

# HYPER PARAMETER TUNING

Mi AI

# HYPERPARAMETERS

## PARAMETERS VS. HYPERPARAMETERS

Parameters are learned through the training procedure. For example, the weights of a neural network.

Hyperparameters are set before training starts, can be tuned through grid search or related methods.

ChrisAlbon

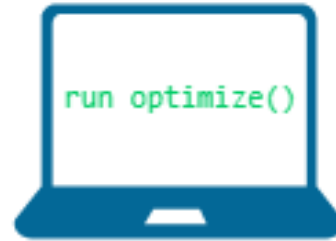
# HYPERPARAMETERS

PARAMETER	HYPERPARAMETER
Estimated during the training with historical data.	Values are set beforehand.
It is a part of the model.	External to the model.
The estimated value is saved with the trained model.	Not a part of the trained model and hence the values are not saved.
Dependent on the dataset that the system is trained with.	Independent of the dataset.

# HYPERPARAMETERS



Hyperparameters



Parameters



Score



n\_layers = 3  
n\_neurons = 512  
learning\_rate = 0.1



Weights  
optimization



85%



n\_layers = 3  
n\_neurons = 1024  
learning\_rate = 0.01



Weights  
optimization



80%



n\_layers = 5  
n\_neurons = 256  
learning\_rate = 0.1



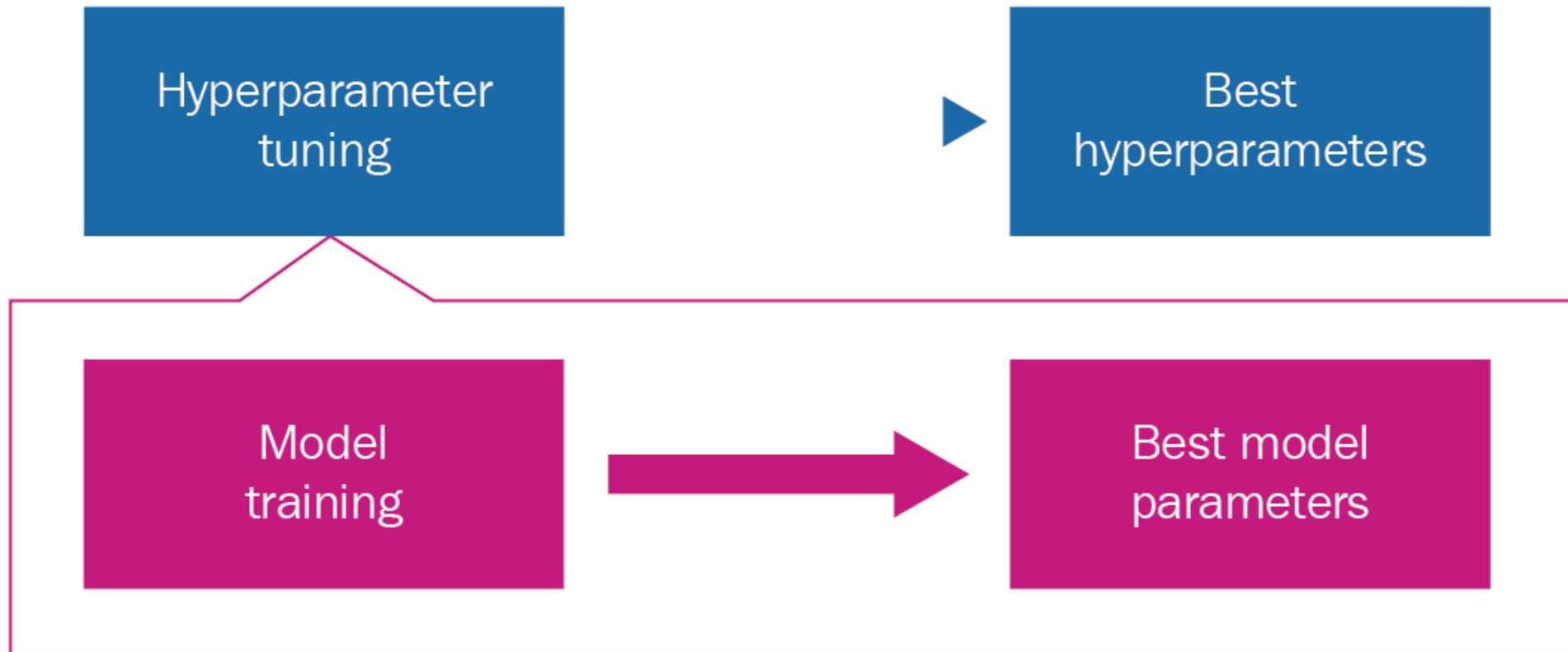
Weights  
optimization



92%

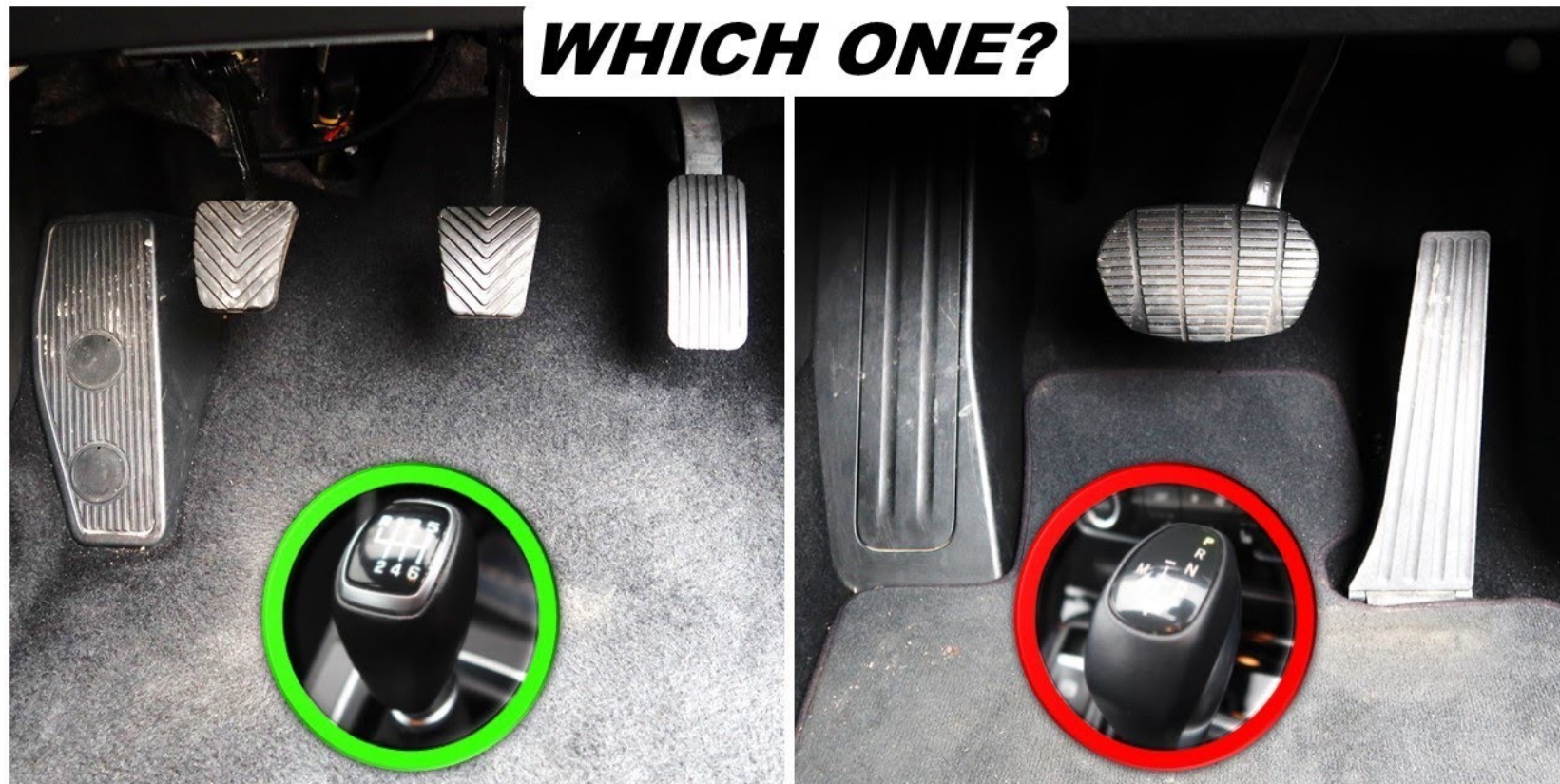
# HYPERPARAMETERS

Hyperparameter tuning vs. model training





# HYPERPARAMETER TUNING



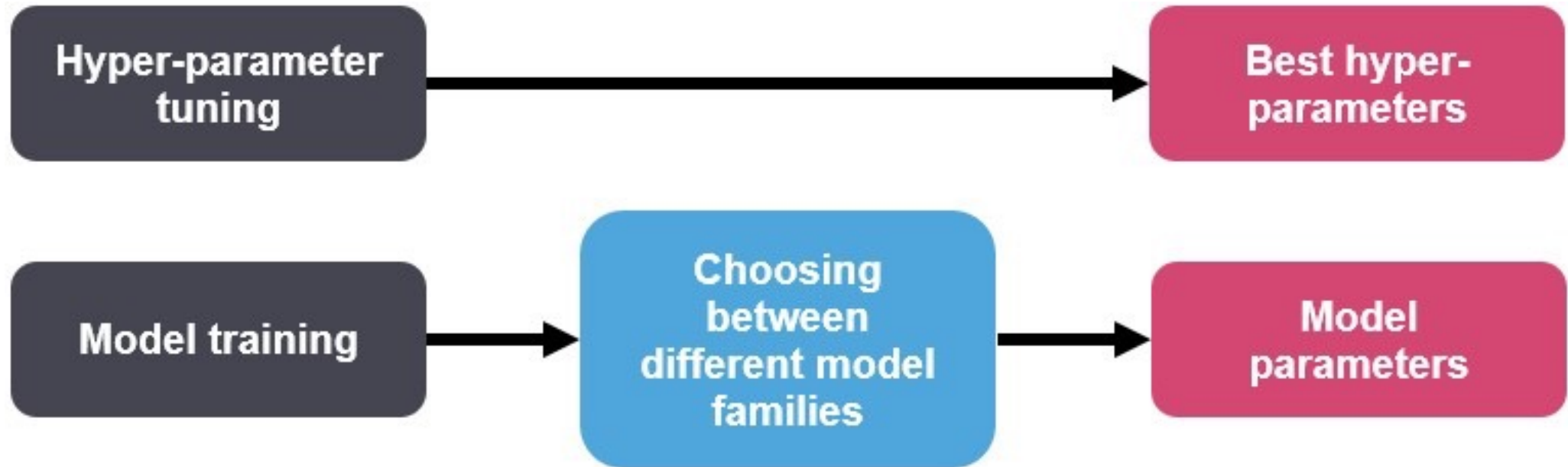
**MANUAL**

**AUTO**

# HYPERPARAMETER TUNING

	Data Source	Hyperparameter1		Hyperparameter2		NDCG	Precision	Recall	F1
		Mode	Value	Mode	Value				
1	IR	topn	500	topn	50	0.7882	0.7173	0.8953	0.7956
2	All	topn	50	topn	20	0.8004	0.7111	0.9020	0.7947
3	Official website	topn	50	topn	20	0.7725	0.7117	0.8921	0.7912
4	Official website	thr.	0.4	topn	20	0.7812	0.7134	0.8871	0.7903
5	IR	topn	500	hitxt	9.0	0.8009	0.7029	0.9156	0.7897
6	Official website	thr.	0.3	topn	50	0.8019	0.6952	0.9138	0.7882
7	IR	topn	50	topn	20	0.7896	0.7095	0.8877	0.7873
8	IR	thr.	0.3	topn	50	0.7859	0.6979	0.9052	0.7864
9	All	thr.	0.4	topn	20	0.7822	0.7112	0.8795	0.7861
10	All	thr.	0.3	topn	50	0.8001	0.6980	0.9020	0.7857
11	Official website	topn	200	topn	50	0.8011	0.6941	0.9088	0.7856
12	IR	topn	200	hitxt	7.0	0.7928	0.7067	0.8931	0.7853
13	All	topn	200	topn	50	0.8037	0.6963	0.9046	0.7851
14	IR	topn	500	hitxt	8.0	0.7956	0.7020	0.9031	0.7851
15	IR	topn	500	hitxt	13.0	0.7981	0.6861	0.9331	0.7842
16	IR	topn	500	hitxt	10.0	0.8163	0.6891	0.9256	0.7841
17	Official website	thr.	0.4	hitxt	6.0	0.7944	0.7069	0.8855	0.7840
18	IR	topn	1000	topn	50	0.7749	0.7257	0.8553	0.7837
19	IR	topn	2000	hitxt	8.0	0.7770	0.6986	0.9057	0.7825
20	IR	topn	500	hitxt	11.0	0.8051	0.6870	0.9256	0.7824

# HYPERPARAMETER TUNING





# HYPERPARAMETER TUNING

## **Advantages of manual hyperparameter optimization:**

- Tuning hyperparameters manually means more control over the process.
- If you're researching or studying tuning and how it affects the network weights then doing it manually would make sense.

## **Disadvantages of manual hyperparameter optimization:**

- Manual tuning is a tedious process since there can be many trials and keeping track can prove costly and time-consuming.
- This isn't a very practical approach when there are a lot of hyperparameters to consider.

# HYPERPARAMETER TUNING

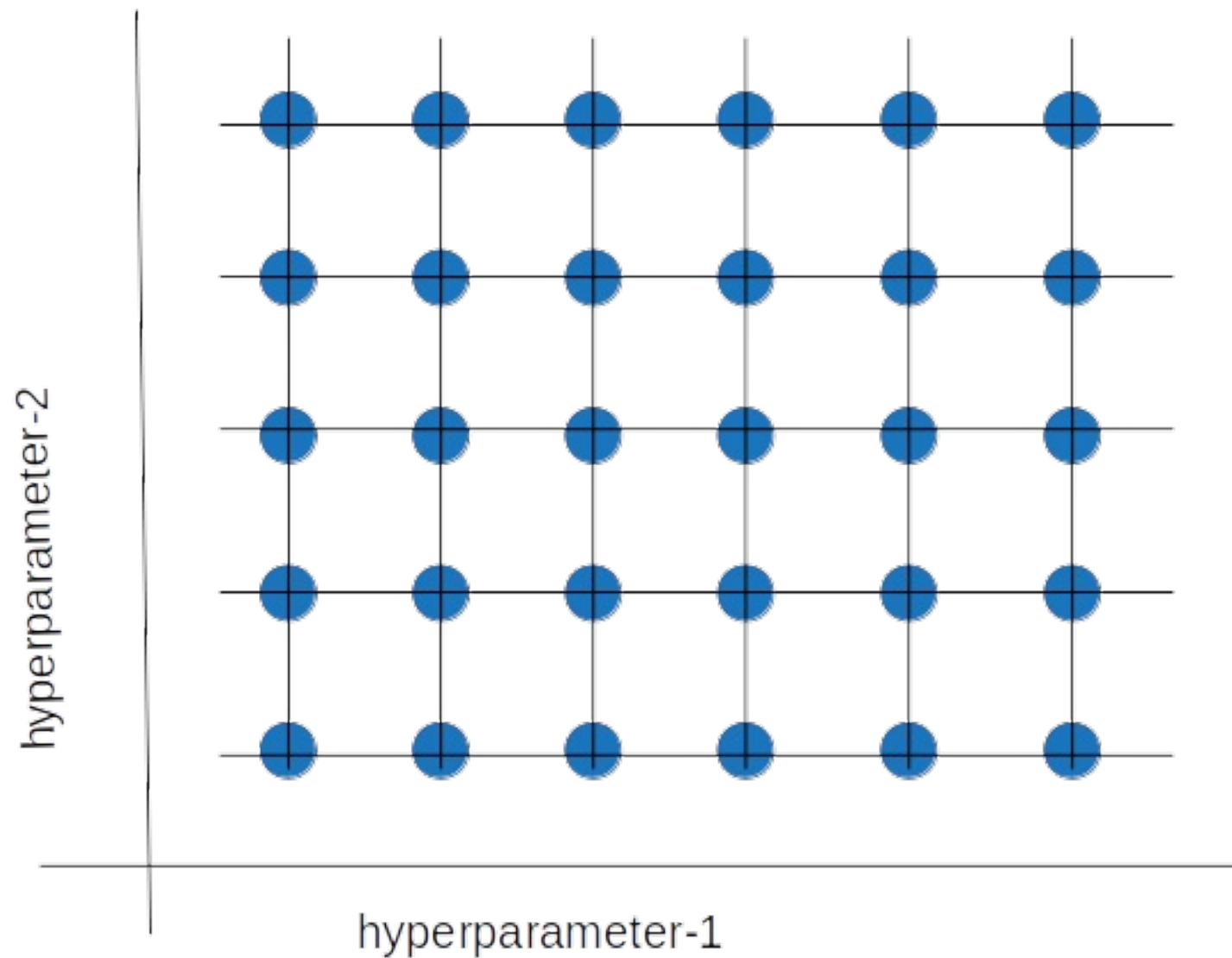
Automated hyperparameter tuning utilizes already existing algorithms to automate the process. The steps you follow are:

- First, specify a set of hyperparameters and limits to those hyperparameters' values (note: every algorithm requires this set to be a specific data structure, e.g. dictionaries are common while working with algorithms).
- Then the algorithm does the heavy lifting for you. It runs those trials and fetches you the best set of hyperparameters that will give optimal results.

# HYPERPARAMETER TUNING

- Grid Search
- Keras Tuner

# GRID SEARCH



# GRID SEARCH

	C=0.001	C=0.01	...	C=10
gamma=0.001	SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	...	SVC(C=10, gamma=0.001)
gamma=0.01	SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	...	SVC(C=10, gamma=0.01)
...	...	...	...	...
gamma=100	SVC(C=0.001, gamma=100)	SVC(C=0.01, gamma=100)	...	SVC(C=10, gamma=100)



# KERAS TUNER

## KerasTuner API

The **Hyperparameters** class is used to specify a set of hyperparameters and their values, to be used in the model building function.

The **Tuner** subclasses corresponding to different tuning algorithms are called directly by the user to start the search or to get the best models.

The **Oracle** subclasses are the core search algorithms, receiving model evaluation results from the Tuner and providing new hyperparameter values.

The **HyperModel** subclasses are predefined search spaces for certain model families like ResNet and XceptionNet.

# HYPERPARAMETERS CLASS

## HyperParameters

- HyperParameters class
- Boolean method
- Choice method
- Fixed method
- Float method
- Int method
- conditional\_scope method
- get method

# TUNER CLASS

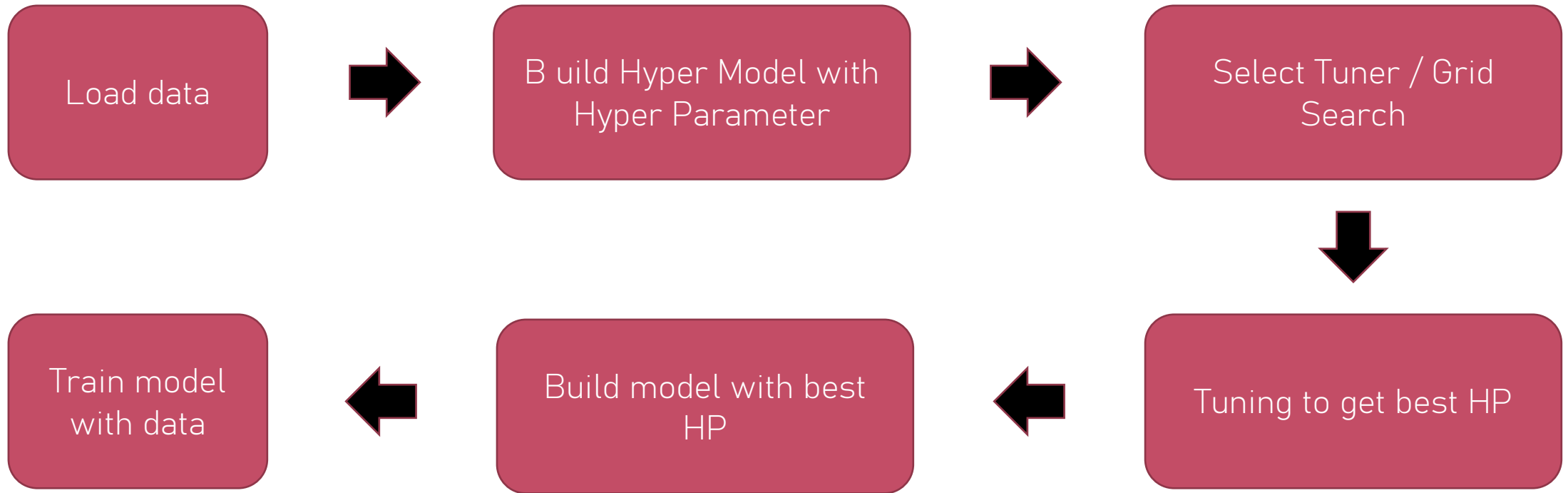
## Tuners

- The base Tuner class
- RandomSearch Tuner
- BayesianOptimization Tuner
- Hyperband Tuner
- Sklearn Tuner

# KERAS TUNER

[https://keras.io/api/keras\\_tuner/](https://keras.io/api/keras_tuner/)

# WORKFLOW





HANDS ON