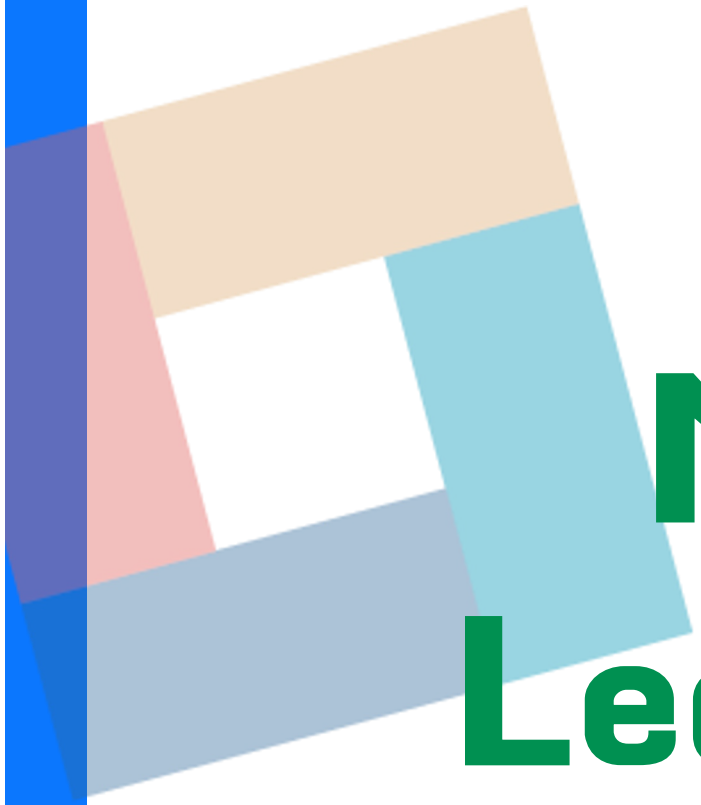


Machine Learning 09

Kihyun Shin
DMSE, HBNU

TheMAO

Bias and variance

Speech recognition example

“What is today’s weather?”

”Coffee shops near me”

Training error J_{train} : 10.8 %

Cross validation error J_{CV} : 14.8 %

0.2 %

4.0 %

Speech recognition example

"What is today's weather?"

"Coffee shops near me"

"What is today's --------?"

Human level performance	: 10.6 %	0.2 %
Training error J_{train}	: 10.8 %	4.0 %
Cross validation error J_{CV}	: 14.8 %	

Establishing a baseline level of performance

What is the level of error you can reasonable hope to get to ?

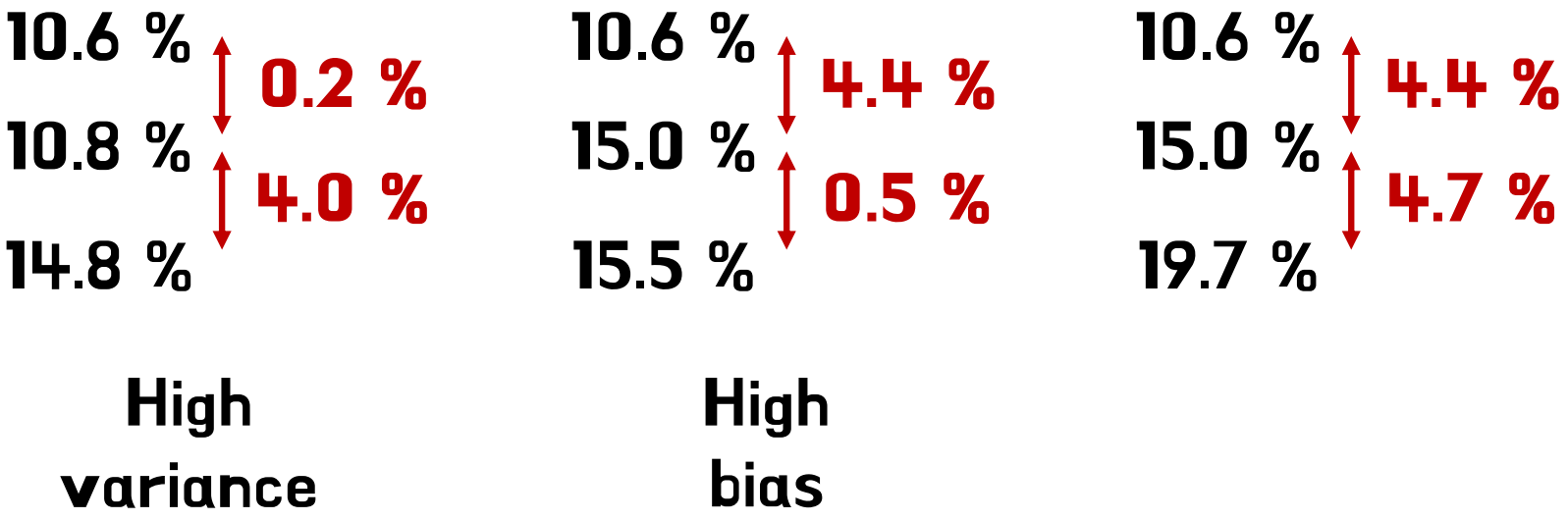
- Human level performance
- Competing algorithm performance
- Guess based on experience

Bias/variance examples

Baseline performance :

Training error (J_{train}) :

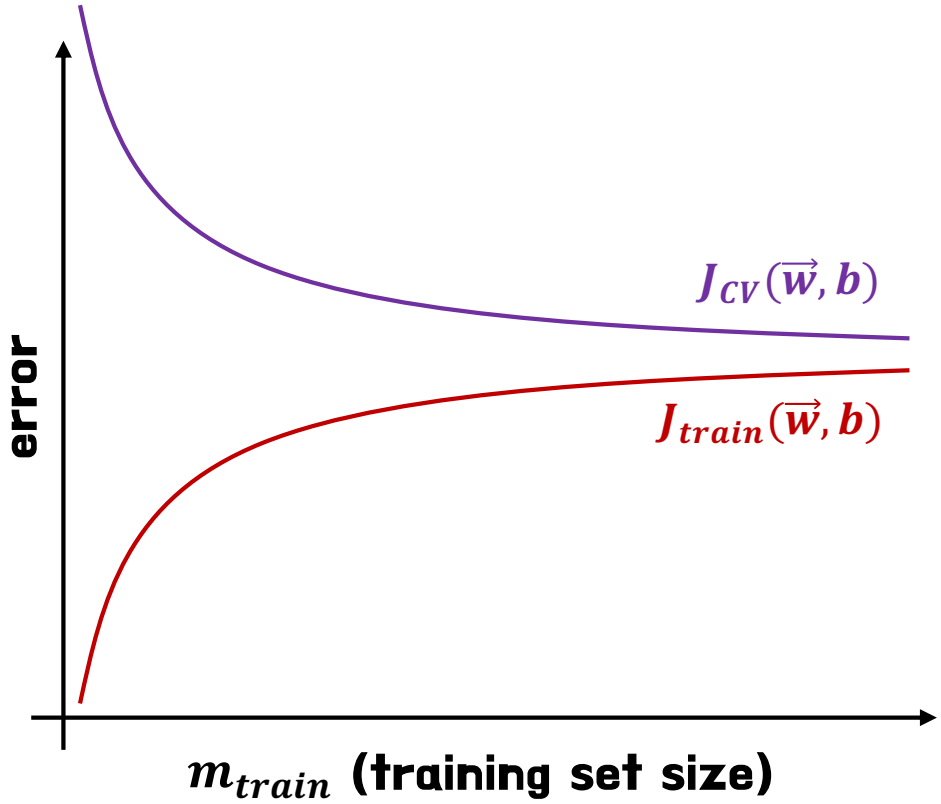
Cross validation error (J_{CV}) :



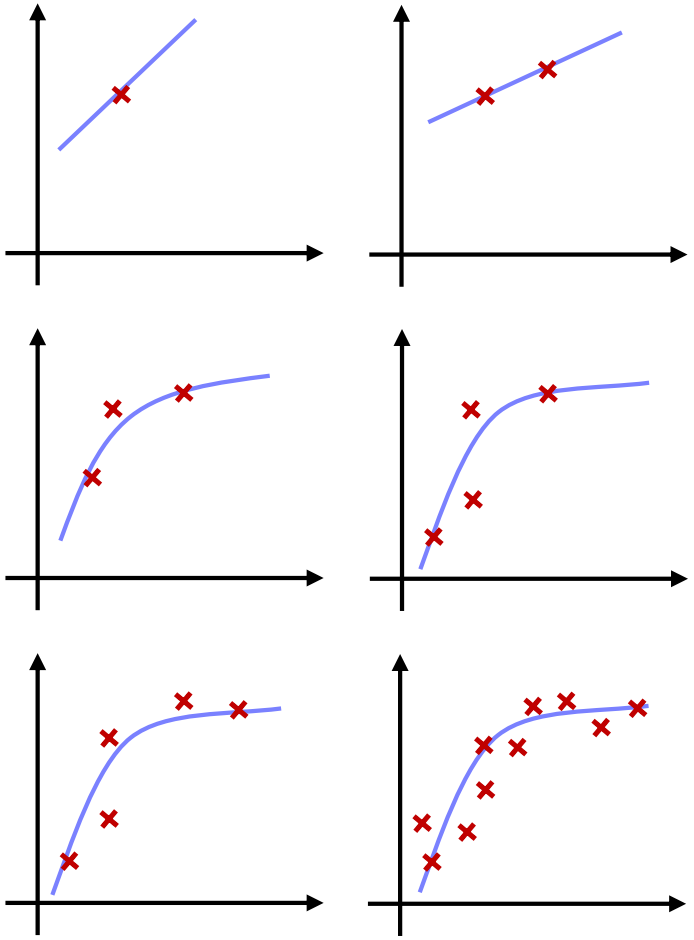
Learning curves

J_{train} = training error

J_{CV} = cross validation error

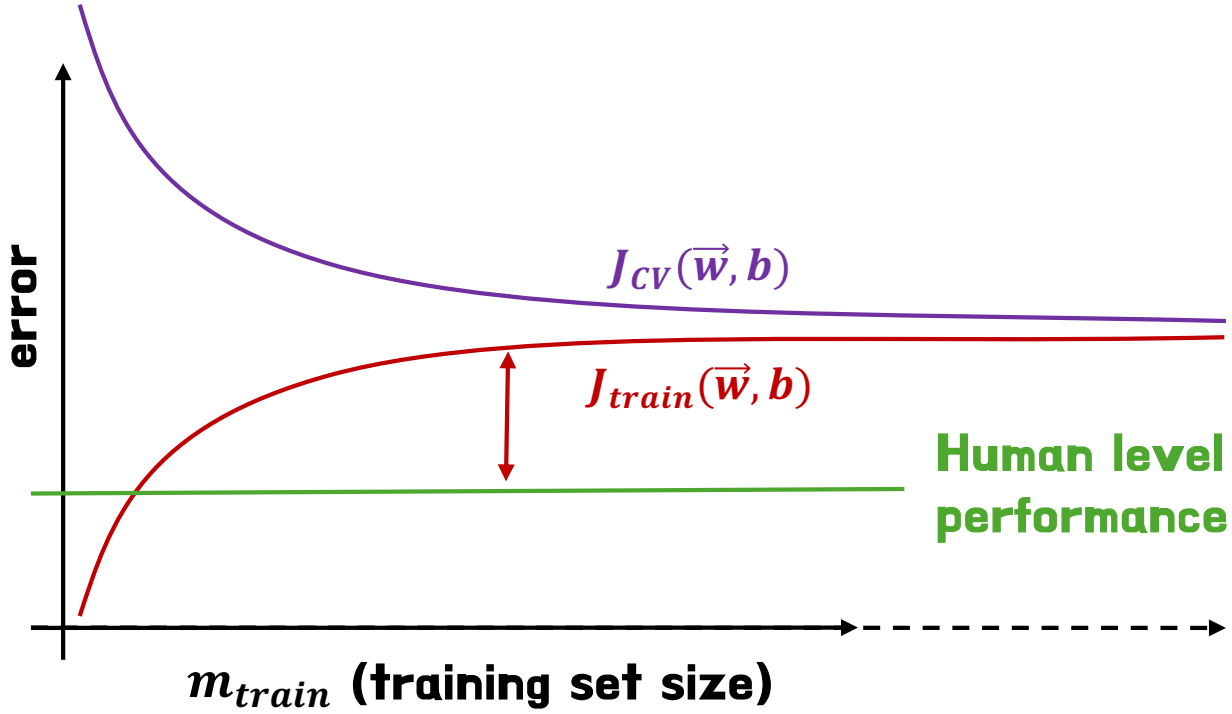


$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + b$

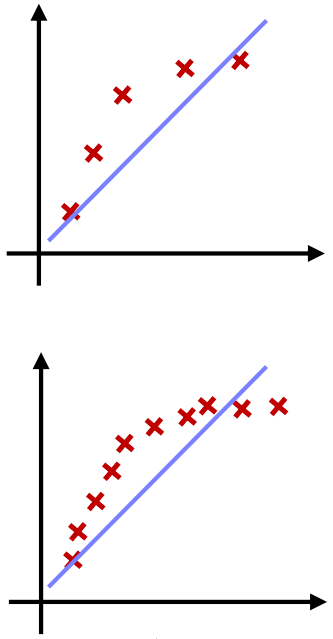


TheMAC

High bias



$f_{\vec{w},b}(x) = w_1 x + b$



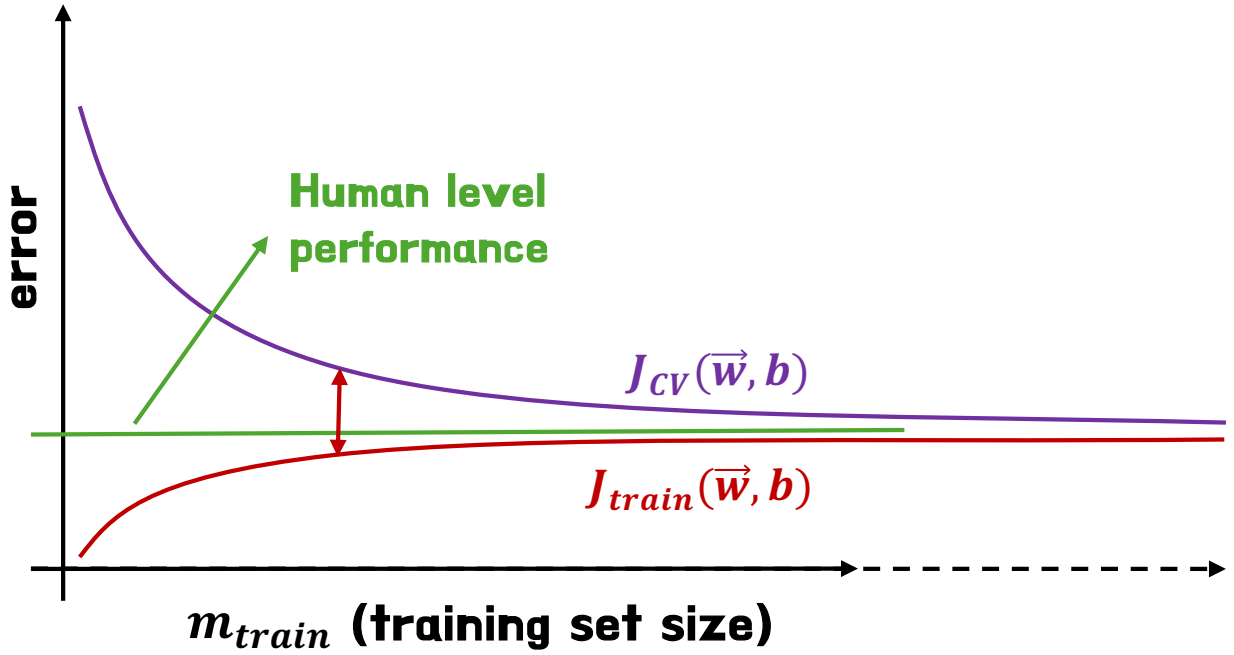
If a learning algorithm (model) suffers from high bias,

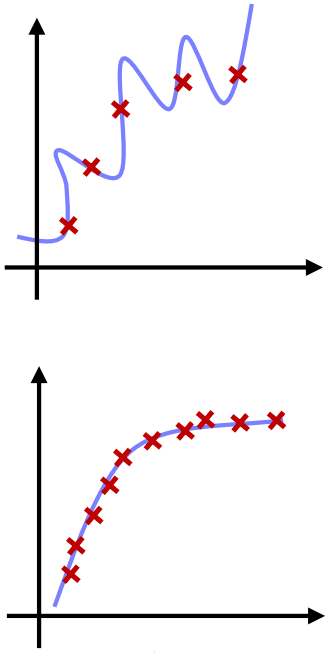
Getting more training data will not help much

High variance

$$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

(with small λ)





If a learning algorithm (model) suffers from high variance

Getting more training data will help much

TheMAD

Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

$$J(\vec{w}, b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$

But it makes unacceptably large errors in predictions.
What do you try next?

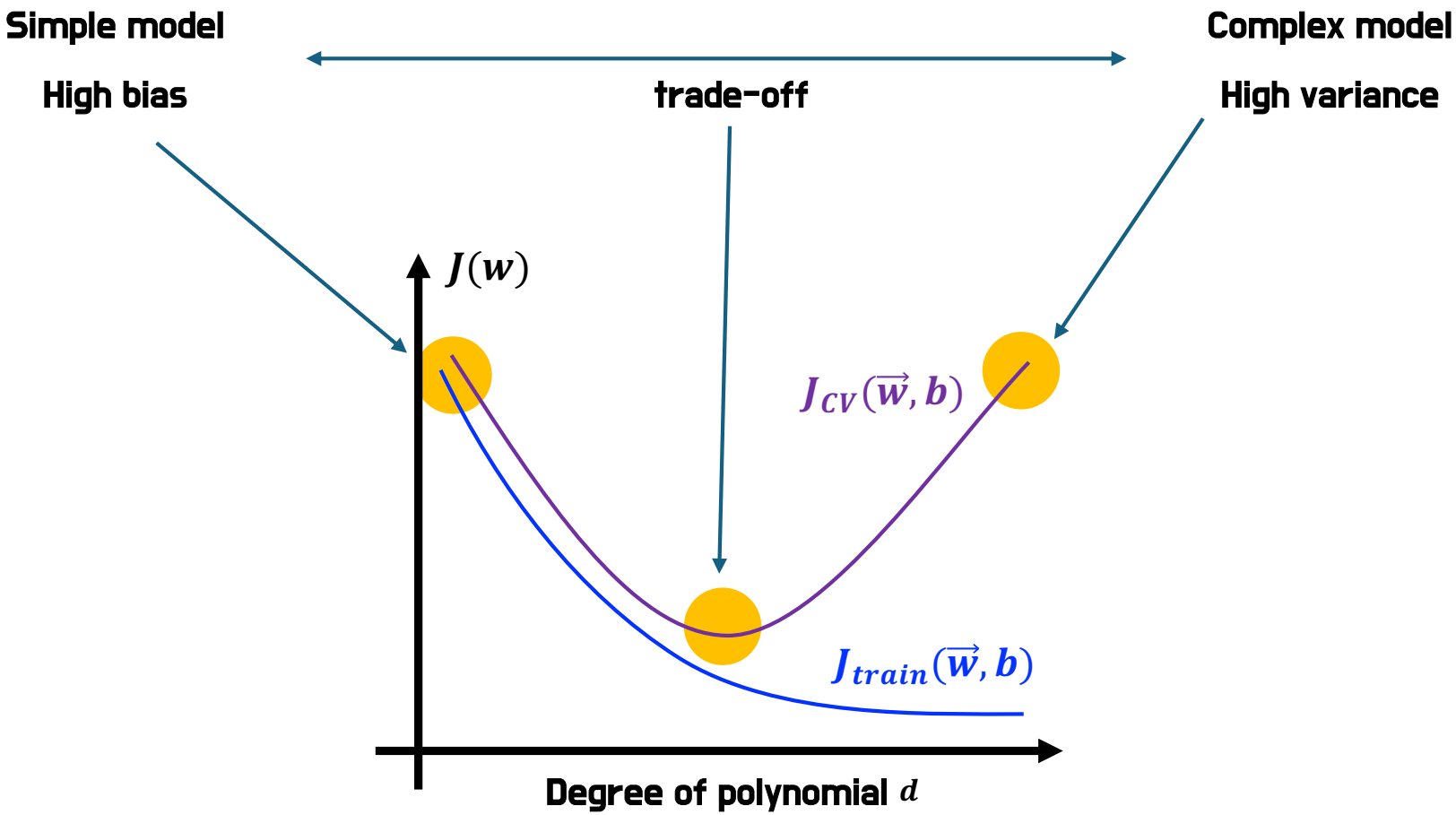
- Get more training examples	Fixes high variance
- Try smaller sets of features	Fixes high variance
- Try getting additional features	Fixes high bias
- Try adding polynomial features ($x_1^2, x_2^2, x_1x_2, etc$)	Fixes high bias
- Try decreasing λ	Fixes high bias
- Try increasing λ	Fixes high variance

The bias variance tradeoff

$f_{\vec{w},b}(x) = w_1 x + b$

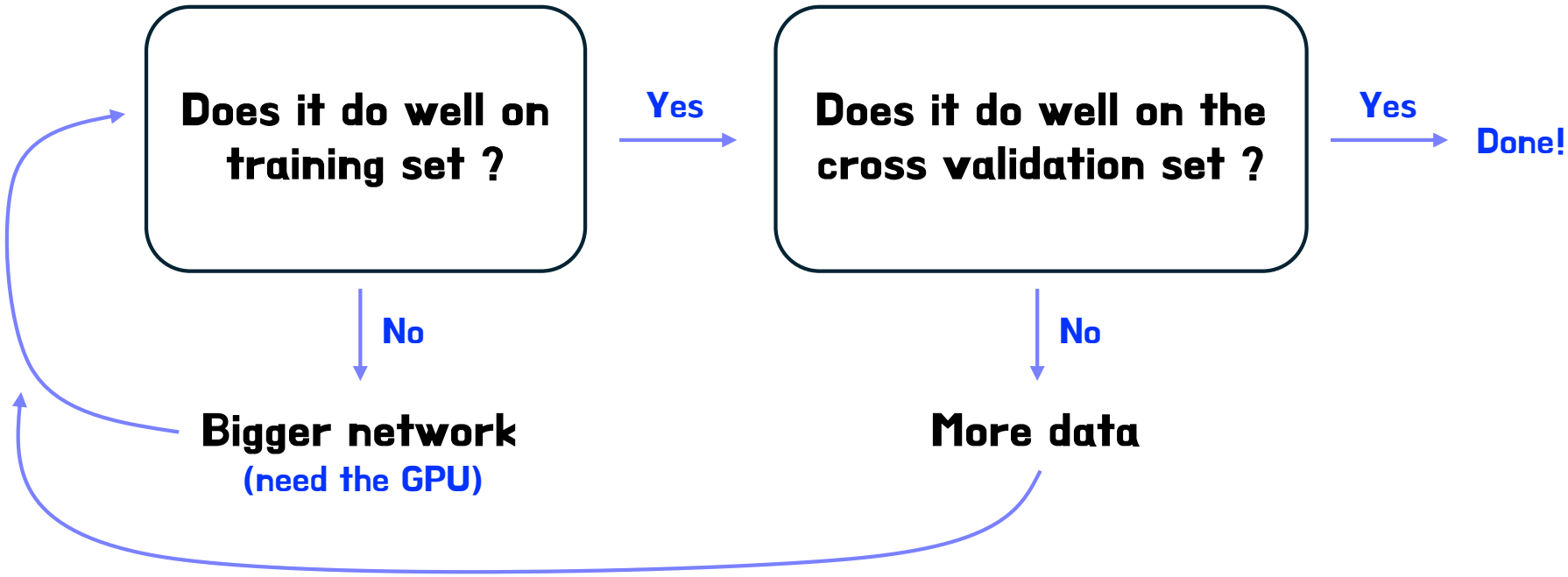
$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + b$

$f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

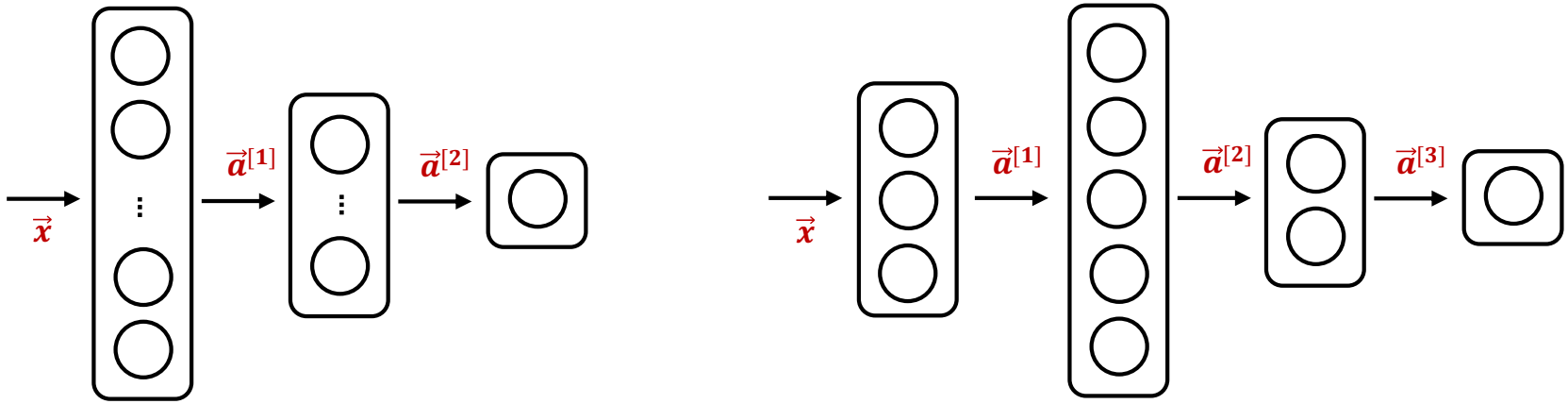


Neural network and bias variance

Large neural networks are low bias machines



Neural network and regularization



3 layers

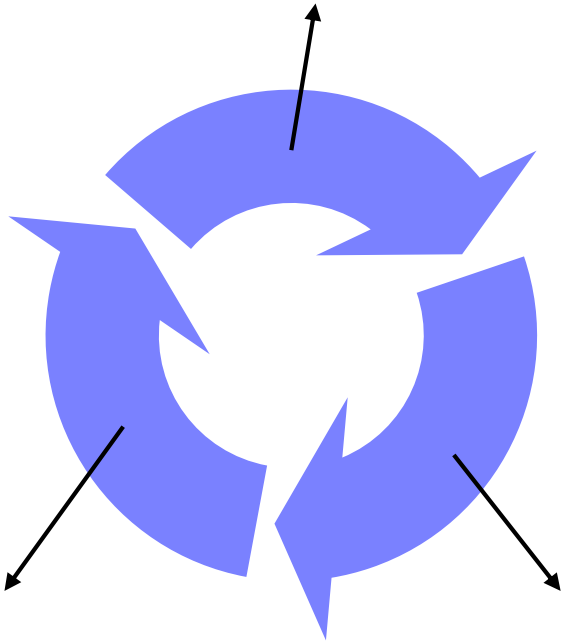
4 layers

A Large neural network will usually do as well or better than a smaller one so long as regularization is chosen appropriately

Machine Learning Development Process

Iterative loop of ML development

Choose architecture
(model, data, etc.)



Train model

Diagnostics
(bias, variance and
error analysis)

Spam classification example

From: cheapsales@buystuffromme.com

To: Kihyun Shin

Subject: Buy Now !

Deal of the week ! Buy now !

Rolex **w4tchs** - $100

Med1cine (any kind) - $50

also low cost **M0rgages**

available.

From: John Doe

To: Kihyun Shin

Subject: Collaboration meeting ?

Hello Kihyun

Do you have any available time in this Thursday ?

I want to discuss about calculation results you sent before.

Building a spam classifier

Supervised learning: $\vec{x}$ = features of email

y = spam (1) or not spam (0)

Features: list the top 10,000 words to compute $x_1, x_2, \ldots, x_{10,000}$

$$\vec{x} = \begin{bmatrix} 0 \\ 1 \\ 1 \text{ or } 2 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \begin{matrix} \text{a} \\ \text{kihyun} \\ \text{buy} \\ \text{deal} \\ \text{discount} \\ \vdots \end{matrix}$$

From: cheapsales@buystuffromme.com

To: Kihyun Shin

Subject: Buy Now !

Deal of the week ! Buy now !

Rolex **w4tchs** - $100

Med1cine (any kind) - $50

also low cost **M0rgages**

available.

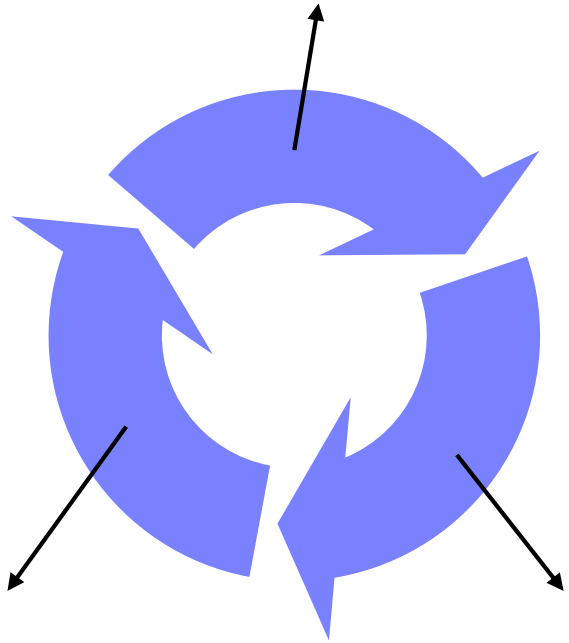
Building a spam classifier

How to try to reduce your spam classifier's error ?

- Collect more data. e. g., "Honeypot" project.
- Develop sophisticated features based on e-mail routing (from e-mail header)
- Define sophisticated features from e-mail body. (e.g., should "discounting" and "discount" be treated as the same word.
- Design algorithms to detect misspellings. (e.g., w4tches, med1cine, m0rtgage.)

Iterative loop of ML development

Choose architecture
(model, data, etc.)



Train model

Diagnostics
(bias, variance and
error analysis)

Error analysis

$m_{CV} = 500\ (or\ 5{,}000)$ **examples in cross validation set.**

Algorithm misclassifies $100\ (or\ 1{,}000)$ **of them**

Manually examine 100 **examples and categorize them**

Based on common traits.

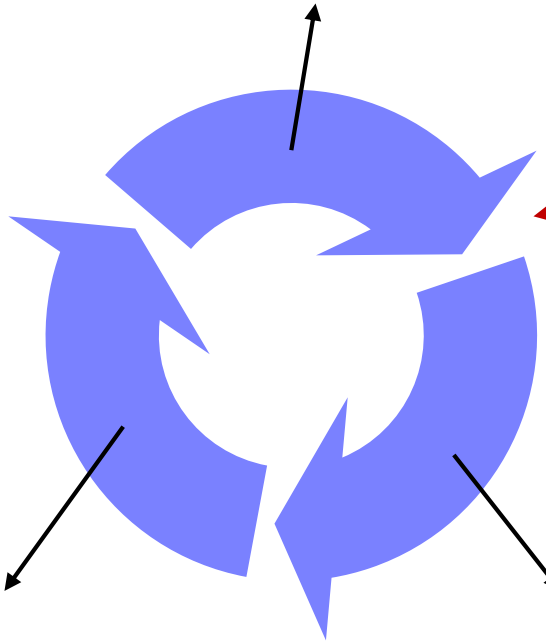
- **Pharmaceutical sales** : **21**
- **Deliberate misspellings** : **3**
- **Unusual email routing** : **7**
- **Steal passwords (phishing)** : **18**
- **Spam message in embedded image** : **5**

Building a spam classifer

How to try to reduce your spam classifier's error ?

- Collect more data. e. g., "Honeypot" project.
- Develop sophisticated features based on e-mail routing (from e-mail header)
- Define sophisticated features from e-mail body. (e.g., should "discounting" and "discount" be treated as the same word.
- Design algorithms to detect misspellings. (e.g., w4tches, med1cine, m0rtgage.)

Iterative loop of ML development



Choose architecture
(model, data, etc.)

Collect more
data for
specific target

Train model

Diagnostics
(bias, variance and
error analysis)

Adding data

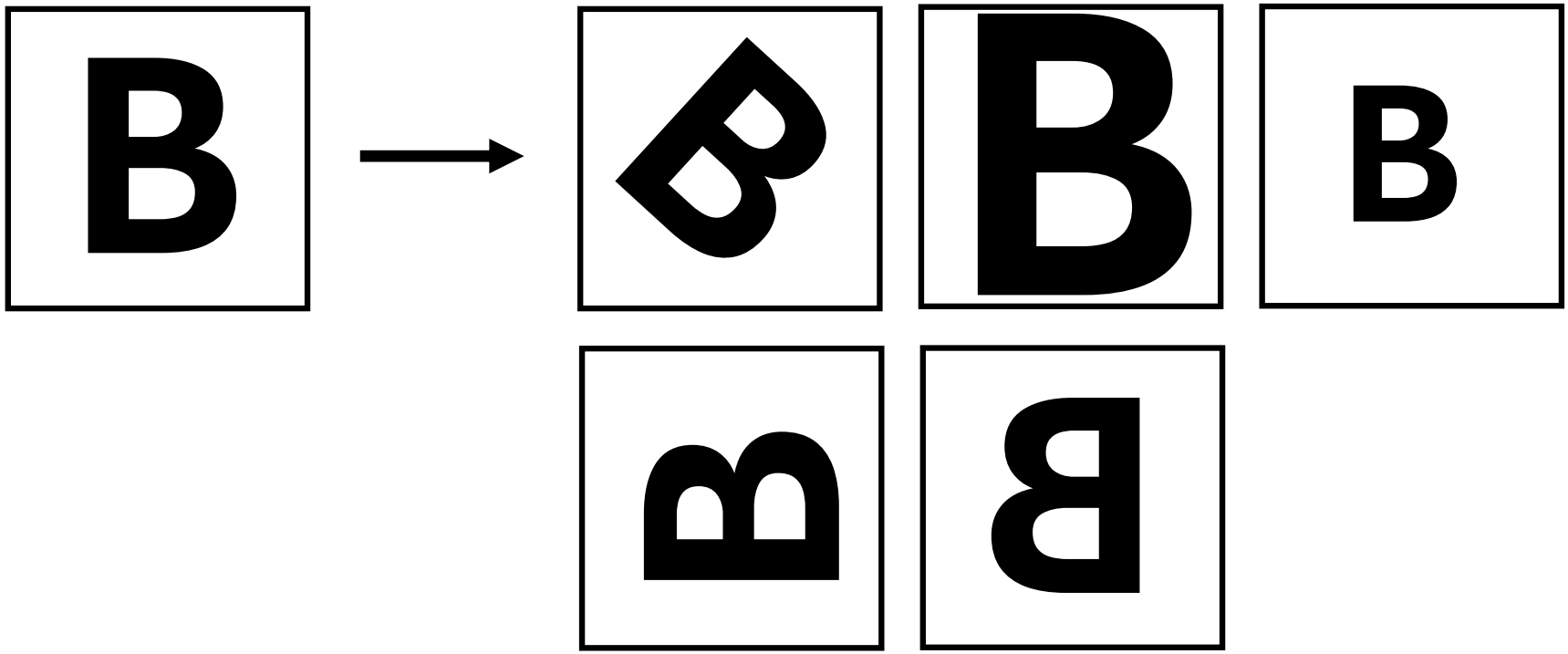
1. Add more date of everything (e.g. “honeypot” project)

2. Add more date of the types where error analysis has indicated it might help. (e.g. Go to unlabeled data and find more examples of Pharma related spam)

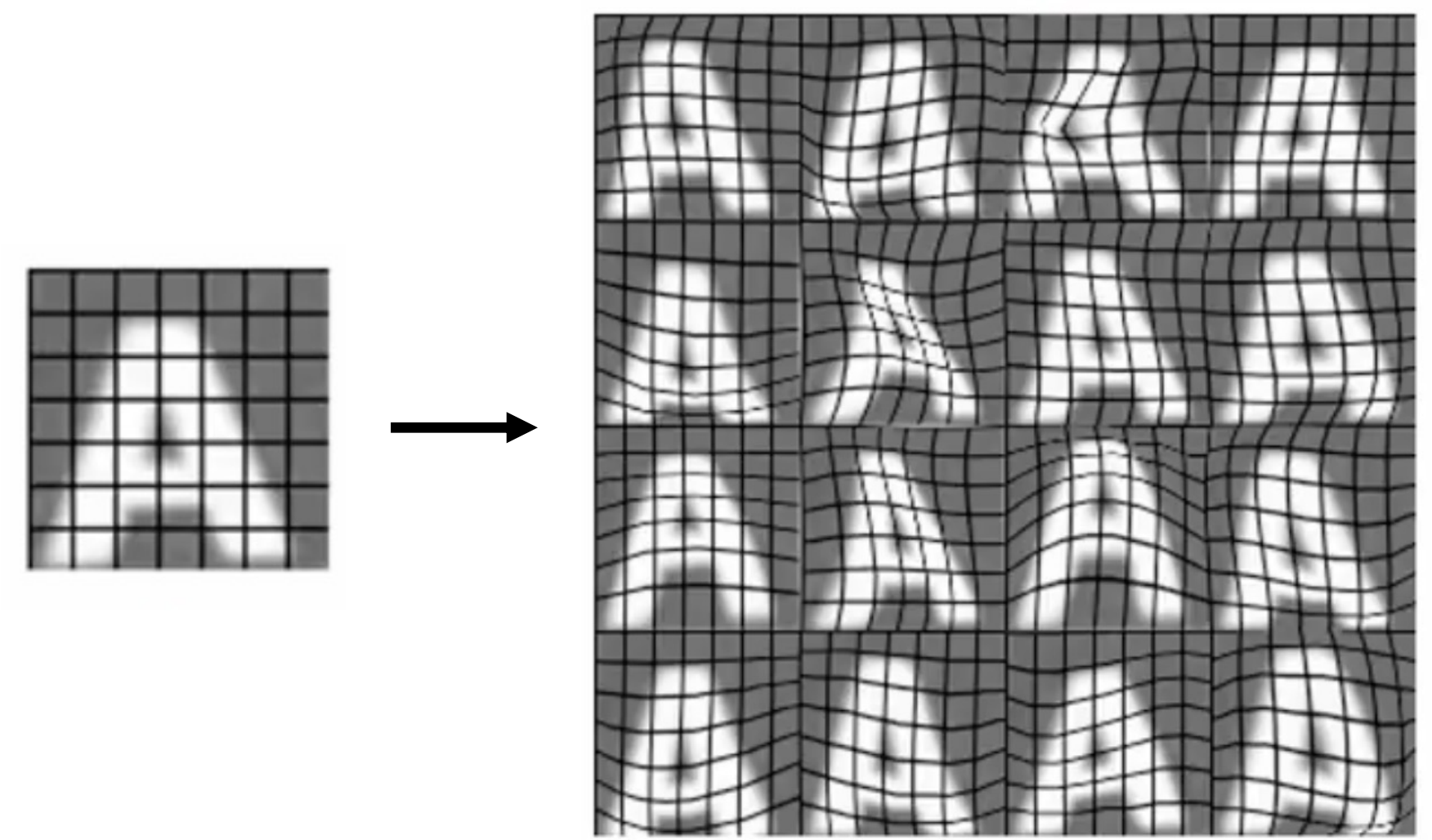
3. Beyond getting brand new training examples (x,y), another technique: Data augmentation.

Data augmentation

Augmentation: modifying an existing training example to create a new training example.



Data augmentation by introducing distortions



Data augmentation for speech

Speech recognition example

Original audio : “What is today’s weather ?”

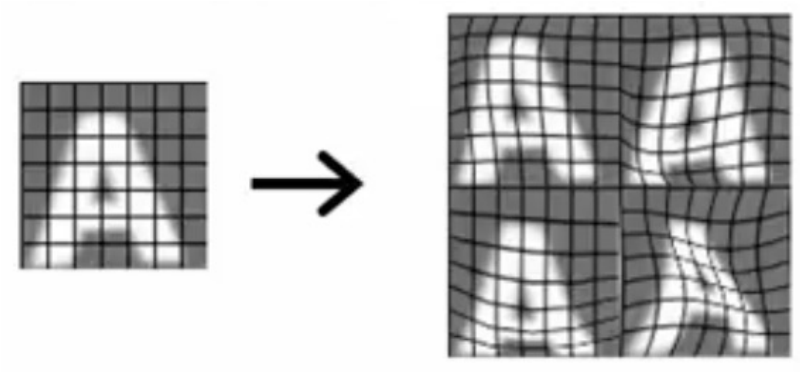
“What is today’s weather ?” + Crowd noise

“What is today’s weather ?” + Car noise

“What is today’s weather ?” + Cell phone

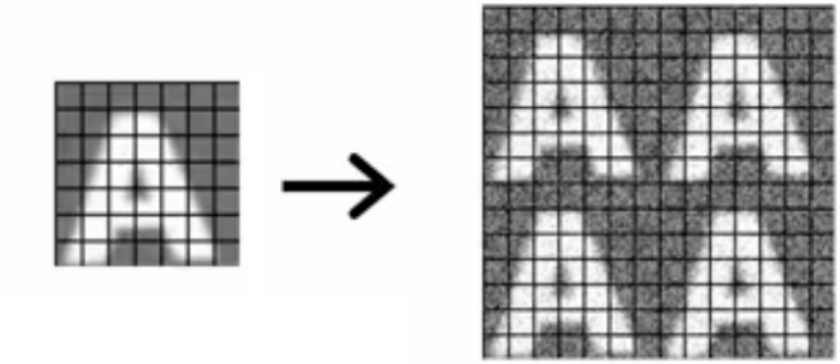
Data augmentation by introducing distortions

Distortion introduced should be representation of the type of noise/distortions in the test set.



Audio:
Background noise,
Bad cellphone connection

Usually does not help to add purely random/meaningless noise to your data.



x_i = **Intensity (brightness) of pixel**
$x_i \leftarrow x_i +$ **random noise**

Data synthesis

Synthesis: using artificial data inputs to create a new training example

Artificial data synthesis for photo OCR

Artificial data synthesis for photo OCR

Artificial data synthesis for photo OCR

Real data from OCR

Artificial data synthesis for photo OCR

Real data from OCR

Synthetic data

Engineering the data used by your system

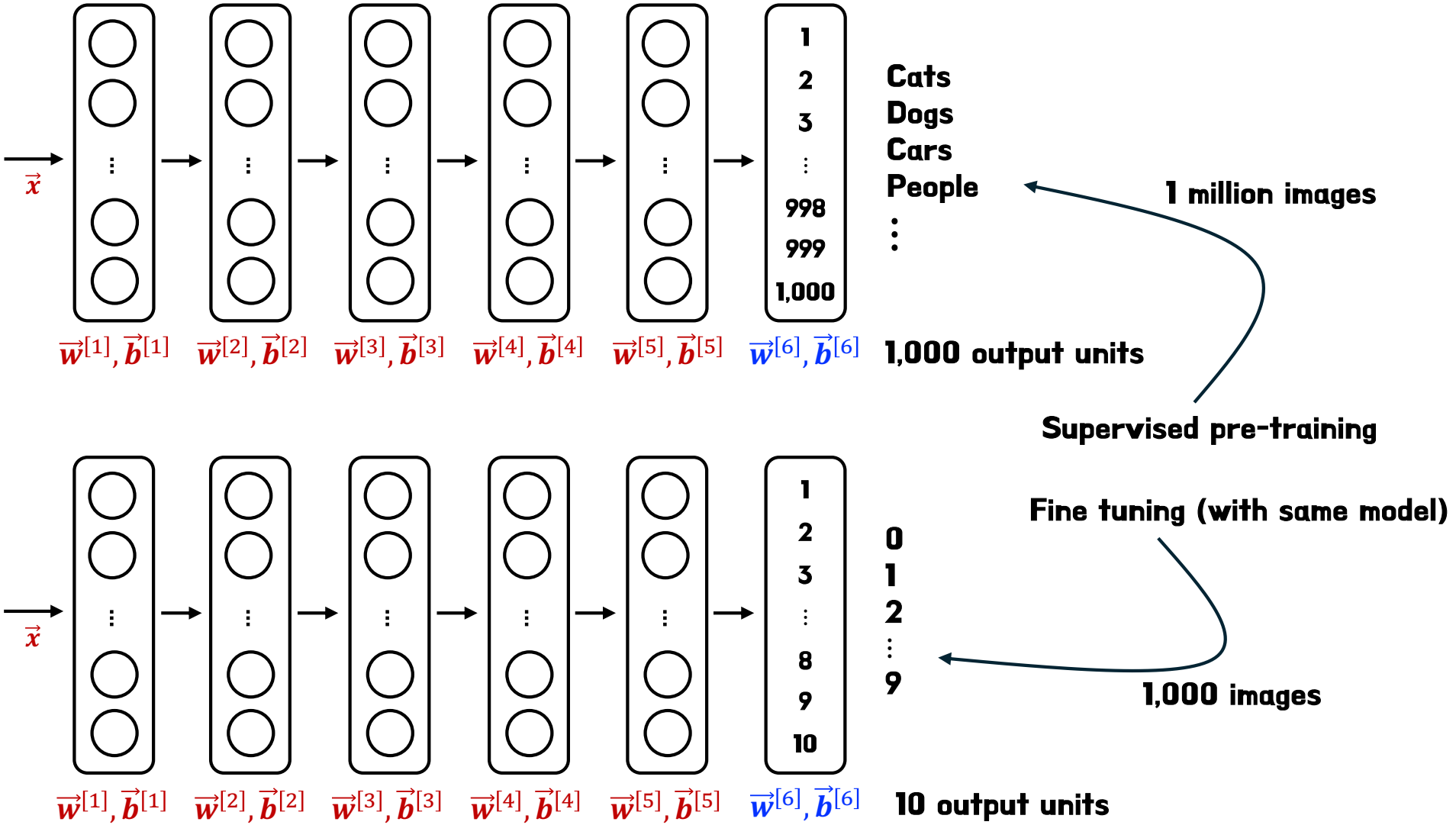
Conventional model-centric approach:

AI = Code + Data
(model/algorithm)

Data-centric approach:

AI = Code + Data
(model/algorithm)

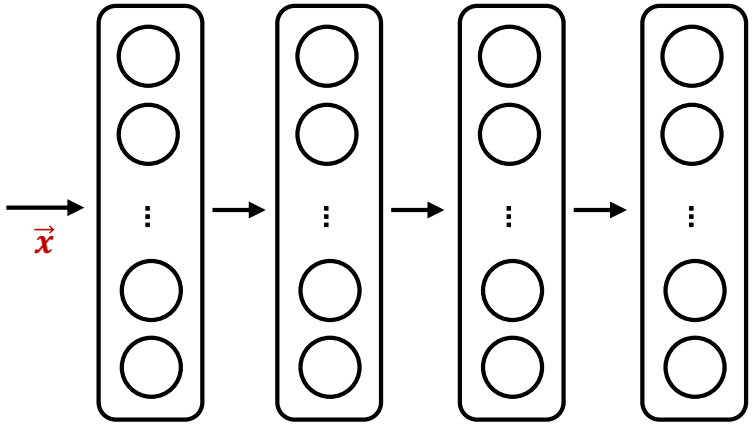
Transfer learning



Option 1: only train output layers parameters

Option 2: train all parameters

Why does transfer learning work ?

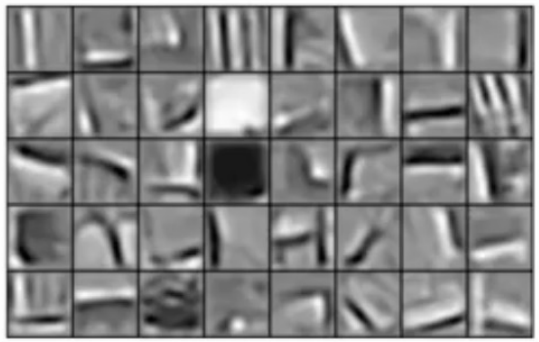


Detects edges

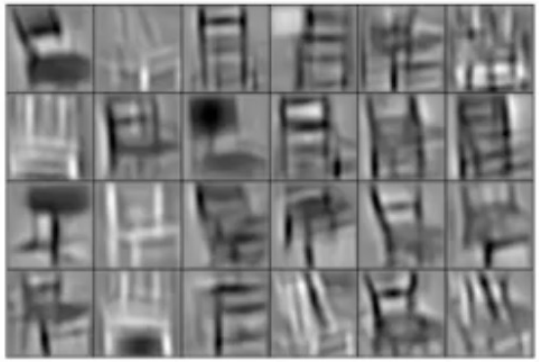
Detects corners

Detects Curves/ Basic shapes

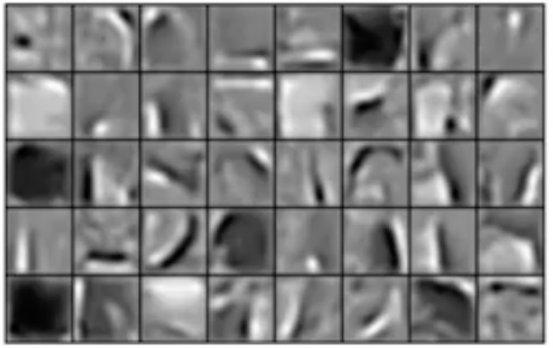
When there are same input types (text, image, audio)



Edges



Corners



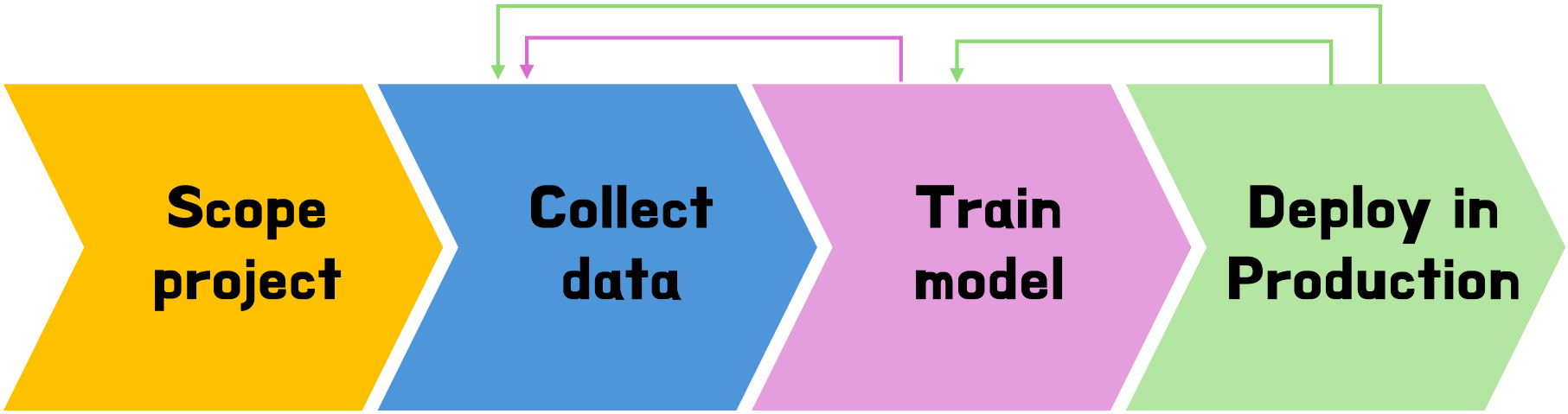
Curves/basic shapes

Transfer learning summary

1. Download neural network parameters pretrained on a large dataset with same input type (e.g., images, audio, text)

2. Further train (fine tune) the network on your own data. (e.g. universal potential, ChatGPT)

Full cycle of a machine learning project



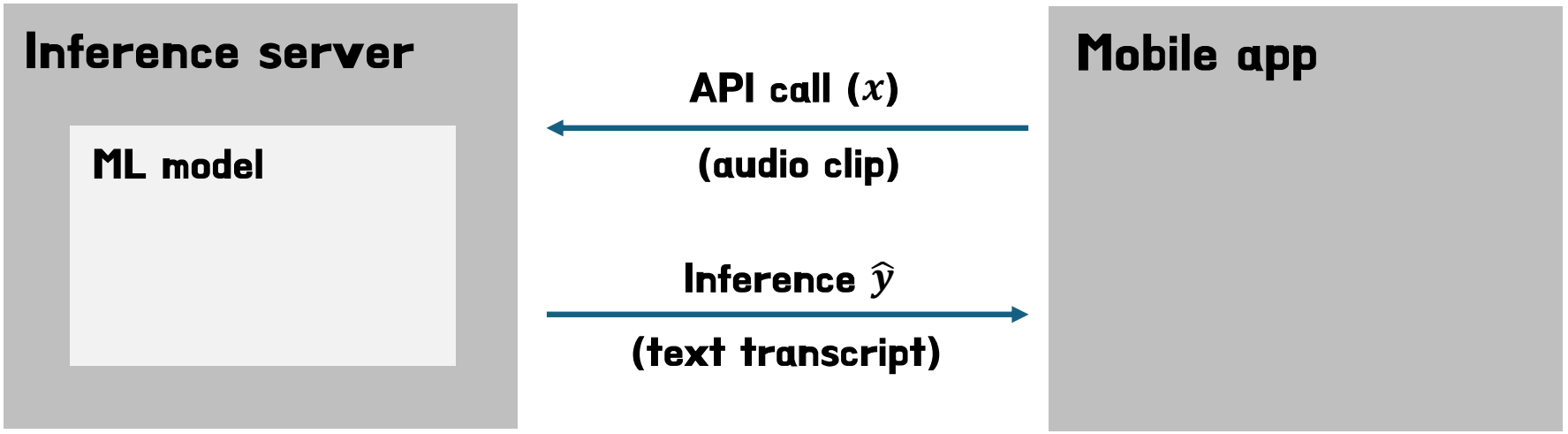
Define project

Define and Collect data

Training, Error analysis & Iterative improvement

Deploy, Monitor and Maintain system

Deployment



- Software engineering may be needed for:
  - Ensure reliable and efficient predictions
  - Scaling
  - Logging
  - System monitoring
  - Model updates

MLOps
Machine learning operations

Bias

- Hiring tool that discriminates against women.
- Facial recognition system matching dark skinned individuals to criminal mugshots.
- Biased bank loan approvals.
- Toxic effect of reinforcing negative stereotypes.

Adverse use cases

Deepfakes

Adverse use cases

Deepfakes

- Spreading toxic/incendiary speech through optimizing for engagement.
- Generating fake content for commercial or political purposes.
- Using ML to build harmful products, commit fraud etc.

Spam vs anti-spam : fraud vs anti-fraud

Guidelines

- Get a diverse team to brainstorm things that might go wrong, with emphasis on possible harm to vulnerable groups.
- Carry out literature search on standards/guidelines for your industry
- Audit systems against possible harm prior to deployment.

- Develop mitigation plan (if applicable), and after deployment, monitor for possible harm.

Skewed datasets

Rare disease classification example

Train classifier $f_{\vec{w},b}(\vec{x})$ $\begin{cases} y = 1 & if\ desease\ present \\ y = 0 & otherwise \end{cases}$

1. Find that you've got 1 % error on test set

(99 % correct diagnoses)

Error ?

Usefulness ?

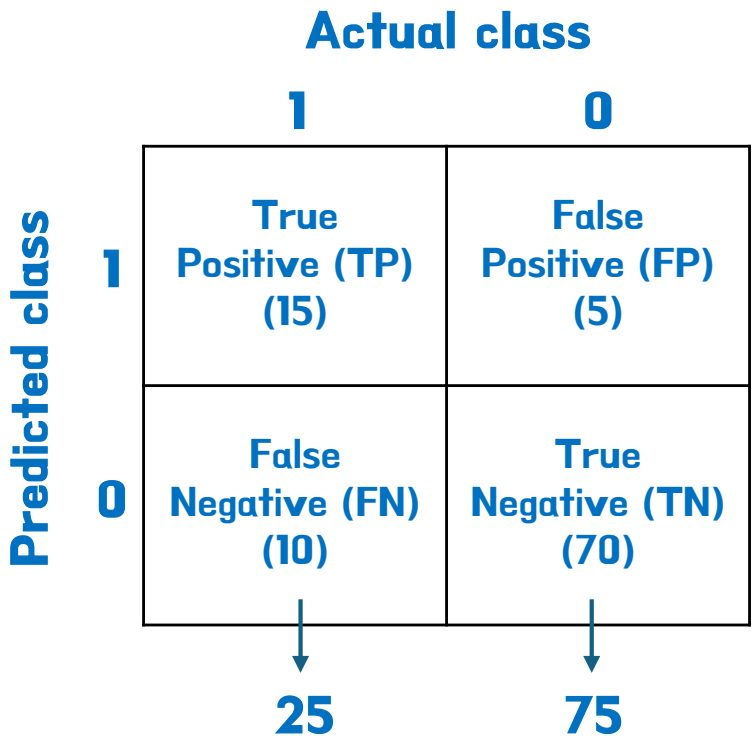
2. Let's say 0.5 % of patents have the disease

And If you made the model $'print(\text{“}y = 0\text{”})'$

(99.5 % accuracy, 0.5 % error)

Precision/recall

$y = 1$ in presence of rare class we want to detect



How about $'print(\text{“}y = 0\text{”})'$?

Precision:
(of all patients where we predicted $y = 1$, what fraction actually have the rare disease?)

$$\frac{True\ positive}{\#\ predicted\ positive} = \frac{TP}{TP + FP} = \frac{15}{15 + 5} = 0.75$$

Recall:
(of all patients that actually have the rare disease, what fraction did we correctly detect as having it?)

$$\frac{True\ positive}{\#\ actual\ positive} = \frac{TP}{TP + FN} = \frac{15}{15 + 10} = 0.6$$

Trading off precision and recall

Logistic regression: $0 < f_{\vec{w},b}(\vec{x}) < 1$

Predict 1 **if** $f_{\vec{w},b}(\vec{x}) \geq 0.5$

Predict 0 **if** $f_{\vec{w},b}(\vec{x}) < 0.5$

$$precision = \frac{True\ positive}{\#\ predicted\ positive}$$

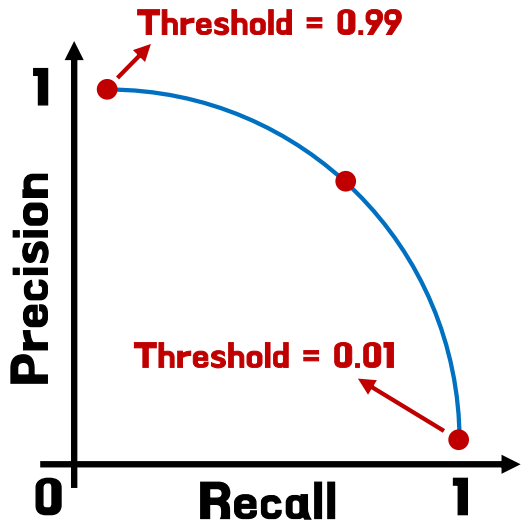
$$recall = \frac{True\ positive}{\#\ actual\ positive}$$

Suppose we want to predict $y = 1$ **(rare disease) only if very confident.**

Higher precision, lower recall

Suppose we want to avoid missing too many case of rare disease (when in doubt predict $y = 1$**)**

Lower precision, higher recall



More generally predict 1 **if:** $f_{\vec{w},b}(\vec{x}) \geq threshold$

F1 score

How to compare precision/recall numbers ?

	Precision (P)	Recall (R)	Average	F_1 score
Algorithm 1	0.5	0.4	0.45	0.444
Algorithm 2	0.7	0.1	0.4	0.175
Algorithm 3	0.02	1.0	0.501	0.0392

$'print(\text{“}y = 1\text{”})'$

$$\text{Average} = \frac{P+R}{2}$$

$$F_1 \text{ score} = \frac{1}{\frac{1}{2}\left(\frac{1}{P}+\frac{1}{R}\right)} = 2\frac{PR}{P+R}$$

Harmonic mean
: emphasizing smaller value