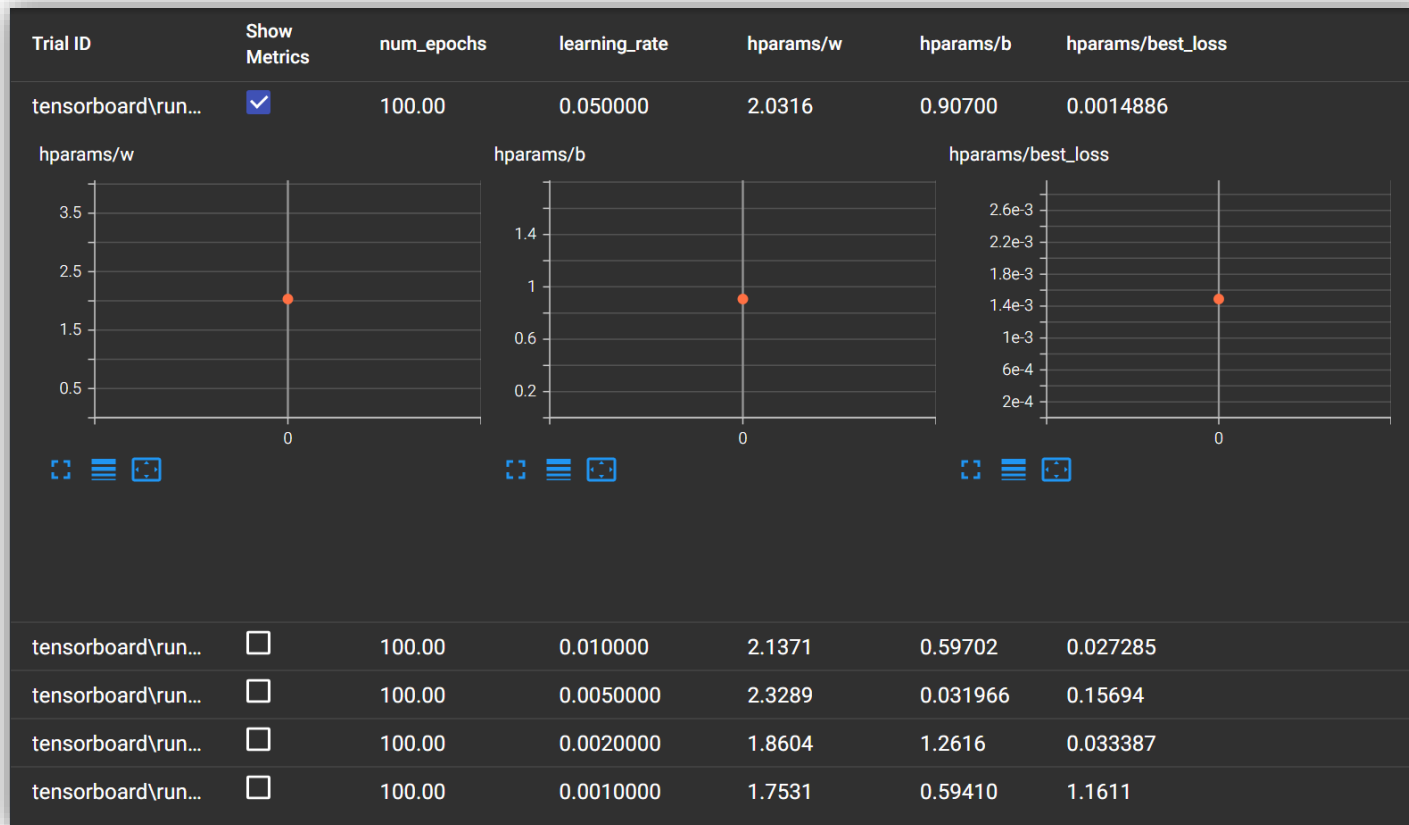
Nell'esempio sono mostrati *loss*, *bias* e *weight* raggiunte al termine di diversi esperimenti.



TensorBoard

Loggare iper-parametri

Una visualizzazione di dettaglio è consultabile nella sezione **Hparams**.
Table view riporta per ogni esperimento gli iper-parametri monitorati.

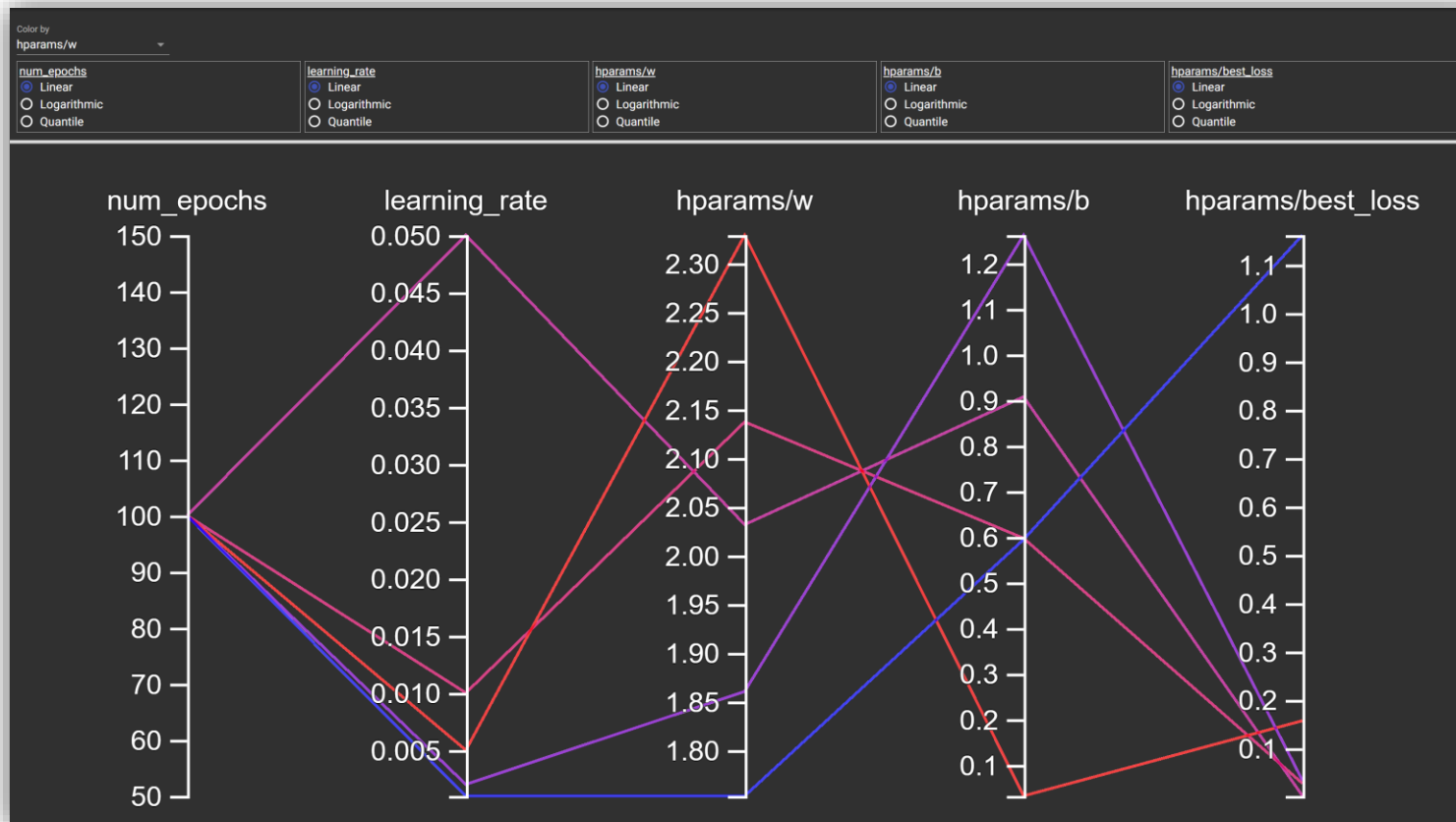




TensorBoard

Loggare iper-parametri

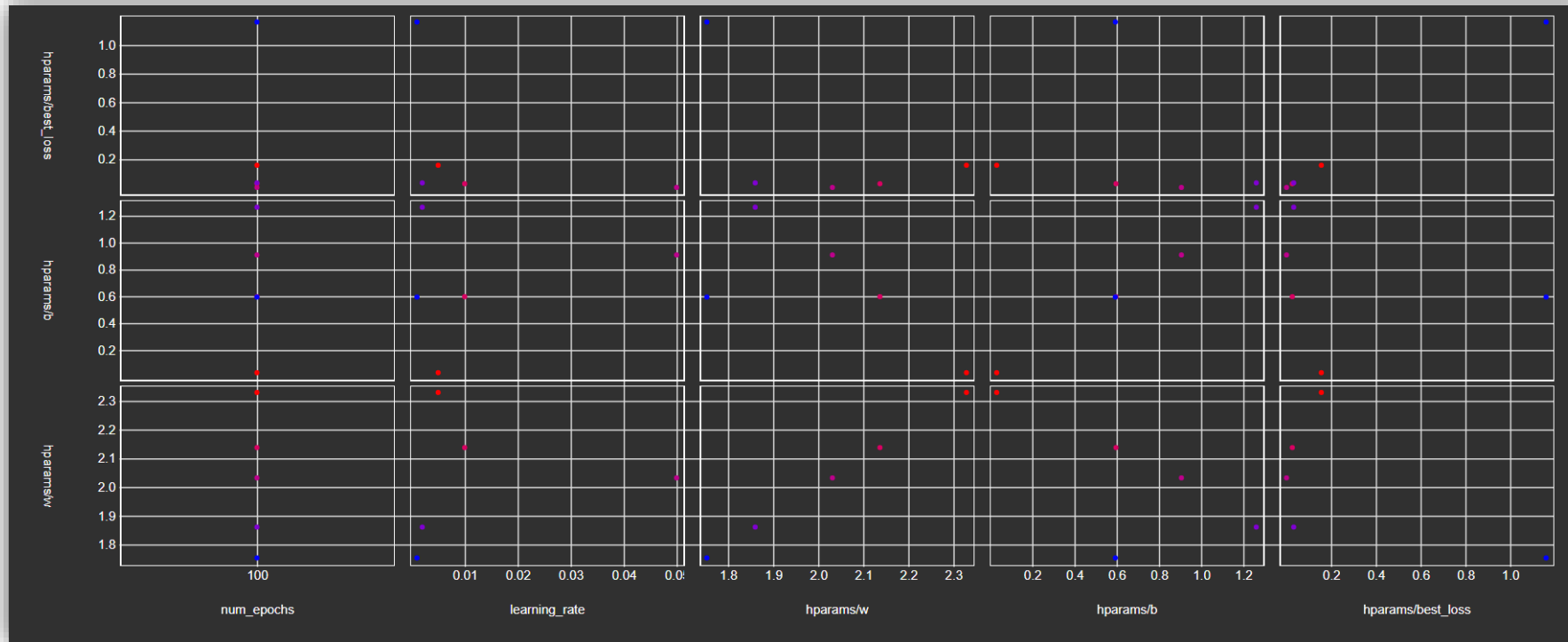
In **Parallel Coordinates View** è riportata in modo schematico la variazione degli iper-parametri.





TensorBoard

Loggare iper-parametri



In ***Scatter Plot Matrix View*** sono riportati gli scatter-plot degli iperparametri e del confronto fra essi.



TensorBoard

Loggare embeddings

Con il metodo `writer.add_embeddings()` è possibile loggare nella **TensorBoard** una codifica / una rappresentazione appresa sui dati, ad esempio, visti in addestramento.

Il termine **embedding** sta ad indicare appunto questo:

*Una tecnica di visualizzazione che mira a rappresentare dati **n-dimensionali** in uno spazio dimensionale inferiore pur mantenendo valide le relazioni che legano i dati stessi.*

La codifica, se appresa correttamente, rappresenterà in maniera significativa le relazioni che legano i dati fra loro...i pattern...le caratteristiche importanti che permettono di distinguerli nello spazio.

Proviamo?

