



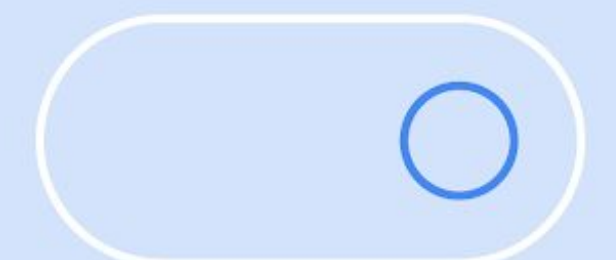
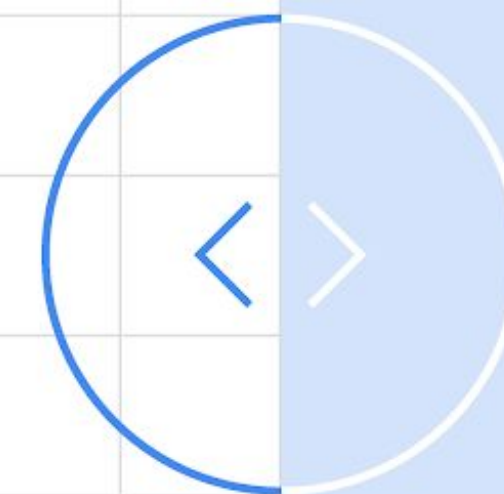
Weights and Biases for better machine learning

Make the best outta your ML models



Sayak Paul
PyImageSearch
[@RisingSayak](https://twitter.com/RisingSayak)

Google Developers

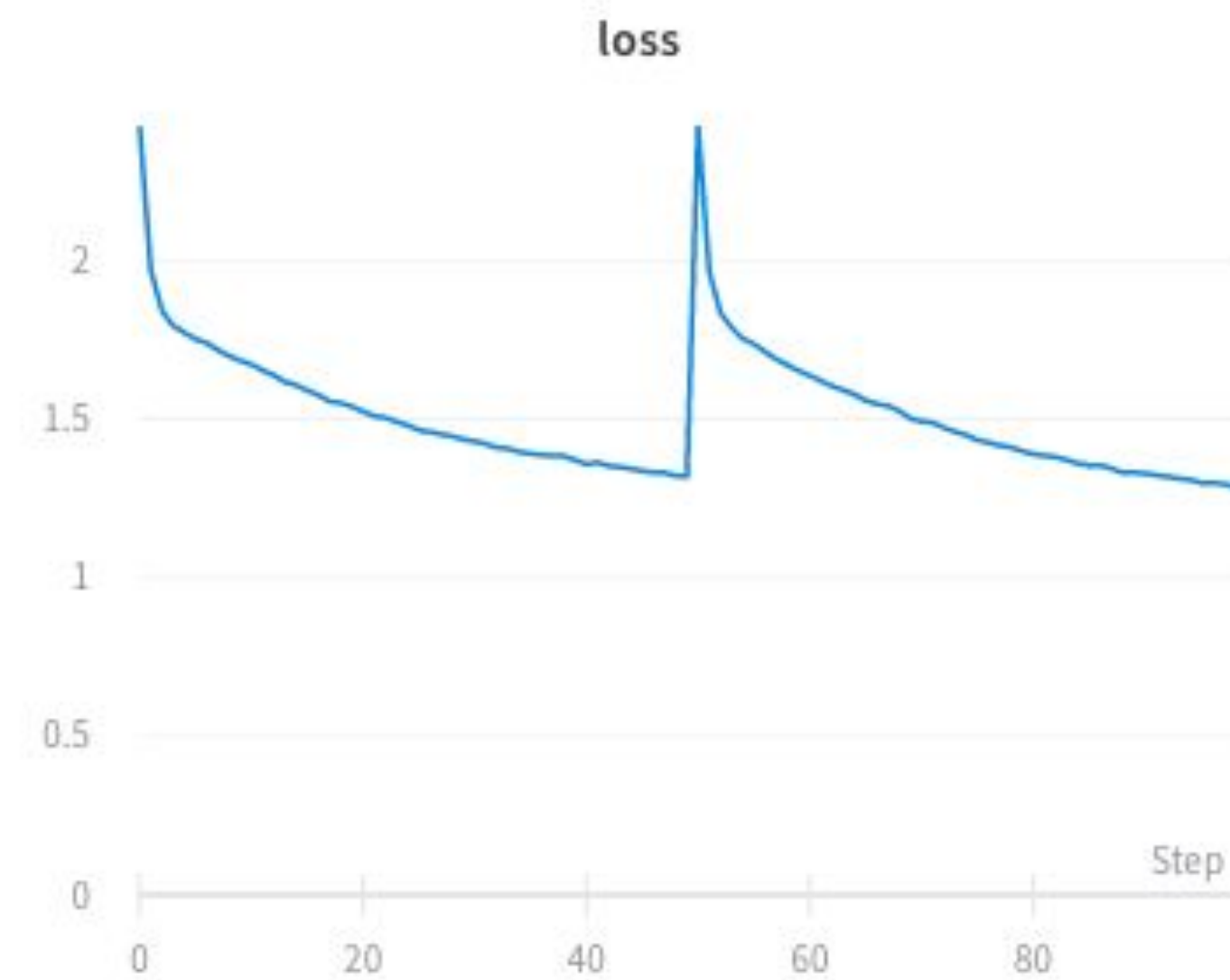
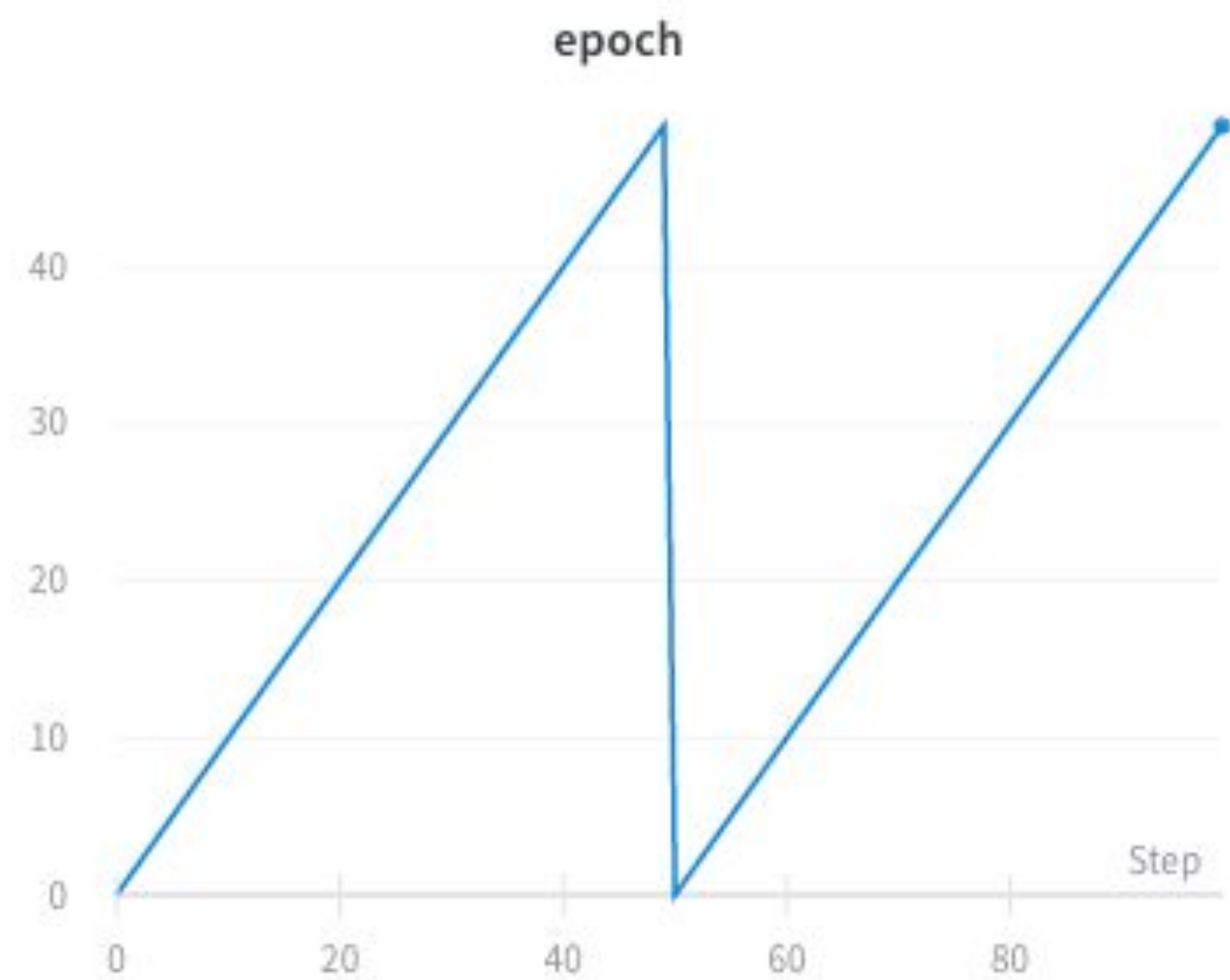
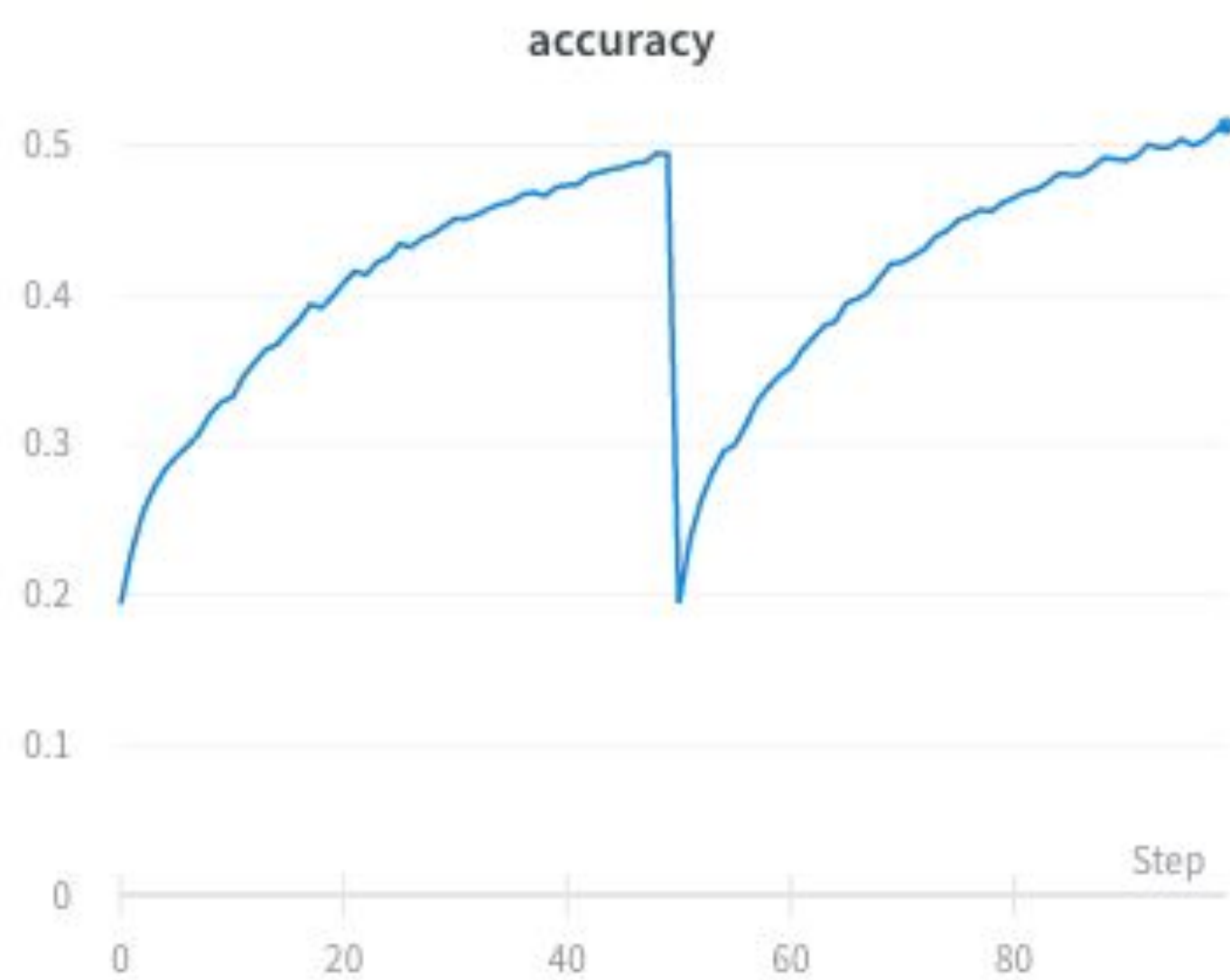


Machine learning involves rapid experimentation

- Lots of experiments, some of them are good, not all of them

Machine learning involves rapid experimentation

- Same boilerplate code to
 - Generate plots (TensorBoard but customization is a bit more involved)



Machine learning involves rapid experimentation

- Same boilerplate code to
 - Monitor system metrics like CPU & GPU usage, memory footprint, and so on
 -

CPU Utilization (%)



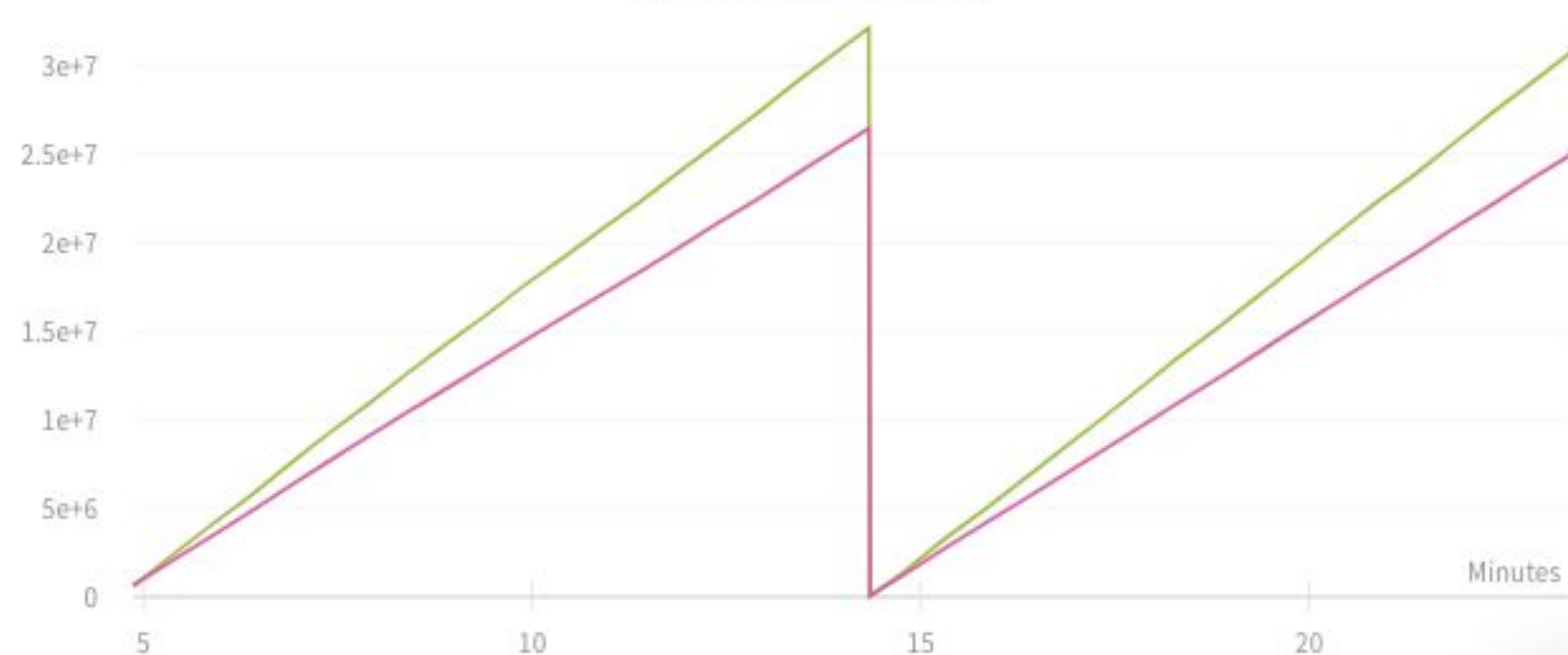
System Memory Utilization (%)



Disk I/O Utilization (%)



Network Traffic (bytes)



Machine learning involves rapid experimentation

- Same boilerplate code to
 - Save the best model weights
 -

Machine learning involves rapid experimentation

- Same boilerplate code to
 - Save the best model weights
 - Analyze the prediction dynamics of a model
 -



Happy



Neutral



Happy

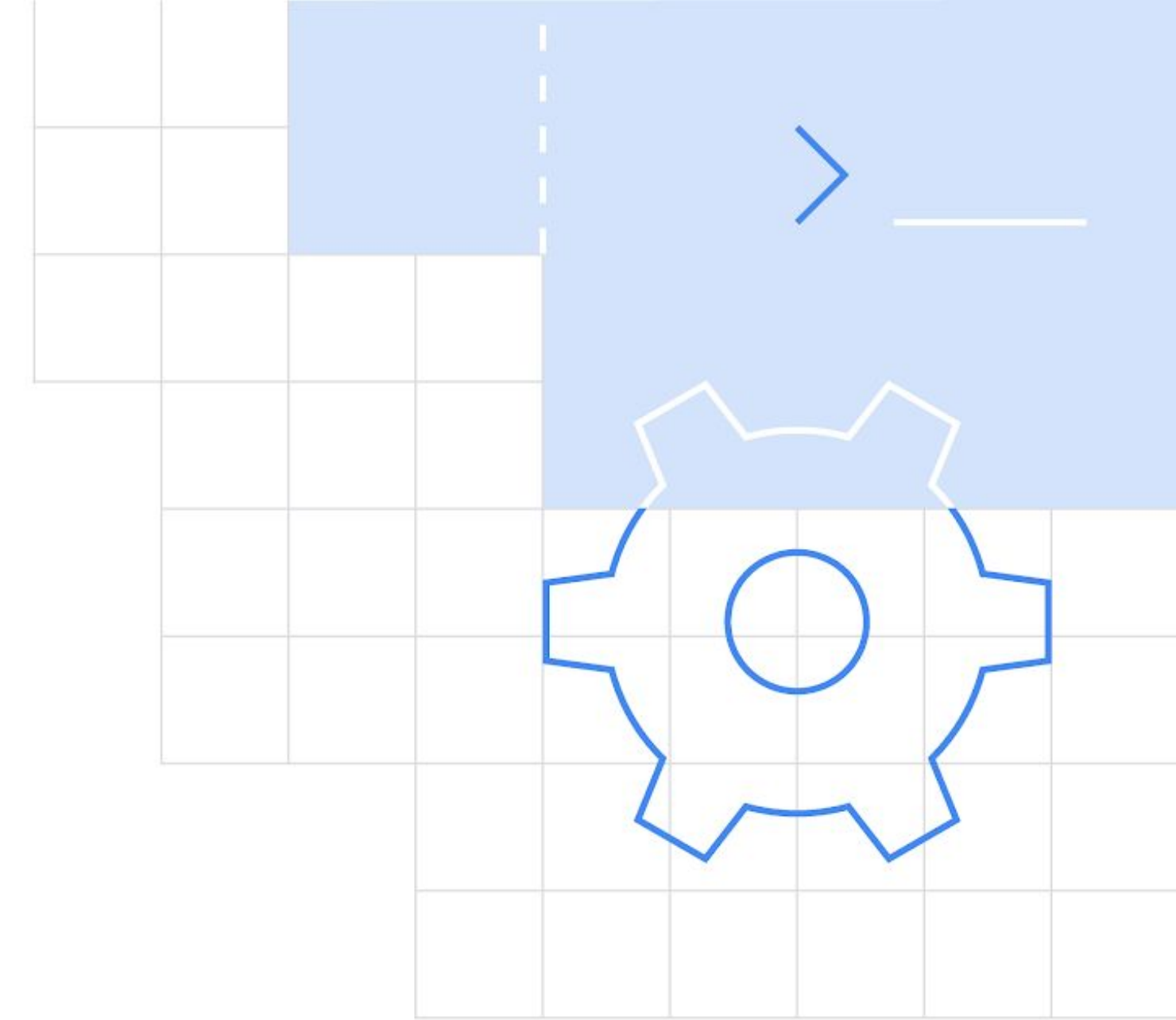


Neutral



Fear

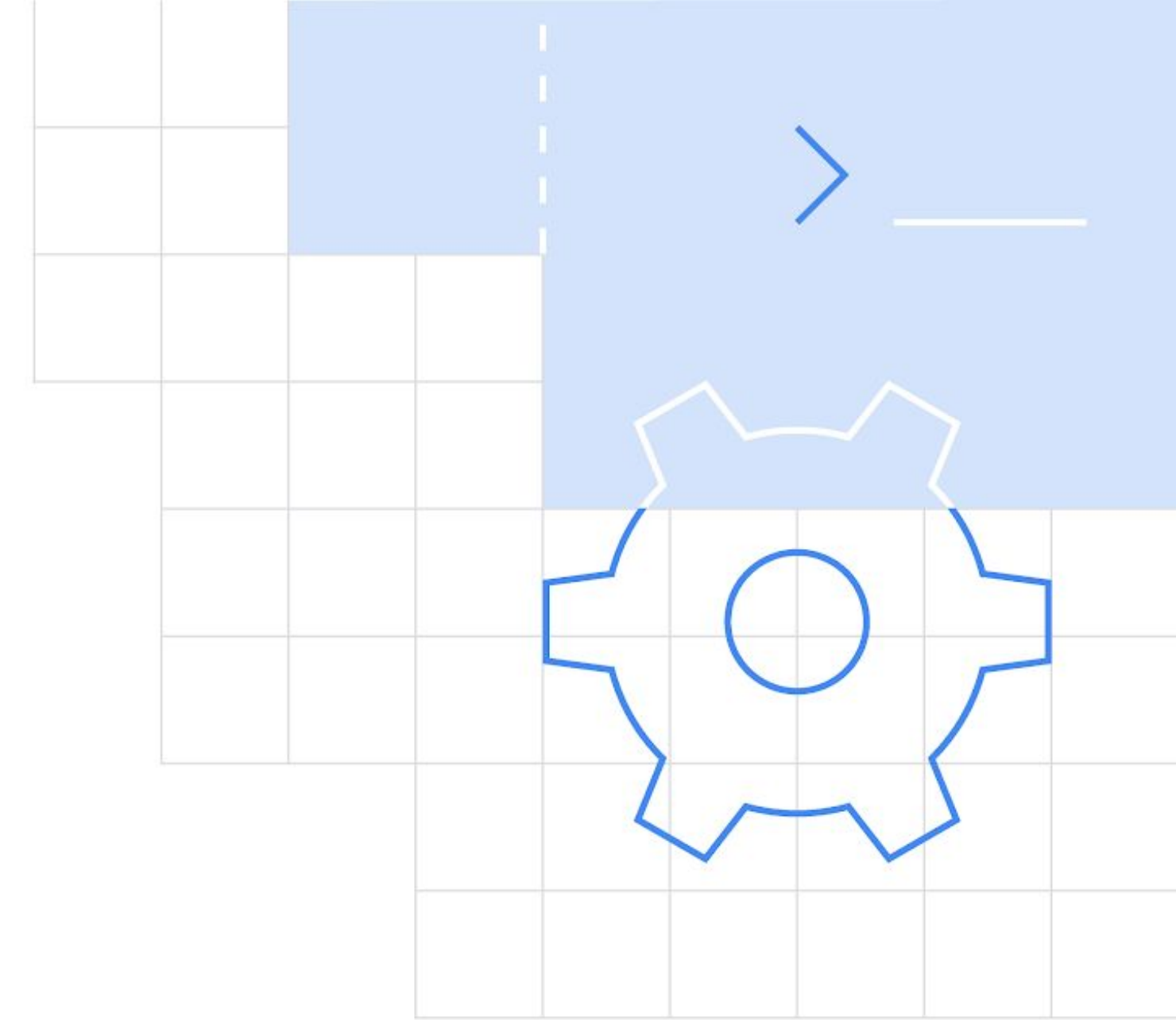
What if all of these
could be done by ...



[illegible]

And *done!*

http://bit.ly/wandb_demo



But that's not all!

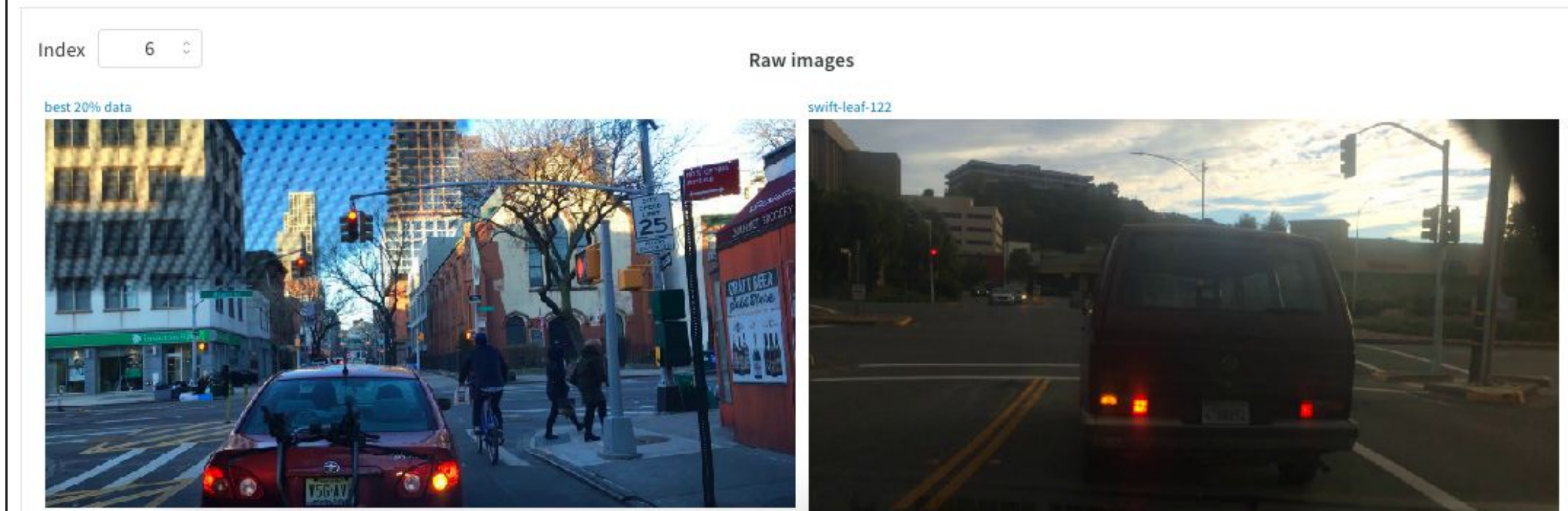
- Create astounding reports!

Understand a dashboard scene with semantic segmentation

A self-driving car must functionally understand the road and its environment the way a human would from the driver's seat. One promising computer vision approach is **semantic segmentation**: parse visual scenes from a car dashboard camera into relevant objects (cars, pedestrians, traffic signs), foreground (road, sidewalk), and background (sky, building). Semantic segmentation annotates an image with object types, labeling meaningful subregions as a tree, bus, cyclist, etc. For a given car dashboard photo, this means labeling *every pixel* as belonging to a subregion.

Below you can see examples in two columns: raw images, the model's predictions, and the ground truth (correct labeling). Buildings are orange, cars are pink, road is cobalt blue, and pedestrians are beige. In the left column, the model can't differentiate between a pedestrian and a rider on a bicycle (magenta and cyan in ground truth, beige in prediction). Note how the hazy conditions in the right column make the model predictions blurry around the boundaries between dashboard and road, or vehicle and road).

Example segmentation maps



But that's not all!

- Customize the data generated by Weights and Biases.

CHARTS



Line Plot



Bar Chart



Histogram



Parameter
Importance

GALLERIES



Table



Image



Video



Audio



HTML



Plotly



3D Object

DATA FRAMES



Confusion
Matrix



Data Frame
Table




















PAGE ELEMENTS



Markdown

But that's not all!

- Get a tabular summary of your experiments.

<input type="checkbox"/>  Name (10 visualized)	State	Notes	User	Tags	Created ▾	Runtime	Sweep	accuracy	epoch	loss	val_accu...	val_loss
•   mini-vgg-wo-data-aug-sgd-lsuv	finished	Add notes	saya...		6d ago	3m 10s	-	0.607	49	1.044	0.4497	1.553
•   mini-vgg-prog-channels-wda-lsuv	finished	Add notes	saya...		6d ago	2m 31s	-	0.5504	49	1.181	0.4351	1.532
•   mini-vgg-prog-channels-wo_da	finished	Add notes	saya...		6d ago	2m 23s	-	0.5016	49	1.291	0.4407	1.488
•   mini-vgg-prog-channels-2	finished	Add notes	saya...		6d ago	7m 52s	-	0.3934	49	1.552	0.4007	1.556
•   mini-vgg-prog-channels	finished	Add notes	saya...		6d ago	2m 53s	-	0.3283	17	1.67	0.3362	1.683
•   mini-vgg-net-daug-sgd-clr	finished	Add notes	saya...		6d ago	6m 38s	-	0.3129	37	1.791	0.3566	1.655
•   mini-vgg-net-da-sgd	finished	Add notes	saya...		6d ago	9m 7s	-	0.5129	49	1.287	0.5572	1.183
•   mini-vgg-net	finished	Add notes	saya...		6d ago	1m 26s	-	0.7534	10	0.6755	0.5493	1.479
•   miniinception-net	finished	Add notes	saya...		6d ago	8m 45s	-	0.8818	12	0.3208	0.501	1.79

But that's not all!

- Seamless hyperparameter tuning!

Steps

Configuration File

Check out our [docs](#), or follow these steps to get started:

- 1) Set up the sweep by choosing the metric to optimize and the hyperparameters to change.
- 2) Download the [configuration file](#) and save it as `sweep.yaml`. Run the command below from your training scripts directory to initialize a sweep.

```
$ wandb sweep sweep.yaml
```

- 3) Run the wandb agent command to start a sweep.

```
$ wandb agent <sweep_id>
```

This will cause the agent to execute something like:

```
$ train.py --epochs=11 --layers=47 \
  --method=grid --batch_size=30
```

Once the run finishes, the agent will run another command with different parameter values.

You can run multiple wandb agents on a single machine or different machines.

Settings

Parameters

Training Script

train.py

Parameter Search Algorithm

Bayes Search

Metric to Optimize

loss

Goal

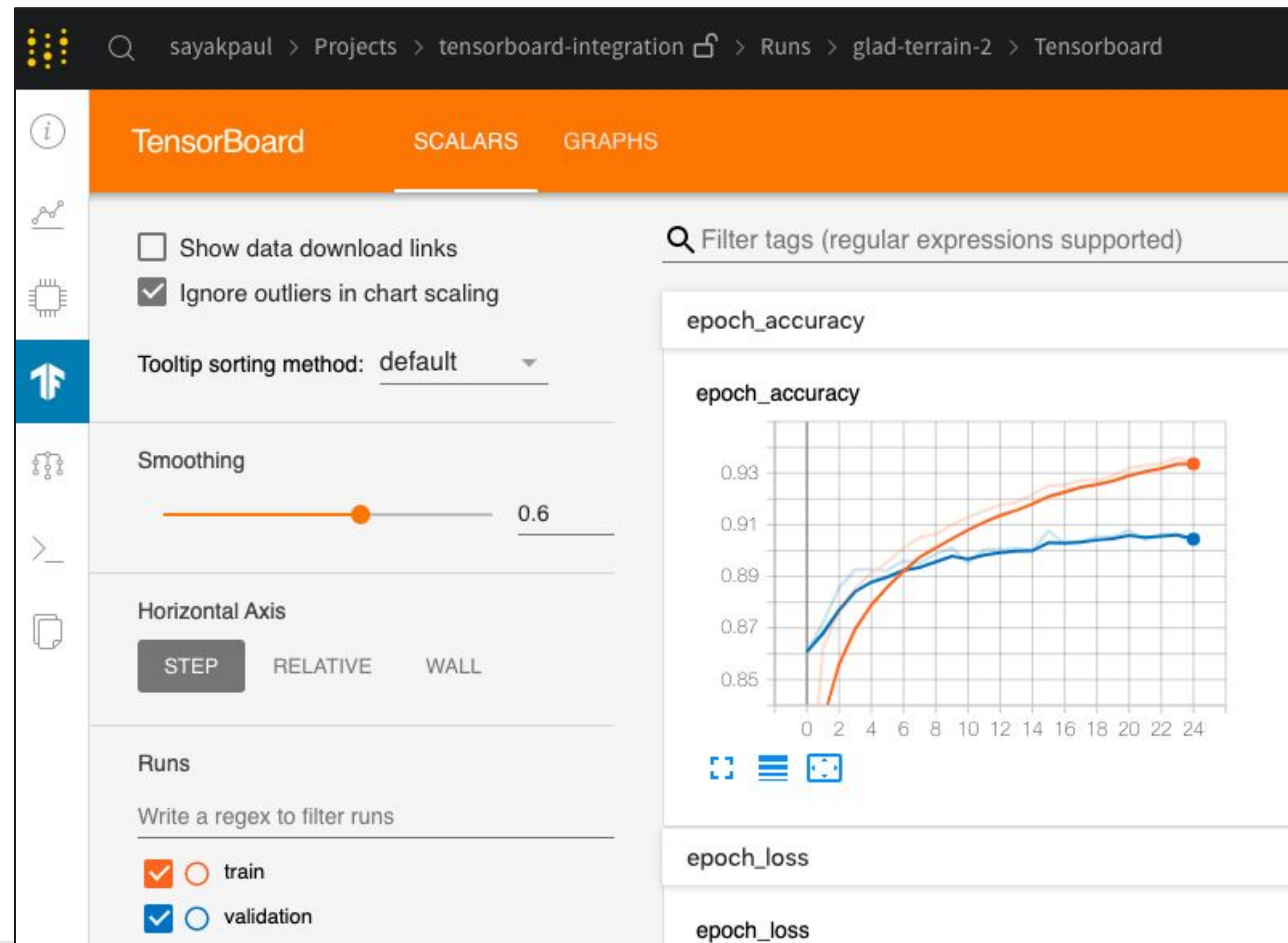
Minimize

Early Termination Algorithm

None

But that's not all!

- Get your TensorBoard hosted automatically.

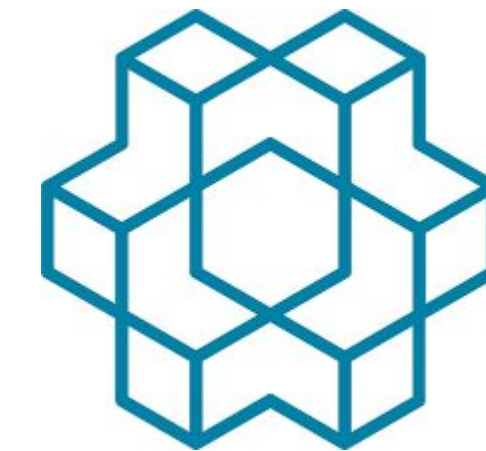


But that's not all!

- Prediction dynamics for different input modalities.

text		
Step 18 ▾		
Text ▲	Predicted Label	True Label
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"	["cs.CL: 99.21%","cs.LG: 21.01%"]	"cs.CL"
"MultiFiT: Efficient Multi-lingual Language Model Fine-tuning"	["cs.CL: 99.84%","cs.LG: 4.30%"]	"cs.CL, cs.LG"
"On the Variance of the Adaptive Learning Rate and Beyond"	["cs.LG: 91.80%","stat.ML: 88.42%"]	"cs.LG, stat.ML"

Works with all of these



It's not framework-limited

- [Mask R-CNN semantic segmentation](#)
- [Visualizing 3D Bounding Boxes](#)

Get started!

- `pip install wandb`
- <https://www.wandb.com/articles>

