EMOCON 2017 S/S

Machine learning debugging with TensorFlow examples

@mingkim

- **Mingkim(김명주)**
  - 멋쟁이 사자처럼 2기 학생
  - 주니어 개발자
  - 머신러닝에 관심이 많은 1인
  - 공유하고 싶은 1인

이 발표는

- 대상 : 머신러닝/텐서플로에 대한 기본 지식이 있는 분들
- 무엇을 : 머신러닝 디버깅 접근법
- 제가 정리한 것을 공유

모델이 학습을 제대로 하지 못했을 때

보통 어떻게 하시나요?

학습도중 문제가 발생했을 때

- 데이터를 본다.
- 그래프를 그려본다.
- 구현 코드 로직을 살펴본다.
- …
- 어떻게 해야 할지 잘 모르겠다.

디버깅이 쉽지 않습니다.

디버깅시 고려해야 하는 것들

- Data
- Feature
- Algorithm
- Learning rate
- epoch
- batch size
- Hyperparameter
- …

멘붕의 구체적인 이유들

- 고려해야 하는 변수가 많음
- 모델은 각 데이터에 맞게 최적화됨
- 도출된 정확도에 대한 수학적 해석이 어려움
- 디버깅 결과를 확인하는데 시간이 오래 걸림

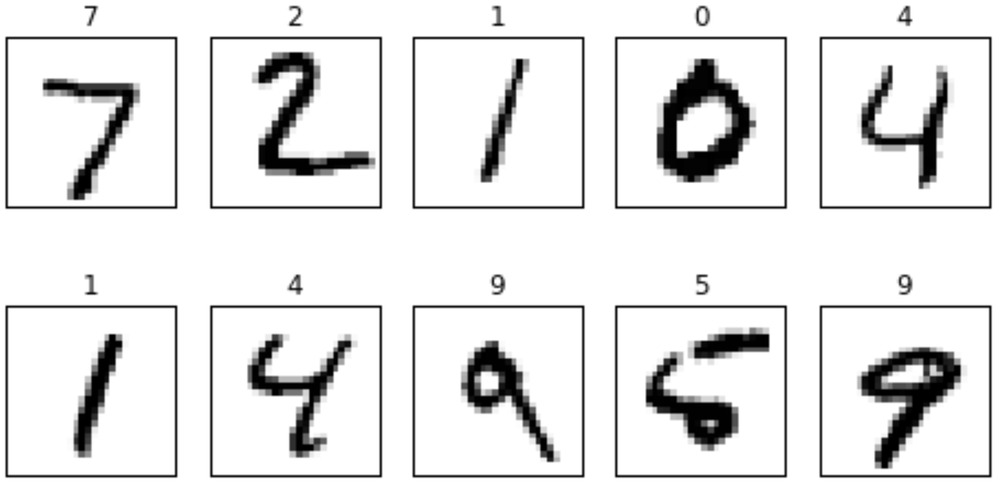
디버깅이 쉽지 않습니다.2

손글씨 예제

Image Classification Problem

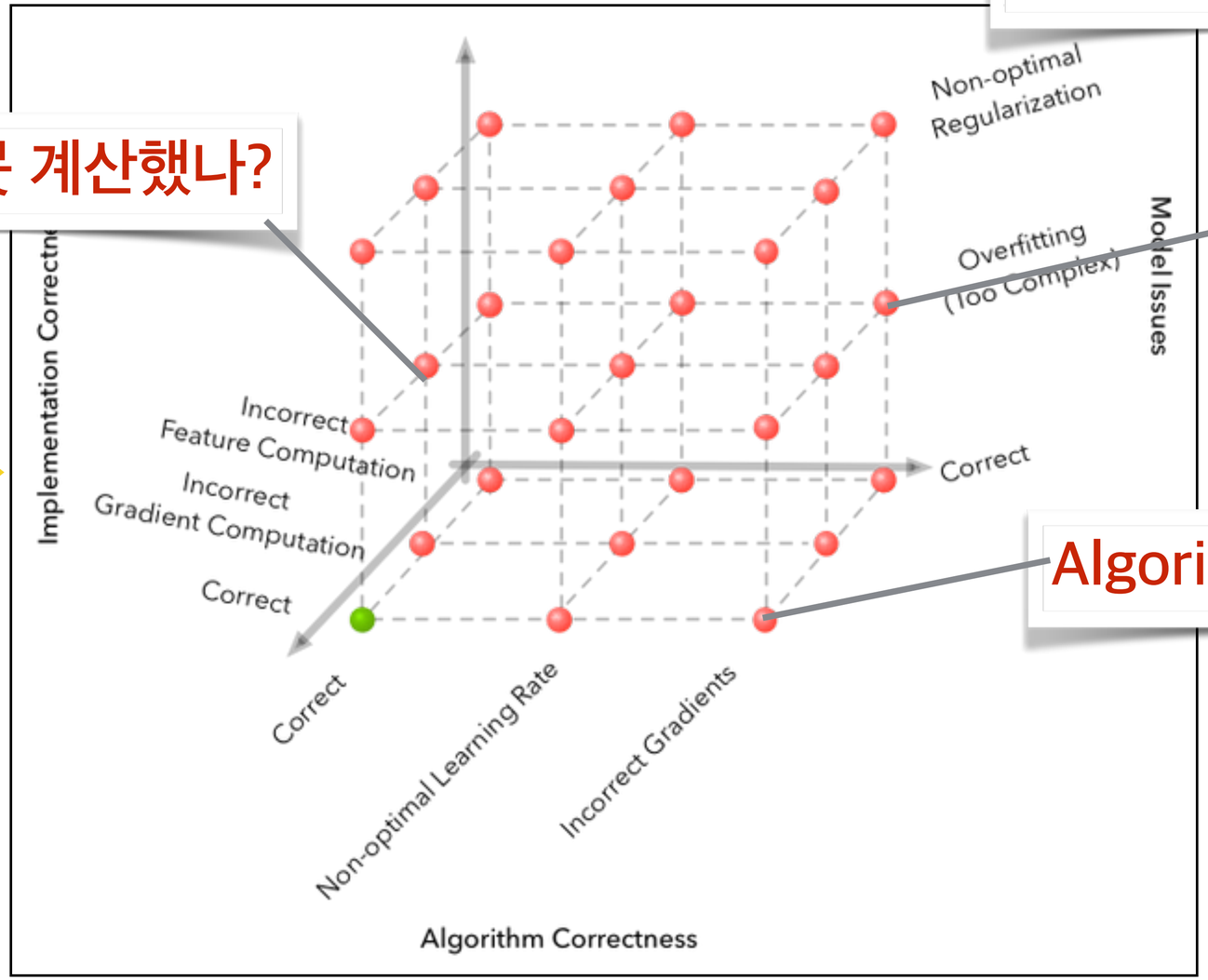
Train

Predict



손글씨 예제

Image Classification Problem Failed



Data가 잘못 되었나?

Loss를 잘못 계산했나?

Overfitting 인가?

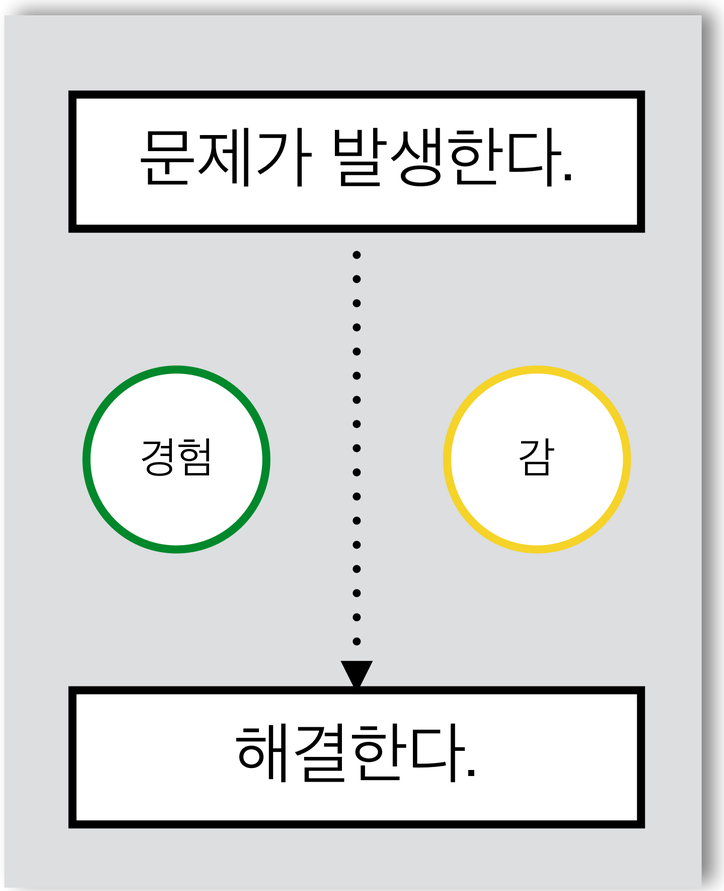
Algorithm을 잘못 구현했나?

http://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html

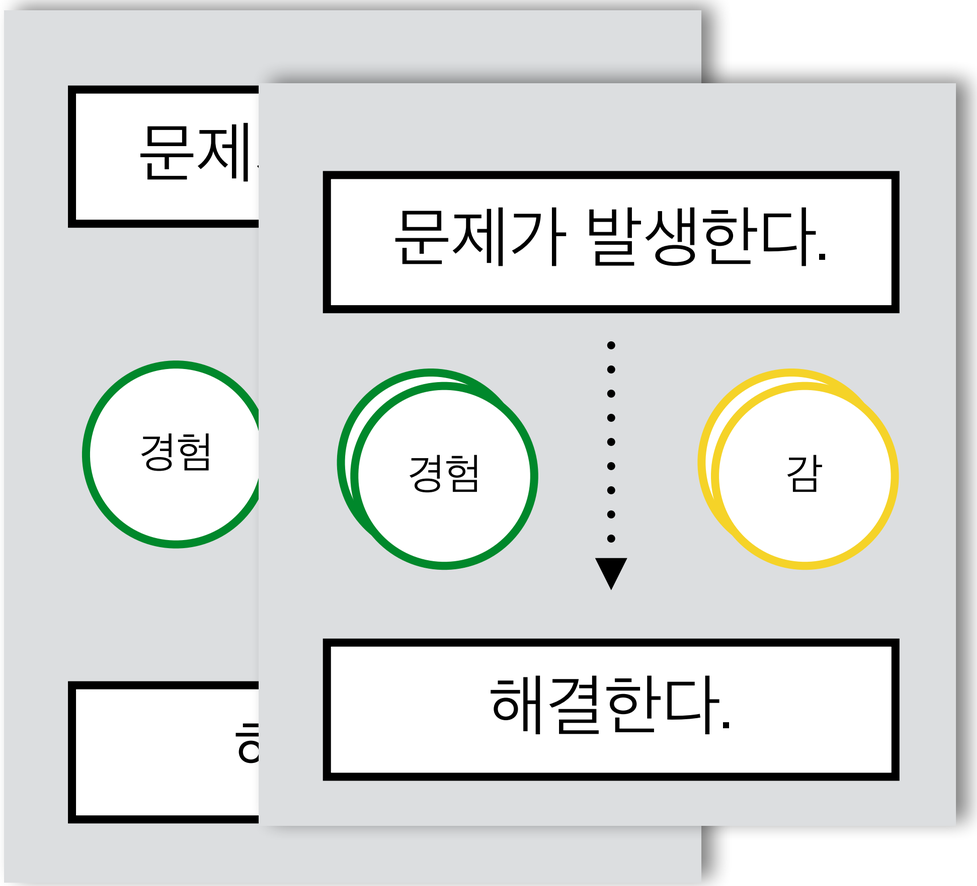
머신러닝 디버깅 - It’s like…

조금이라도 정리해 봅시다.

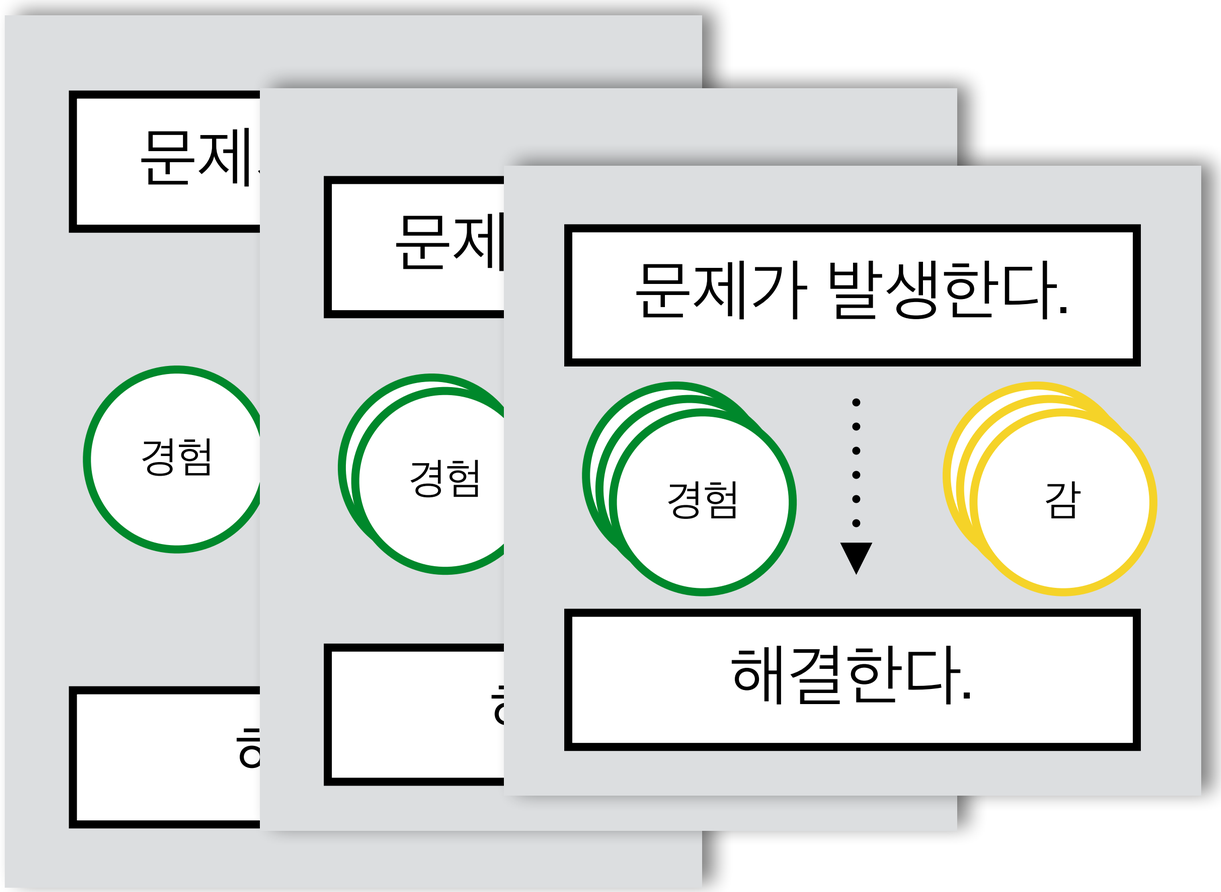
일반적인 문제 해결법



일반적인 문제 해결법



일반적인 문제 해결법



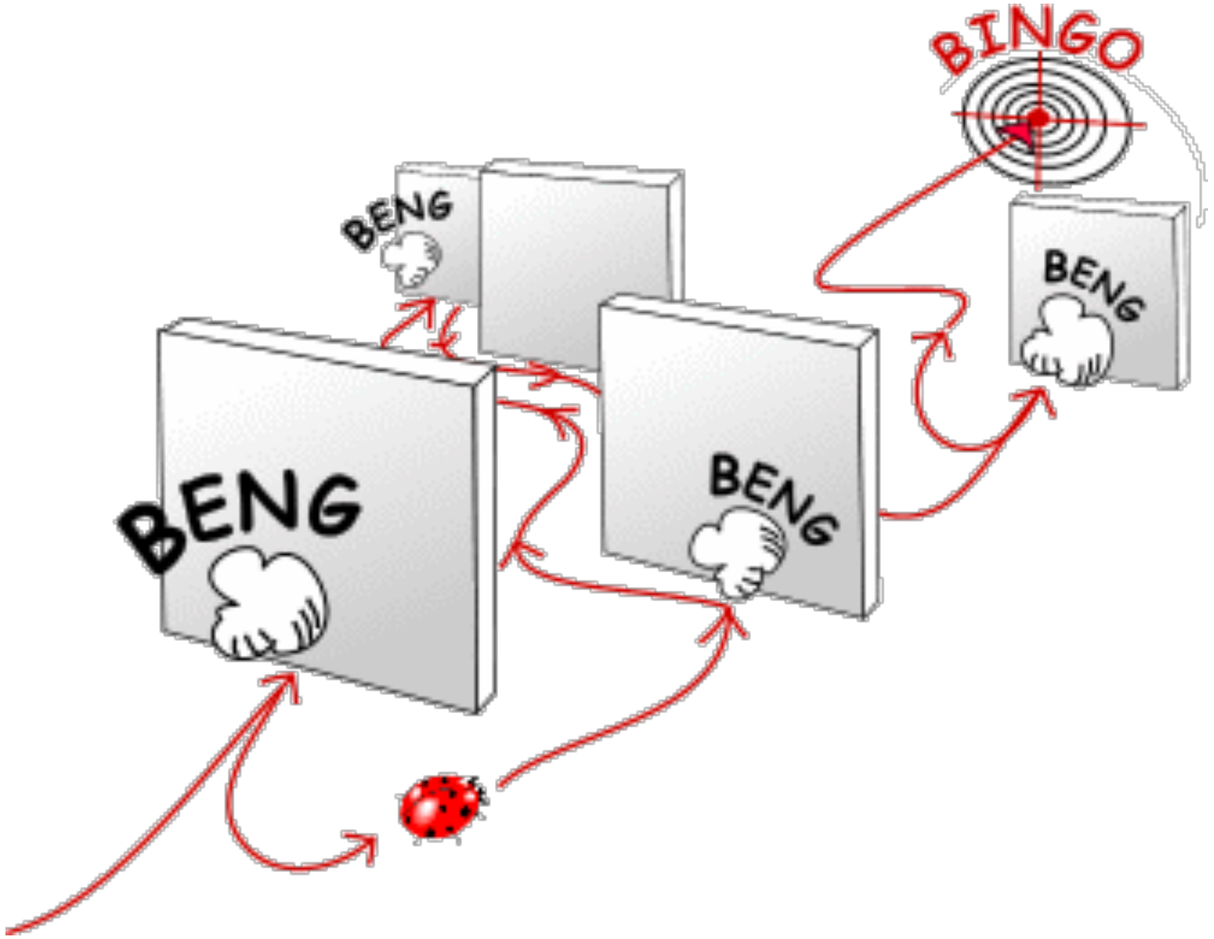
일반적인 문제 해결법

비슷한 유형의 문제 해결을 반복

경험이 쌓이고 감이 생김

문제 해결을 빨리 할 수 있게 됨

Heuristic Method



https://www.careermetis.com/wp-content/uploads/2015/10/heuristic_approach.gif

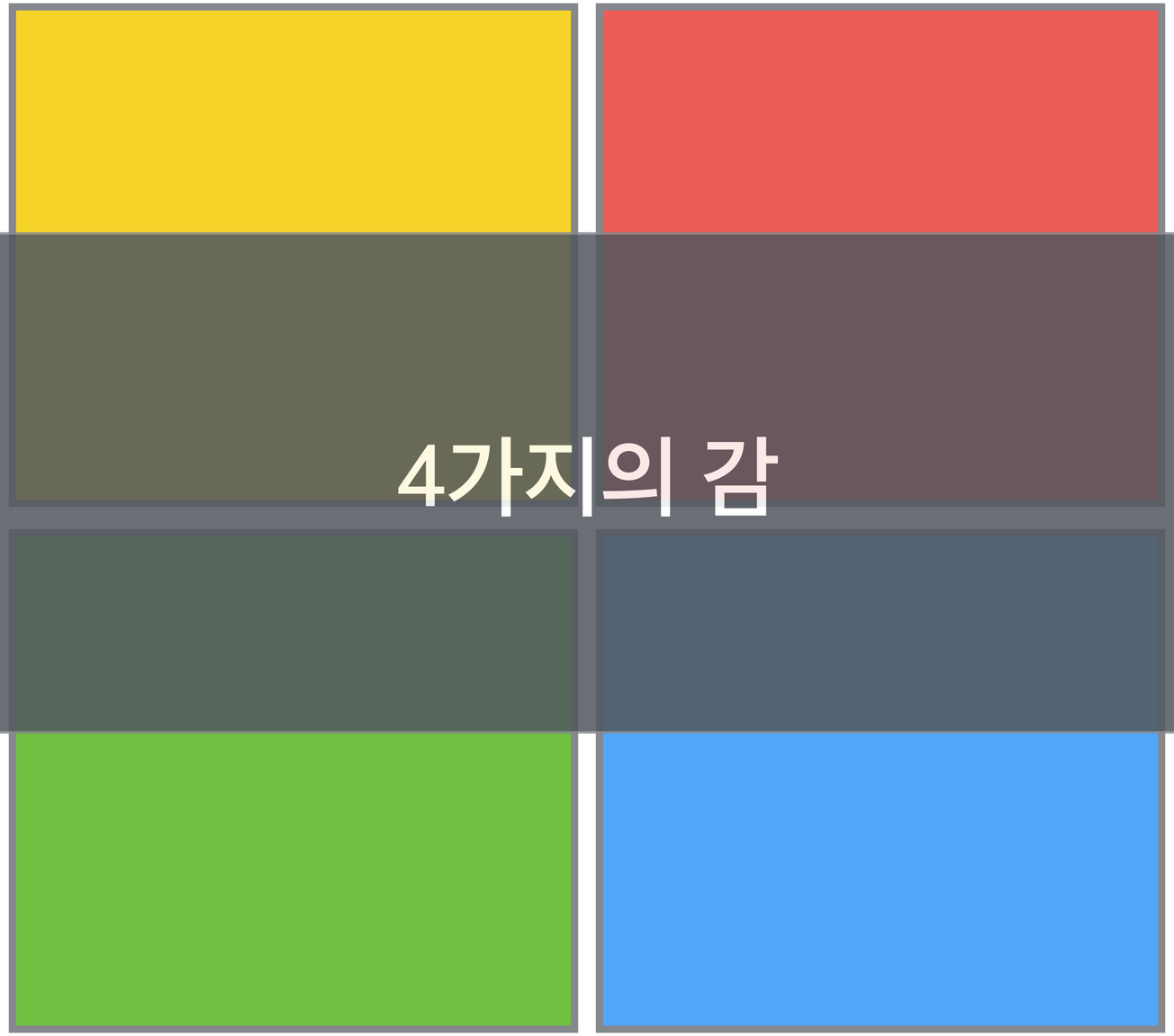
머신러닝 문제 해결법

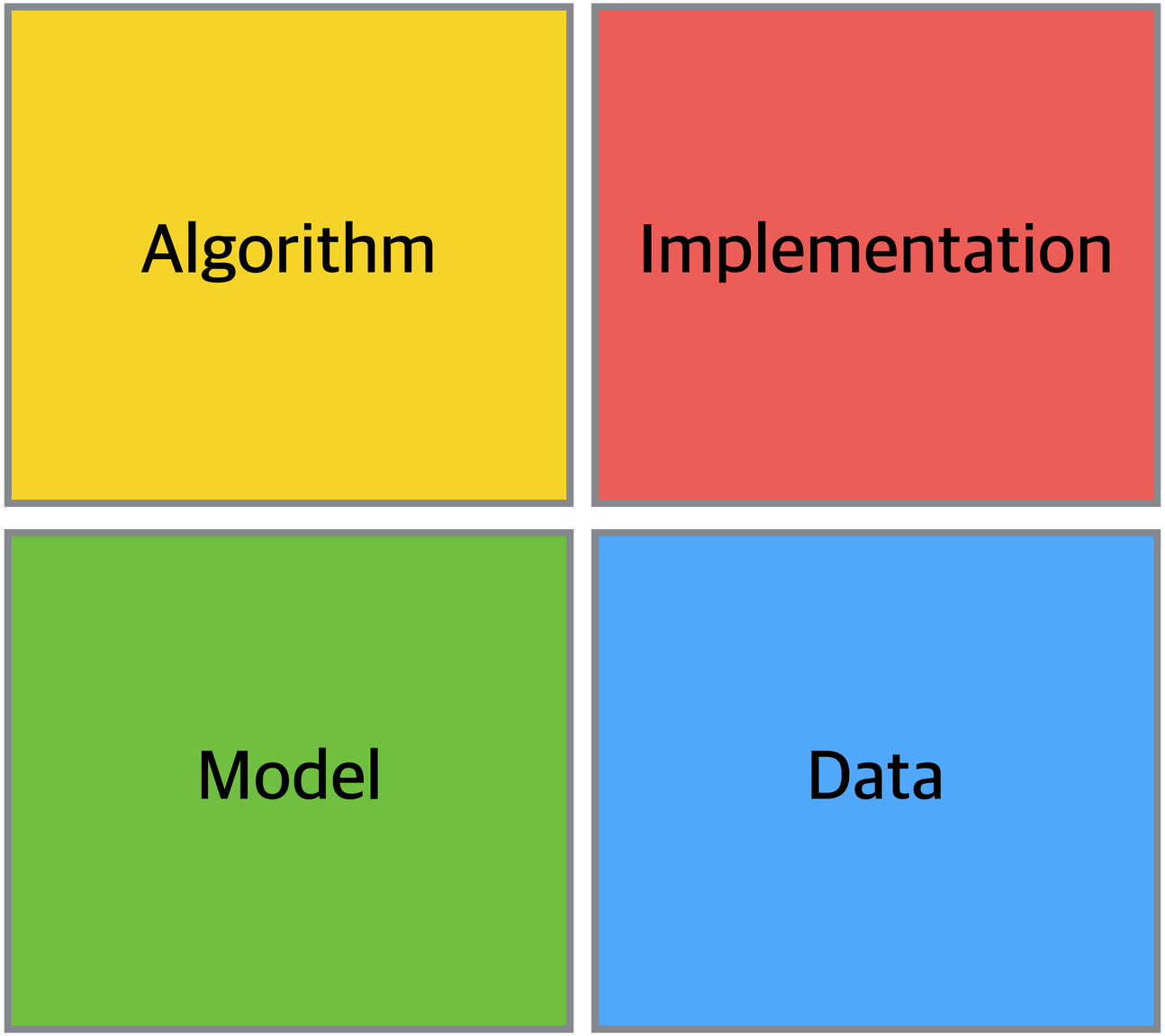
- 경험과 노하우가 많다.

  -> 해결

- 경험과 노하우가 부족하다.

  -> 하지만, 디버깅에 사용되는 기본적인 감을 활용해서 해결

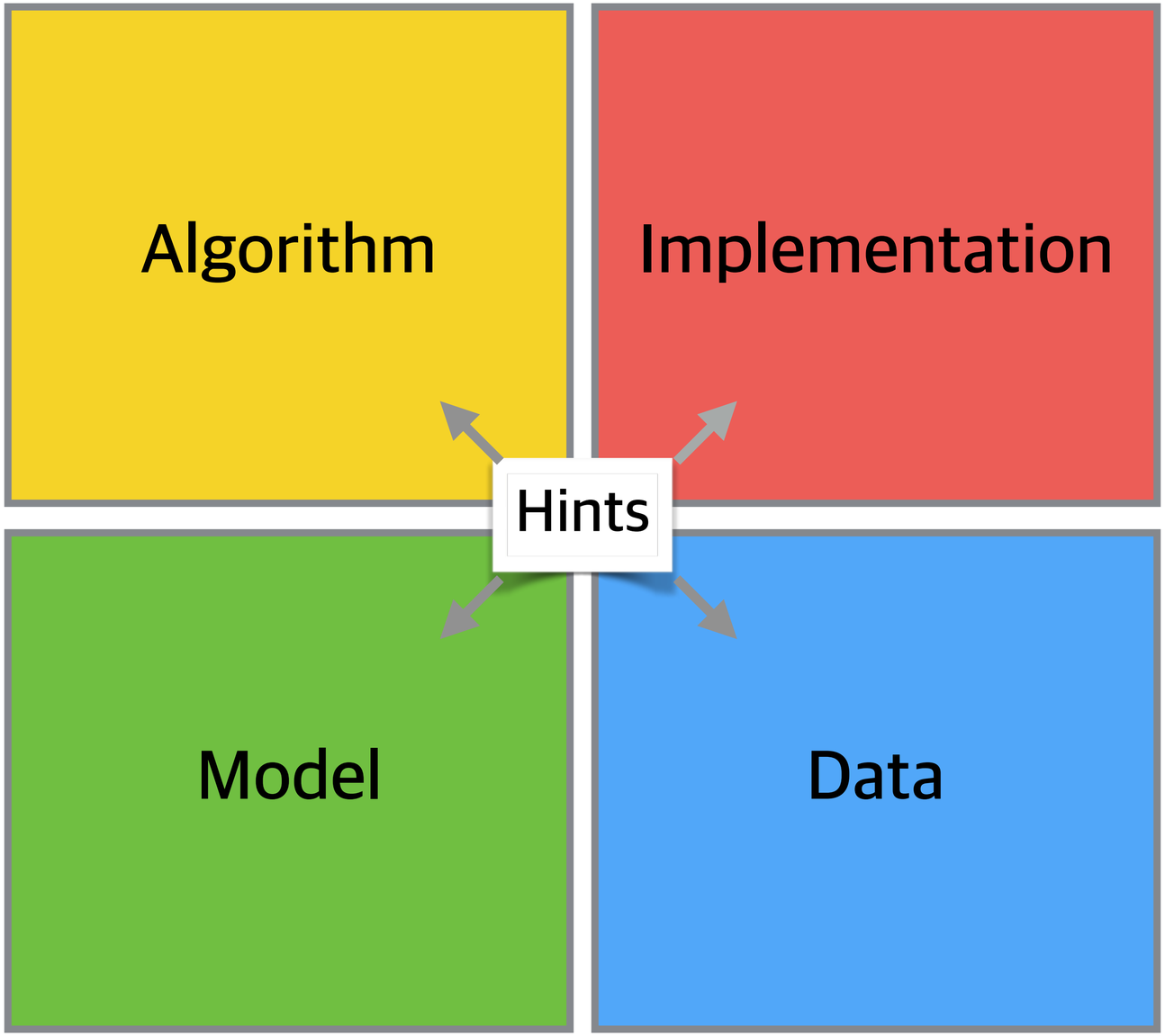

4가지의 감


Algorithm
Implementation
Model
Data

문제를 많이 해결하지 않아도

알 수 있으니 이득이네요!


Algorithm
Implementation
어떤걸 봐야 하나?
Model
Data


Algorithm
Implementation
Hints
Model
Data

힌트

- 문제에서 말해주는 비정상적인 특징
- 4가지 중 어떤것들을 봐야하는지 말해줌

힌트

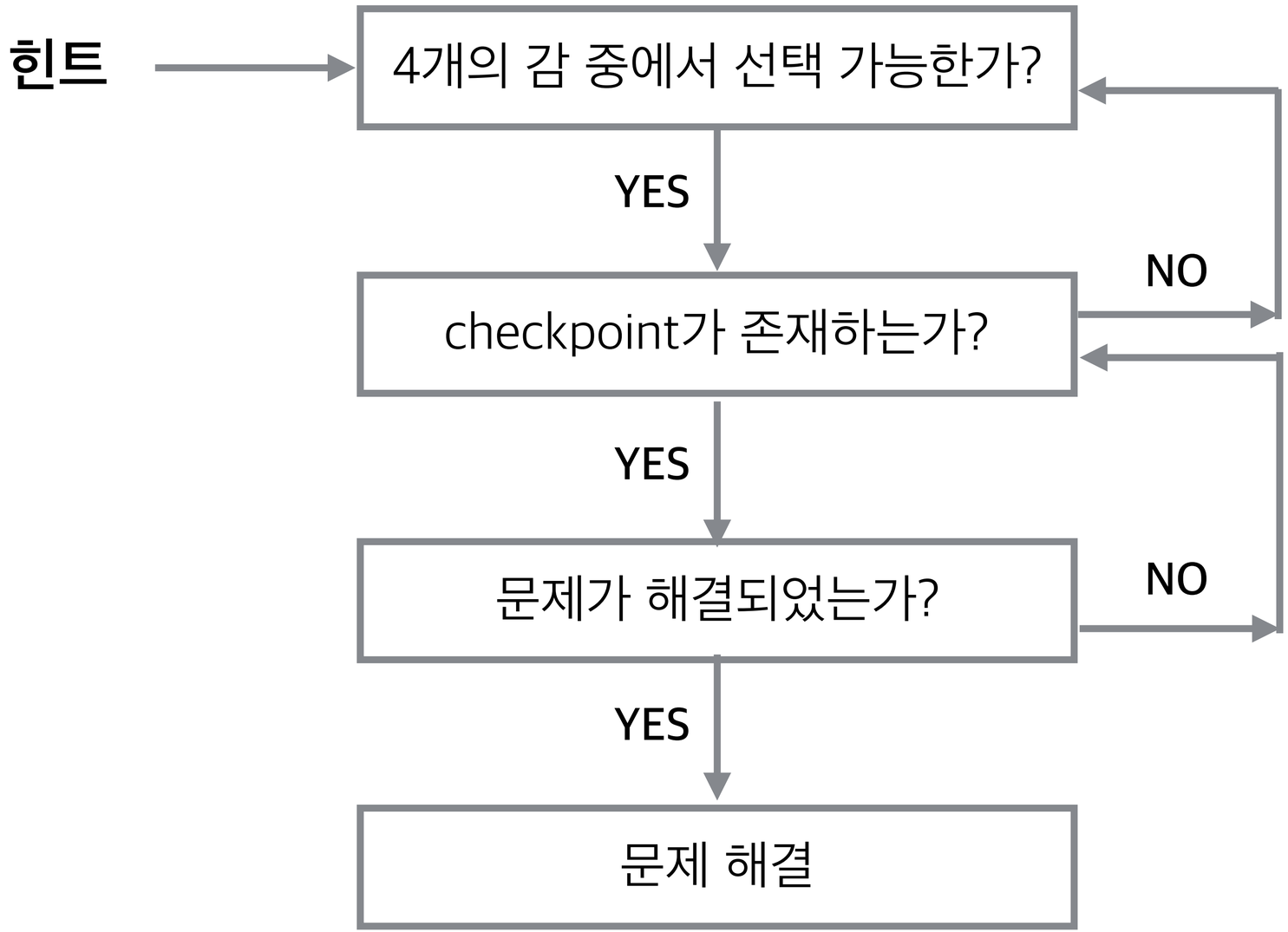
- Loss plot
- Accuracy plot
- 콘솔 에러 메시지
- 실제 output data
- 중간 변수 계산 통계치
- Human level error - Training set error
- Training set error - Validation error
- …

힌트

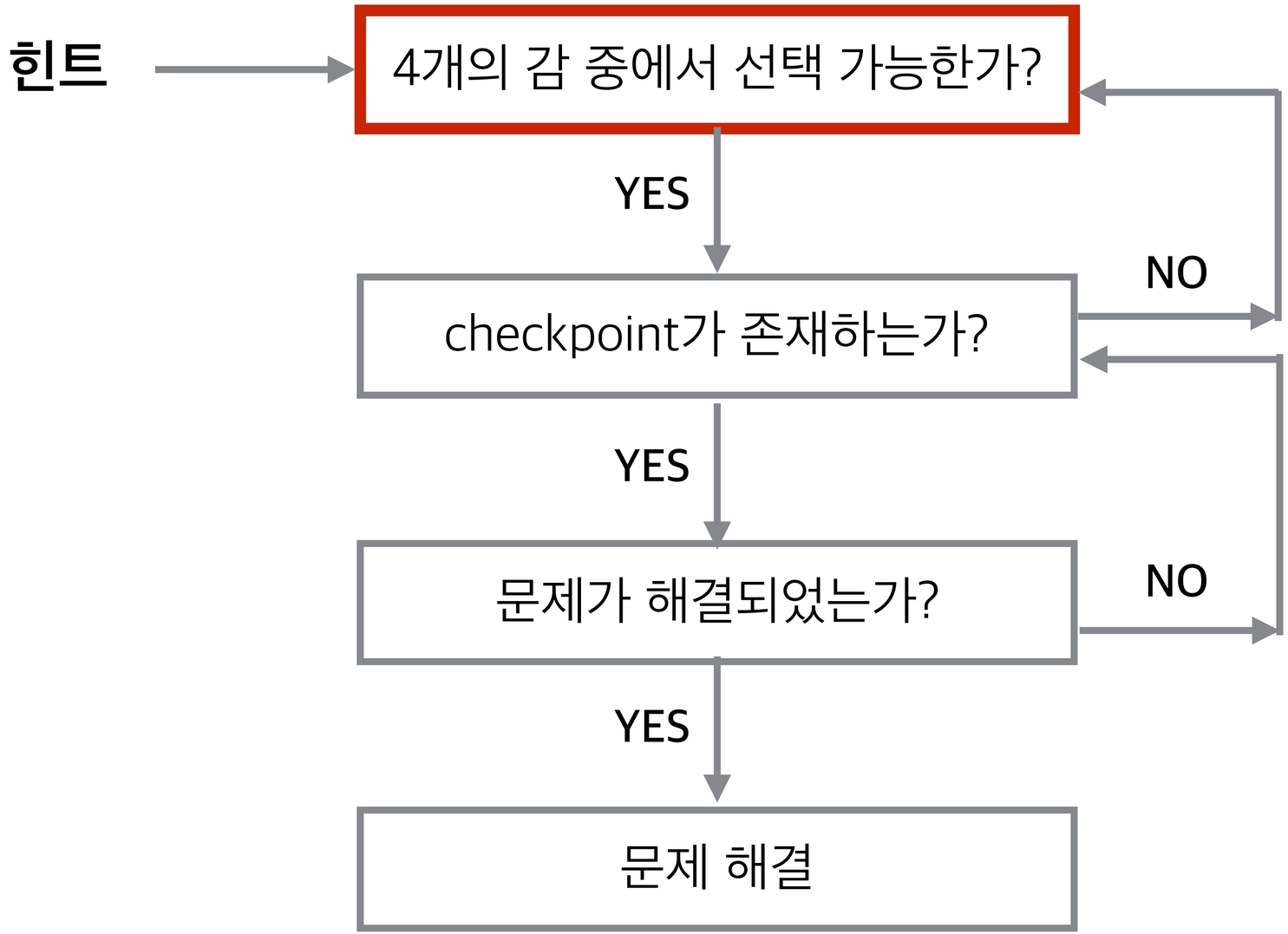
- Loss plot
- Accuracy plot
- 콘솔 에러 메시지
- 실제 output data
- 중간 변수 계산 통계치
- Human level error - Training set error
- Training set error - Validation error
- …

“ 해결 방법은 어떻게 찾나요? ”

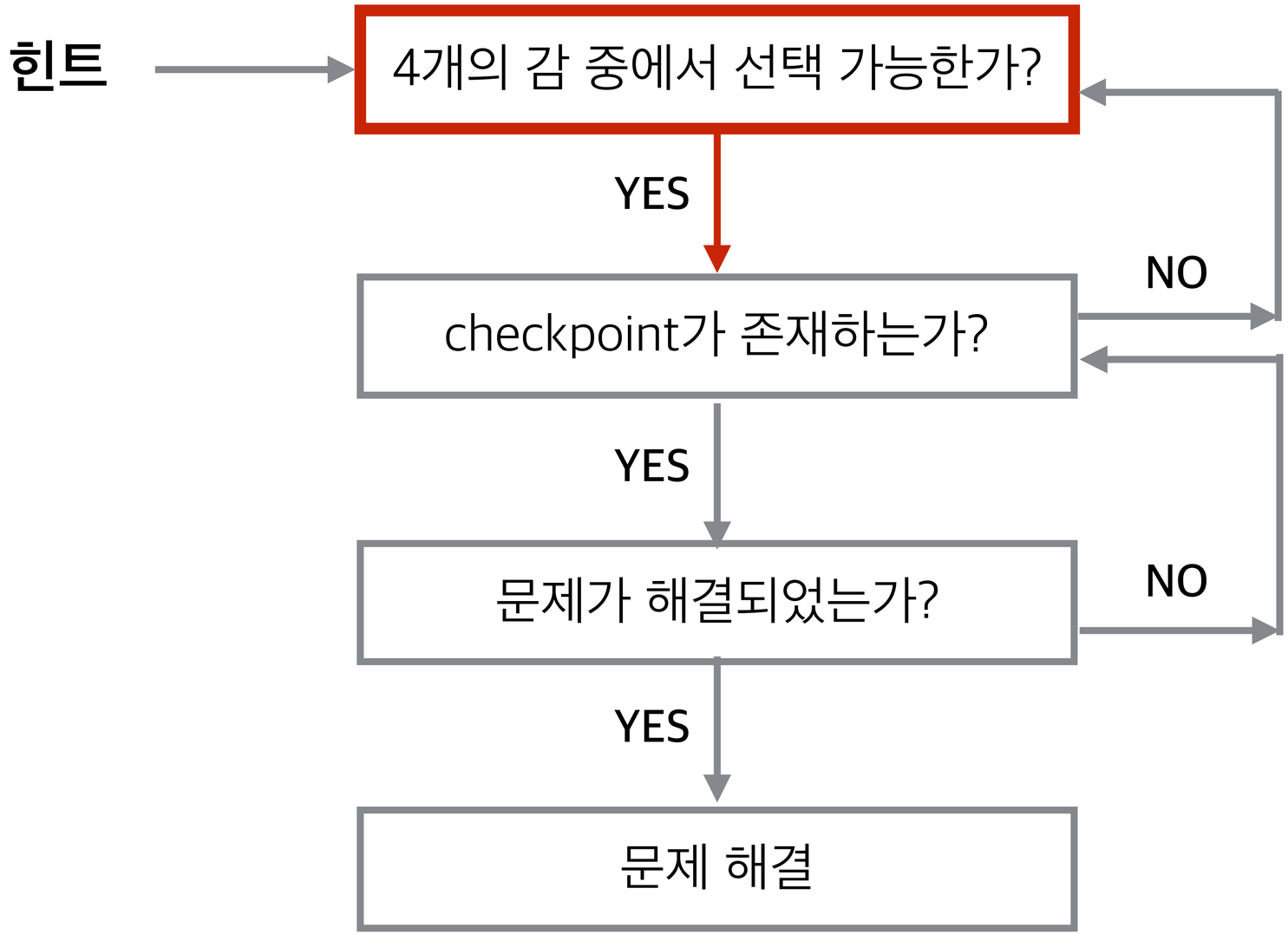
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

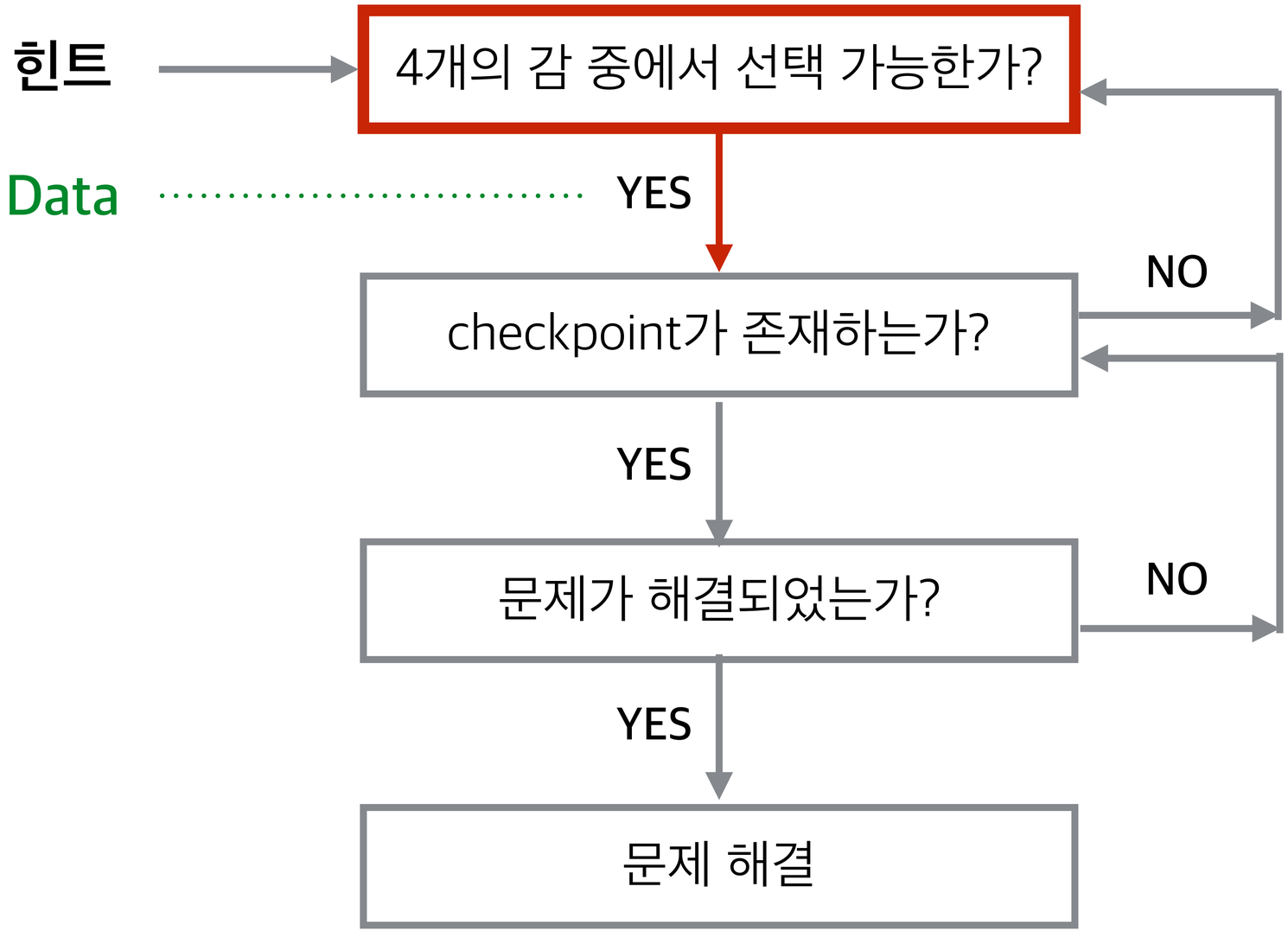
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

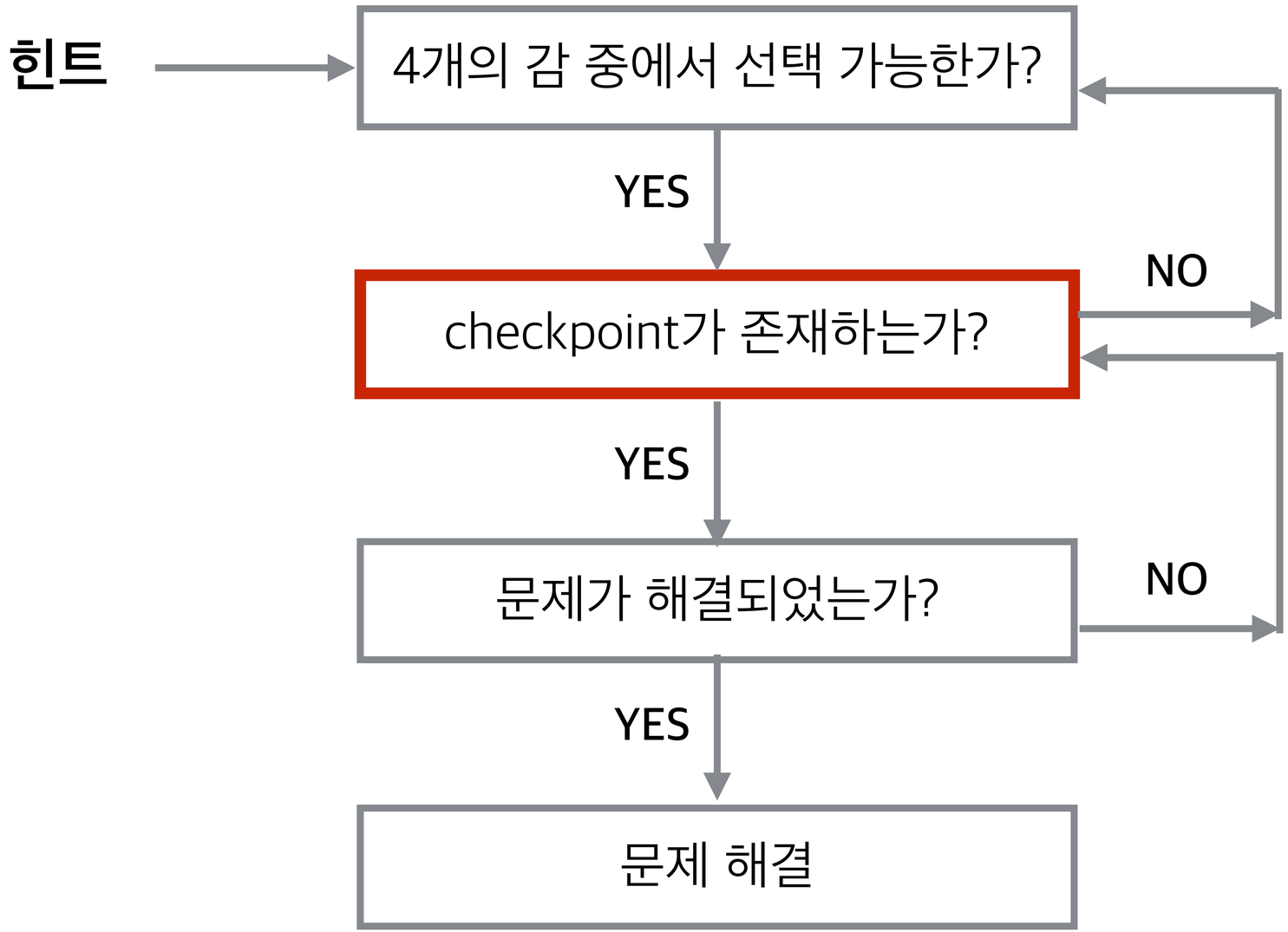
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

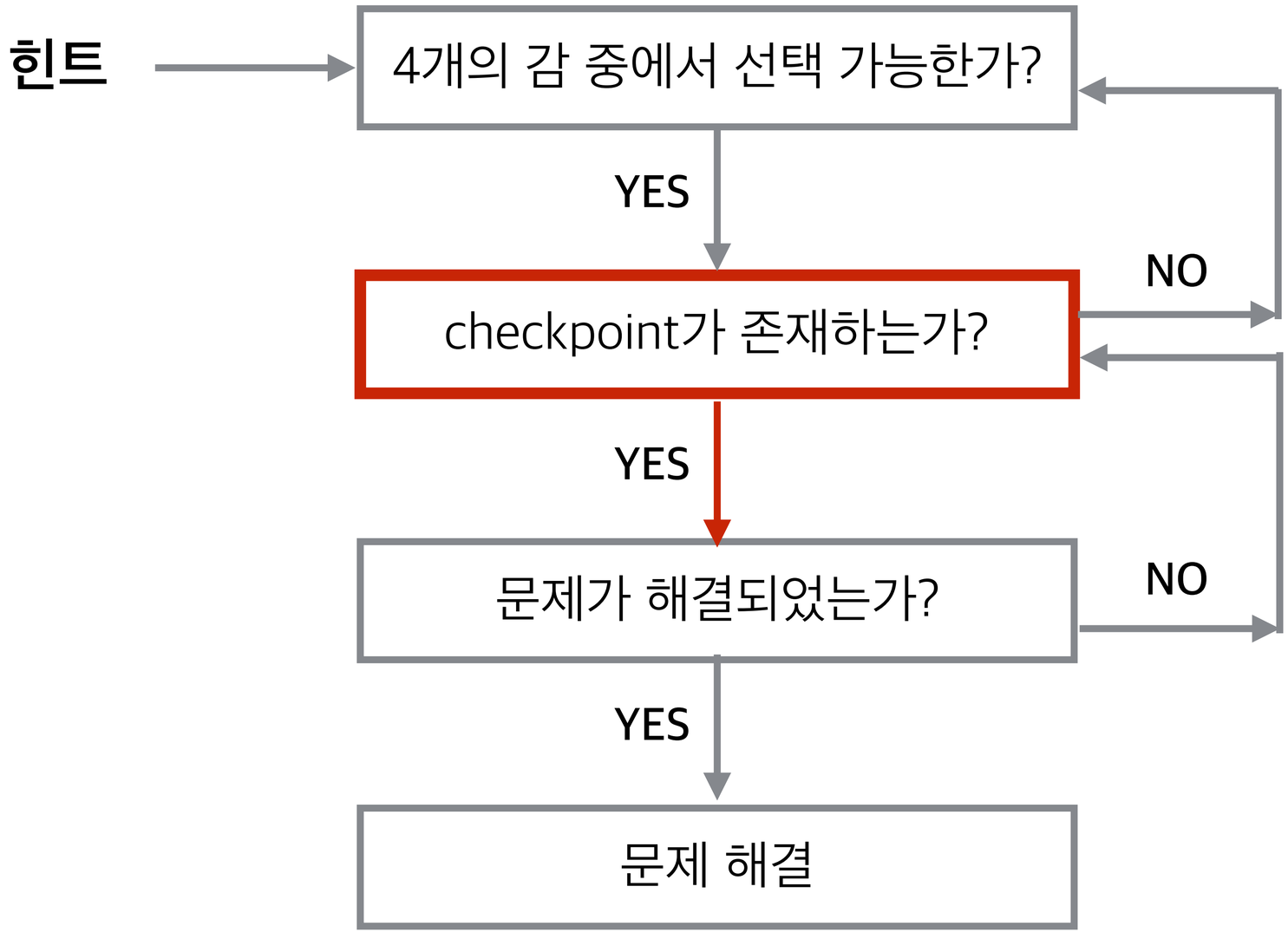
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
Data
YES
NO
checkpoint가 존재하는가?
YES
문제가 해결되었는가?
NO
YES
문제 해결

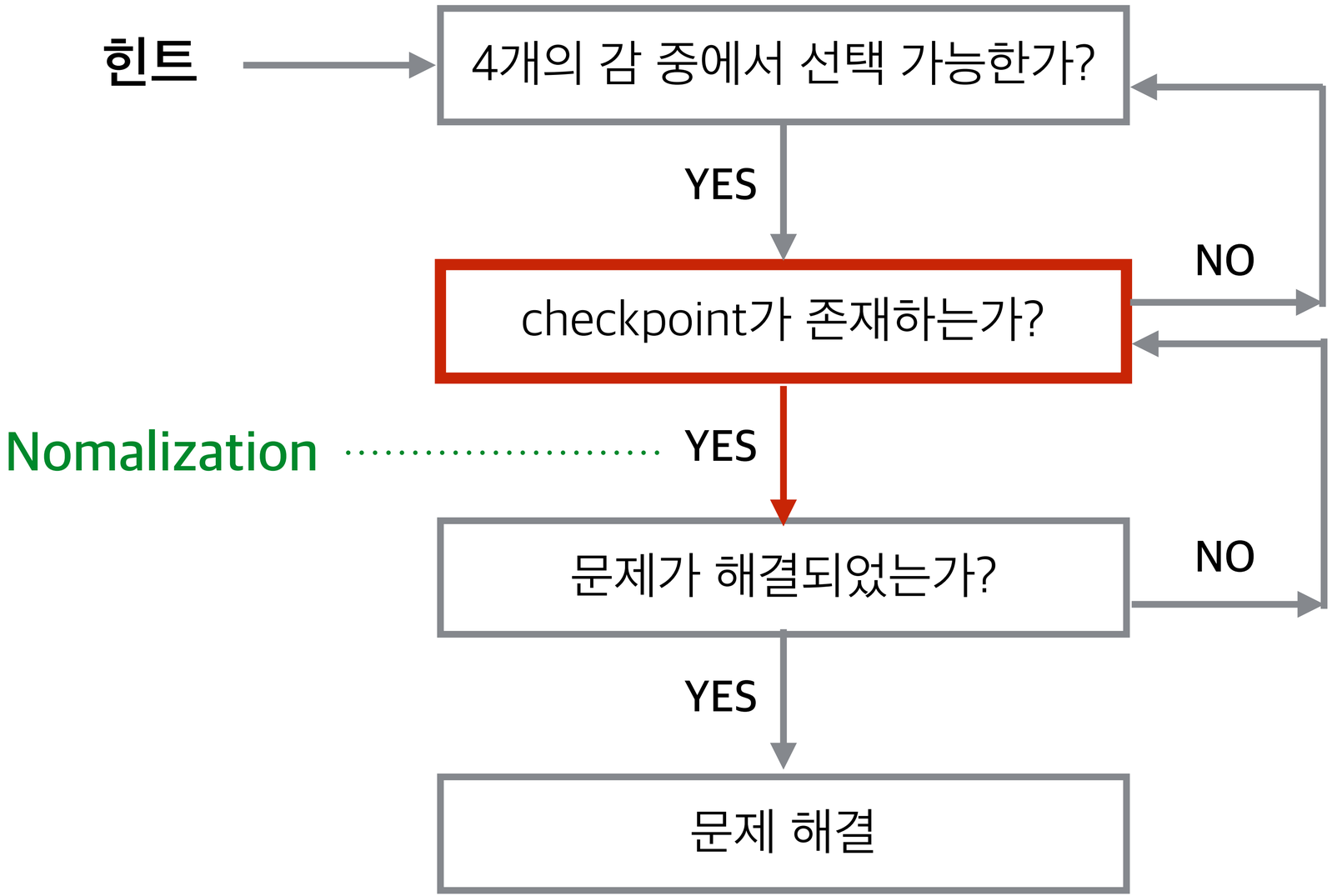
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
문제가 해결되었는가?
NO
YES
문제 해결

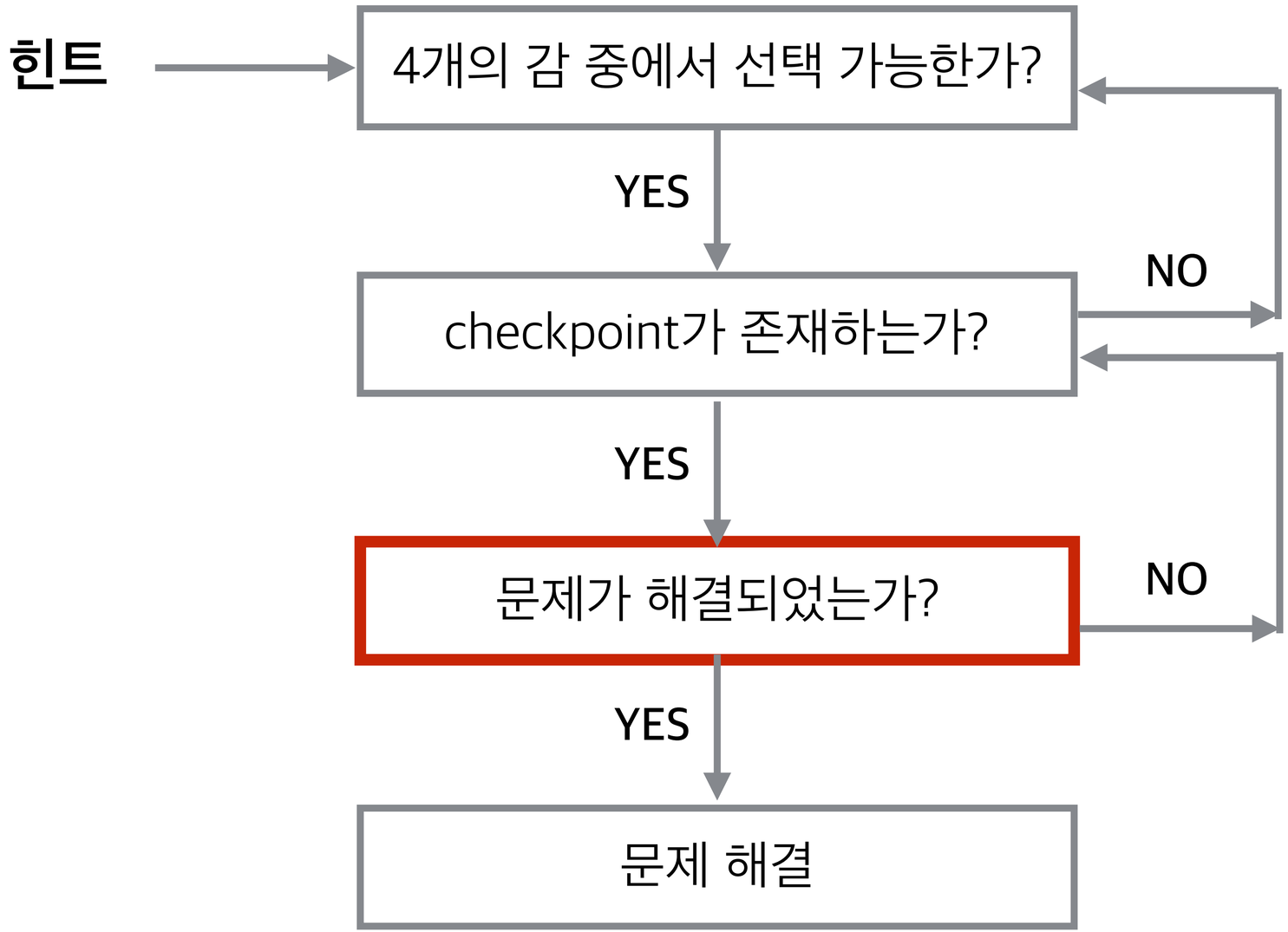
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
문제가 해결되었는가?
NO
YES
문제 해결

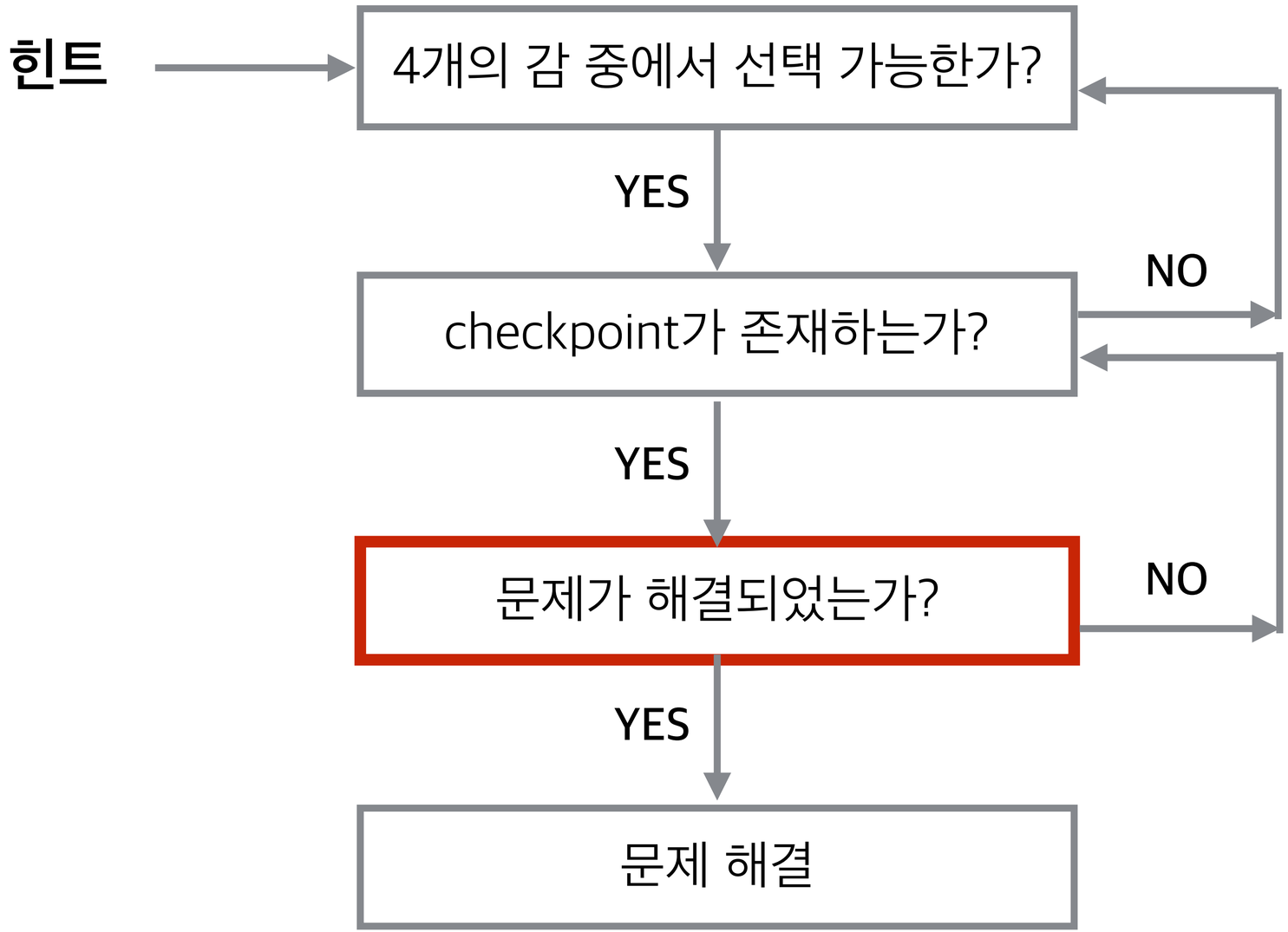
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
Nomalization
YES
문제가 해결되었는가?
NO
YES
문제 해결

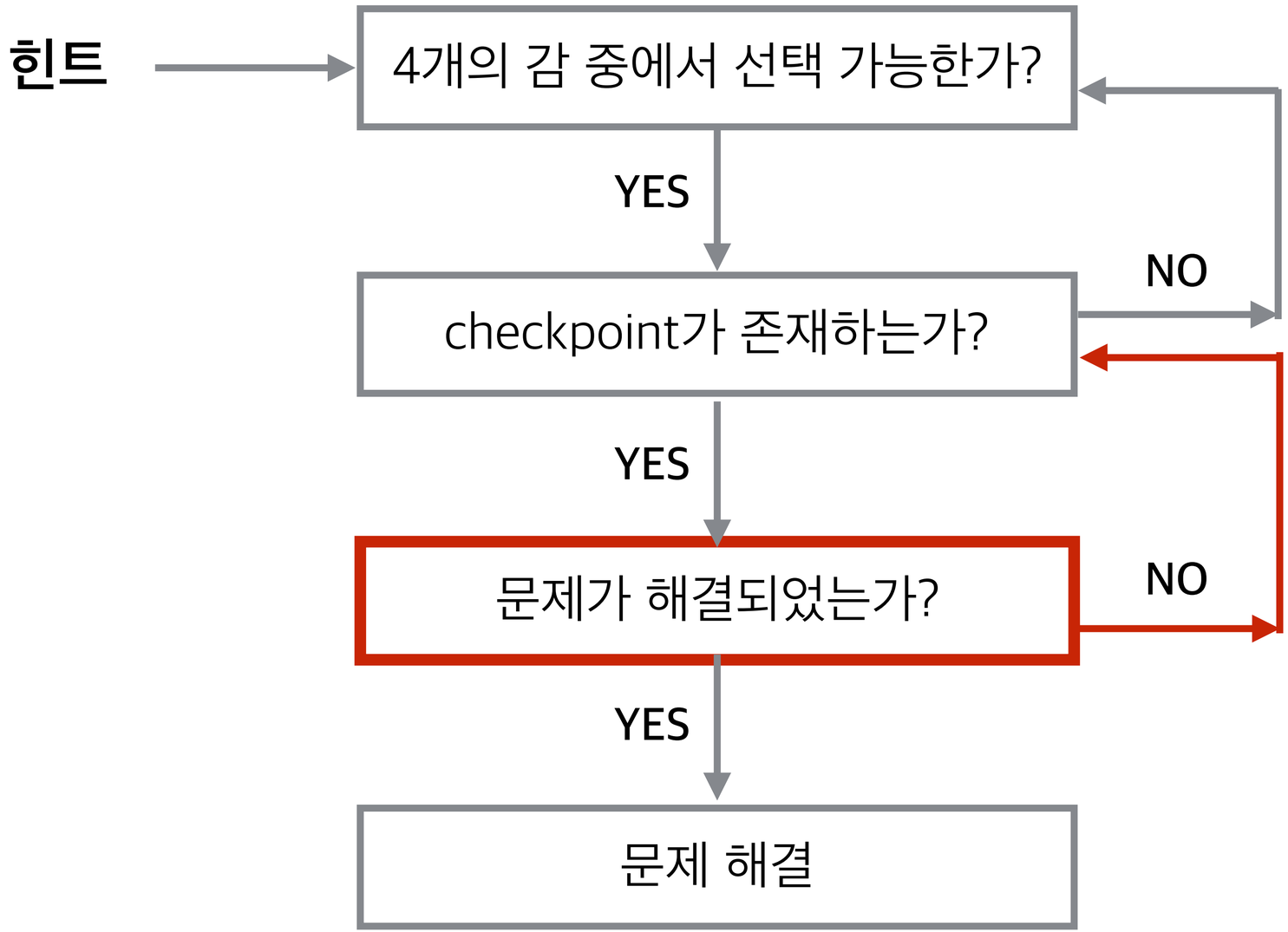
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

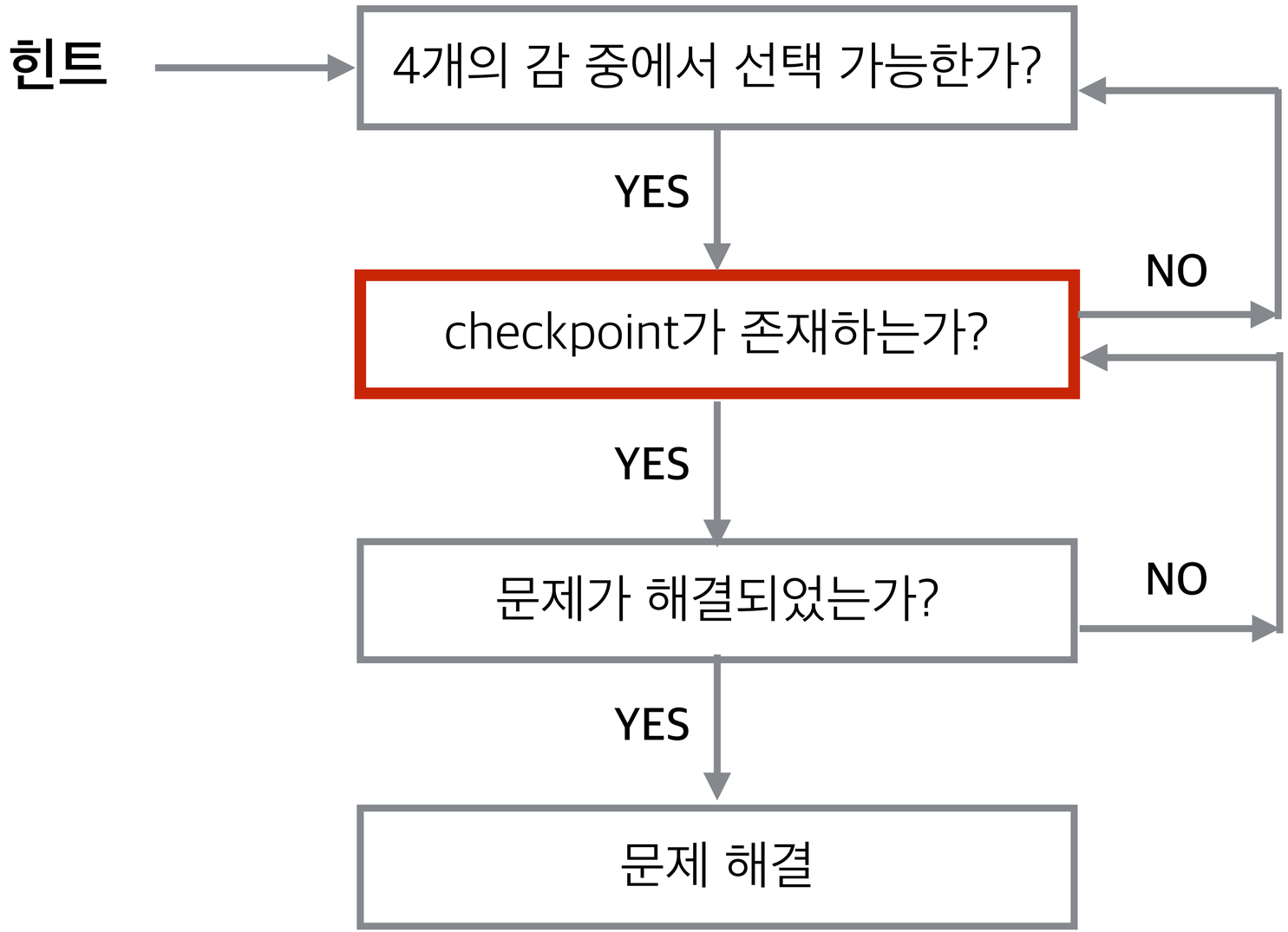
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

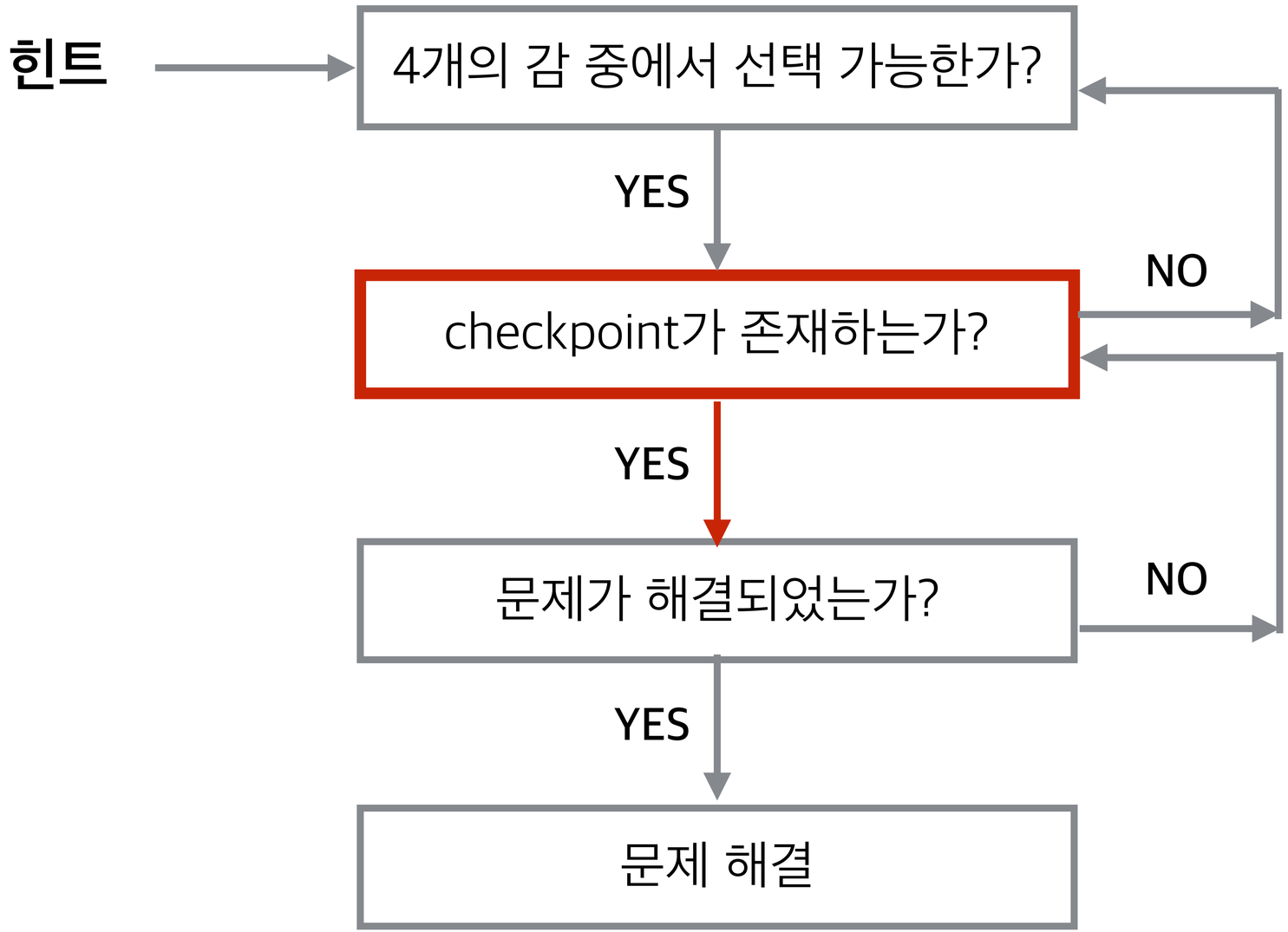
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

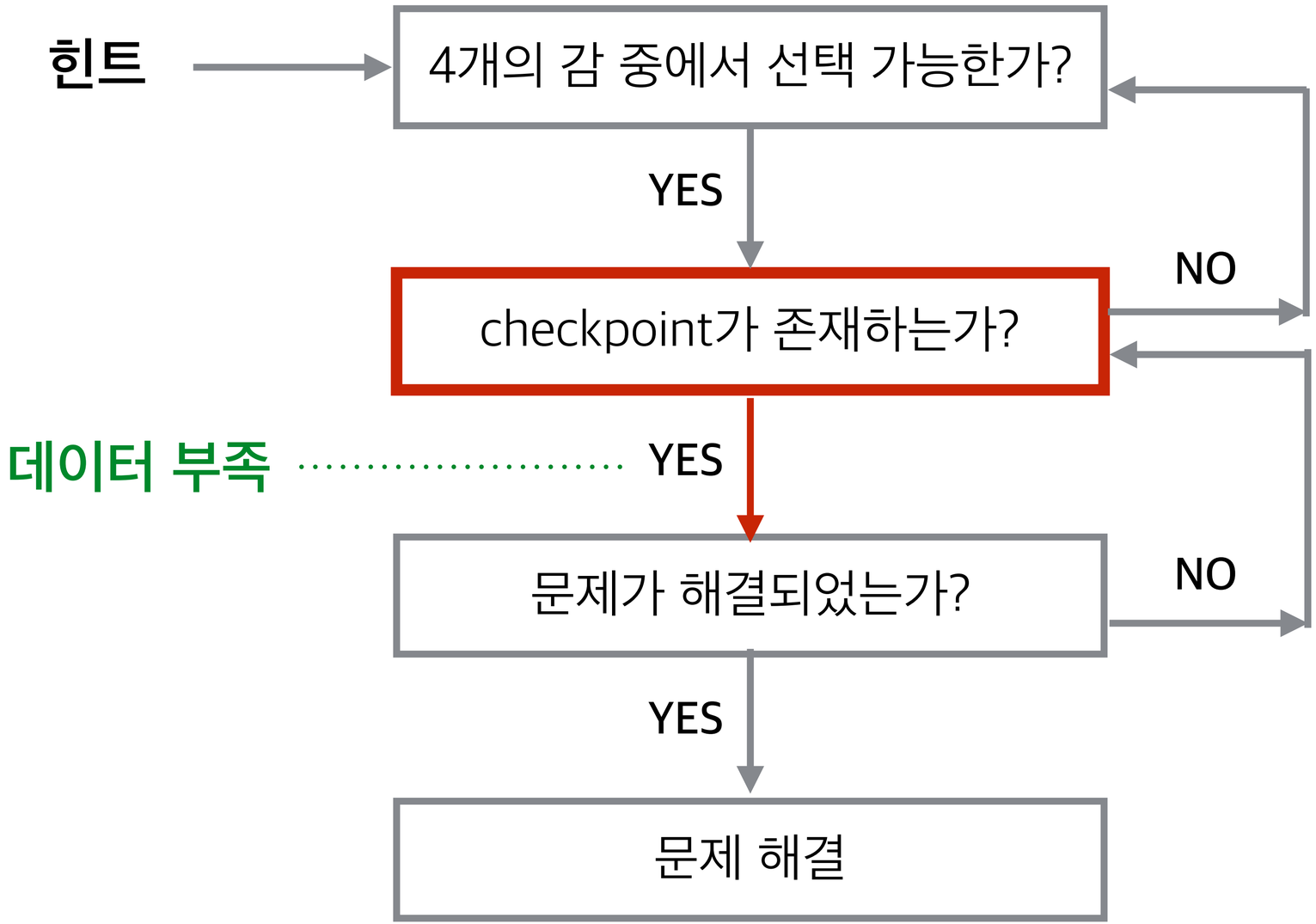
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
문제가 해결되었는가?
NO
YES
문제 해결

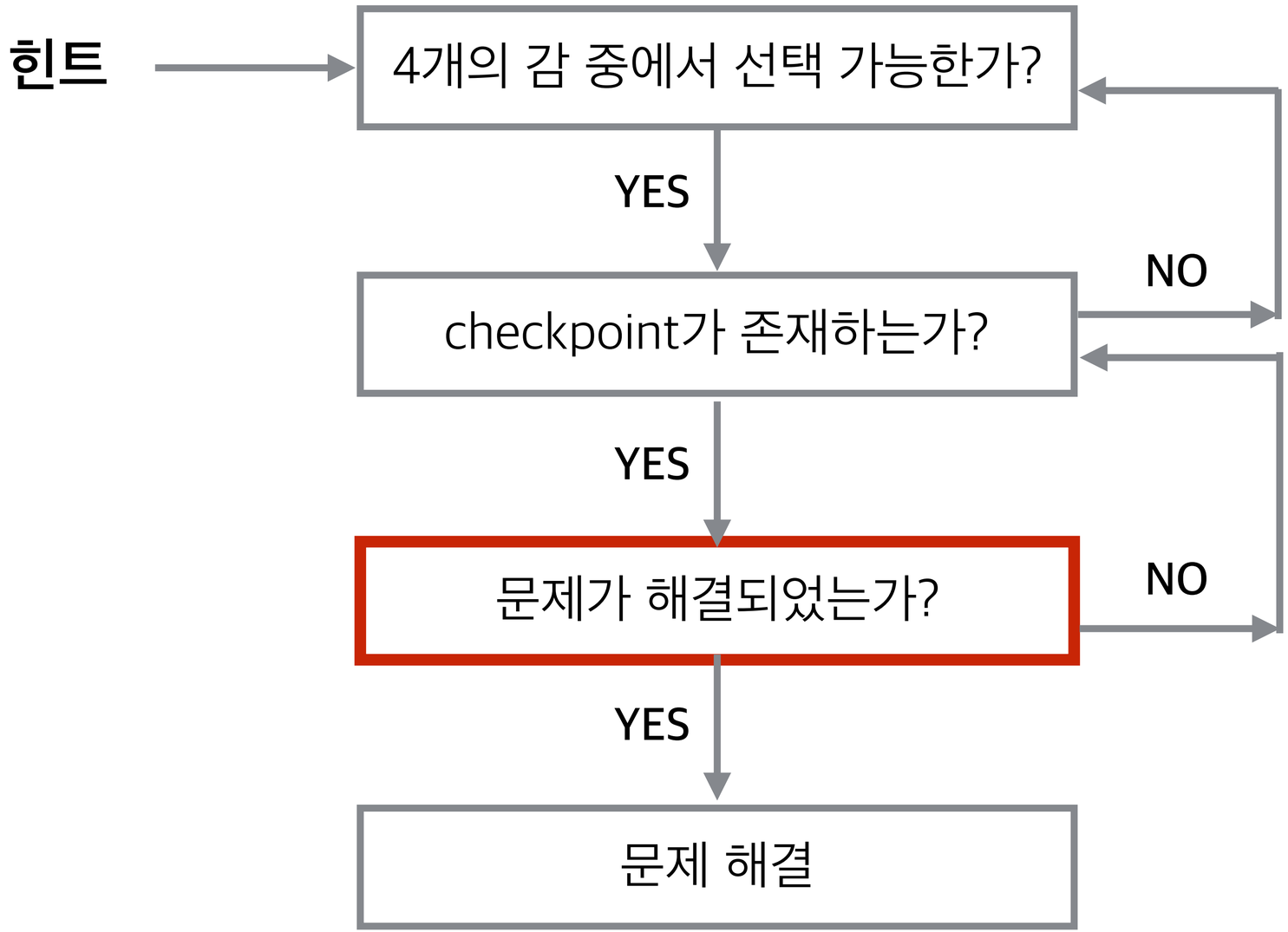
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
문제가 해결되었는가?
NO
YES
문제 해결

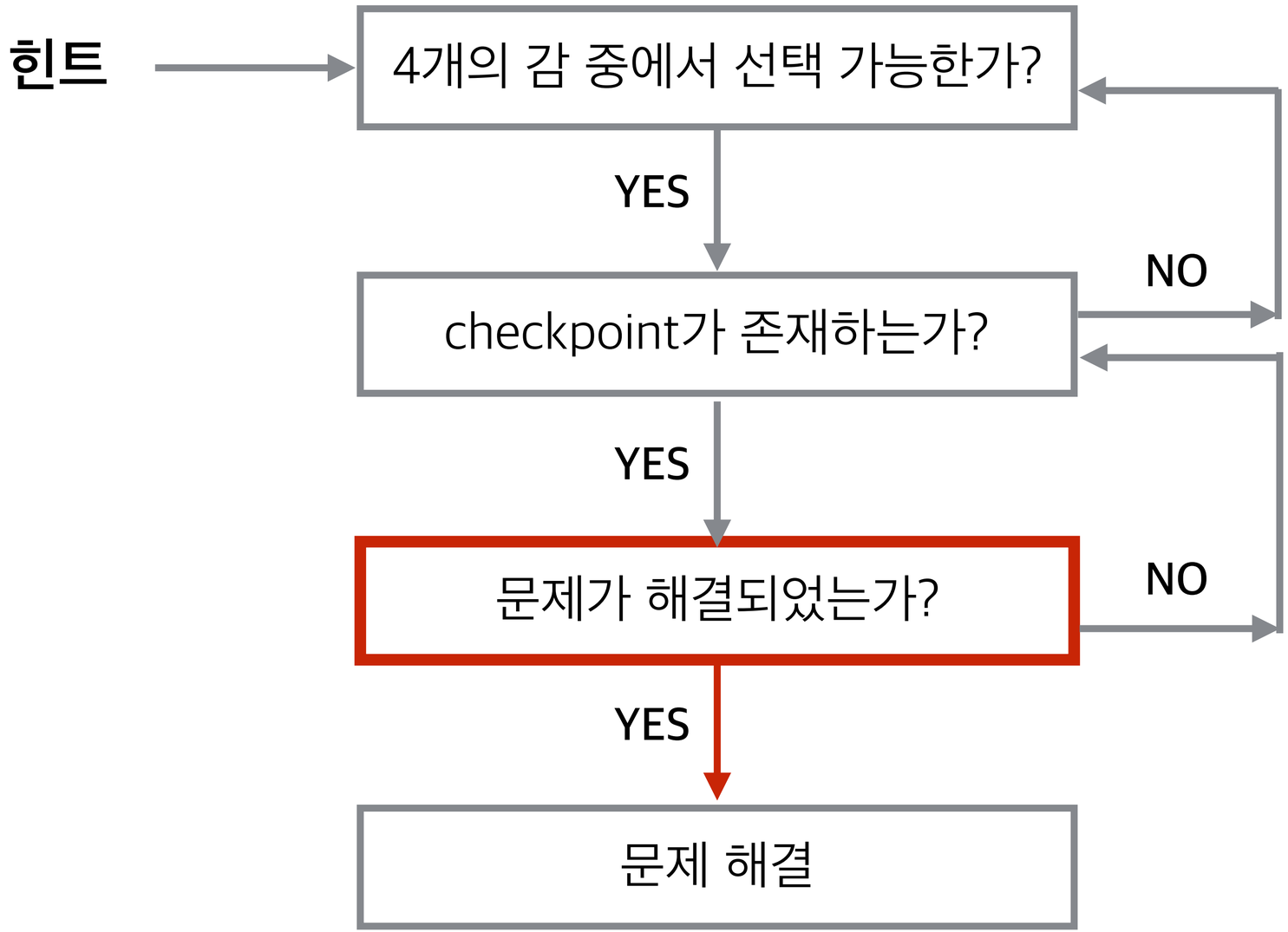
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
데이터 부족
YES
문제가 해결되었는가?
NO
YES
문제 해결

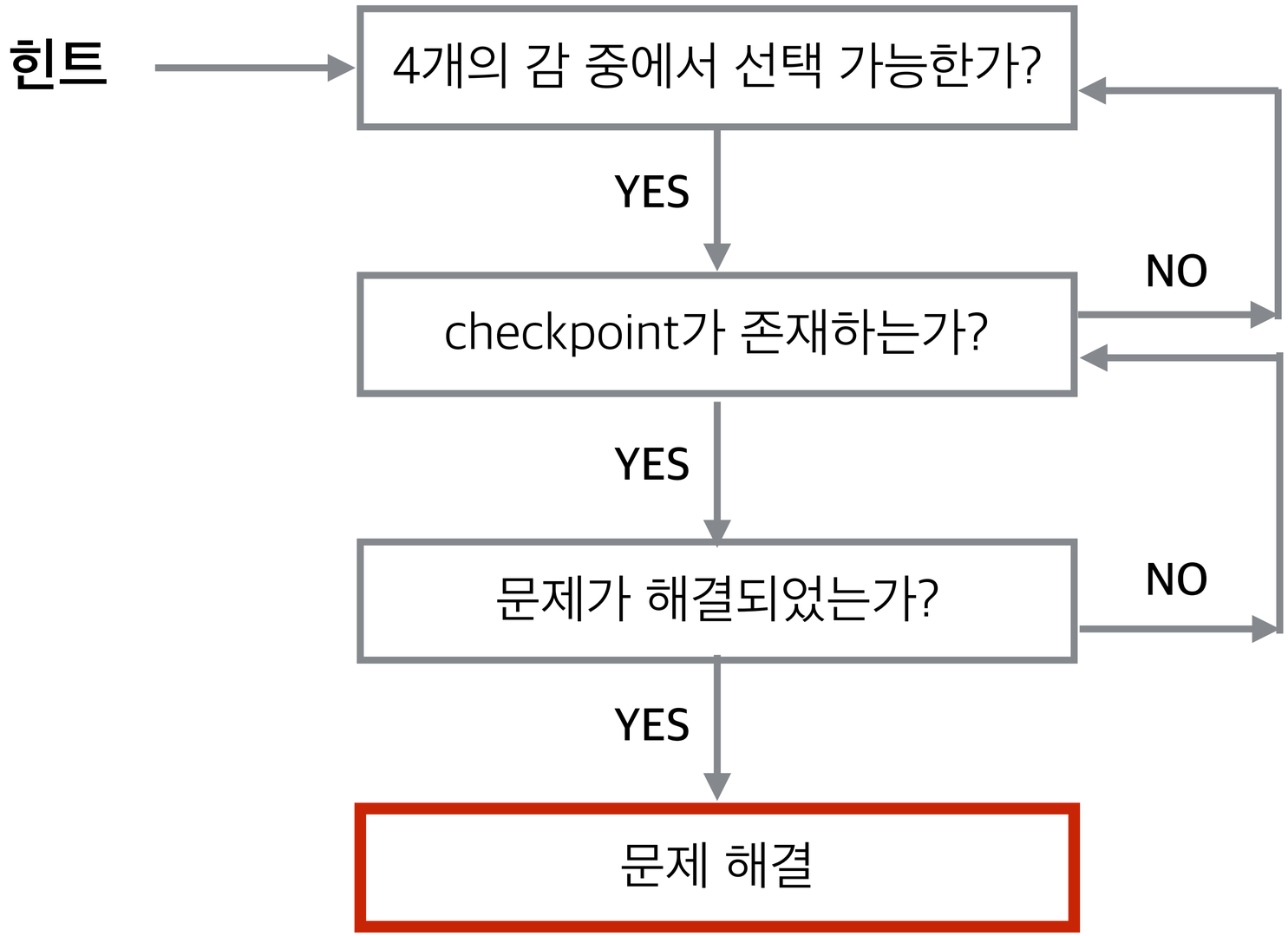
해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
NO
checkpoint가 존재하는가?
YES
NO
문제가 해결되었는가?
YES
문제 해결

해결방법 찾기


힌트
4개의 감 중에서 선택 가능한가?
YES
checkpoint가 존재하는가?
NO
YES
문제가 해결되었는가?
NO
YES
문제 해결

Algorithm Checkpoint

- Loss function
- Gradient descent update error
- Non optional learning rate
- …

Model Checkpoint

- Overfitting
- Model architecture
- Non optional regularization
- Small model
- …

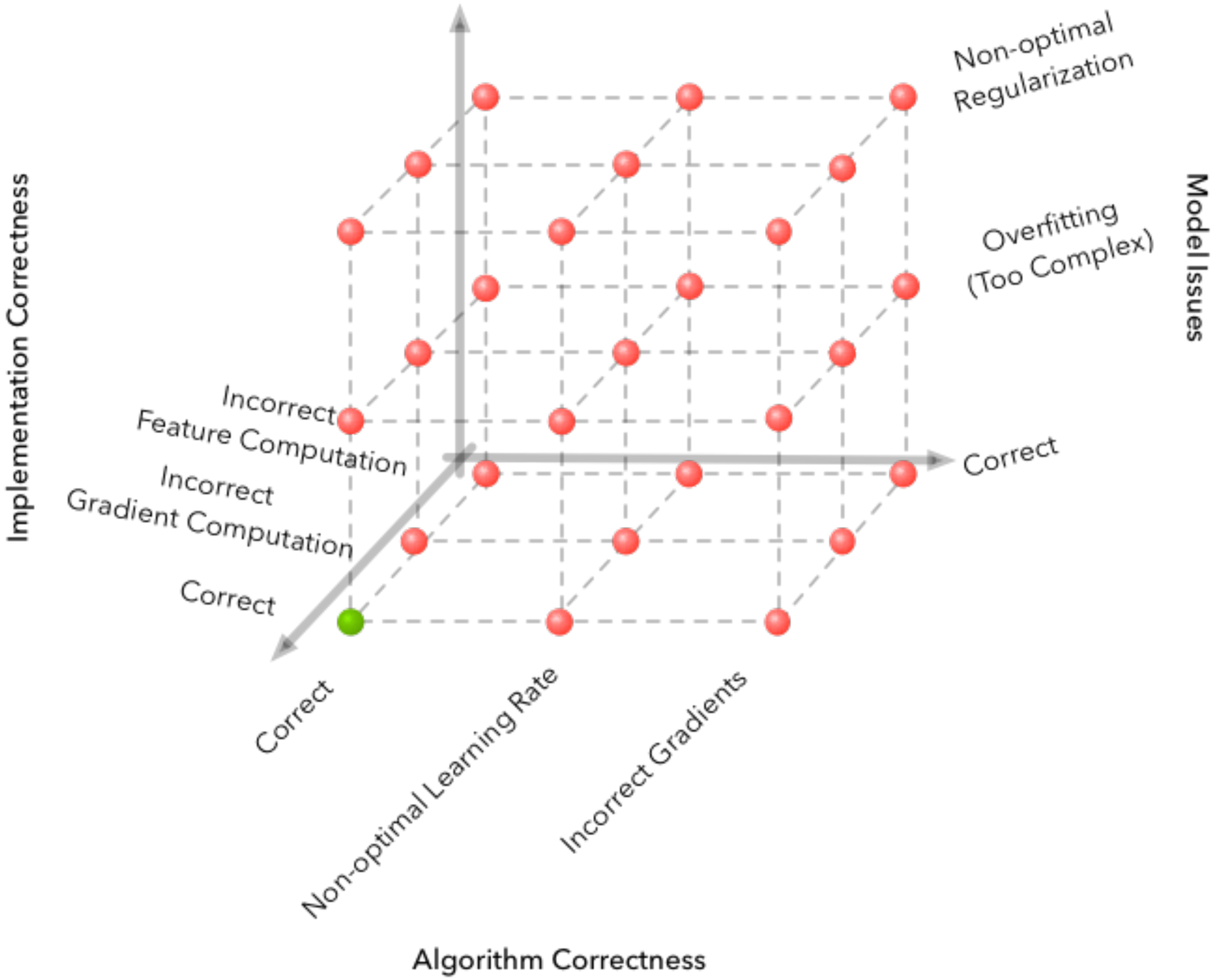
Implementation Checkpoint

- Learning rate
- Training too short
- Value initialization
- Incorrect feature computation
- Incorrect gradient computation
- …

Data Checkpoint

- Normalization
- Not enough data
- Data initialization
- Noisy labels
- Mistake in preprocessing
- Too large/small value
- Feature counts
- …

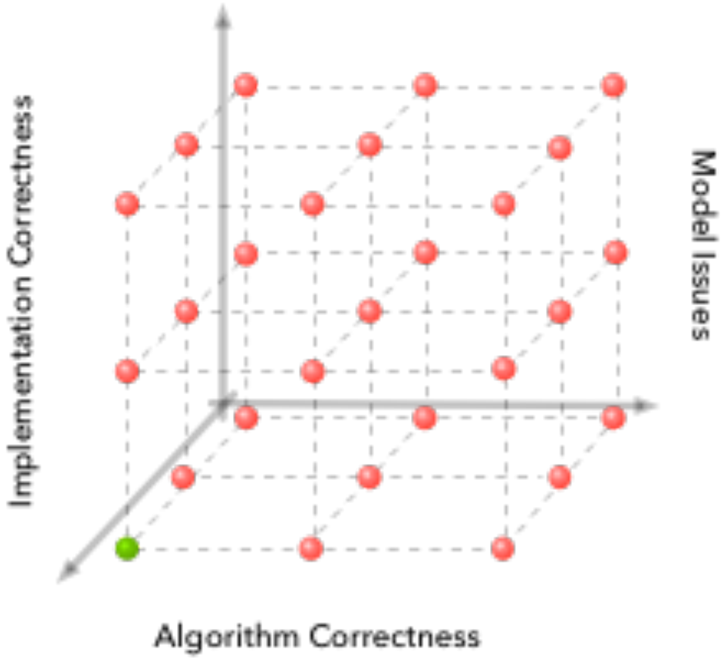
3가지 감에 대한 checkpoint 시각화



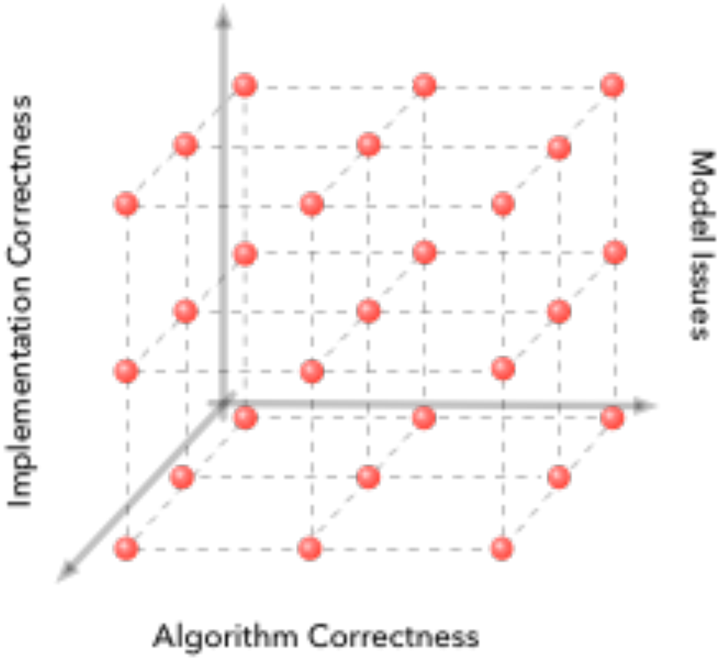
http://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html

4가지 감에 대한 checkpoint 시각화

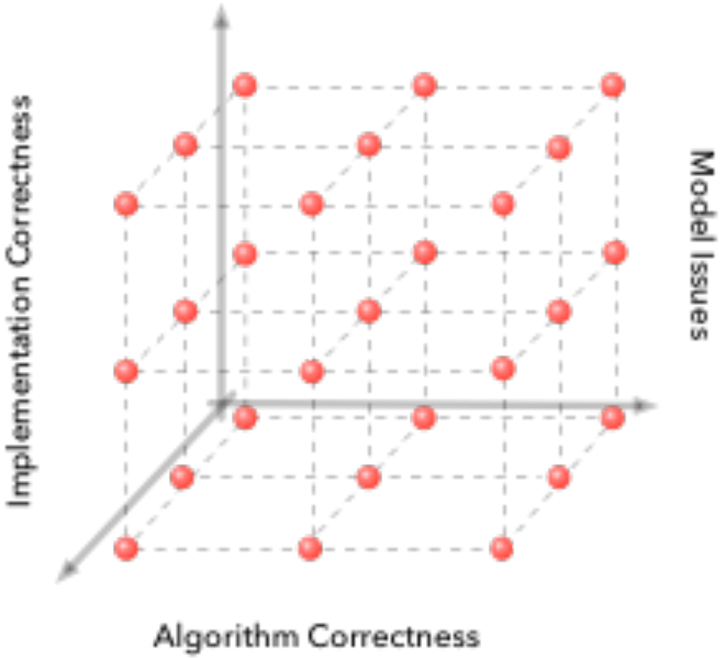
Enough Correct Data



Not Enough Data



Weak Labels

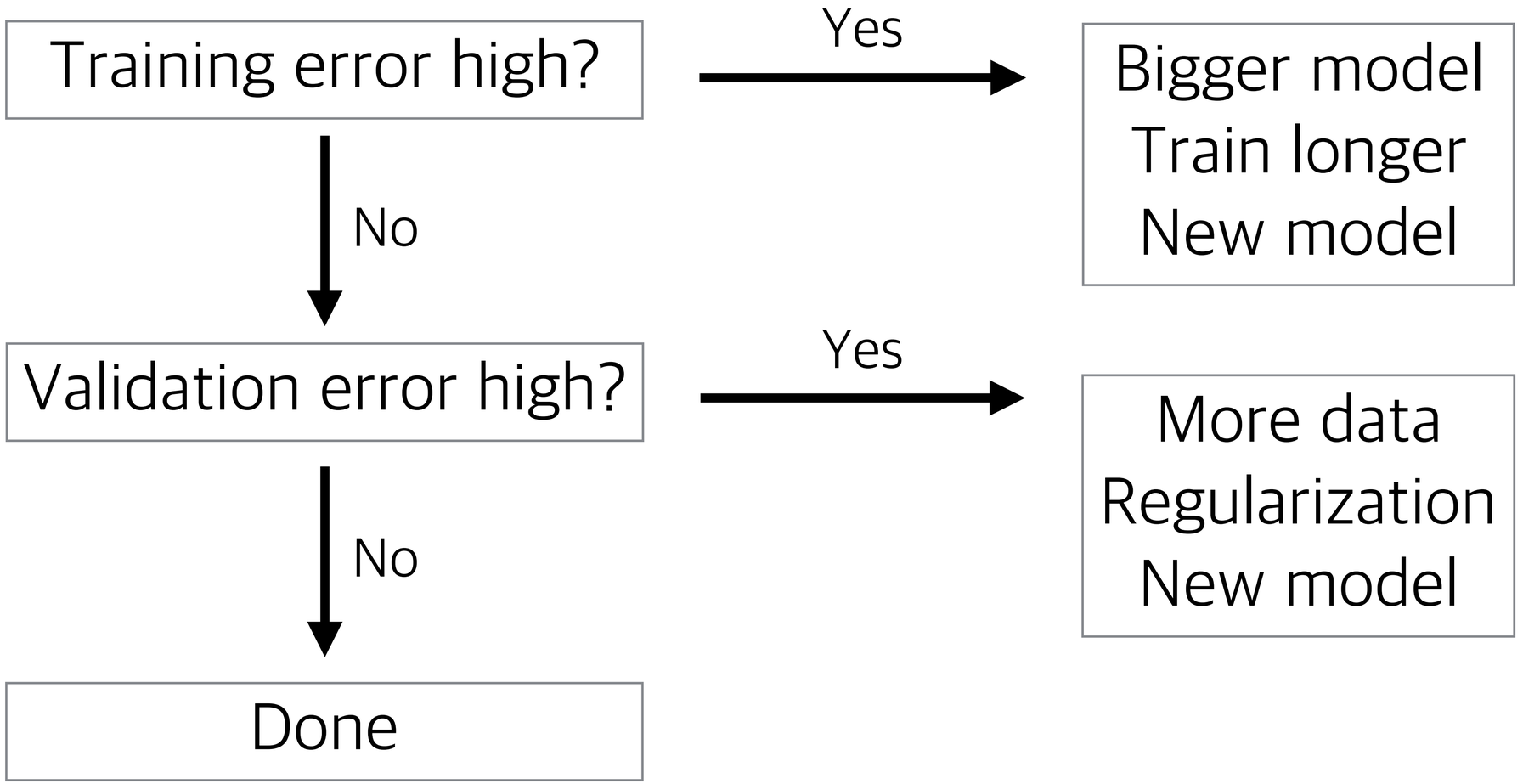


http://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html

적절한 checkpoint를 고르기 위해

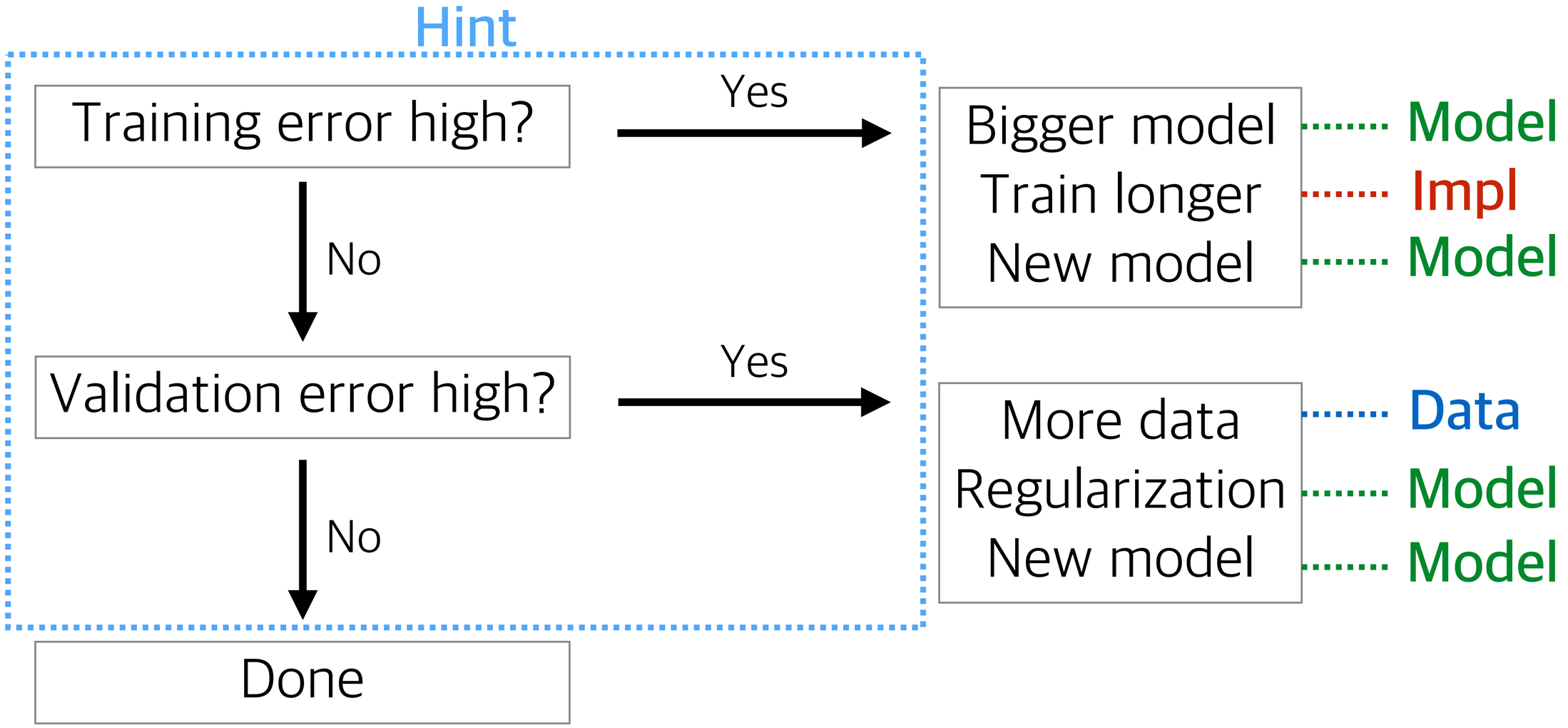
힌트를 잘 봐야 합니다.

So how? - example



Nuts and Bolts of Applying Deep Learning (Andrew Ng)

So how? - example



Nuts and Bolts of Applying Deep Learning (Andrew Ng)

이론적인 접근법이라

실제 적용엔 차이가 있을 수 있습니다.

디버깅 접근법 정리

Hint를 관찰

이에 맞는 감이 뭔지 생각

맞는 checkpoint를 찾아서 확인

Example 1

Example1

간단한 Linear model

```python
import tensorflow as tf

W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)

x = tf.placeholder(tf.float32)
linear_model = W * x + b
y = tf.placeholder(tf.float32)

loss = tf.reduce_sum(tf.square(linear_model - y))

optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(loss)

x_train = [1, 2, 3, 4]
y_train = [0, -1, -2, -3]

init = tf.global_variables_initializer()
sess = tf.Session()

sess.run(init)
for i in range(1000):
    sess.run(train, {x:x_train, y:y_train})

curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
```

Example1

간단한 Linear model

```
1  import tensorflow as tf
2
3  W = tf.Variable([.3], tf.float32)
4  b = tf.Variable([-.3], tf.float32)
5
6  x = tf.placeholder(tf.float32)
7  linear_model = W * x + b
8  y = tf.placeholder(tf.float32)
```

Example1

간단한 Linear model

```
10  loss = tf.reduce_sum(tf.square(linear_model - y))
11
12  optimizer = tf.train.GradientDescentOptimizer(0.01)
13  train = optimizer.minimize(loss)
```

Example1

간단한 Linear model

```
18  init = tf.global_variables_initializer()
19  sess = tf.Session()
20
21  sess.run(init)
22  for i in range(1000):
23      sess.run(train, {x:x_train, y:y_train})
24
25  curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
26  print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
```

Example1

```
1  import tensorflow as tf
2
3  W = tf.Variable([.3], tf.float32)
4  b = tf.Variable([-.3], tf.float32)
5
6  x = tf.placeholder(tf.float32)
7  linear_model = W * x + b
8  y = tf.placeholder(tf.float32)
9
10 loss = tf.reduce_sum(tf.square(linear_model - y))
11
12 optimizer = tf.train.GradientDescentOptimizer(0.01)
13 train = optimizer.minimize(loss)
14
15 x_train = [1, 2, 3, 4]
16 y_train = [0, -1, -2, -3]
17
18 init = tf.global_variables_initializer()
19 sess = tf.Session()
20
21 sess.run(init)
22 for i in range(1000):
23     sess.run(train, {x:x_train, y:y_train})
24
25 curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
26 print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
```

Example1

x_train = [10, 20, 30, 40],

y_train = [0, -10, -20, -30]

로 변경하니 다음과 같은 결과 출력

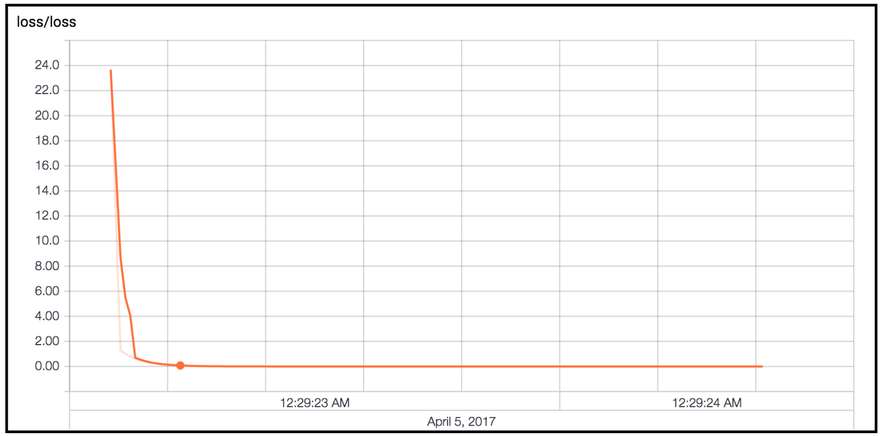
“ W: [nan] b: [nan] loss: nan "

```python
import tensorflow as tf

W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)

x = tf.placeholder(tf.float32)
linear_model = W * x + b
y = tf.placeholder(tf.float32)

loss = tf.reduce_sum(tf.square(linear_model - y))

optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(loss)

x_train = [1, 2, 3, 4]
y_train = [0, -1, -2, -3]

init = tf.global_variables_initializer()
sess = tf.Session()

sess.run(init)
for i in range(1000):
    sess.run(train, {x:x_train, y:y_train})

curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
```

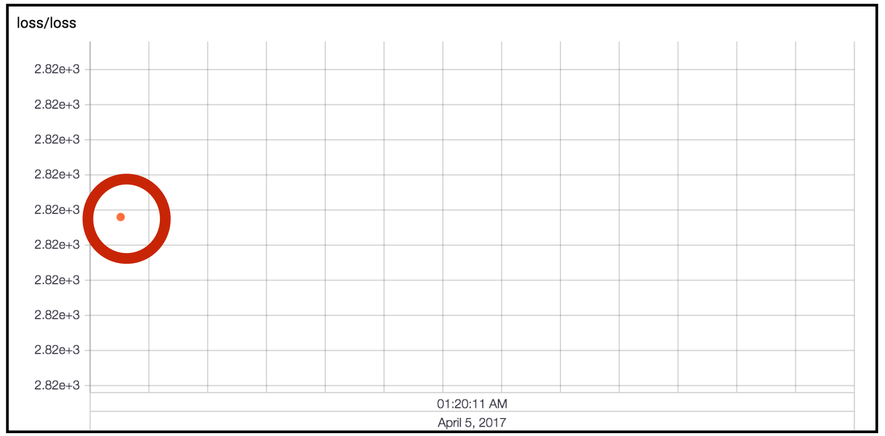
Example1 - hint

- Loss plot

x_train, y_train 변경 전



x_train, y_train 변경 후



- **콘솔 출력**
  - W: [nan] b: [nan] loss: nan

Example1 - hint

힌트 분석

```
1  import tensorflow as tf
2
3  W = tf.Variable([.3], tf.float32)
4  b = tf.Variable([-.3], tf.float32)
5
6  x = tf.placeholder(tf.float32)
7  linear_model = W * x + b
8  y = tf.placeholder(tf.float32)
9
10 loss = tf.reduce_sum(tf.square(linear_model - y))
11
12 optimizer = tf.train.GradientDescentOptimizer(0.01)
13 train = optimizer.minimize(loss)
14 x_train = [1, 2, 3, 4]
15 y_train = [0, -1, -2, -3]
16
17 init = tf.global_variables_initializer()
18 sess = tf.Session()
19
20 sess.run(init)
21 for i in range(1000):
22     sess.run(train, {x:x_train, y:y_train})
23
24 curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
25 print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
26
```

Example1 - hint

힌트 분석

Data

```python
import tensorflow as tf

W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)

x = tf.placeholder(tf.float32)
linear_model = W * x + b
y = tf.placeholder(tf.float32)

loss = tf.reduce_sum(tf.square(linear_model - y))

optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(loss)

x_train = [1, 2, 3, 4]
y_train = [0, -1, -2, -3]

init = tf.global_variables_initializer()
sess = tf.Session()

sess.run(init)
for i in range(1000):
    sess.run(train, {x:x_train, y:y_train})

curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
```

Example1 - hint

힌트 분석

Data

```python
import tensorflow as tf

W = tf.Variable([.3], tf.float32)
b = tf.Variable([-.3], tf.float32)

x = tf.placeholder(tf.float32)
linear_model = W * x + b
y = tf.placeholder(tf.float32)

loss = tf.reduce_sum(tf.square(linear_model - y))

optimizer = tf.train.GradientDescentOptimizer(0.01)
train = optimizer.minimize(loss)

x_train = [1, 2, 3, 4]
y_train = [0, -1, -2, -3]

init = tf.global_variables_initializer()
sess = tf.Session()

sess.run(init)
for i in range(1000):
    sess.run(train, {x:x_train, y:y_train})

curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
```

Example1 - hint

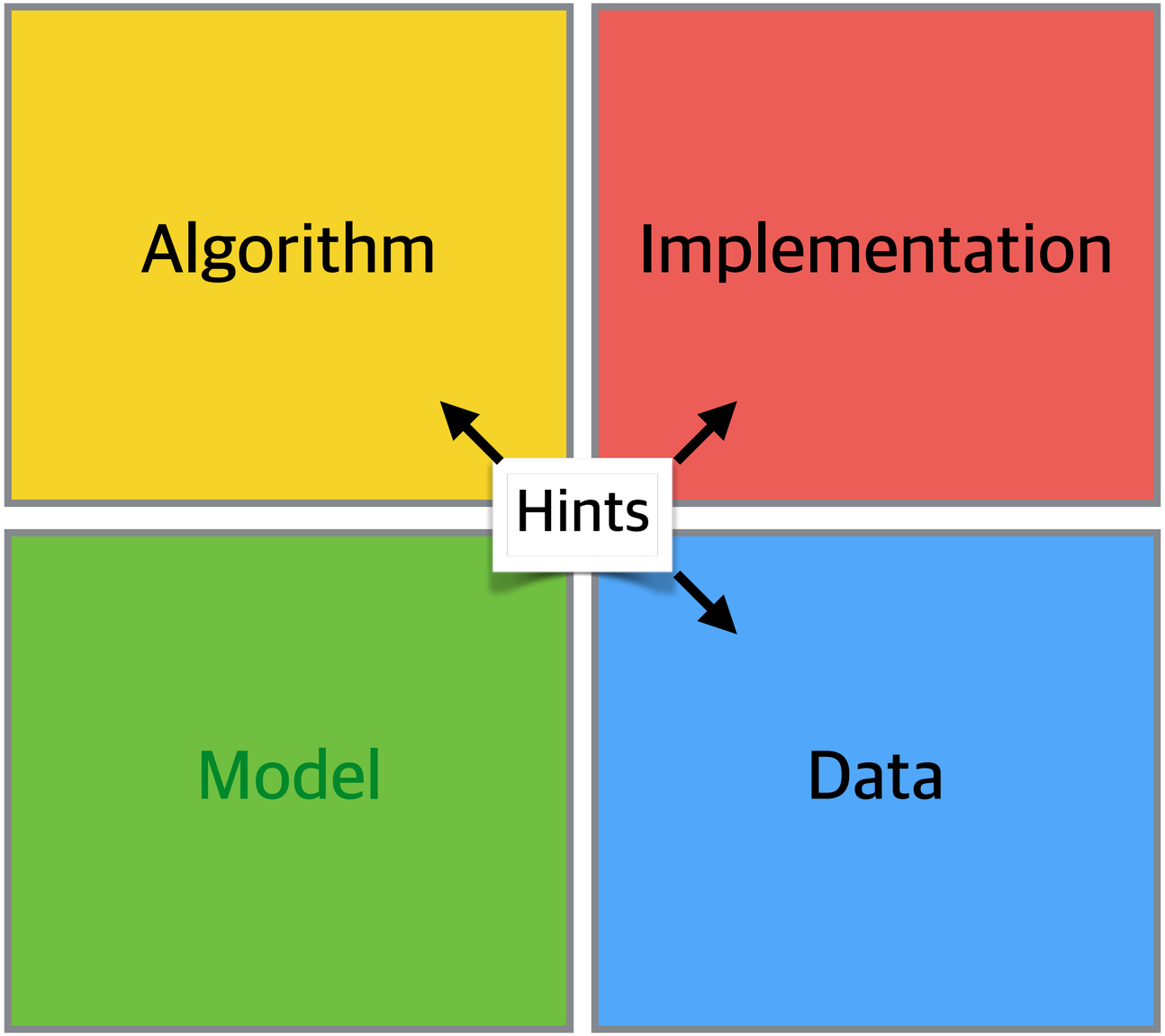
힌트 분석

Data

Imple/Algorithm

```
1  import tensorflow as tf
2
3  W = tf.Variable([.3], tf.float32)
4  b = tf.Variable([-.3], tf.float32)
5
6  x = tf.placeholder(tf.float32)
7  linear_model = W * x + b
8  y = tf.placeholder(tf.float32)
9
10 loss = tf.reduce_sum(tf.square(linear_model - y))
11
12 optimizer = tf.train.GradientDescentOptimizer(0.01)
13 train = optimizer.minimize(loss)
14 x_train = [1, 2, 3, 4]
15 y_train = [0, -1, -2, -3]
16
17 init = tf.global_variables_initializer()
18 sess = tf.Session()
19
20 sess.run(init)
21 for i in range(1000):
22     sess.run(train, {x:x_train, y:y_train})
23
24 curr_W, curr_b, curr_loss = sess.run([W, b, loss], {x:x_train, y:y_train})
25 print("W: %s b: %s loss: %s" % (curr_W, curr_b, curr_loss))
26
```

So how?



Example1 - solution1(Data)

- Data Nomalization
  - tf.nn.l2_normalize(data, dim=0)

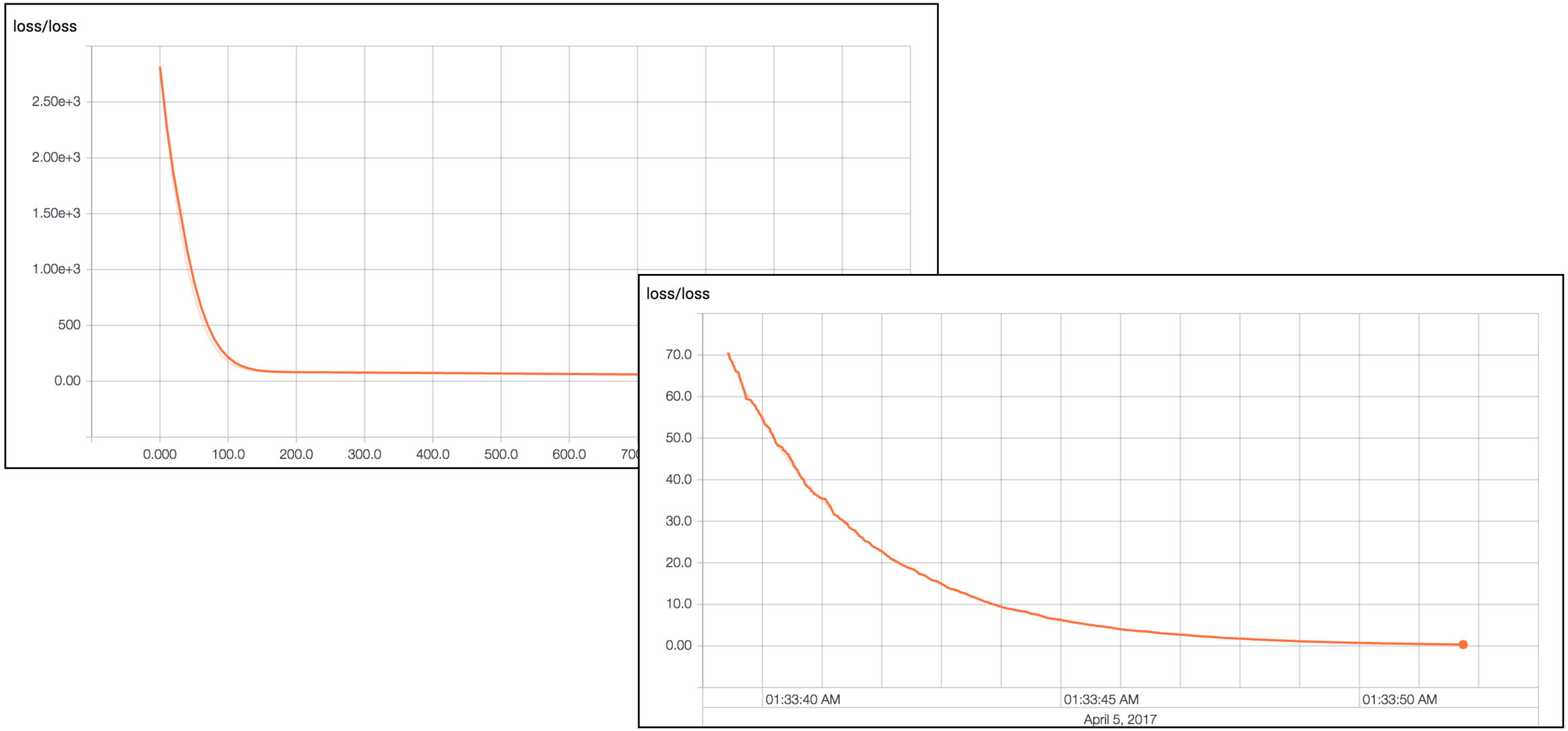
Example1 - solution2(Algorithm)

- 알고리즘의 종류
- GradientDescentOptimizer -> AdamOptimizer

Example1 - solution3(Implementation)

- Learning rate 0.01 -> 0.001
- Iterate 1000 -> 20000

Example1 - solution



Example 2

Example2

5 layers ReLU

```
...

X = tf.placeholder(tf.float32, [None, 28, 28, 1])
Y_ = tf.placeholder(tf.float32, [None, 10])
lr = tf.placeholder(tf.float32)

W1 = tf.Variable(tf.truncated_normal([784, 200], stddev=0.1))  # 784 = 28 * 28
B1 = tf.Variable(tf.ones([L])/10)
W2 = tf.Variable(tf.truncated_normal([200, 100, stddev=0.1))
B2 = tf.Variable(tf.ones([M])/10)
W3 = tf.Variable(tf.truncated_normal([100, 60], stddev=0.1))
B3 = tf.Variable(tf.ones([N])/10)
W4 = tf.Variable(tf.truncated_normal([60, 30], stddev=0.1))
B4 = tf.Variable(tf.ones([O])/10)
W5 = tf.Variable(tf.truncated_normal([30, 10], stddev=0.1))
B5 = tf.Variable(tf.zeros([10]))

# The model
XX = tf.reshape(X, [-1, 784])
Y1 = tf.nn.relu(tf.matmul(XX, W1) + B1)
Y2 = tf.nn.relu(tf.matmul(Y1, W2) + B2)
Y3 = tf.nn.relu(tf.matmul(Y2, W3) + B3)
Y4 = tf.nn.relu(tf.matmul(Y3, W4) + B4)
Ylogits = tf.matmul(Y4, W5) + B5
Y = tf.nn.softmax(Ylogits)

...

```

https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_2.1_five_layers_relu_lrdecay.py

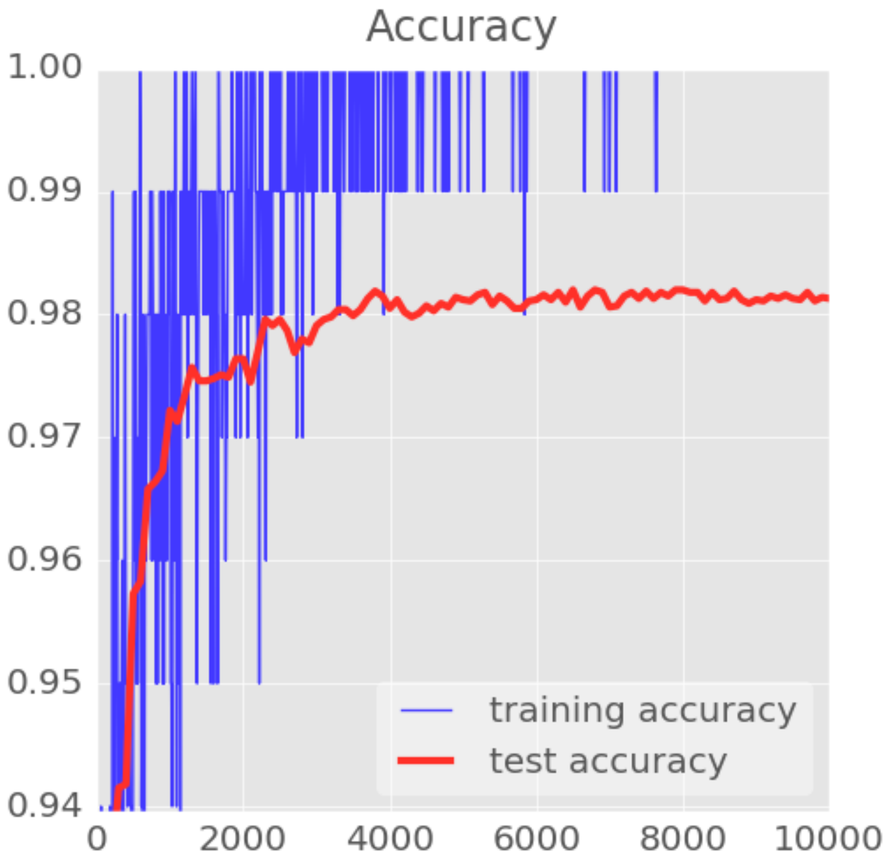
Example2

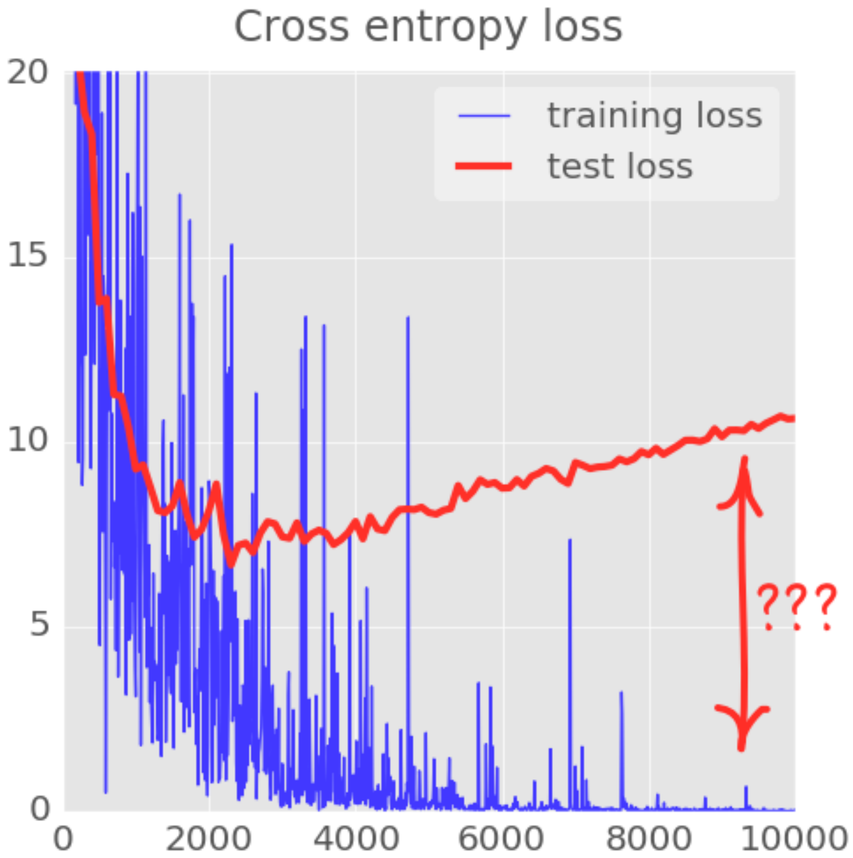
5 layers ReLU

```
18  # The model
19  XX = tf.reshape(X, [-1, 784])
20  Y1 = tf.nn.relu(tf.matmul(XX, W1) + B1)
21  Y2 = tf.nn.relu(tf.matmul(Y1, W2) + B2)
22  Y3 = tf.nn.relu(tf.matmul(Y2, W3) + B3)
23  Y4 = tf.nn.relu(tf.matmul(Y3, W4) + B4)
24  Ylogits = tf.matmul(Y4, W5) + B5
25  Y = tf.nn.softmax(Ylogits)
26
27  ...
28
```

https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_2.1_five_layers_relu_lrdecay.py

Example2 - hint

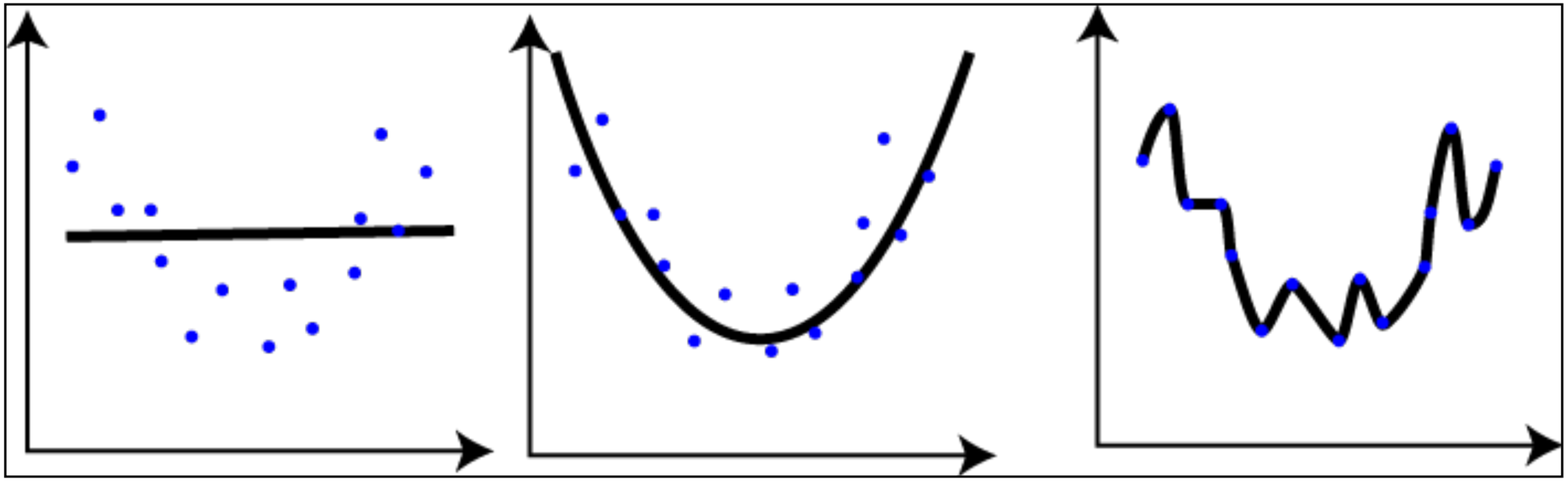




https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#9

Example2 - hint

- Overfitting



https://shapeofdata.wordpress.com/2013/03/26/general-regression-and-over-fitting/

Example2 - solution1(Model)

- Dropout
  - keep_prob = tf.placeholder(tf.float32)
  - h_drop = tf.nn.dropout(h_fc, keep_prob)

Example2 - solution1(Model)

```
18  # The model
19  XX = tf.reshape(X, [-1, 28*28])
20
21  Y1 = tf.nn.relu(tf.matmul(XX, W1) + B1)
22  Y1d = tf.nn.dropout(Y1, pkeep)
23
24  Y2 = tf.nn.relu(tf.matmul(Y1d, W2) + B2)
25  Y2d = tf.nn.dropout(Y2, pkeep)
26
27  Y3 = tf.nn.relu(tf.matmul(Y2d, W3) + B3)
28  Y3d = tf.nn.dropout(Y3, pkeep)
29
30  Y4 = tf.nn.relu(tf.matmul(Y3d, W4) + B4)
31  Y4d = tf.nn.dropout(Y4, pkeep)
32
33  Ylogits = tf.matmul(Y4d, W5) + B5
34  Y = tf.nn.softmax(Ylogits)
35
36  ...
37
```

https://github.com/martin-gorner/tensorflow-mnist-tutorial/blob/master/mnist_2.1_five_layers_relu_lrdecay.py

Example2 - solution1(Model)

- L1 and L2 regularization
  - tf.contrib.layers.l1_regularizer
  - tf.contrib.layers.l2_regularizer
  - tf.contrib.layers.sum_regularizer

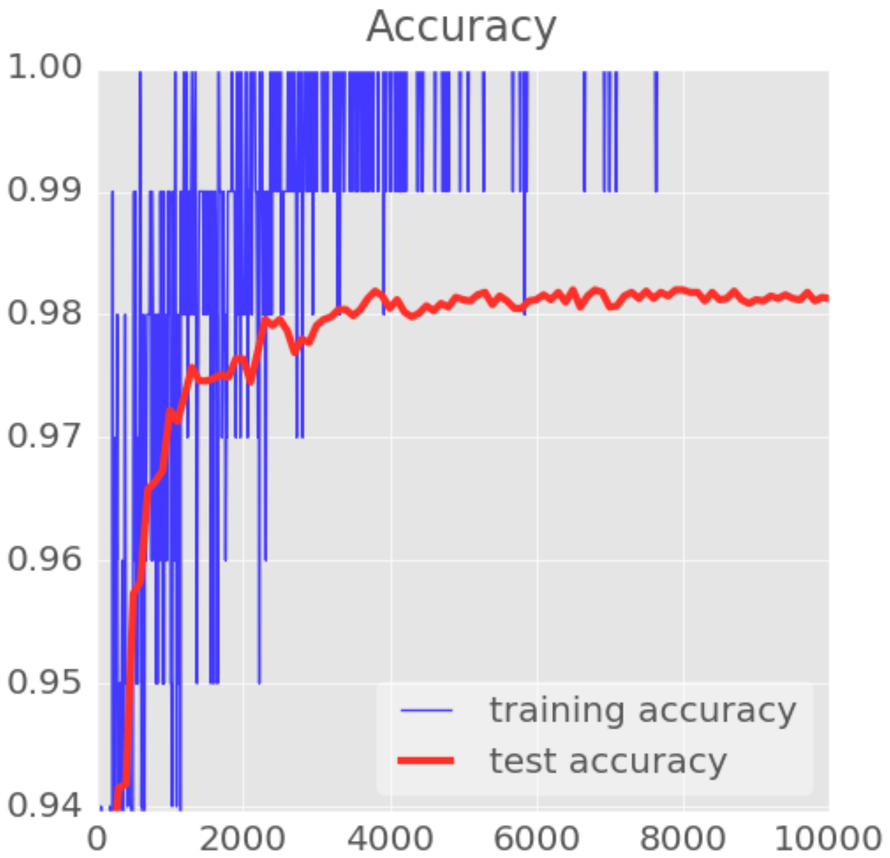
Example2 - solution1(Model)

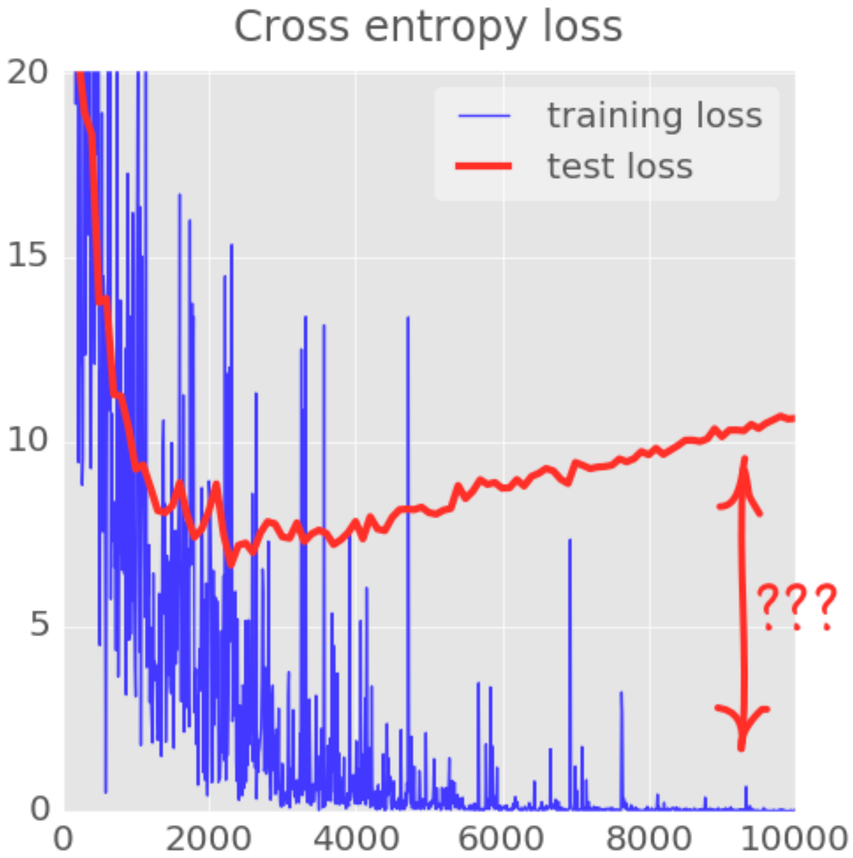
- Make network smaller
- Early stopping

Example2 - solution2(Data)

- Not enough data
- Traning set shuffle
  - (mini batch stochastic training 할 경우)

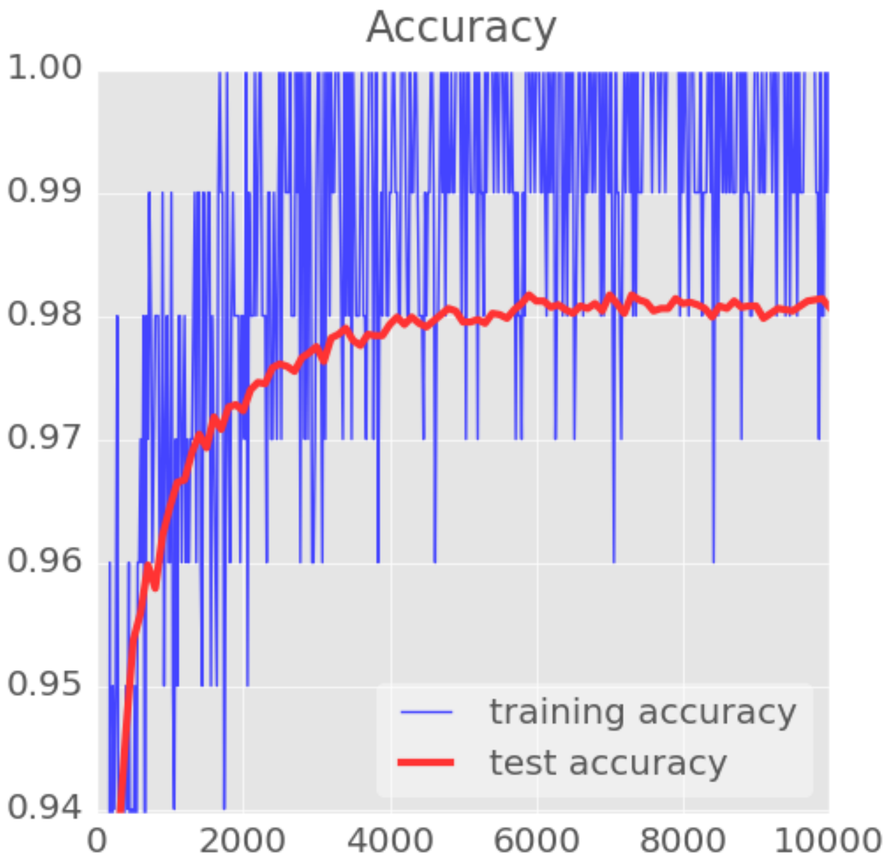
Example2 - solution

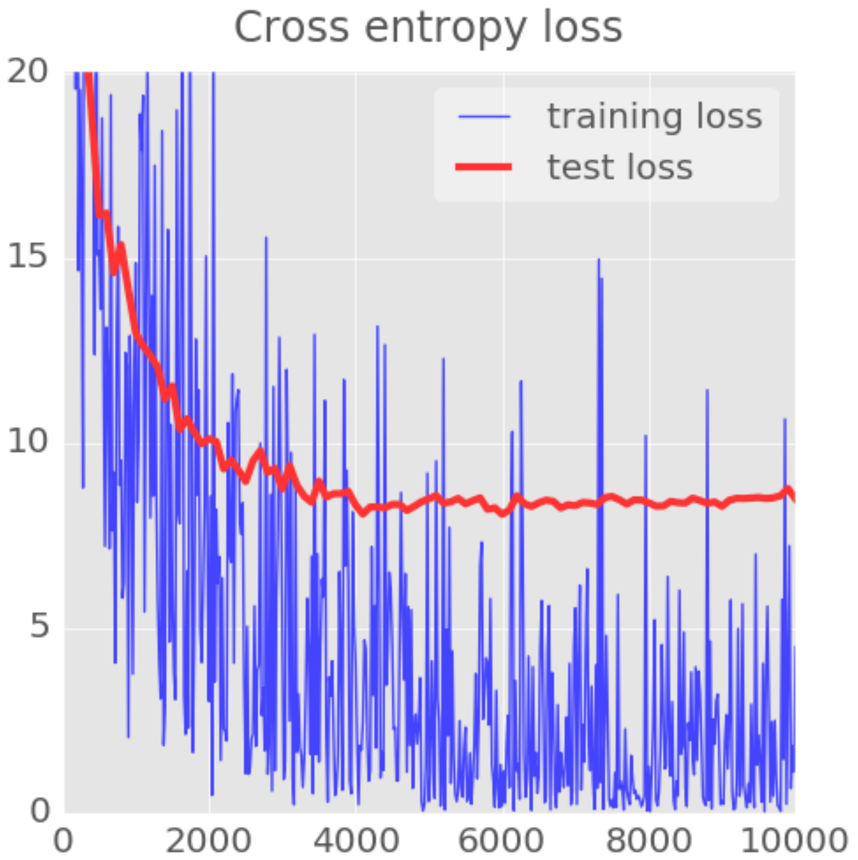




https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#9

Example2 - solution





https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#9

“ 힌트가 잘 안보여요 ”

기타 - Cheating feature 사용

- Cheating feature
  - label과 정확히 일치하는 feature

기타 - Cheating feature 사용

- **Cheating feature만 넣어서 학습**
  - Overfitting 되지 않는다면

    -> 다른 optimizer로 변경하거나 data를 추가

기타 - Cheating feature 사용

- **Cheating feature 1개로 시작해 기존 feature를 하나씩 추가**
  - 어느 시점에 정확도가 올라가지 않는다면

    -> feature 가 너무 많거나 data가 너무 적을 가능성 존재

기타 - Training data 쪼개기

- **Training data를 반으로 쪼개서 학습**
  - 반으로 쪼갰더니 정확도가 많이 떨어진다면

    -> 아직 best 정확도 가 아닐 가능성 존재

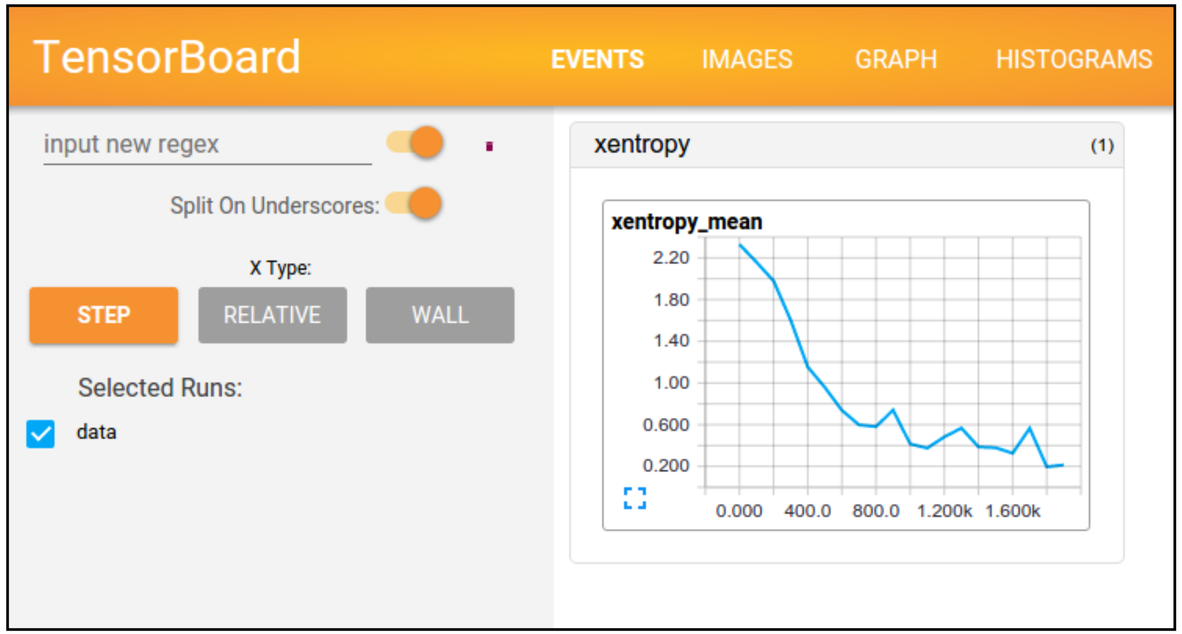
    -> data를 좀더 추가해볼 필요

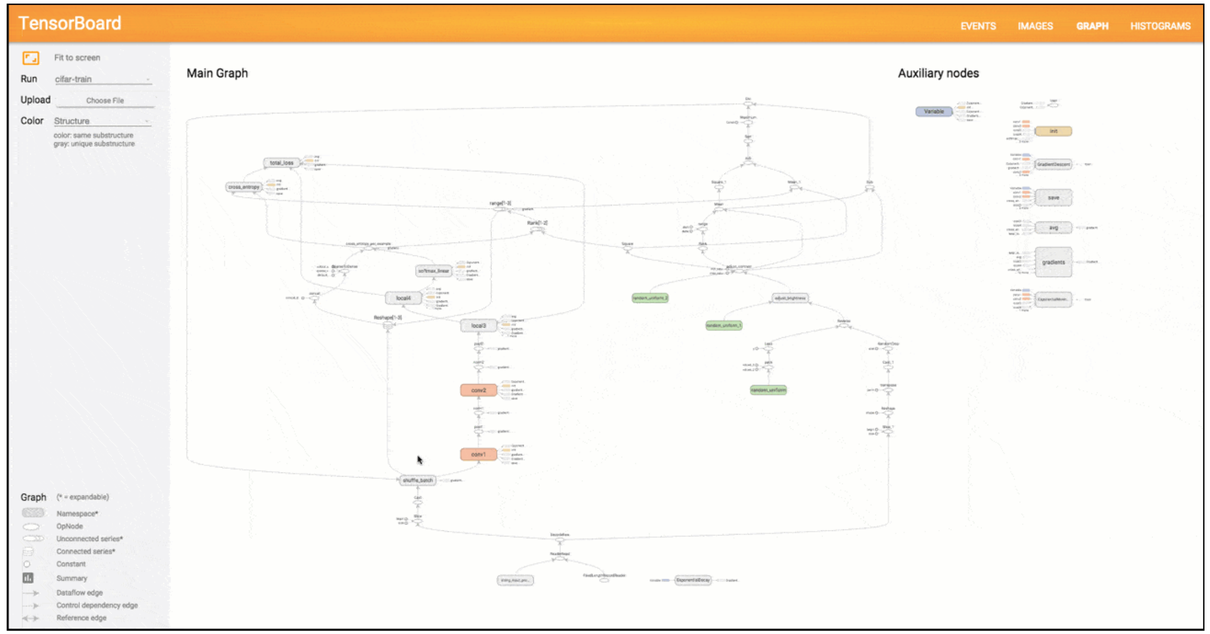
좀 더 편하게 !

기타 - TensorFlow 디버깅 툴

TensorBoard

- 학습 시각화 도구





https://www.tensorflow.org/get_started/summaries_and_tensorboard

기타 - TensorFlow 디버깅 툴

tf.contrib.learn - Logging and Monitor API

- tf.logging.set_verbosity(tf.logging.INFO)

```
INFO:tensorflow:Training steps [0,200)
INFO:tensorflow:global_step/sec: 0
INFO:tensorflow:Step 1: loss_1:0 = 1.48073
INFO:tensorflow:training step 100, loss = 0.19847 (0.001 sec/batch).
INFO:tensorflow:Step 101: loss_1:0 = 0.192693
INFO:tensorflow:Step 200: loss_1:0 = 0.0958682
INFO:tensorflow:training step 200, loss = 0.09587 (0.003 sec/batch).
```

- tf.contrib.learn.monitors.ValidationMonitor

```
validation_monitor = tf.contrib.learn.monitors.ValidationMonitor(
    test_set.data,
    test_set.target,
    every_n_steps=50,
    metrics=validation_metrics,
    early_stopping_metric="loss",
    early_stopping_metric_minimize=True,
    early_stopping_rounds=200)
```

기타 - TensorFlow 디버깅 툴

Debug Operations

- tf.is_finite
- tf.is_inf
- tf.is_nan
- tf.verify_tensor_all_finite
- tf.check_numerics
- tf.Assert
- tf.Print

기타 - TensorFlow 디버깅 툴

tfdbg(Experimental)

- 실행되고 있는 그래프의 내부 구조와 상태를 관찰

```
1. python -m tensorflow.python.debug.examples.debug_mnist --debug (python)
--- run-end: run #12: 1 fetch (train/Adam); 2 feeds ---------------------------
| list_tensors | node_info | print_tensor | list_inputs | list_outputs | run_i
Tensor "cross_entropy/Log:0:DebugIdentity":
  dtype: float32
  shape: (100, 10)

array([[ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
       [ nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
@[16,0]  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan,  nan],
--- Scroll (PgDn): 0.00% -[16,0] ------------------------------- Mouse: ON --
tfdbg> pt cross_entropy/Log:0
```

```
1. python -m tensorflow.python.debug.examples.debug_mnist --debug (python)
--- run-end: run #12: 1 fetch (train/Adam); 2 feeds ---------------------------
| list_tensors | node_info | print_tensor | list_inputs | list_outputs | run_i
Node cross_entropy/Log

  Op: Log
  Device: /job:localhost/replica:0/task:0/cpu:0

  1 input(s) + 0 control input(s):
    1 input(s):
      [Softmax] softmax/Softmax

  3 recipient(s) + 0 control recipient(s):
    3 recipient(s):
      [Shape] train/gradients/cross_entropy/mul_grad/Shape_1
      [Mul] cross_entropy/mul
      [Mul] train/gradients/cross_entropy/mul_grad/mul


Traceback of node construction:
0: /System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/runpy
.py
  Line:     162
  Function: _run_module_as_main
--- Scroll (PgDn): 0.00% ------------------------------------------ Mouse: ON --
tfdbg> ni -t cross_entropy/Log
```

기타 - TensorFlow 디버깅 툴

sklearn GridSearchCV 활용

- 입력된 parameter 값으로 완전 탐색

```
...

class GridSearchTest(test.TestCase):
  """Grid search tests."""

  def testIrisDNN(self):
    if HAS_SKLEARN:
      random.seed(42)
      iris = datasets.load_iris()
      feature_columns = learn.infer_real_valued_columns_from_input(iris.data)
      classifier = learn.DNNClassifier(
          feature_columns=feature_columns,
          hidden_units=[10, 20, 10],
          n_classes=3)
      grid_search = GridSearchCV(
          classifier, {'hidden_units': [[5, 5], [10, 10]]},
          scoring='accuracy',
          fit_params={'steps': [50]})
      grid_search.fit(iris.data, iris.target)
      score = accuracy_score(iris.target, grid_search.predict(iris.data))
      self.assertGreater(score, 0.5, 'Failed with score = {0}'.format(score))
```

tensorflow/tensorflow/contrib/learn/python/learn/grid_search_test.py

Summary

Hint를 관찰

train set error, test set error plot

↓

4가지 중 맞는게 뭔지 생각

Data, Model, Implementation, Algorithm

↓

맞는 checkpoint로 원인을 확인

Regularization, More Data, New Model Architecture…

Reference

- https://www.tensorflow.org/
- http://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html
- https://youtu.be/F1ka6a13S9I
- https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#9
- http://machinelearningmastery.com/machine-learning-checklist/
- https://people.ucsc.edu/~praman1/2015/02/07/Debugging-In-MachineLearning.html
- https://nlpers.blogspot.kr/2016/08/debugging-machine-learning.html

Q & A

Thanks

Mingkim

ming3772@gmail.com