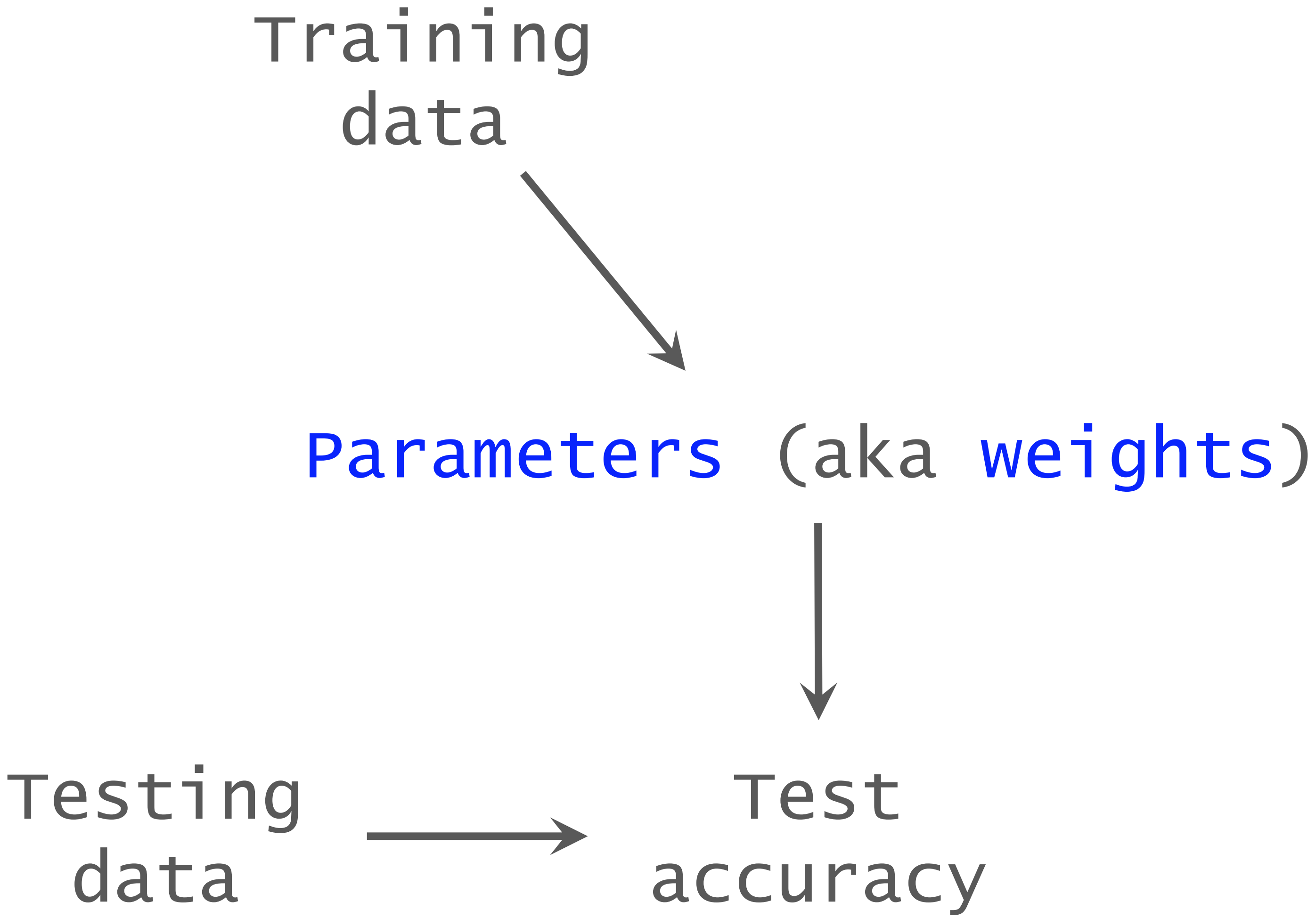
Neural Architecture Search Basics

Shusen Wang

http://wangshusen.github.io/

- Parameters （参数）
- Hyper-parameters （超参数）

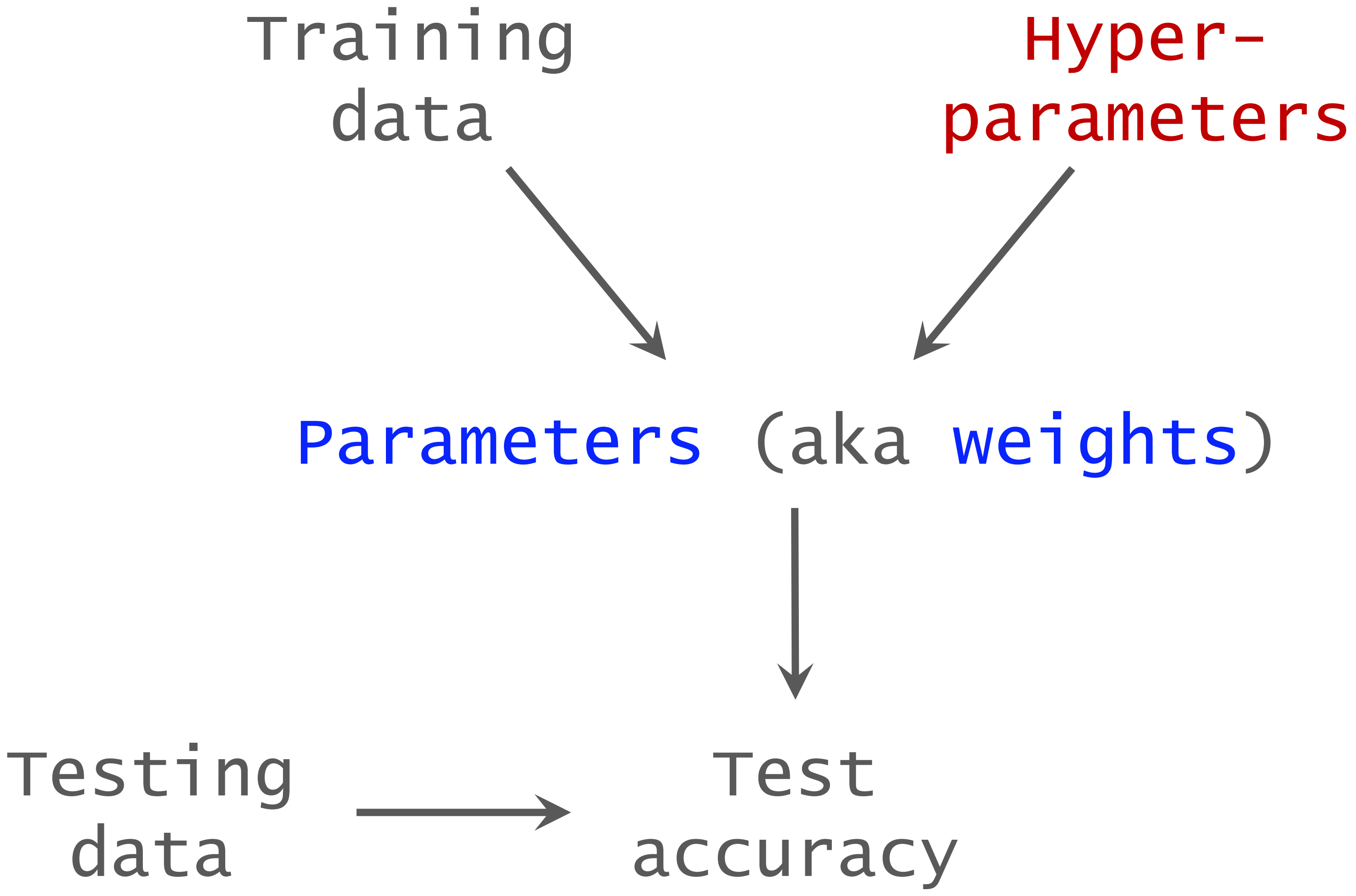


Training data

Parameters (aka weights)

Testing data

Test accuracy



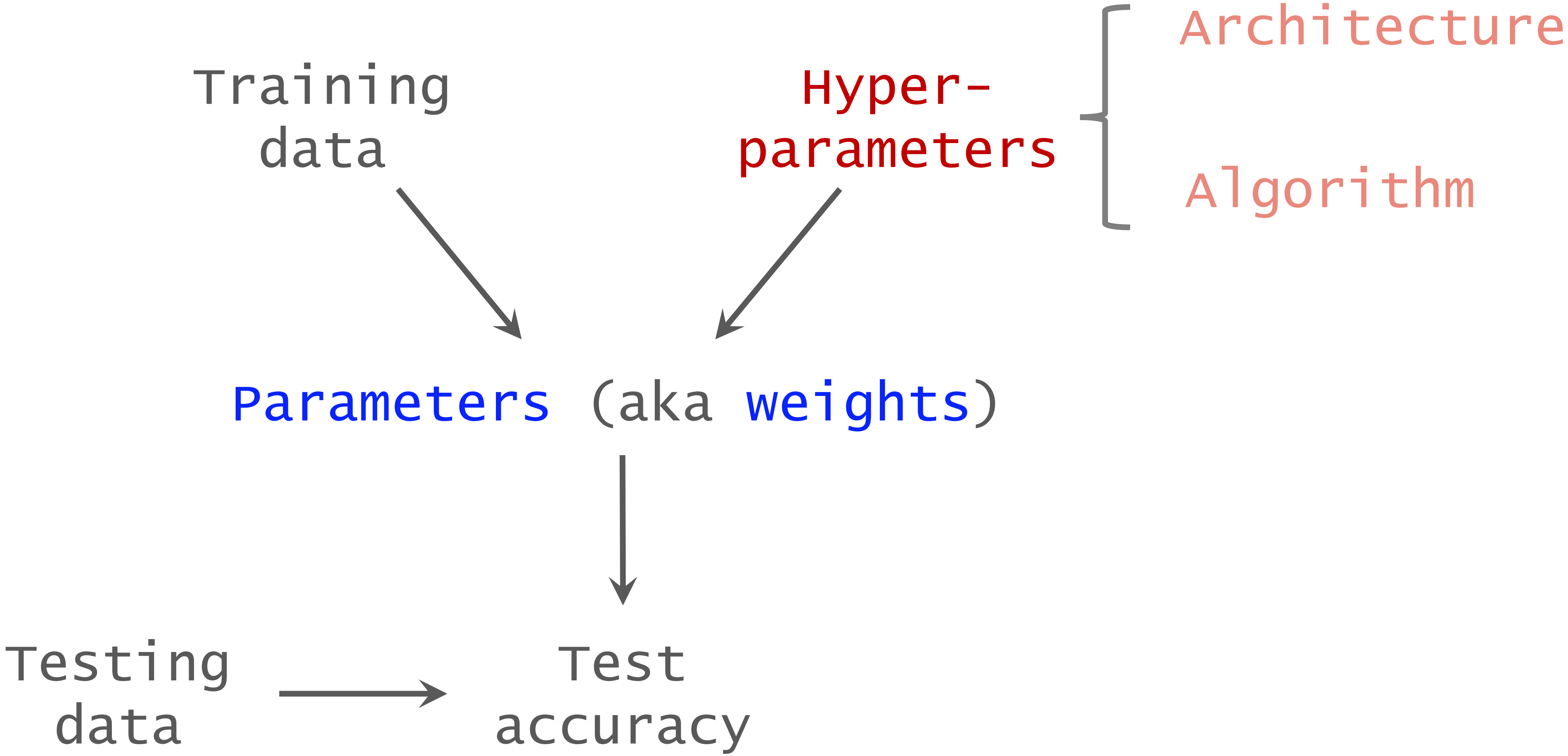
Training data

Hyper-parameters

Parameters (aka weights)

Testing data

Test accuracy



Training data

Hyper-parameters

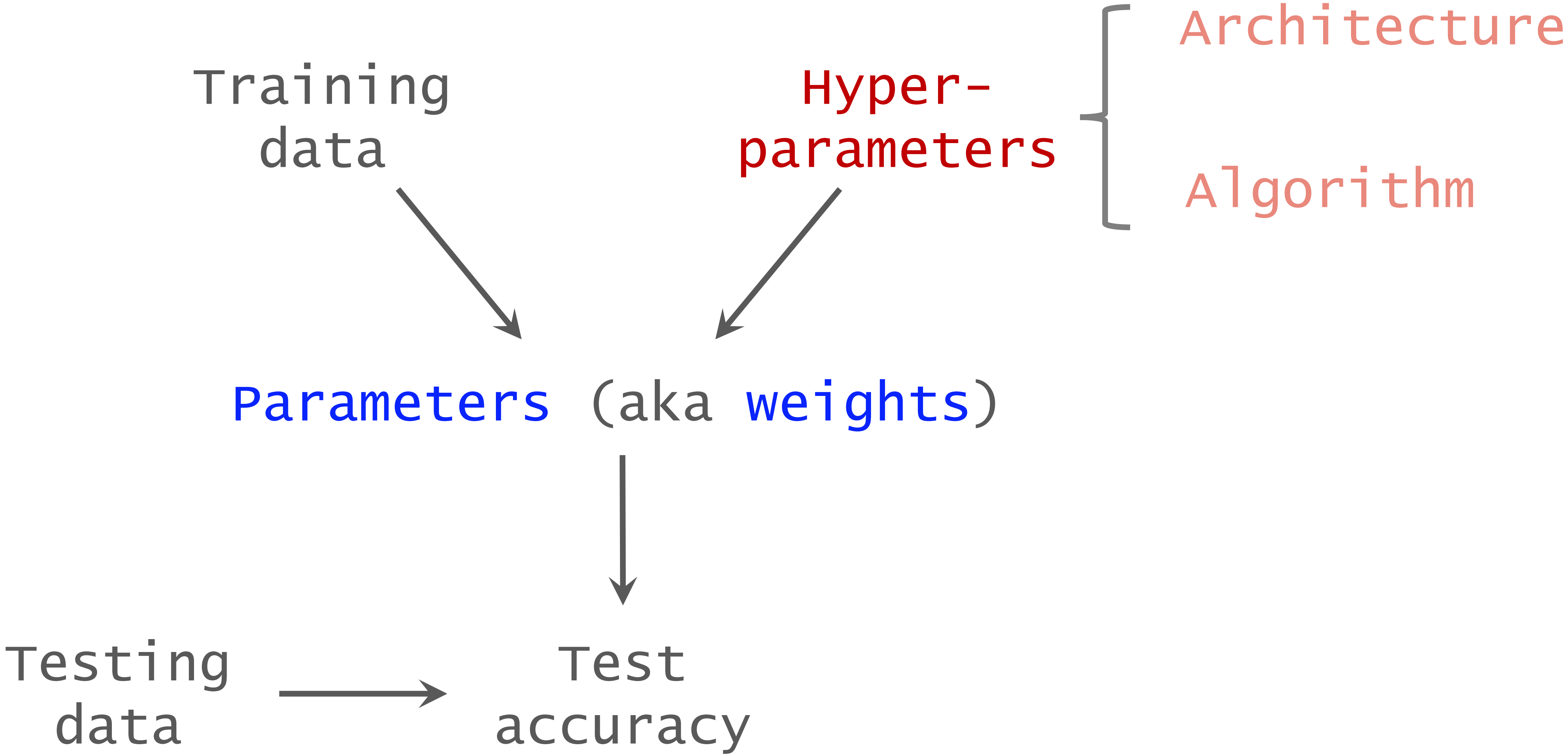
Architecture

Algorithm

Parameters (aka weights)

Testing data

Test accuracy



Training data

Hyper-parameters

Architecture

Algorithm

Parameters (aka weights)

Testing data

Test accuracy

CNN Architectures

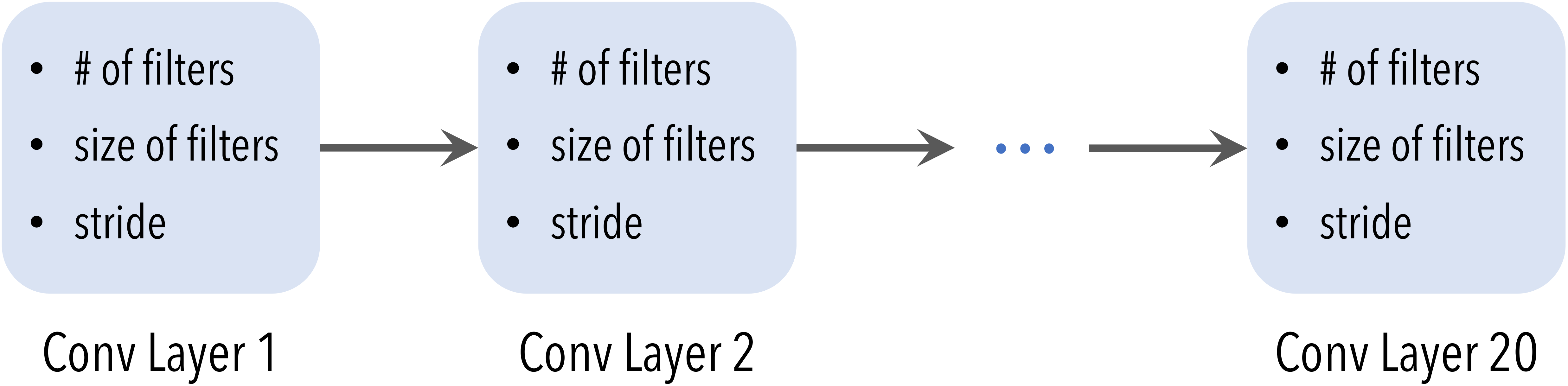
- Architectural hyper-parameters of a CNN include
  - numbers of conv and dense layers,
  - number of filters, size of filters, and stride in each conv layer,
  - width of each dense layer.

CNN Architectures

- Architectural hyper-parameters of a CNN include
  - numbers of conv and dense layers,
  - number of filters, size of filters, and stride in each conv layer,
  - width of each dense layer.

- Popular CNN architectures are manually designed.
  - E.g., ResNet, MobileNet, etc.
  - Manually tuning the architectural hyper-parameters.

CNN Architectures



- # of filters
- size of filters
- stride

Conv Layer 1

- # of filters
- size of filters
- stride

Conv Layer 2

. . .

- # of filters
- size of filters
- stride

Conv Layer 20

Neural Architecture Search (NAS)

Definition: Neural Architecture Search (NAS).

Find the architecture that leads to the best validation accuracy (or other metrics such as efficiency.)

- **Example:** ResNet has better accuracy than VGG.
- **Example:** MobileNet is more efficient than ResNet, although MobileNet has lower accuracy.

Search Space

Hyper-parameter Types	Candidates
# of filters	{24, 36, 48, 64}
size of filters	{3×3, 5×5, 7×7}
stride	{1, 2}

# of filters ∈ { 10, 11, 12, 13, ..., 98, 99, 100 }

Search Space

Hyper-parameter Types	Candidates
# of filters	{24, 36, 48, 64}
size of filters	{3×3, 5×5, 7×7}
stride	{1, 2}

Search Space

Hyper-parameter Types	Candidates
# of filters	$\{24, \quad 36, \quad 48, \quad 64\}$
size of filters	$\{3\times3, \quad 5\times5, \quad 7\times7\}$
stride	$\{1, \quad 2\}$

Search space: The set containing all the possible architectures.

- We want to build a CNN with 20 Conv layers.
- Search space:

$$\{24, 36, 48, 64\}^{20} \times \{3\times3, 5\times5, 7\times7\}^{20} \times \{1, 2\}^{20}.$$

Search Space

Hyper-parameter Types	Candidates
# of filters	{24, 36, 48, 64}
size of filters	{3×3, 5×5, 7×7}
stride	{1, 2}

Search space: The set containing all the possible architectures.

- We want to build a CNN with 20 Conv layers.
- Search space:

$$\{24, 36, 48, 64\}^{20} \times \{3\times 3, 5\times 5, 7\times 7\}^{20} \times \{1, 2\}^{20}.$$

- Size of search space (i.e., number of possible architectures):

$$(4\times 3\times 2)^{20}$$

Search Space

Hyper-parameter Types	Candidates
# of filters	$\{24, \quad 36, \quad 48, \quad 64\}$
size of filters	$\{3\times3, \quad 5\times5, \quad 7\times7\}$
stride	$\{1, \quad 2\}$

Search space: The set containing all the possible architectures.

- We want to build a CNN with 20 Conv layers.
- Search space:

$$\{24,36,48,64\}^{20} \times \{3\times3,5\times5,7\times7\}^{20} \times \{1,2\}^{20}.$$

- Size of search space (i.e., number of possible architectures):

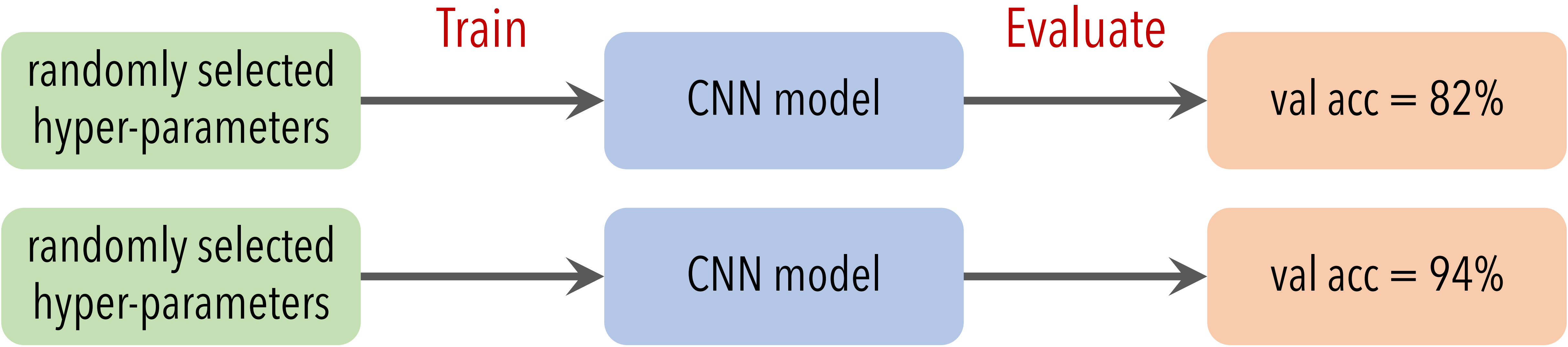
$$(4\times3\times2)^{20} = \boxed{4\times10^{27}.}$$

Outcome of NAS?

- For example, this is an outcome of NAS:

	Layer 1	Layer 2	⋯	Layer 20
# of filters	24	48	⋯	64
Size of filters	5×5	3×3	⋯	3×3
Stride	1	1	⋯	2

Baseline: Random Search



randomly selected hyper-parameters

Train

CNN model

Evaluate

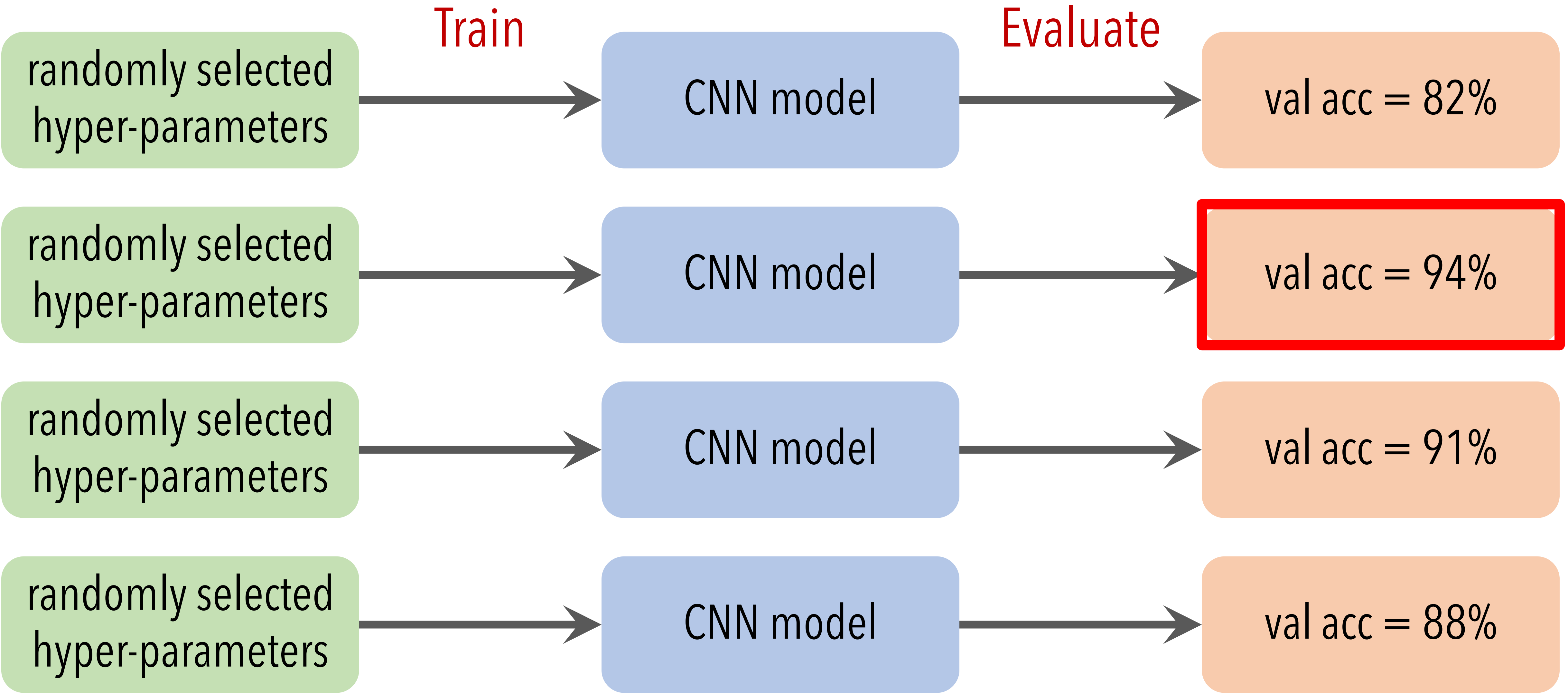
val acc = 82%

randomly selected hyper-parameters

CNN model

val acc = 94%

Baseline: Random Search



Train

Evaluate

randomly selected hyper-parameters → CNN model → val acc = 82%

randomly selected hyper-parameters → CNN model → val acc = 94%

randomly selected hyper-parameters → CNN model → val acc = 91%

randomly selected hyper-parameters → CNN model → val acc = 88%

Challenges in NAS

Challenge 1: Each trial is expensive.

- Training a CNN from scratch takes hours or days, if a single GPU is used.

Challenge 2: The search space is too big.

- Number of possible architectures:

$$(4\times3\times2)^{20} = 4\times10^{27}.$$

Thank You!

http://wangshusen.github.io/