Framework of ML

Training data: \[ \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots, (x^N, \hat{y}^N)\} \]

Testing data: \[ \{x^{N+1}, x^{N+2}, \ldots, x^{N+M}\} \]

---

**Speech Recognition**

\[ \begin{align*} 
    x: & \quad \text{声波} \\
    \hat{y}: & \quad \text{音素} 
\end{align*} \]

**Image Recognition**

\[ \begin{align*} 
    x: & \quad \text{画像} \\
    \hat{y}: & \quad \text{スープ} 
\end{align*} \]

---

**Speaker Recognition**

\[ \begin{align*} 
    x: & \quad \text{音声} \\
    \hat{y}: & \quad \text{John} \\
    & \text{(speaker)}
\end{align*} \]

**Machine Translation**

\[ \begin{align*} 
    x: & \quad \text{痛みを知れ} \\
    \hat{y}: & \quad \text{了解痛苦吧}
\end{align*} \]
Framework of ML

Training data: \( \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots, (x^N, \hat{y}^N)\} \)

Training:

Step 1: function with unknown \( y = f_\theta(x) \)

Step 2: define loss from training data \( L(\theta) \)

\( \theta^* = \arg \min_{\theta} L \)

Step 3: optimization

Testing data: \( \{x^{N+1}, x^{N+2}, \ldots, x^{N+M}\} \)

Use \( y = f_{\theta^*}(x) \) to label the testing data

\( \{y^{N+1}, y^{N+2}, \ldots, y^{N+M}\} \) Upload to Kaggle
**General Guide**

- **loss on training data**
  - large
  - optimize
    - make your model complex
  - small
    - overfitting
      - make your model simpler
    - mismatch
      - not in HWs except HW 11
- **loss on testing data**
  - large
  - not in HWs
  - small

- **trade-off**
  - more training data (not in HWs)
  - data augmentation
  - make your model simpler

- Split your training data into training set and validation set for model selection
### Model Bias

- The model is too simple.

\[ f_{\theta_1}(x) \quad y = f_{\theta}(x) \]

find a needle in a haystack ...

... but there is no needle
depending on the error function, the model can be too small ...

- Solution: redesign your model to make it more flexible

\[ y = b + w x_1 \quad \text{More features} \quad y = b + \sum_{j=1}^{56} w_j x_j \]

Deep Learning
(more neurons, layers)

\[ y = b + \sum_i c_i \text{ sigmoid} \left( b_i + \sum_j w_{ij} x_j \right) \]
General Guide

- **loss on training data**
  - **large**
  - **small**
    - model bias
      - make your model complex
    - optimization
      - Next Lecture
      - overfitting
        - more training data (not in HWs)
        - data augmentation
      - mismatch
        - Not in HWs, except HW 11
        - trade-off
          - make your model simpler
          - Split your training data into training set and validation set for model selection
Optimization Issue

• Large loss not always imply model bias. There is another possibility ...

\[ L(\theta^*) \text{ large} \]

A needle is in a haystack ... 
... Just cannot find it.

\[ y = f_\theta(x) \]

\[ f_{\theta^1}(x), f_{\theta^2}(x), f_{\theta^*}(x) \]
Model Bias

find a needle in a haystack ...
... but there is no needle

Which one???

Optimization Issue

A needle is in a haystack ...
... Just cannot find it.
Model Bias v.s. Optimization Issue

- Gaining the insights from comparison

Ref: http://arxiv.org/abs/1512.03385
Optimization Issue

• Gaining the insights from comparison
• Start from shallower networks (or other models), which are easier to optimize.
• If deeper networks do not obtain smaller loss on training data, then there is optimization issue.

<table>
<thead>
<tr>
<th>1 layer</th>
<th>2 layer</th>
<th>3 layer</th>
<th>4 layer</th>
<th>5 layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017 – 2020</td>
<td>0.28k</td>
<td>0.18k</td>
<td>0.14k</td>
<td>0.10k</td>
</tr>
</tbody>
</table>

• Solution: More powerful optimization technology (next lecture)
Split your training data into training set and validation set for model selection.

model bias

loss on training data

optimization

loss on testing data

overfitting

mismatch

Trade-off:

- make your model simpler
- more training data (not in HWs)
- data augmentation

General Guide

Not in HWs, except HW 11

Next Lecture
Split your training data into training set and validation set for model selection.
Overfitting

- Small loss on training data, large loss on testing data. Why?

**An extreme example**

Training data: \( \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), ..., (x^N, \hat{y}^N)\} \)

\[
f(x) = \begin{cases} 
\hat{y}^i & \exists x^i = x \\
\text{random} & \text{otherwise}
\end{cases}
\]

Less than useless ...

This function obtains zero training loss, but large testing loss.
Overfitting

Real data distribution (not observable)

- Training data
- Testing data

Flexible model

“freestyle”

Large loss
Overfitting

More training data (cannot do it in HWs)

Data augmentation (you can do that in HWs)
Overfitting

Real data distribution
(not observable)

- Training data
- Testing data

\[ y = a + bx + cx^2 \]
Overfitting

Real data distribution (not observable)

Training data
Testing data

$y = a + bx + cx^2$
Overfitting

- Less parameters, sharing parameters
- Less features
- Early stopping
- Regularization
- Dropout

\[ y = a + bx + cx^2 \]
Overfitting

Real data distribution (not observable)
- Training data
- Testing data

$y = a + bx$

Back to model bias...
Bias-Complexity Trade-off

Model becomes complex (e.g. more features, more parameters)

Training loss

Testing loss

select this one
Homework

The extreme example again

\[ f_k(x) = \begin{cases} \hat{y}^i & \exists x^i = x \\ \text{random} & \text{otherwise} \end{cases} \quad k: 1 - 10000000000000000000 \]

It is possible that \( f_{56789}(x) \) happens to get good performance on public testing set.

So you select \( f_{56789}(x) \) ...... Random on private testing set
Homework

What will happen?
http://www.chioka.in/how-to-select-your-final-models-in-a-kaggle-competitio/

This explains why machine usually beats human on benchmark corpora. 😊

Top 10 in public leaderboard

Why?

May be poor ...

Pick this one!
Cross Validation

How to split?

Training Set

Validation set

Training Set

Public

Testing Set

Private

Testing Set

Using the results of public testing data to select your model
You are making public set better than private set.

Model 1 → mse = 0.9
Model 2 → mse = 0.7
Model 3 → mse = 0.5

mse > 0.5 → mse > 0.5

Not recommend
N-fold Cross Validation

Training Set
- Train
- Val
- Train
- Val
- Train
- Train

Model 1
- mse = 0.2
- mse = 0.4
- mse = 0.3
Avg mse = 0.3

Model 2
- mse = 0.4
- mse = 0.5
- mse = 0.5
Avg mse = 0.5

Model 3
- mse = 0.4
- mse = 0.6
- mse = 0.3
Avg mse = 0.4

Testing Set
- public
- private
General Guide

loss on training data
  large
  model bias
    - make your model complex
  optimization
    - Next Lecture
  small

loss on testing data
  large
  overfitting
    - more training data (not in HWs)
    - data augmentation
  small
  mismatch
    - Not in HWs, except HW 11

trade-off
- Split your training data into training set and validation set for model selection
- make your model simpler
Let’s predict no. of views of 2/26!

Red: real, Blue: predicted

2/26

e = 2.58k
**General Guide**

- **loss on training data**
  - large
  - small
    - **model bias**
    - **optimization**
    - **overfitting**
      - make your model complex
      - **Next Lecture**
        - more training data (not in HWs)
        - data augmentation
        - make your model simpler
      - mismatch
        - Not in HWs, except HW 11
      - trade-off
        - Split your training data into training set and validation set for model selection
Mismatch

- Your training and testing data have different distributions. Be aware of how data is generated.

Most HWs do not have this problem, except HW11

**Training Data**

- horse
- bed
- clock
- apple
- cat
- plane
- television
- dog
- dolphin
- spider

Simply increasing the training data will not help.

**Testing Data**

- horse
- bed
- clock
- apple
- cat
- plane
- television
- dog
- dolphin
- spider
General Guide

loss on training data

large
model bias
make your model complex

optimization
Next Lecture

loss on testing data

large
overfitting
more training data (not in HWs)
data augmentation
make your model simpler

small
mismatch
Not in HWs, except HW 11

trade-off
Split your training data into training set and validation set for model selection