# Machine Learning HW5

ML TAs
ntu-ml-2022-spring-ta@googlegroups.com

# Outline

1. Machine translation
2. Workflow
3. Training tips
4. Requiements
5. Report
6. JudgeBoi Guide
7. Regulation and Grading policy

# Machine Translation

# Machine Translation

In this homework, we’ll translate English to Traditional Chinese

e.g.

- Thank you so much, Chris. -> 非常謝謝你，克里斯。

Since sentences are with different length in different languages, the seq2seq framework is applied to this task.

# Training datasets

- Paired data
  - TED2020: TED talks with transcripts translated by a global community of volunteers to more than 100 language
  - We will use (en, zh-tw) aligned pairs
- Monolingual data
  - More TED talks in traditional Chinese

# Evaluation

source: Cats are so cute

target: 貓咪真可愛

output: 貓好可愛

## BLEU

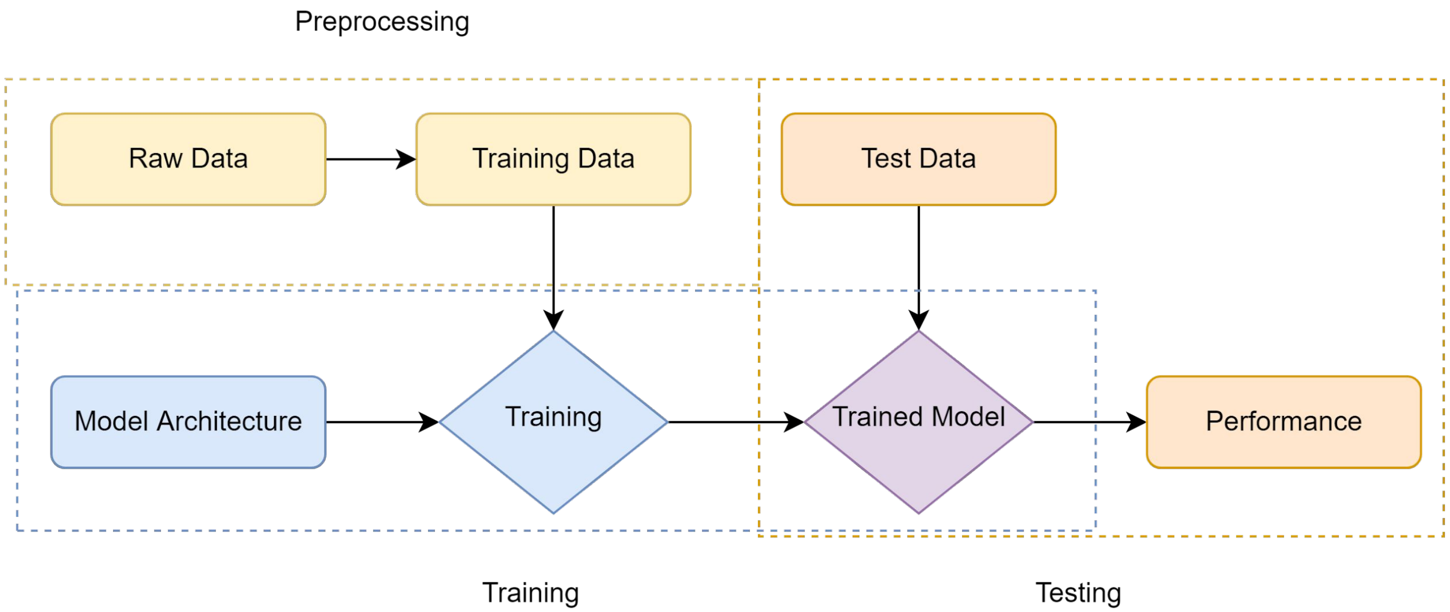
- Modified n-gram precision (n = 1~4)
- Brevity penalty: penalizes short hypotheses

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

  - c is the hypothesis length, r is the rerferencwe length
- The BLEU socre is the geometric mean of n-gram precision, miltiplied by brevity penalty

# Workflow

# Workflow


Preprocessing
Raw Data
Training Data
Test Data
Model Architecture
Training
Trained Model
Performance
Training
Testing

# Workflow

1. Preprocessing
   a. download raw data
   b. clean and normalize
   c. remove bad data (too long/short)
   d. tokenization
2. Training
   a. initialize a model
   b. train it with training data
3. Testing
   a. generate translation of test data
   b. evaluate the performance

# Training tips

# Training tips

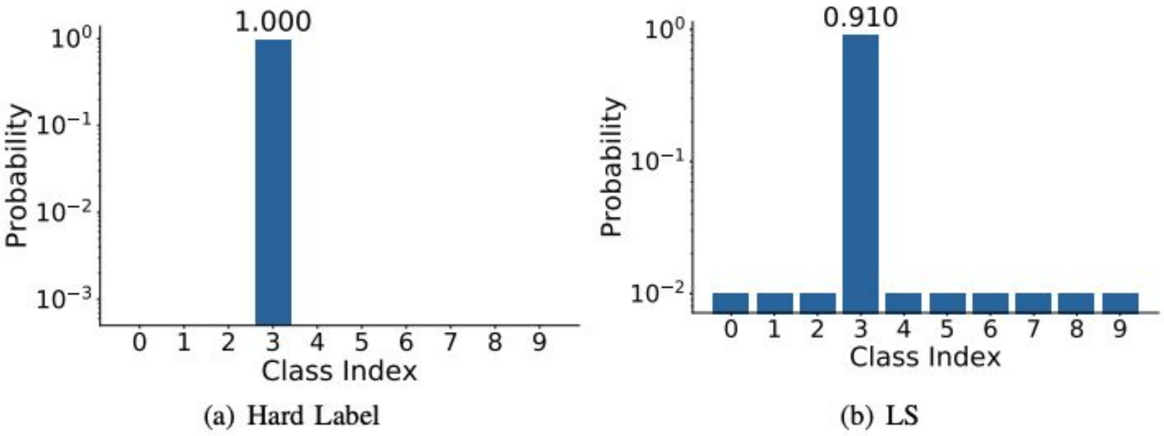
- Tokenize data with sub-word units
- Label smoothing regularization
- Learning rate scheduling
- Back-translation

# Tokenize

- Tokenize data with subword units
  - Reduce the vocabulary size
  - Alleviate the open vocabulary problem
  - example
    - _put _your s el ve s _in _my _po s ition _.
    - Put yourselves in my position.

# Label smoothing

- Label smoothing regularization
  - When calculating loss, reserve some probability for incorrect labels
  - Avoids overfitting

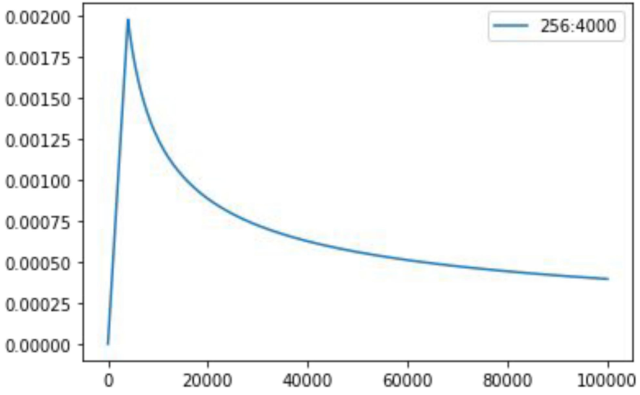


(a) Hard Label

(b) LS

# Leaning rate scheduling

- Learning rate scheduling
  - Increasing the learning rate linearly for the first warmup_steps training steps, and decreasing it thereafter proportionally to the inverse square root of the step number.
  - Stabilizing training for transformers in early stages



# Back translation

Using monolingual data for creating synthetic translation data

1. Train a translation system in the **opposite direction**
2. Collect monolingual data in target side and apply machine traslation
3. Use translated and original monolingual data as  additional parallel data to train stronger translation systems

# Back translation

Some points to note about back-translation

1. Monolingual data should be in the same domain as the parallel corpus
2. The performance of the backward model is critical
3. Increase model capacity since the data amount is increased

# Requirements

# Baselines

| Baseline | Public score | Estimated time(kaggle) |
|---|---|---|
| Simple | 14.58 | 1 hour |
| Medium | 18.04 | 1 hour 40 mins |
| Strong | 25.20 | ~3 hours |
| Boss | 29.13 | > 12hours |

# Baseline Guide

- Simple Baseline: Train a simple RNN seq2seq to acheive translation
- Medium Baseline: Add learning rate scheduler and train longer
- Strong Baseline: Switch to Transformer and tuning hyperparameter
- Boss Baseline: Apply back-translation

# Simple Baseline

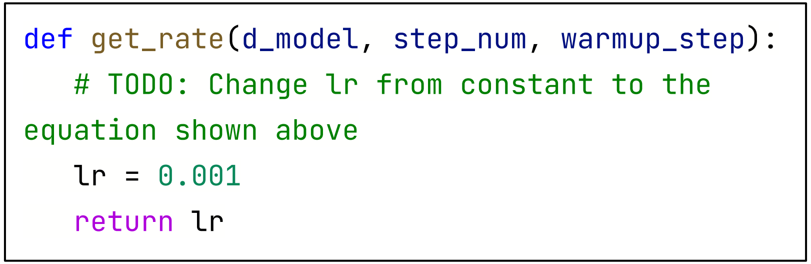
Train a simple RNN seq2seq to acheive translation

- Running the sample code should pass the baseline

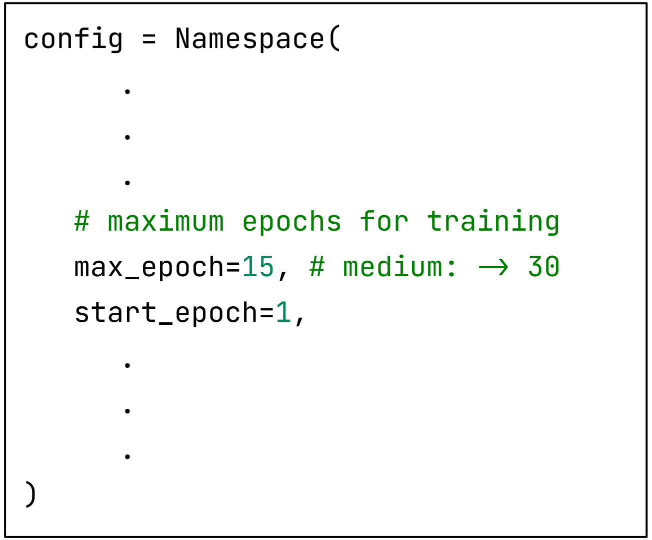
# Medium Baseline

Add learning rate scheduler and train longer

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$



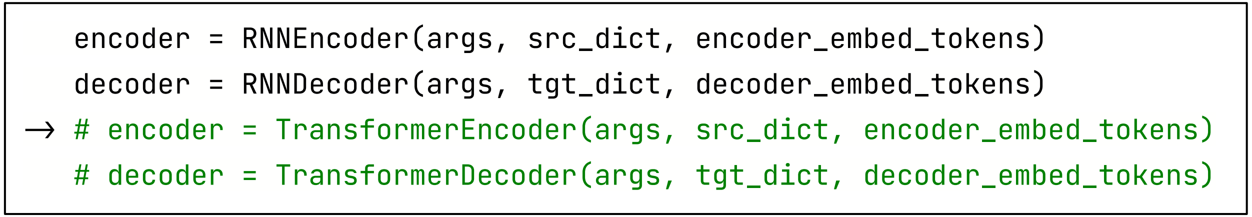
```python
def get_rate(d_model, step_num, warmup_step):
    # TODO: Change lr from constant to the
equation shown above
    lr = 0.001
    return lr
```



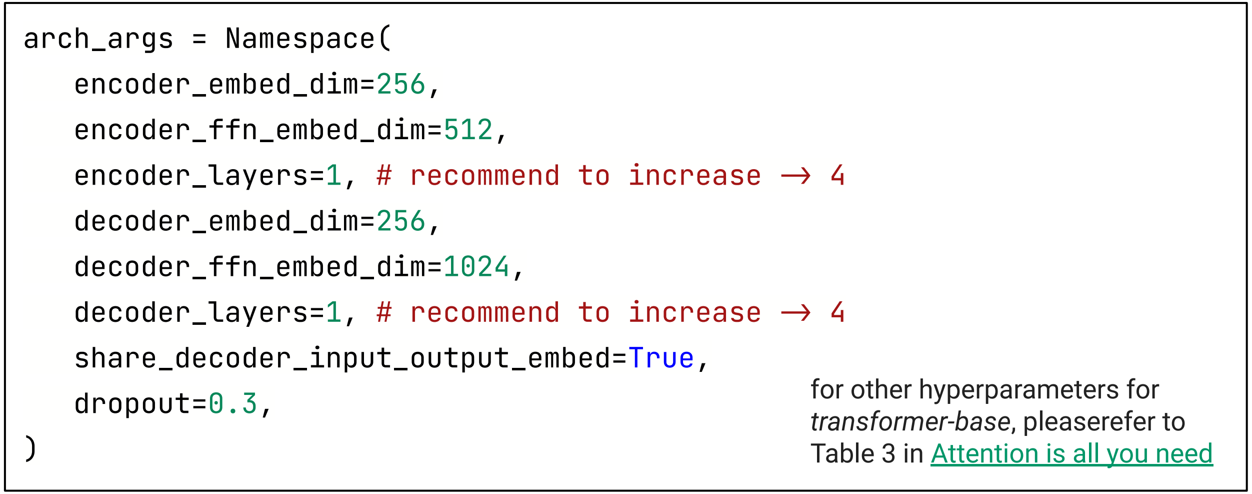
```python
config = Namespace(
        .
        .
        .
    # maximum epochs for training
    max_epoch=15, # medium: → 30
    start_epoch=1,
        .
        .
        .
)
```

# Strong Baseline

Switch to Transformer and tuning hyperparameter



```
    encoder = RNNEncoder(args, src_dict, encoder_embed_tokens)
    decoder = RNNDecoder(args, tgt_dict, decoder_embed_tokens)
 →  # encoder = TransformerEncoder(args, src_dict, encoder_embed_tokens)
    # decoder = TransformerDecoder(args, tgt_dict, decoder_embed_tokens)
```



```
arch_args = Namespace(
    encoder_embed_dim=256,
    encoder_ffn_embed_dim=512,
    encoder_layers=1, # recommend to increase → 4
    decoder_embed_dim=256,
    decoder_ffn_embed_dim=1024,
    decoder_layers=1, # recommend to increase → 4
    share_decoder_input_output_embed=True,
    dropout=0.3,
)
```

for other hyperparameters for *transformer-base*, pleaserefer to Table 3 in Attention is all you need

# Boss Baseline

Apply back-translation

1. Train a backward model by switching languages

   ```
   source_lang = "zh",
   target_lang = "en",
   ```

2. Translate monolingual data with backward model to obtain synthetic data
   1. Complete TODOs in the sample code
   2. All the TODOs can be completed by using commands from earlier cells
3. Train a stronger forward model with the new data
   1. If done correctly, ~30 epochs on new data should pass the baseline

# Links

Colab sample code

Kaggle sample code

JudgeBoi

# Report

# Report Overview

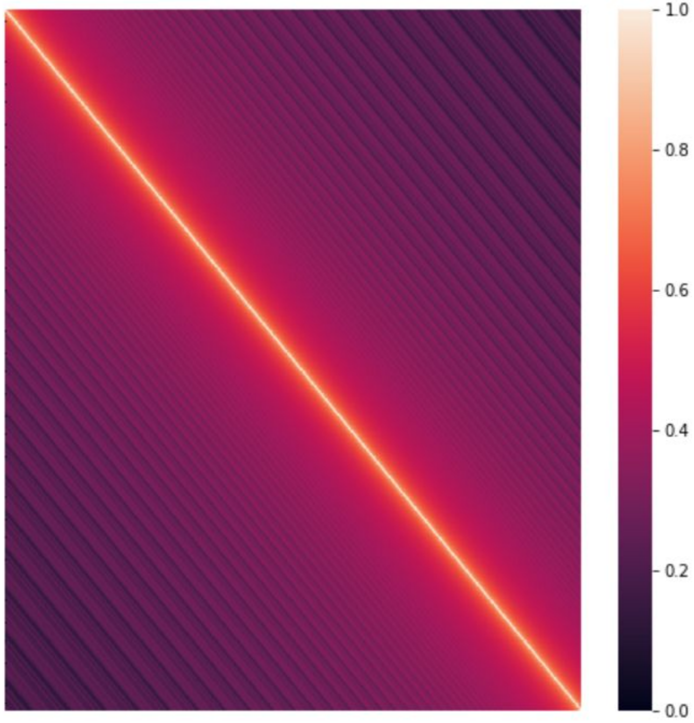
- Problem 1
  - Visualize the similarity between different pairs of positional embedding and briefly explain the result.
- Problem 2
  - Clip gradient norm and visualize the changes of gradient norm in different steps. Circle two places with gradient explosion.

# Problem 1: Visualize Positional Embedding

Given a (N x D) positional embeding lookup table, you aim to get a (N x N) “similarity matrix” by calculating similarity between different pairs of embeddings in the table.

You need to **visualize the similarity matrix and briefly explain the result**.

In this problem, we focus on the positional embeddings of the **decoder**



# Problem 1: Similarity Matrix

|  | p1 | p2 | p3 | p4 | p5 |
|---|---|---|---|---|---|
| p1 | 1 | 0.8 | 0.6 | 0.4 | 0.3 |
| p2 | 0.8 | 1 | 0.8 | 0.6 | 0.4 |
| p3 | 0.6 | 0.8 | 1 | 0.8 | 0.6 |
| p4 | 0.4 | 0.6 | 0.8 | 1 | 0.8 |
| p5 | 0.3 | 0.4 | 0.6 | 0.8 | 1 |

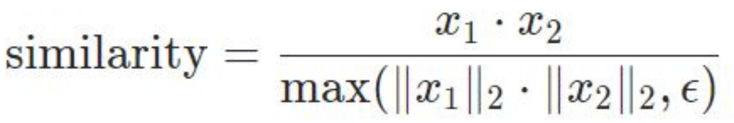
In the sence of encoding positional information, **we expect that the similarity between the embedding of close positions is stronger**.

# Problem 1: Cosine Similarity

We recommend you to measure the similarity between two vectors by cosine similarity.

There is a pytorch implementation of cosine similarity. Check more detail in the following link.

https://pytorch.org/docs/stable/generated/torch.nn.functional.cosine_similarity.html



$$\text{similarity} = \frac{x_1 \cdot x_2}{\max(\|x_1\|_2 \cdot \|x_2\|_2, \epsilon)}$$

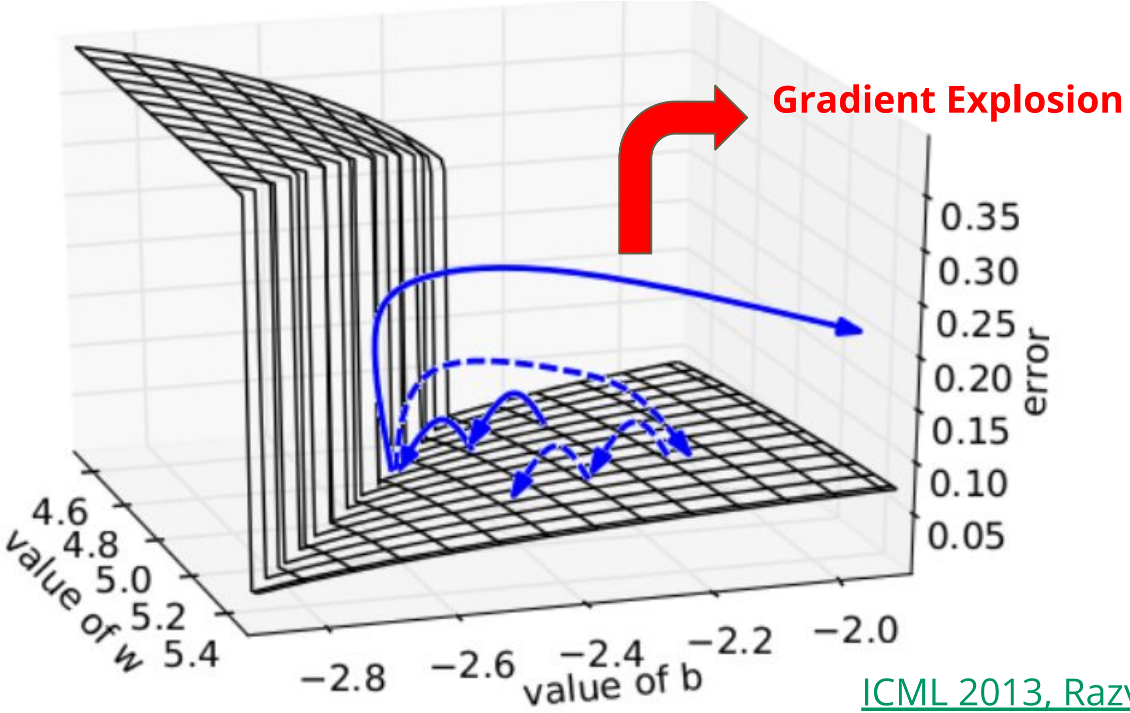
# Problem 1: Tips and Hint

You could get the positional embeddings of decoder by following codes

```
pos_emb = model.decoder.embed_positions.weights.cpu().detach()
```

```
torch.Size( [1026, 256] )
```

# Problem 2: Gradient Explosion



宏毅老師講解: ML2017 - RNN

ICML 2013, Razvan Pascanu

# Problem 2: Clipping Gradient Norm

1. Set up a maximum norm value *max_norm*
2. Collecting the gradient of each parameters to be a vector. Calculate the **p-norm of the vector** to be *Lnorm*
3. If *Lnorm <= max_norm*, do nothing. Otherwise calculate the scale factor *scale_factor = max_norm / Lnorm* and multiply each gradient by the scale factor.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\|\mathbf{x}\|_p := \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p} .$$

p-norm

$$\|\boldsymbol{x}\|_2 := \sqrt{x_1^2 + \cdots + x_n^2}.$$

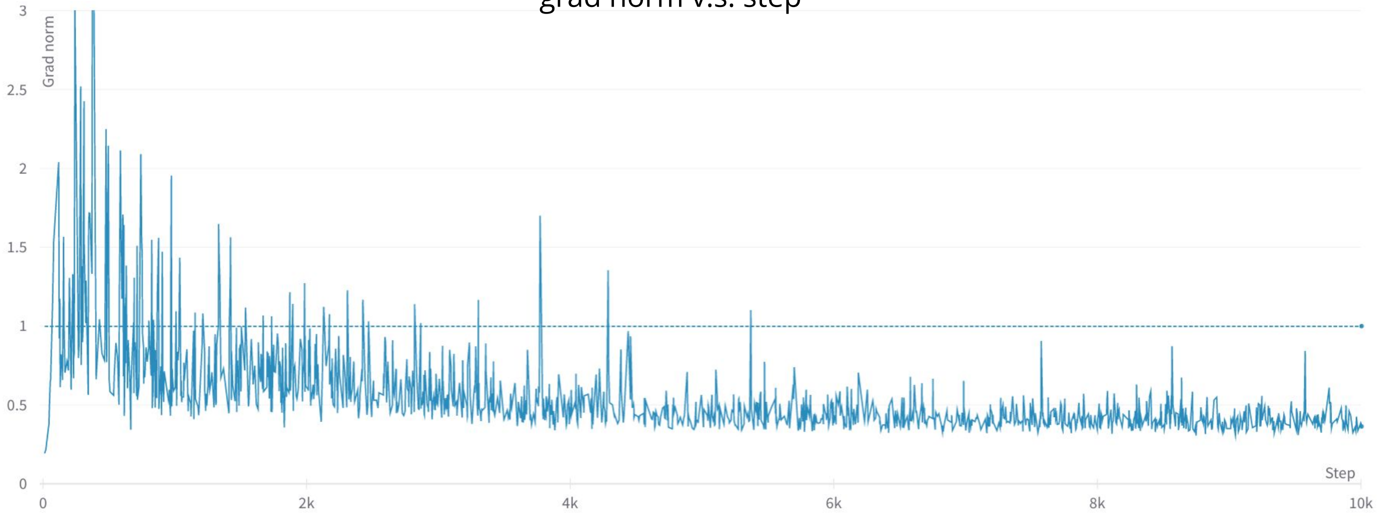
2-norm

# Problem 2: Visualize Gradient Norm

Step1: Apply clips gradient norm and set max_norm = 1.0 .

Step2: Make a plot of “gradient norm v.s step”.

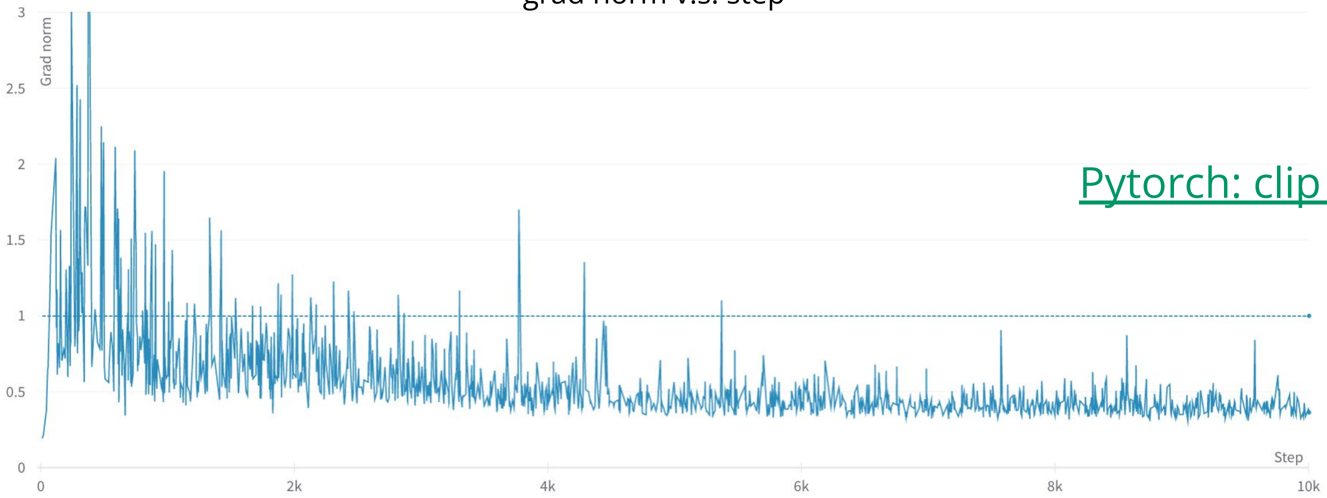
grad norm v.s. step



# Problem 2: Visualize Gradient Norm

Step3: Circle two places with gradient explosion (where the clip_grad_norm function take effect)

grad norm v.s. step

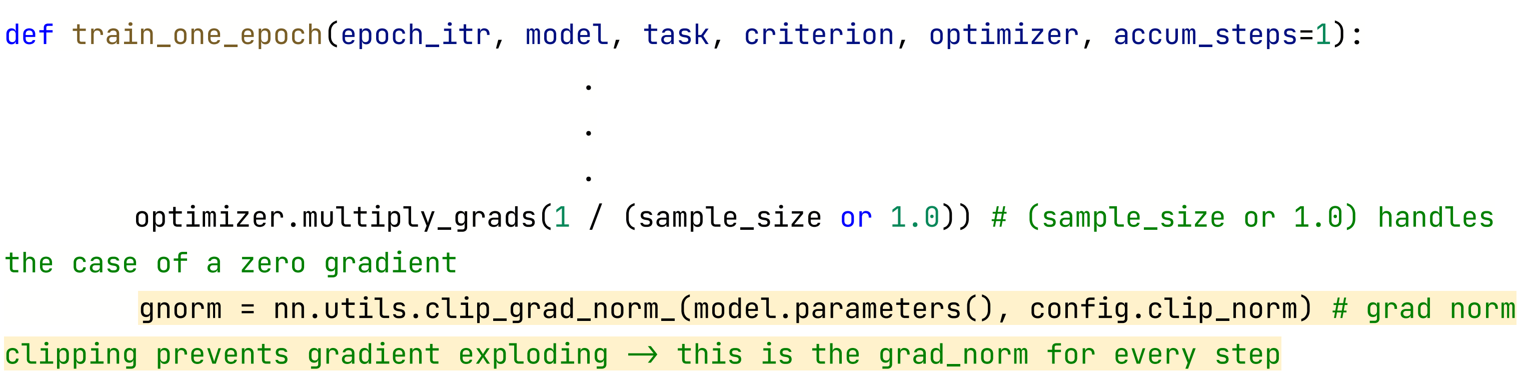


Pytorch: clip_grad_norm_()

# Problem 2: Overview

In this problem, you need to do

1. Plot the grad_norm



```
def train_one_epoch(epoch_itr, model, task, criterion, optimizer, accum_steps=1):
                                              .
                                              .
                                              .
        optimizer.multiply_grads(1 / (sample_size or 1.0)) # (sample_size or 1.0) handles
the case of a zero gradient
        gnorm = nn.utils.clip_grad_norm_(model.parameters(), config.clip_norm) # grad norm
clipping prevents gradient exploding → this is the grad_norm for every step
```

2. Circle two place with gradient explosion (if there is gradient explosion)

# JudgeBoi Guide

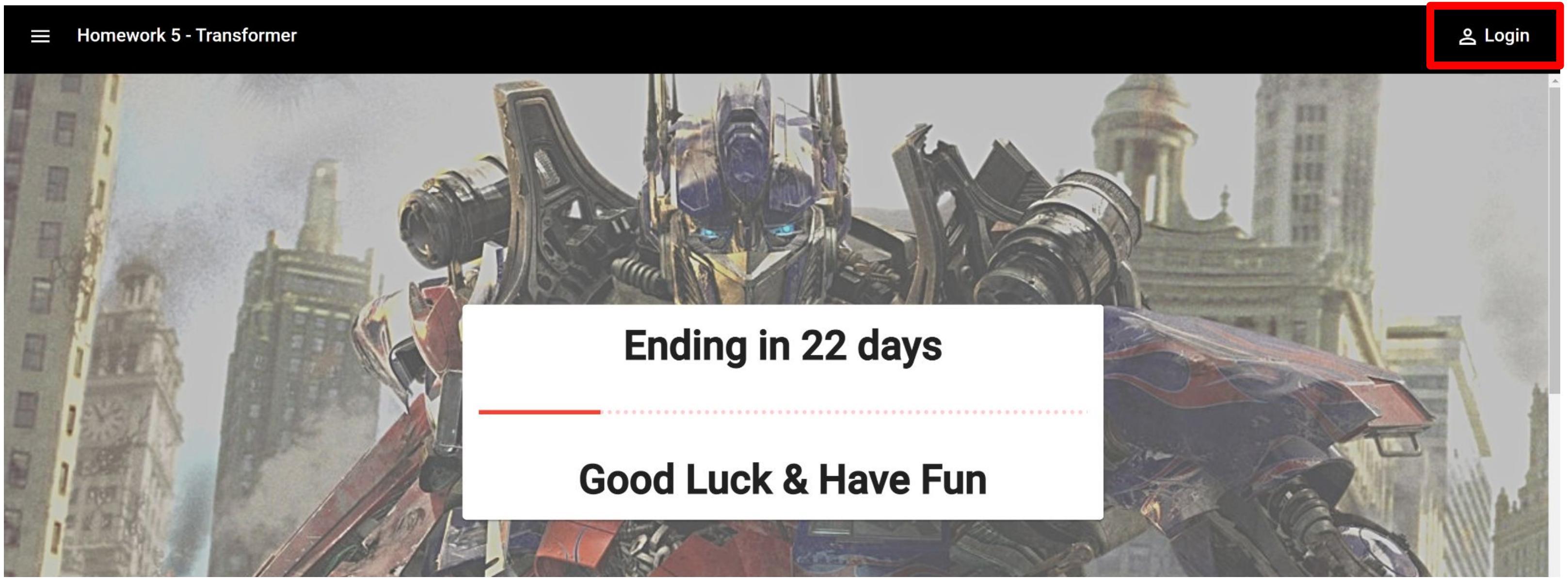
# Previously… Github Account Survey

We have kindly requested everyone to report your github username and ID.

IMPORTANT: You must take this survey in order to submit to JudgeBoi server.

# Step 1: Register for Submission

Go to JudgeBoi to login.

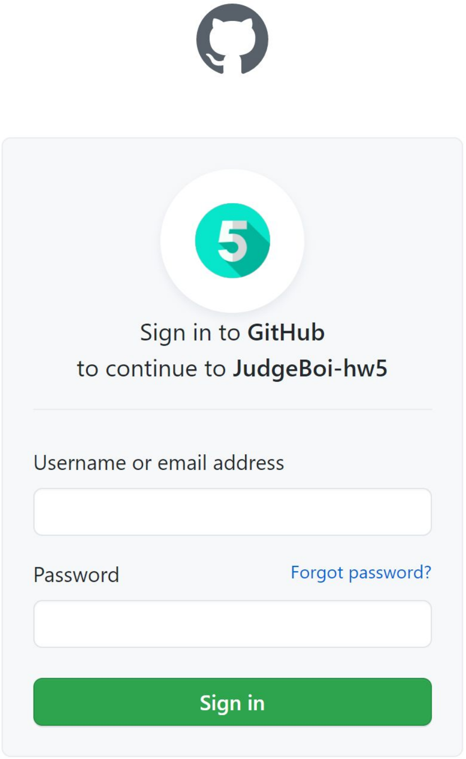


Ending in 22 days

Good Luck & Have Fun

# Step 2: Sign-in with Github

- If you have not completed the Github account survey
  - You can log in
  - You will not be able to submit



Sign in to **GitHub**
to continue to **JudgeBoi-hw5**

Username or email address

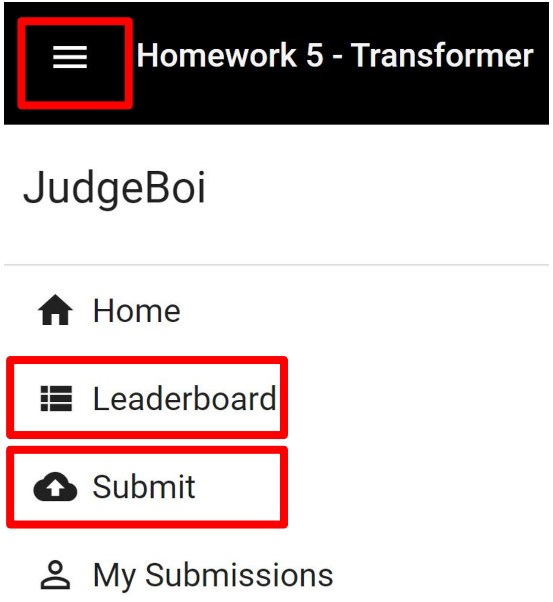
Password

**fill in username >**

**fill in password >**

# Step 3: Submit your Results

You can now submit results to the server and view the leaderboard.



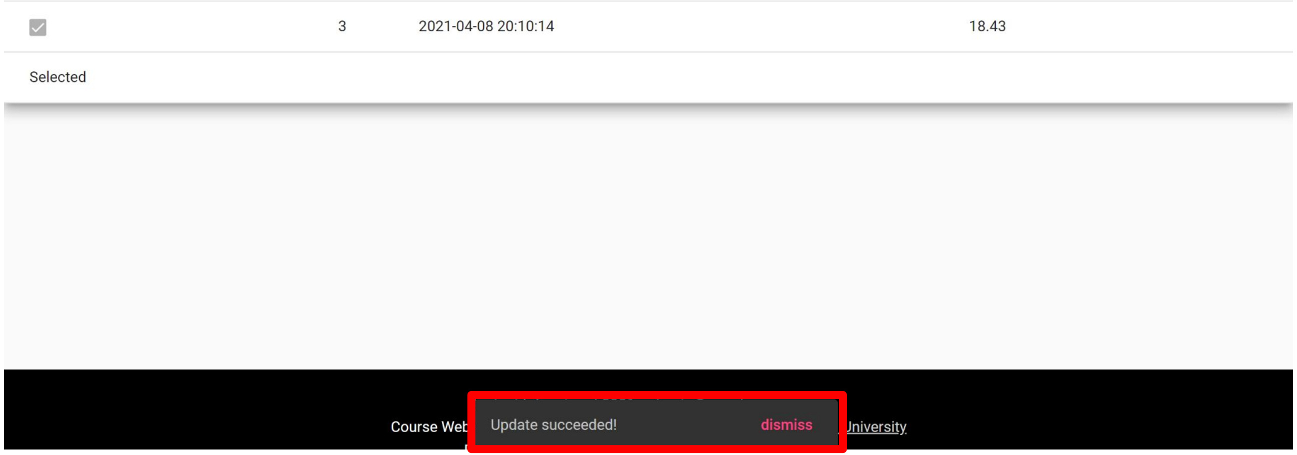
1) Click

3) Check the leaderboard

2) Submit results

# Step 4: Select your submissions

- You can select up to 2 submissions.
- If none of your submissions is chosen, we will use the submission with the best public score.
- If your selection is successful, you will see a message box as follows:



| | | | |
|---|---|---|---|
| ☑ | 3 | 2021-04-08 20:10:14 | 18.43 |

Selected

Update succeeded!

# JudgeBoi Rules

- 5 submission quota per day, reset at **midnight**.
  - Users not in the whitelist will have no quota.
- Only ***.txt** file is allowed, filesize should be smaller than **700kB**.
- The countdown timer on the homepage is for reference only.
- We do limit the number of connections and request rate for each IP.
  - If you cannot access the website temporarily, please wait a moment.
- The system can be very busy as the deadline approaches
  - If this prevents uploads, we do not offer additional opportunities for remediation
- Please do not attempt to attack JudgeBoi.
- Every **Friday** from **6:00 to 9:00** is our system maintenance time.
- For any JudgeBoi issues, please post on NTUCOOL discussion
  - Discussion Link: https://cool.ntu.edu.tw/courses/11666/discussion_topics/91777

# Regulations and Grading Policy

# Grading

| | | |
|---|---|---|
| simple | (public) | +0.5 pts |
| simple | (private) | +0.5 pts |
| medium | (public) | +0.5 pts |
| medium | (private) | +0.5 pts |
| strong | (public) | +0.5 pts |
| strong | (private) | +0.5 pts |
| boss | (public) | +0.5 pts |
| boss | (private) | +0.5 pts |
| code submission | | +2 pts |
| report | | +4 pts |

Total : 10 pts

# Code Submission

- **NTU COOL** (4pts)
    - Compress your code and report into

      **<student ID>_hwX.zip**

      *** e.g. b06901020_hw1.zip**

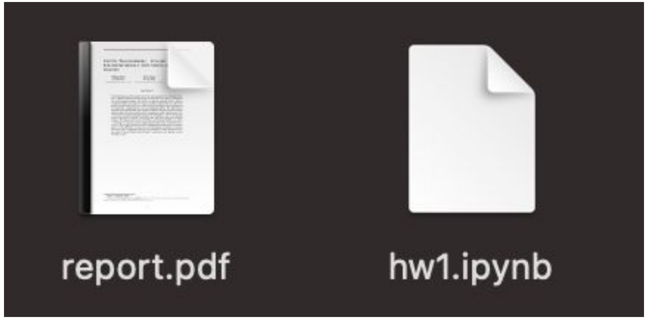
      *** X is the homework number**

    - We can only see your last submission.
    - Do not submit your model or dataset.
    - If your code is not reasonable, your semester grade x 0.9.

# Code Submission

- Your .zip file should include only
  - **Code**: either .py or .ipynb
  - **Report**: .pdf (only for those who got 10 points)

- Example:



# Report Submission

Answer the questions on GradeScope

# Deadlines

**2022/04/08 23:59 (UTC+8)**

# Regulation

- You should NOT plagiarize, if you use any other resource, you should cite it in the reference. ( ∗ )
- You should NOT modify your prediction files manually.
- Do NOT share codes or prediction files with any living creatures.
- Do NOT use any approaches to submit your results more than 5 times a day.
- **Do NOT search or use additional data or pre-trained models.**
- Your **final grade x 0.9** if you violate any of the above rules.
- Prof. Lee & TAs preserve the rights to change the rules & grades.

# If any questions, you can ask us via...

- NTU COOL (recommended)
  - https://cool.ntu.edu.tw/courses/4793
- Email
  - ntu-ml-2021spring-ta@googlegroups.com
  - The title should begin with “[hwX]” (X is the homework number)
- TA hour
  - Each Friday during class