**Keras Tuner**

======================================================================

KerasTuner is an easy-to-use, scalable hyperparameter optimization framework that solves the pain points of hyperparameter search.

**Step 1**: Define the Search Space (build\_model function)

You write a build\_model(hp) function where you:

* Use hp.Int(), hp.Float(), etc. to declare which hyperparameters should be tuned.
* Construct a model using these hyperparameters.
* Compile the model.

This defines the search space: the range of values for each hyperparameter.

--------------------------------------------------------------------------------------------------------------

**Step 2**: Initialize the Tuner

You create a tuner object, e.g., RandomSearch:

*tuner = kt.RandomSearch(*

*build\_model, # The function defining the model & search space*

*objective='val\_mae', # Metric to optimize*

*max\_trials=20, # Number of different combinations to try*

*executions\_per\_trial=2, # Each trial is run multiple times for stability*

*...*

*)*

This prepares the search logic and initializes tracking.

--------------------------------------------------------------------------------------------------------------

**Step 3**: Search Starts (tuner.search(...))

*tuner.search(X\_train, y\_train, validation\_split=0.2, epochs=100, batch\_size=16)*

Now the magic begins:

1. For each trial (up to max\_trials):

Tuner samples a random combination of hyperparameters from the search space.

Calls your build\_model(hp) with those values.

Trains the model for the given epochs, using your data.

Evaluates it on the validation set (e.g., val\_mae).

Stores the results.

1. If executions\_per\_trial > 1:

The same model is trained multiple times to average the results and reduce randomness.

**Step 4:** Trials are Scored and Ranked

* After all trials, Keras Tuner ranks them by the objective metric (e.g., lowest val\_mae).
* You can access the top models or hyperparameters using:

*best\_model = tuner.get\_best\_models(num\_models=1)[0]*

*best\_hps = tuner.get\_best\_hyperparameters(num\_trials=1)[0]*

**Step 5:** Use or Retrain the Best Model

Now that you have the best model or its hyperparameters, you can:

* Retrain it on full data
* Save it for deployment
* Visualize the tuning process

**(Optional) Step 6:** Inspect the Search Results

*tuner.results\_summary()*

This shows a summary of all the top trials, hyperparameters, and metrics.

**Quick Analogy**

Think of Keras Tuner as an AI Chef:

* You give it a recipe template (your model architecture).
* It tries different ingredient amounts (hyperparameters).
* It bakes many versions (training).
* And picks the best tasting one (based on val metric)!