Machine Learning and Having It Deep and Structured

HOMEWORK #1

Training Deep Neural Network

呂相弘　吳彥諶
r03942039@ntu.edu.tw
b99901106@ntu.edu.tw

Outline

- DNN in Speech Recognition
- DNN
- Kaggle
- Dataset and Format
- Submission Requirements
- Grading
- GPU Acceleration
- Tips
- Baselines
- 語音導覽(103-2)

DNN IN SPEECH RECOGNITION

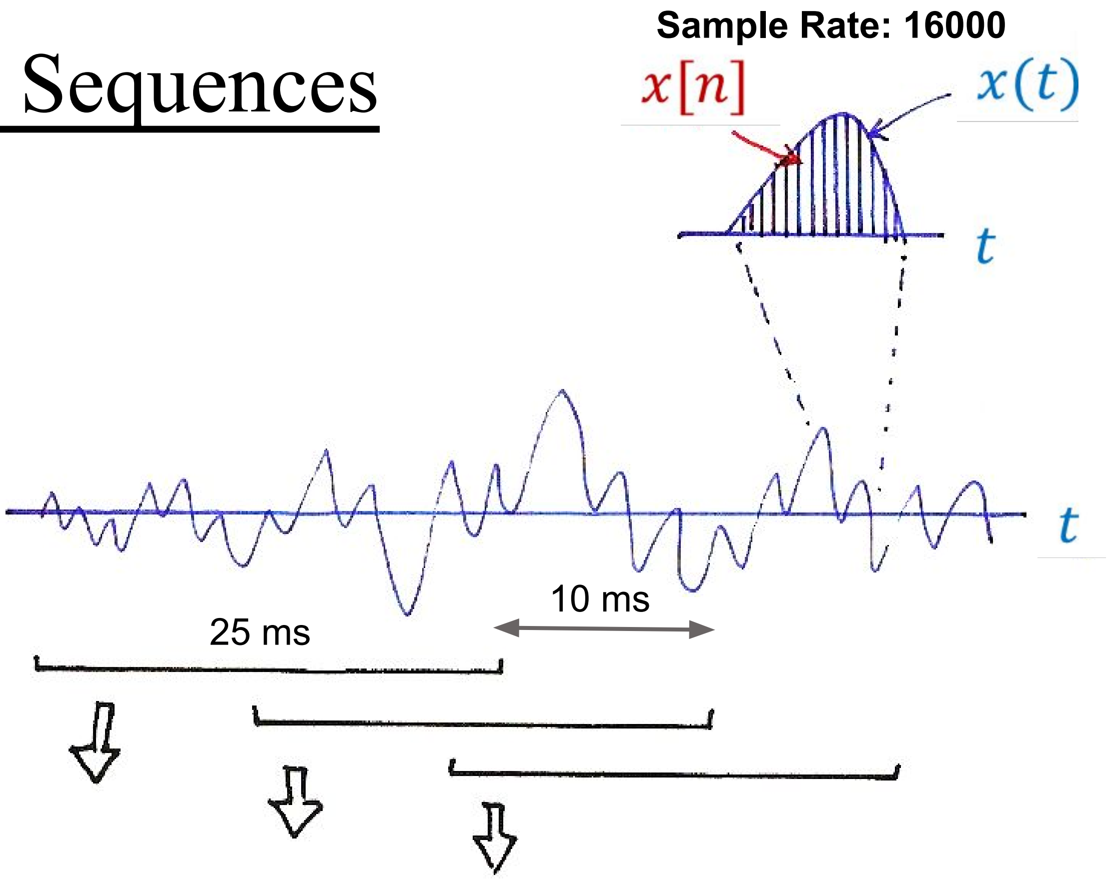
Speech Recognition

- In speech processing...
  - each word consists of syllables
  - each syllable consists of phonemes

“青色”→ “青(ㄑㄧㄥ)色(ㄙㄜˋ)”→ ”ㄑ” (syllables)
青:TSI --I –N (phonemes)
色:S--@ (phonemes)

- Each time frame, with an observance (vector) mapped to a phoneme.

Observation Sequences



sliding window

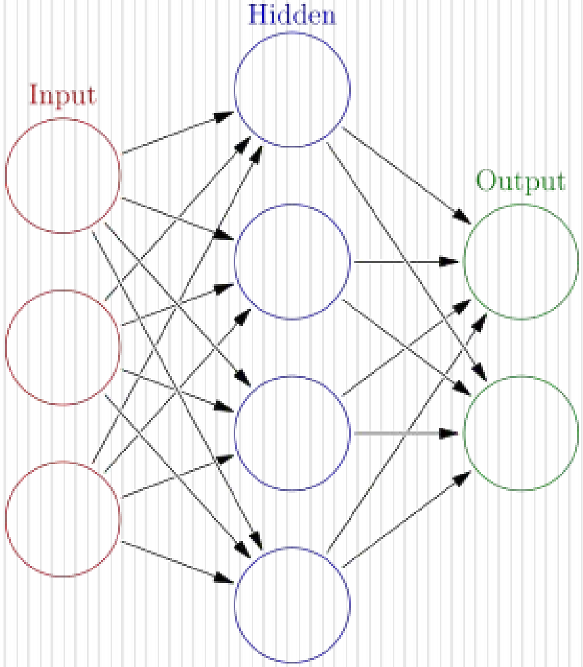
frames of features

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_D \end{bmatrix} = o_1 \quad \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_D \end{bmatrix} = o_2 \quad \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_D \end{bmatrix} = o_3$$

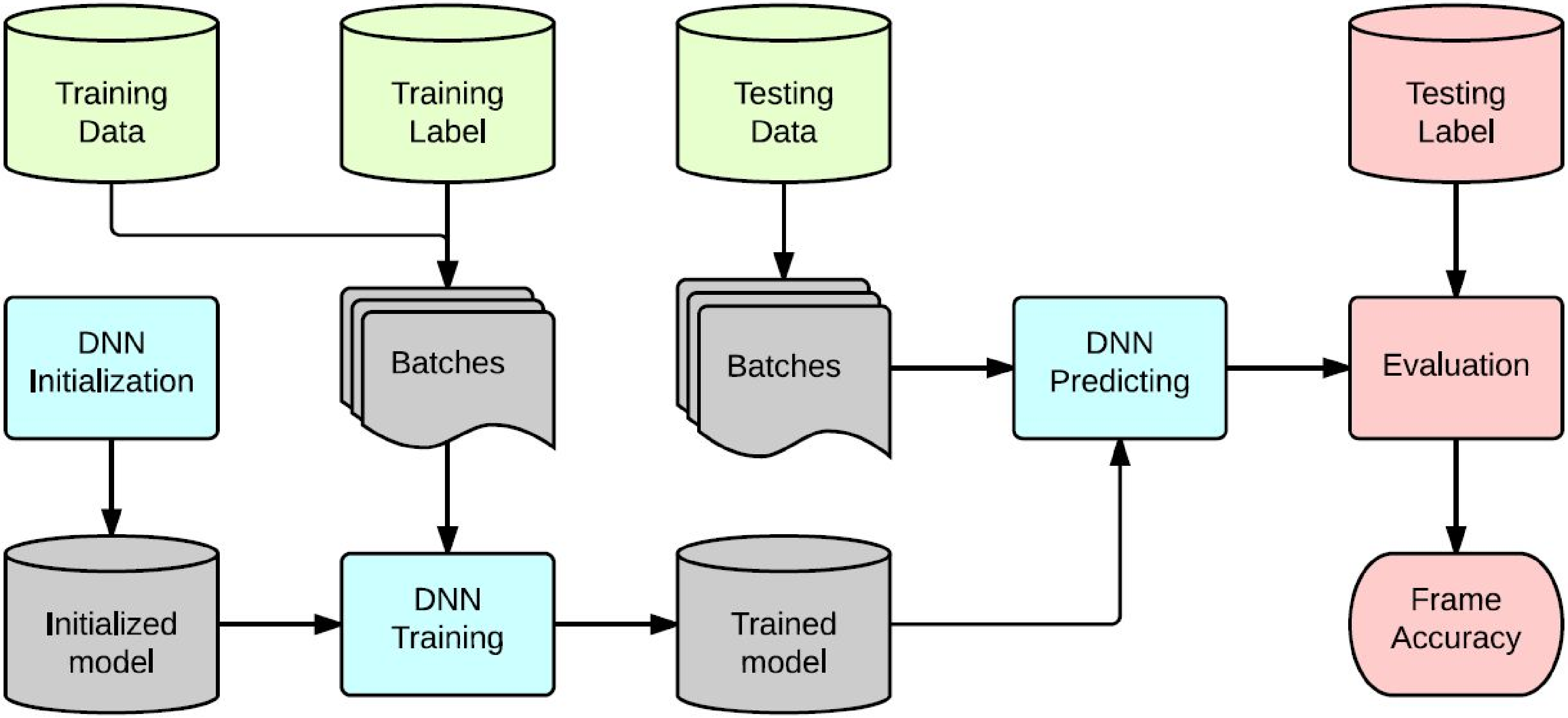
Frame 1 **Frame 2** **Frame 3**

DNN in Speech Recognition

- **Goal**: predict **phoneme** given feature in each time frame.
    - Frame-wise prediction
- **Input**: acoustic features
    - Raw waveform, MFCC, FBANK or…
- **Output**: pronunciation units
    - Phonemes or…
- To know more about Automatic Speech Recognition(ASR), please refer to DSP



Homework 1: Training Deep Neural Network

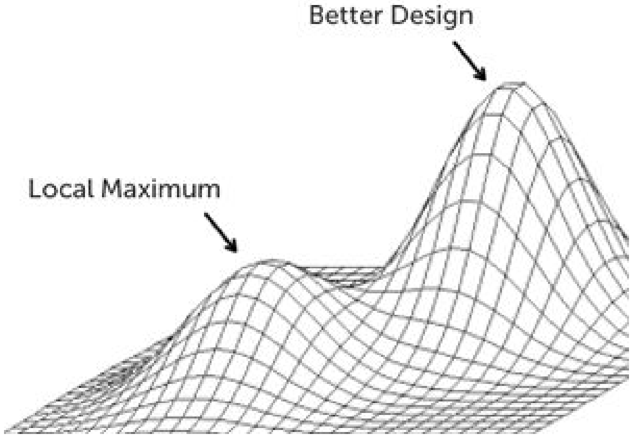

Training Data
Training Label
Testing Data
Testing Label
DNN Initialization
Batches
Batches
DNN Predicting
Evaluation
Initialized model
DNN Training
Trained model
Frame Accuracy

Main Problems in HW1

- Model initialize
- Feedforward
- Backpropagate
- Update
- Predict

Model Initialize

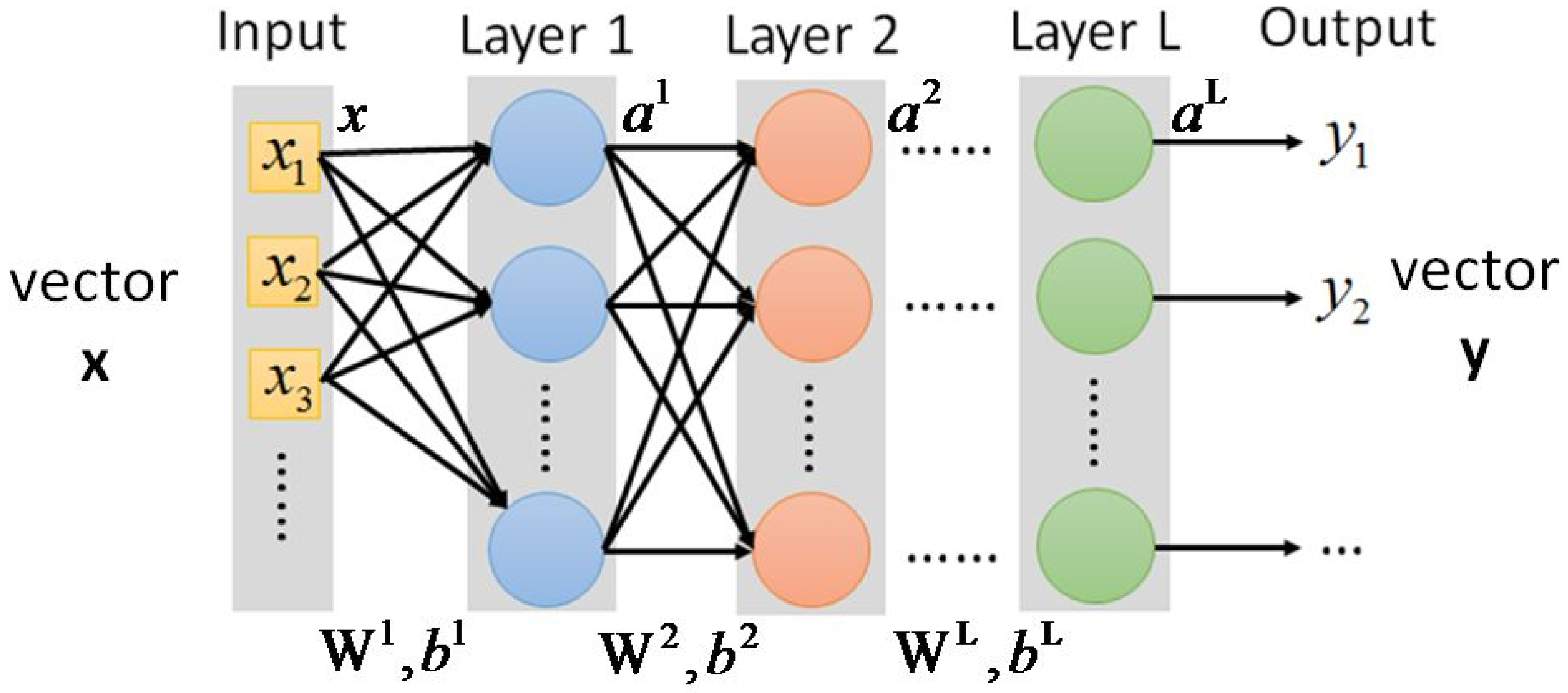
- DNN sometimes fails at local optimum problem, so **initialization matters**.
- Practically, there exists unsupervised pre-training technique on initialization.
- However, in this homework, we recommend you **initialize them randomly for the simplicity and efficiency**.



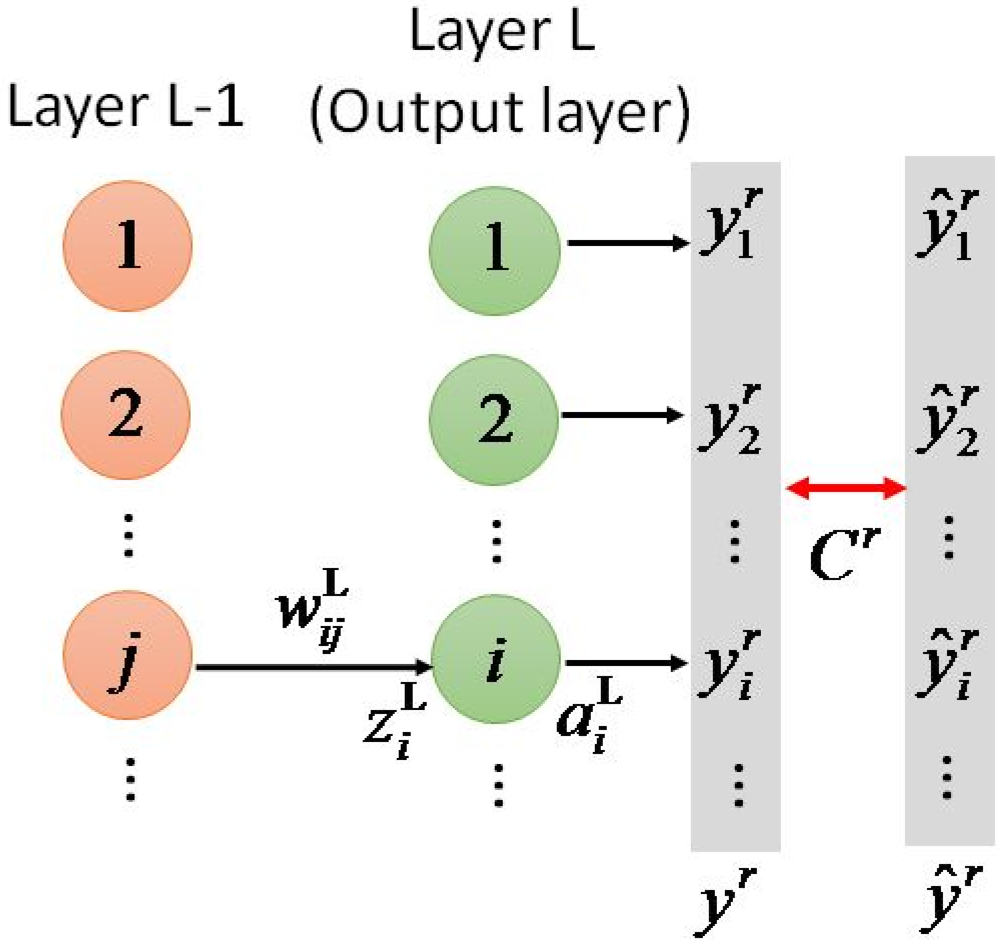
Feedforward

$$y = f(x) = \sigma\left(\mathrm{W}^{L} \ldots \sigma\left(\mathrm{W}^{2} \sigma\left(\mathrm{W}^{1} x + b^{1}\right) + b^{1}\right) \ldots + b^{L}\right)$$

$$\sigma\left(\mathrm{W}^{\mathrm{L}} a^{\mathrm{L\text{-}1}} + b^{\mathrm{L}}\right) = a^{\mathrm{L}} = y$$



Backpropagate



$$\frac{\partial C^r}{\partial w_{ij}^L} = \frac{\partial C^r}{\partial y_i^r}\frac{\partial y_i^r}{\partial z_i^L}\frac{\partial z_i^L}{\partial w_{ij}^L}$$

$$= a_j^{L-1}\sigma'\left(z_i^L\right)\left(y_i^r - \hat{y}_i^r\right)$$

Example: $\left(x^r, \hat{y}^r\right)$

Update

$$\textit{Compute}\ \nabla C\left(\theta^0\right)$$

$$\theta^1 = \theta^0 - \eta \nabla C\left(\theta^0\right)$$

$$\textit{Compute}\ \nabla C\left(\theta^1\right)$$

$$\theta^2 = \theta^1 - \eta \nabla C\left(\theta^1\right)$$

$$\textit{Compute}\ \nabla C\left(\theta^2\right)$$

$$\begin{tikzcd}
\text{Starting Parameters} \arrow[d] \\
\theta^0 \arrow[r] & \theta^1 \arrow[r] & \theta^2 \arrow[r] & \cdots\cdots
\end{tikzcd}$$

$$C\left(\theta^0\right) \quad > \quad C\left(\theta^1\right) \quad > \quad C\left(\theta^2\right) \quad > \quad \cdots\cdots$$

In practice, η can simply set to be constant

η is called "***learning rate***"

η should be small enough, but should not be too small.

Predict

- DNN classifier usually output the **1-best label** as prediction.
- Practically, the DNN may preserve certain number of good candidates for further processing due to the **structure of speech.**
  - That is, the best in one frame would be unacceptable when considering the context frames.
- The sequence “fixing” procedure is actually your HW2, comparable among the world as announced.
- **In HW1, you should only predict the framewise 1-best output.**

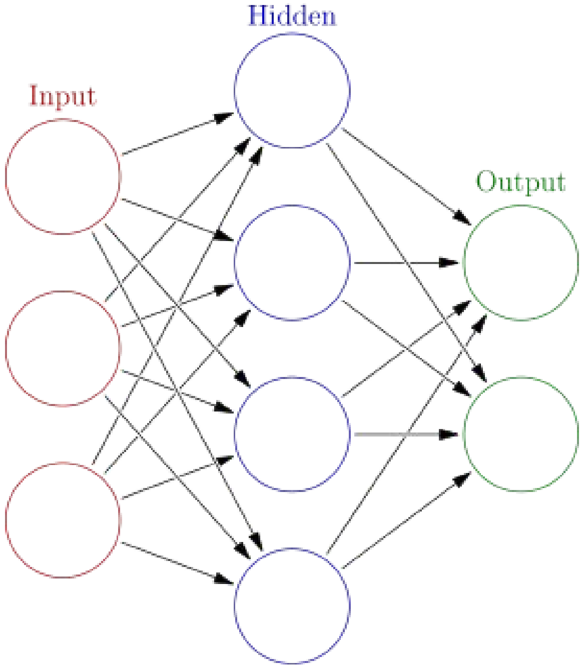
More About Input...

- We provide **3 types** of feature for the DNN input w.r.t each frame.
  - Raw Waveform(16k sampling rate)
  - MFCC(dimension = 39)
  - FBANK(dimension = 69)
- MFCC and FBANK are 2 types of feature extracted from the wave. However, **the details lie beyond the scope of this course.**
- There does exist debates over which feature is better. So the choice is yours.

More About Input...

- You could generate your own features based on the WAV/MFCC/FBANK.
- However, be sure your task can be evaluated in a frame-wise way.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_D \end{bmatrix} = o_1 \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_D \end{bmatrix} = o_2$$



More About Output...

- We provide **3 types** of training target for the DNN output w.r.t each frame.
  - states(index from 0 to 1942)
  - 48 - phonemes
  - 39 - phonemes
- The difference also lies beyond the scope of this course. You could simply view them as more **delicate pronunciation units** at different level.

More About Output…

- Actually, the framewise labels are usually absent. There are only human annotation w.r.t each sentence.
- Practically, we forcibly align the labels to each frame by generative models other than DNN.
- The alignment results are then prepared for DNN as training targets.
- **In this homework, you do not need to bother. The alignments are transformed into framewise labels for you.**

Validation

- **There are two scores displayed when you upload your results on the site.**
- Public score is based on validation set and you can compare yours with others'.
- Private score is based on test set and should determine your final place.
- To better control your training procedure, you may want to test your model on a validation set after each epoch.
  - However, submitting the results onto the scoreboard after each epoch can be tedious,
  - In such case, it is recommended to **split your own validation set** from the training set.

Evaluation

- **Framewise** phoneme prediction
- Frame Accuracy

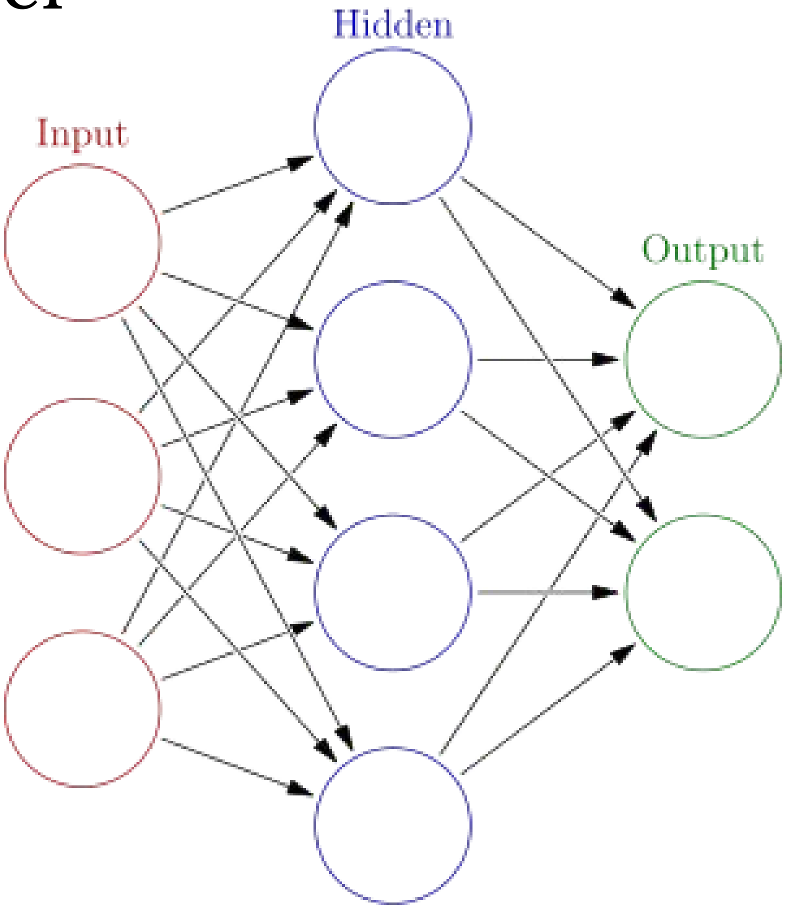
$$Frame\ Accuracy = \frac{\#\ of\ correct\ frames}{\#\ of\ total\ frames}$$

Evaluation

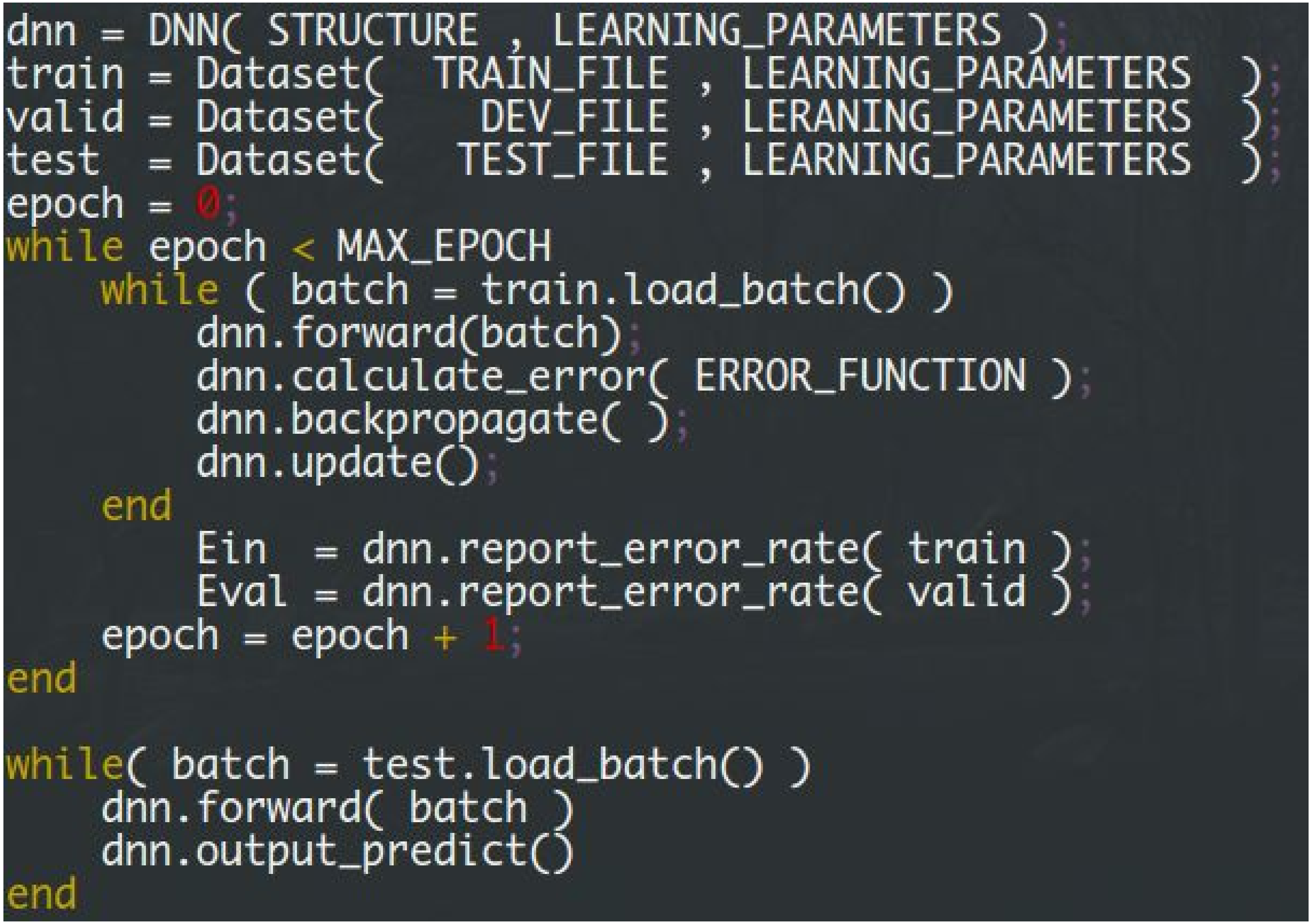
- Despite the type of training target you used, **you have to map all of them to 39 phoneme set for evaluation.**
  - Some training targets would be merged into the same phoneme.
- The frame accuracy is calculated according to the merged prediction.

Baseline

- Only one hidden layer
- 69x128x48
- 69: fbank features
- 128: 1 hidden layer
- 48: 48 phoneme set

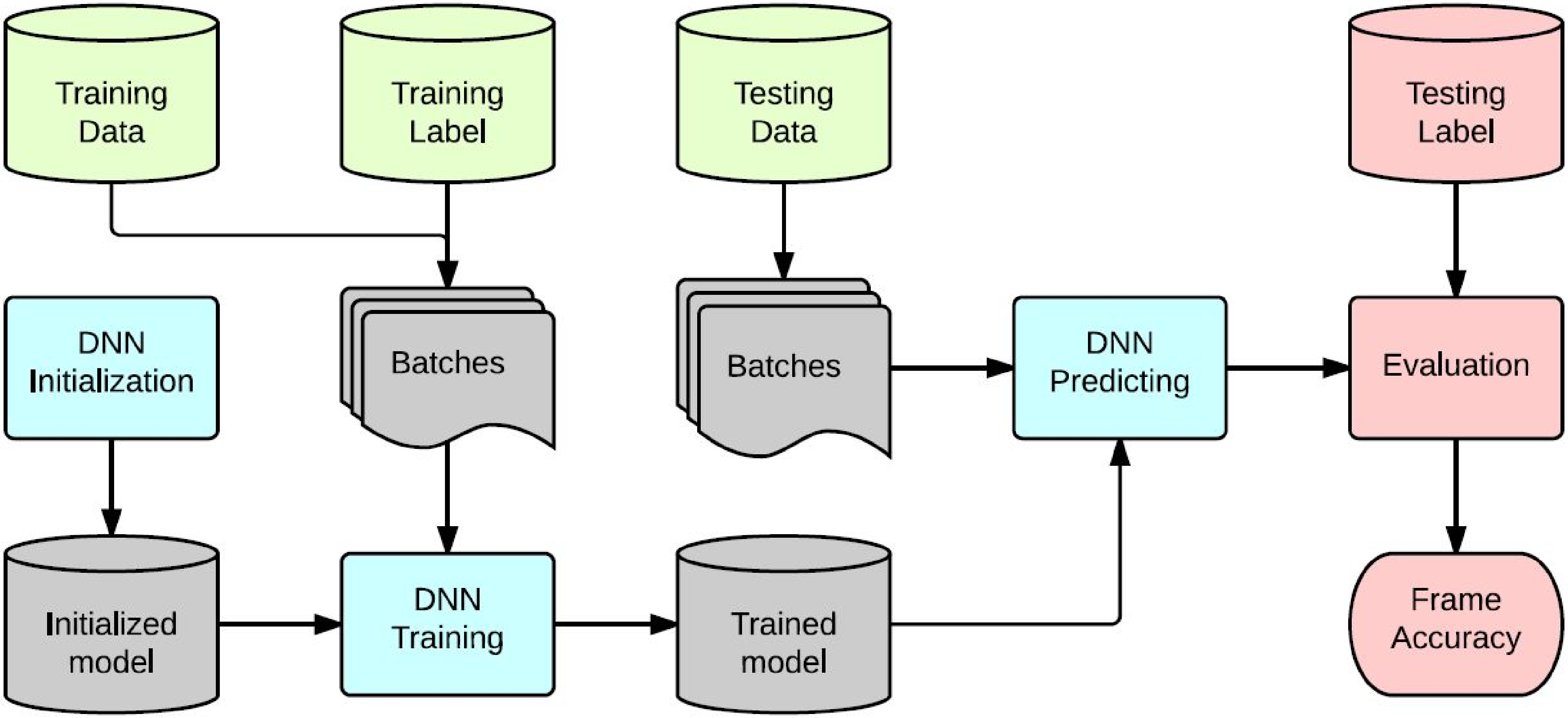


Pseudo Code



```
dnn = DNN( STRUCTURE , LEARNING_PARAMETERS );
train = Dataset(  TRAIN_FILE , LEARNING_PARAMETERS  );
valid = Dataset(    DEV_FILE , LERANING_PARAMETERS  );
test  = Dataset(   TEST_FILE , LEARNING_PARAMETERS  );
epoch = 0;
while epoch < MAX_EPOCH
    while ( batch = train.load_batch() )
        dnn.forward(batch);
        dnn.calculate_error( ERROR_FUNCTION );
        dnn.backpropagate( );
        dnn.update();
    end
        Ein  = dnn.report_error_rate( train );
        Eval = dnn.report_error_rate( valid );
    epoch = epoch + 1;
end

while( batch = test.load_batch() )
    dnn.forward( batch )
    dnn.output_predict()
end
```



Training Data

Training Label

Testing Data

Testing Label

DNN Initialization

Batches

Batches

DNN Predicting

Evaluation

Initialized model

DNN Training

Trained model

Frame Accuracy

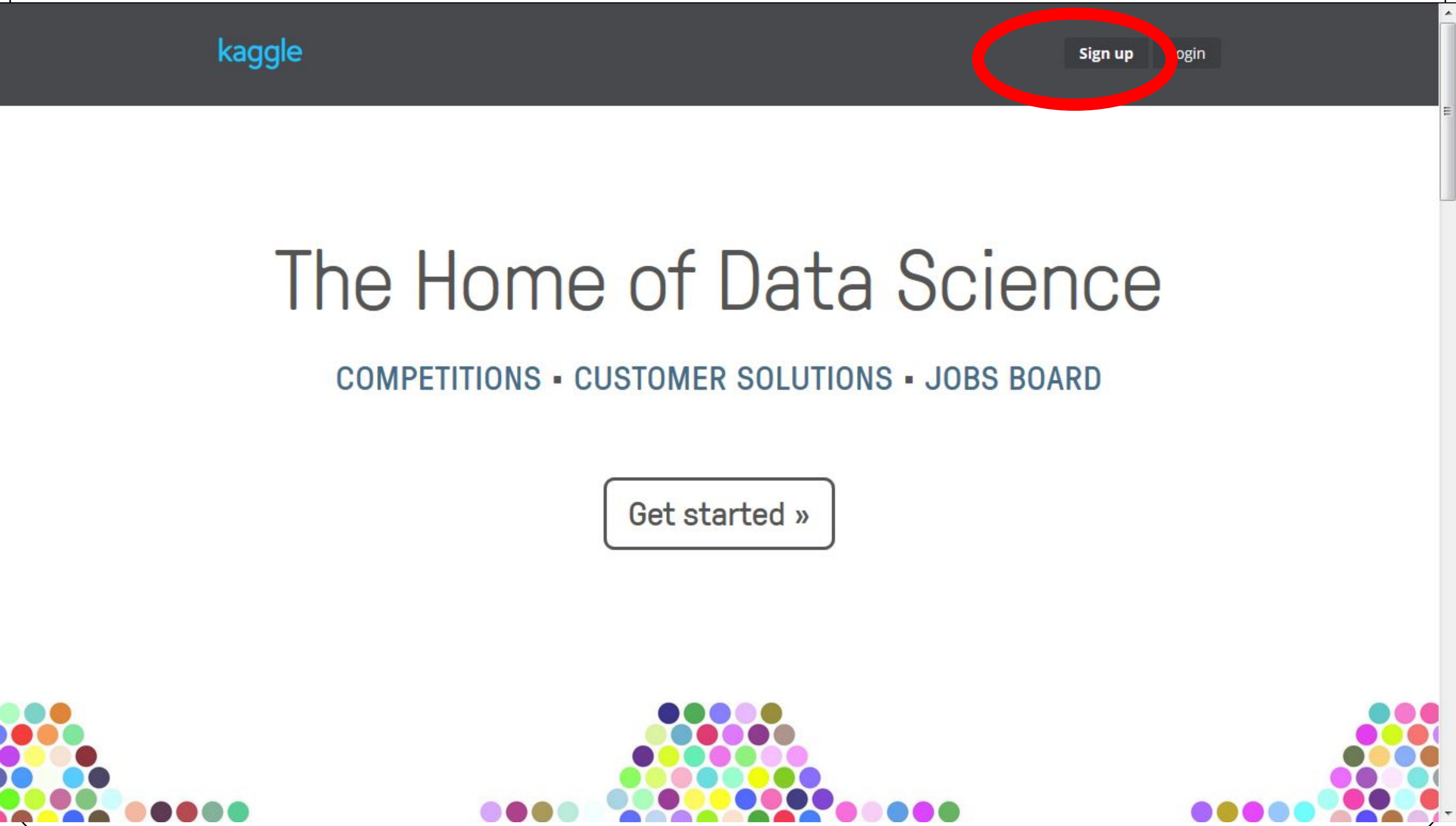
KAGGLE

Kaggle

- http://www.kaggle.com/

- A competition platform used by many academic institutes.
- Evaluate your result on this website.
  - With Scoreboard that can show off your improvement.
- You could upload your predictions **only twice a day**.
  - Be careful!

Registration



kaggle

Sign up

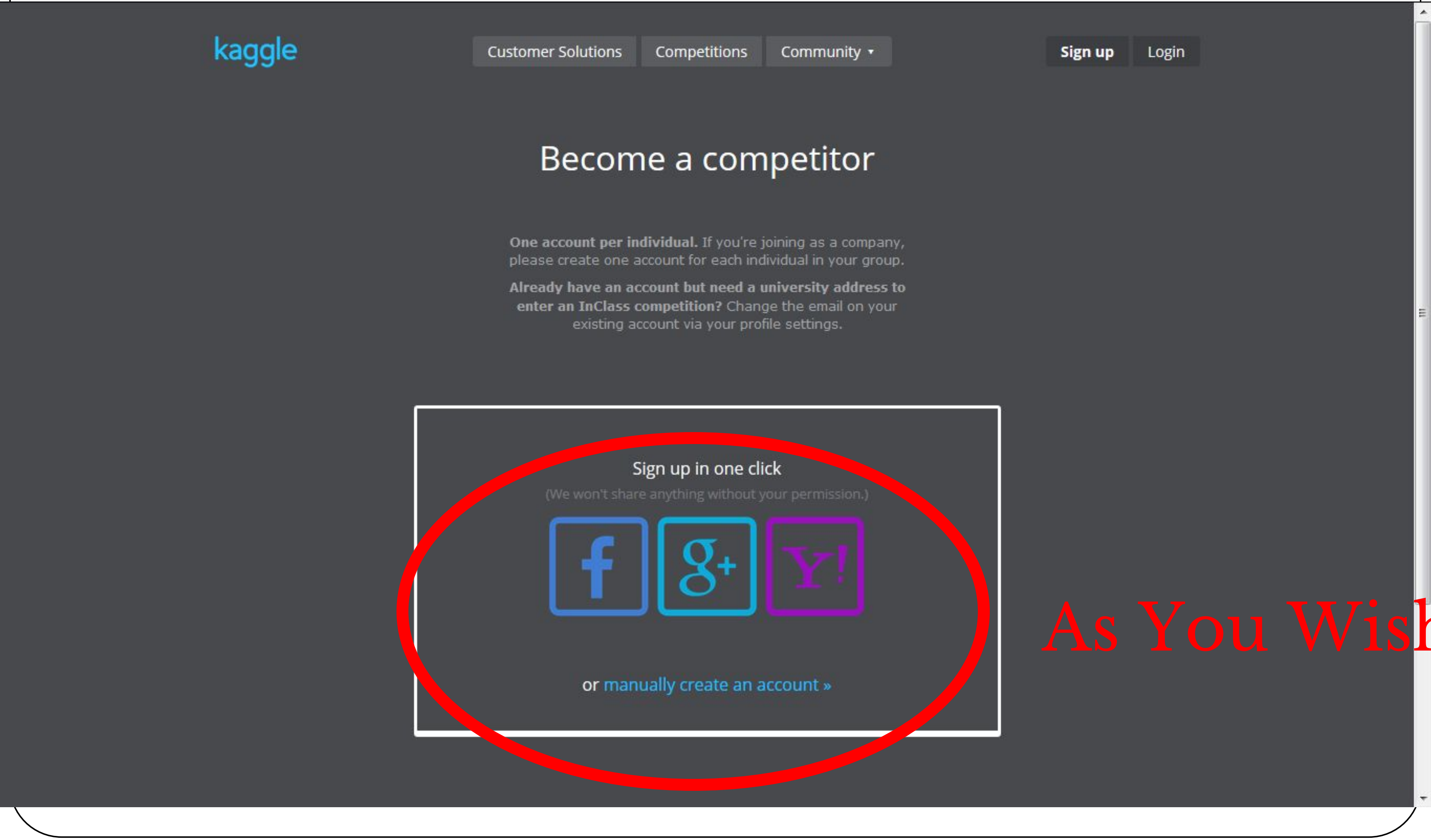
Login

The Home of Data Science

COMPETITIONS ▪ CUSTOMER SOLUTIONS ▪ JOBS BOARD

Get started »

Registration



Become a competitor

One account per individual. If you're joining as a company, please create one account for each individual in your group.

Already have an account but need a university address to enter an InClass competition? Change the email on your existing account via your profile settings.

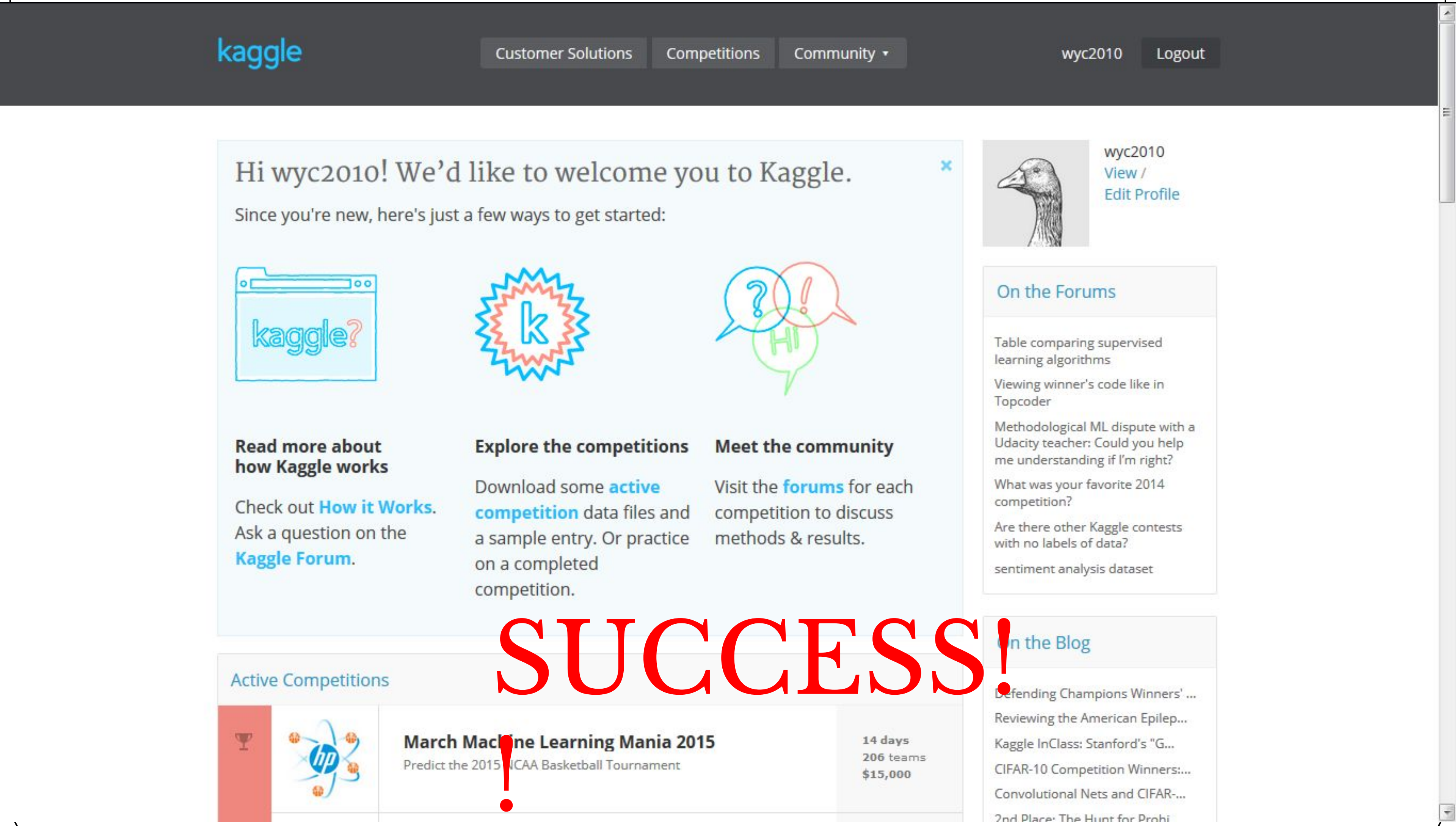
Sign up in one click

(We won't share anything without your permission.)

or manually create an account »

As You Wish

Registration



Hi wyc2010! We'd like to welcome you to Kaggle.

Since you're new, here's just a few ways to get started:

Read more about how Kaggle works

Check out **How it Works**. Ask a question on the **Kaggle Forum**.

Explore the competitions

Download some **active competition** data files and a sample entry. Or practice on a completed competition.

Meet the community

Visit the **forums** for each competition to discuss methods & results.

On the Forums

- Table comparing supervised learning algorithms
- Viewing winner's code like in Topcoder
- Methodological ML dispute with a Udacity teacher: Could you help me understanding if I'm right?
- What was your favorite 2014 competition?
- Are there other Kaggle contests with no labels of data?
- sentiment analysis dataset

On the Blog

- Defending Champions Winners' ...
- Reviewing the American Epilep...
- Kaggle InClass: Stanford's "G...
- CIFAR-10 Competition Winners:...
- Convolutional Nets and CIFAR-...
- 2nd Place: The Hunt for Prohi

Active Competitions

March Machine Learning Mania 2015
Predict the 2015 NCAA Basketball Tournament

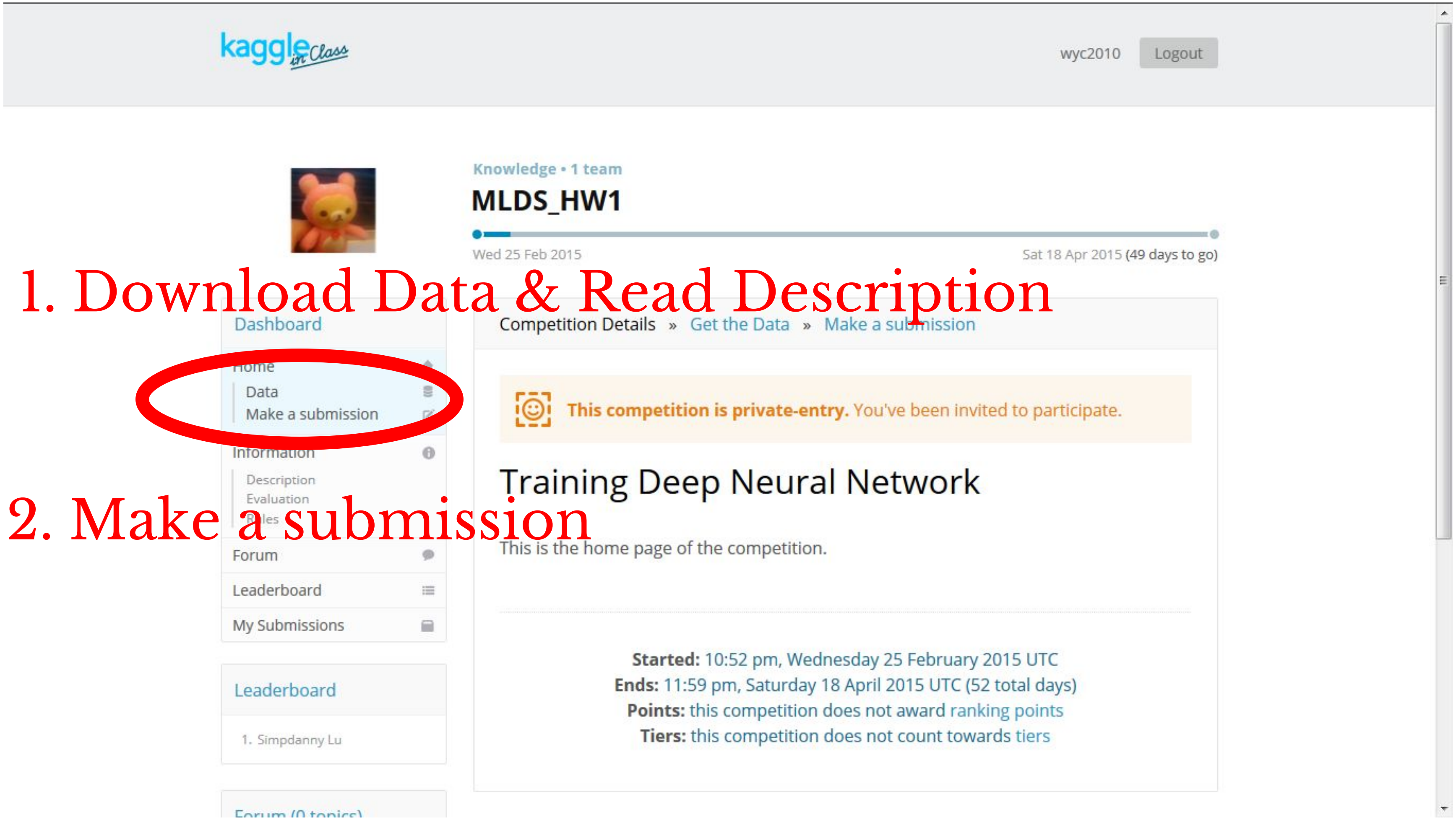
14 days
206 teams
$15,000

SUCCESS!

!

Join the Competition

- https://kaggle.com/join/104_1_mlds_hw1



1. Download Data & Read Description

2. Make a submission

Knowledge • 1 team

MLDS_HW1

Wed 25 Feb 2015 — Sat 18 Apr 2015 (49 days to go)

Competition Details » Get the Data » Make a submission

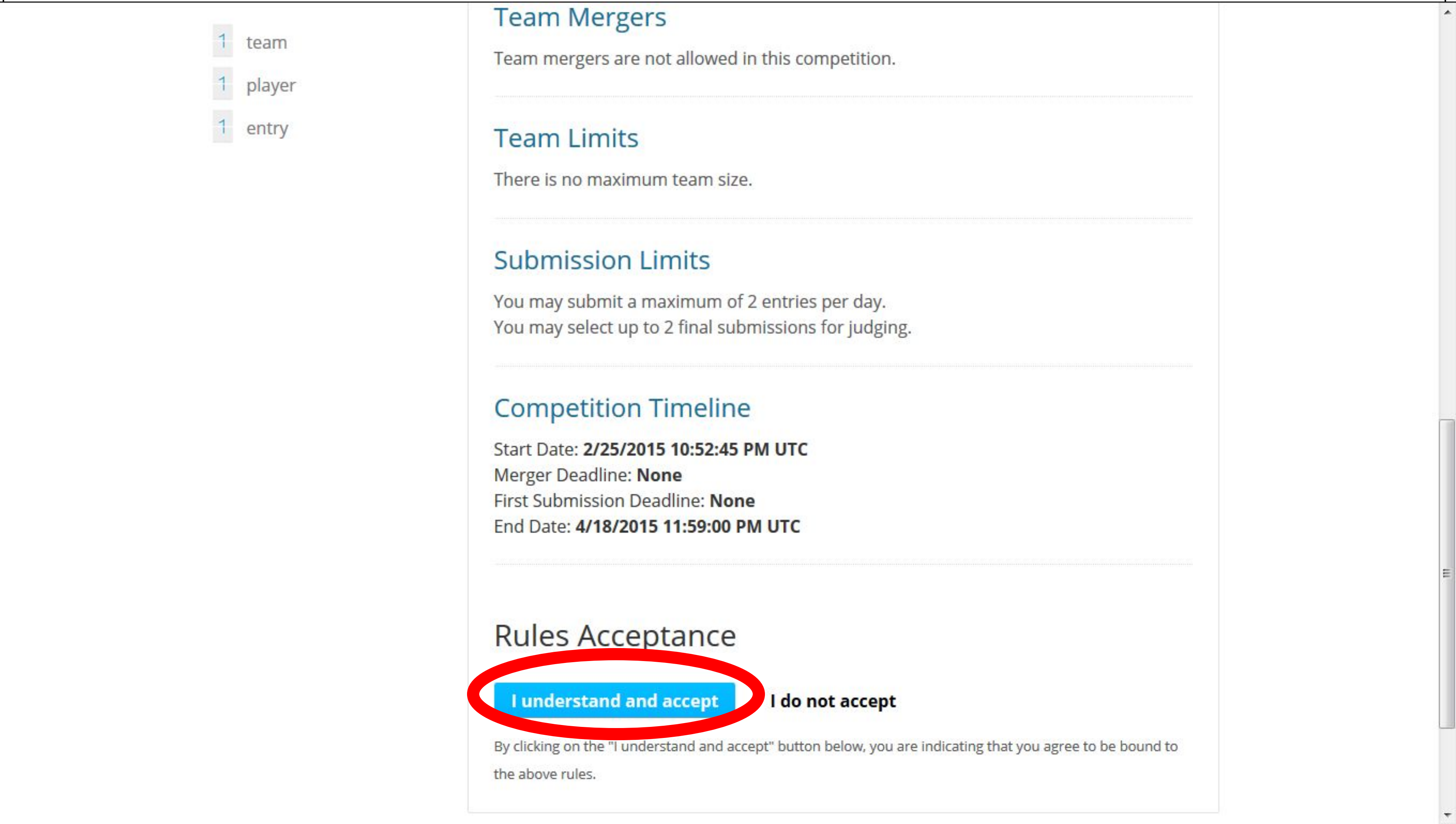
This competition is private-entry. You've been invited to participate.

Training Deep Neural Network

This is the home page of the competition.

Started: 10:52 pm, Wednesday 25 February 2015 UTC
Ends: 11:59 pm, Saturday 18 April 2015 UTC (52 total days)
Points: this competition does not award ranking points
Tiers: this competition does not count towards tiers

Join the Competition



1	team
1	player
1	entry

Team Mergers

Team mergers are not allowed in this competition.

Team Limits

There is no maximum team size.

Submission Limits

You may submit a maximum of 2 entries per day.
You may select up to 2 final submissions for judging.

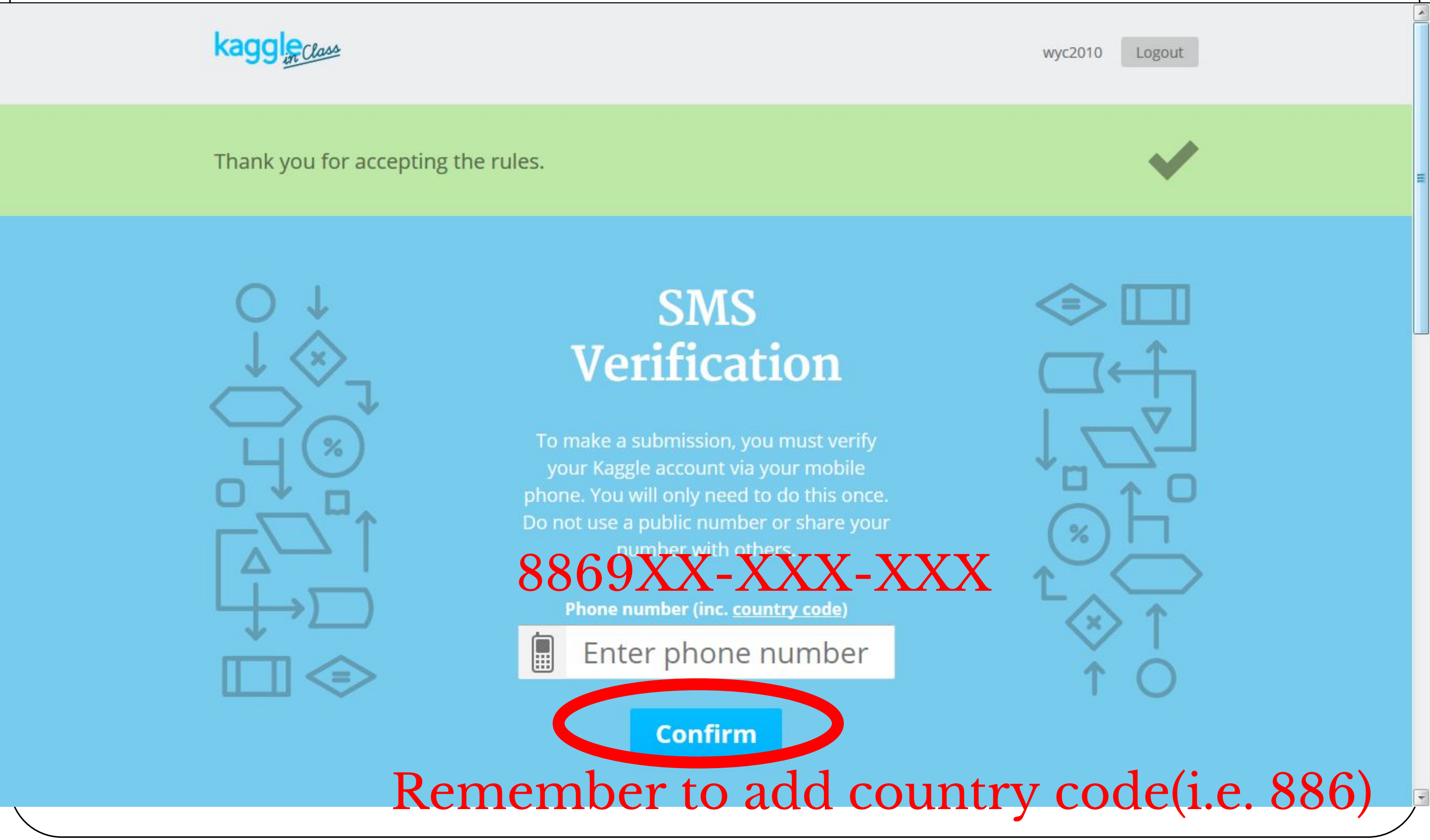
Competition Timeline

Start Date: **2/25/2015 10:52:45 PM UTC**
Merger Deadline: **None**
First Submission Deadline: **None**
End Date: **4/18/2015 11:59:00 PM UTC**

Rules Acceptance

By clicking on the "I understand and accept" button below, you are indicating that you agree to be bound to the above rules.

Join the Competition



Thank you for accepting the rules. ✓

SMS Verification

To make a submission, you must verify your Kaggle account via your mobile phone. You will only need to do this once. Do not use a public number or share your number with others.

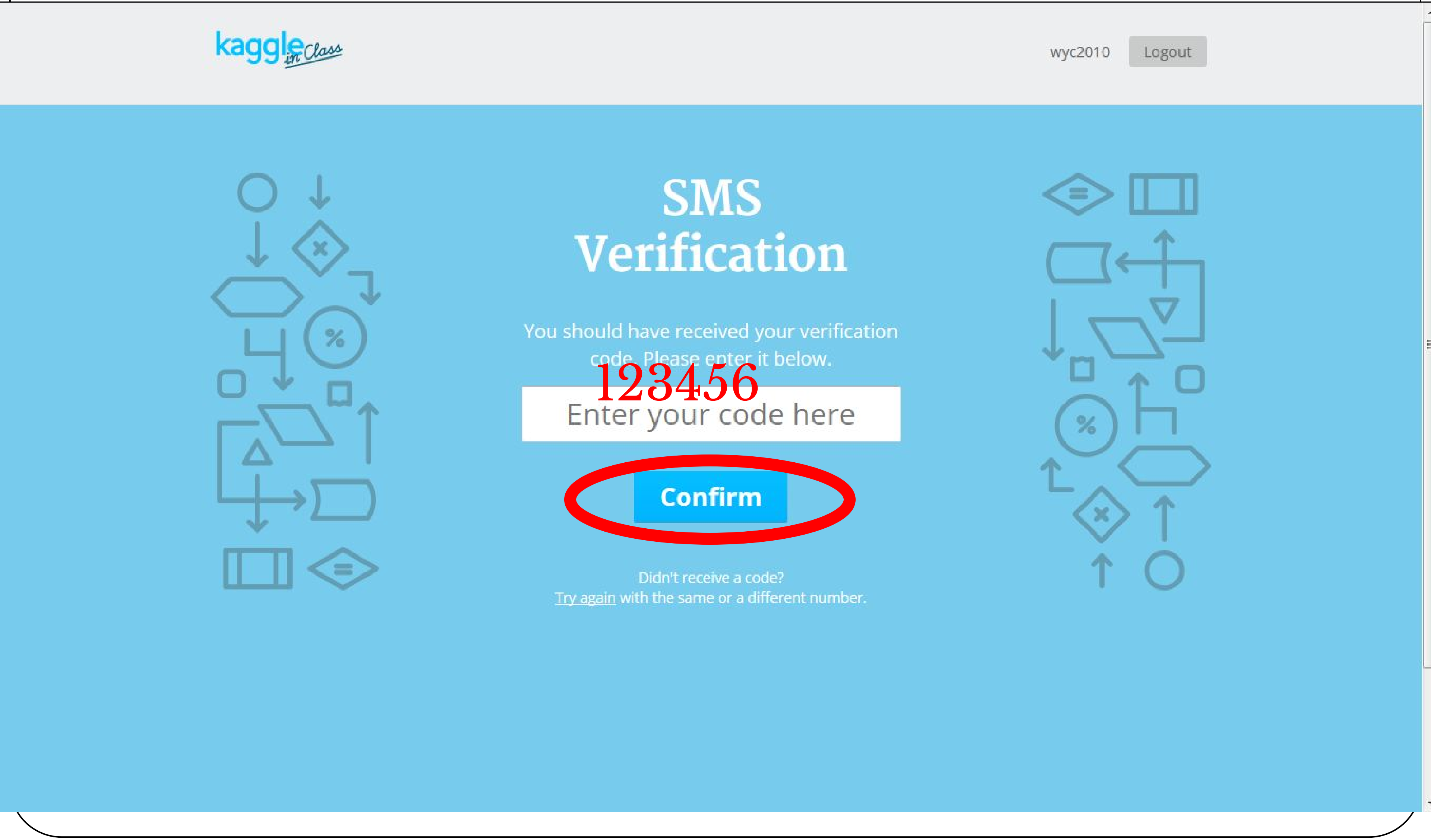
8869XX-XXX-XXX

Phone number (inc. country code)

☎ Enter phone number

Remember to add country code(i.e. 886)

Join the Competition



SMS Verification

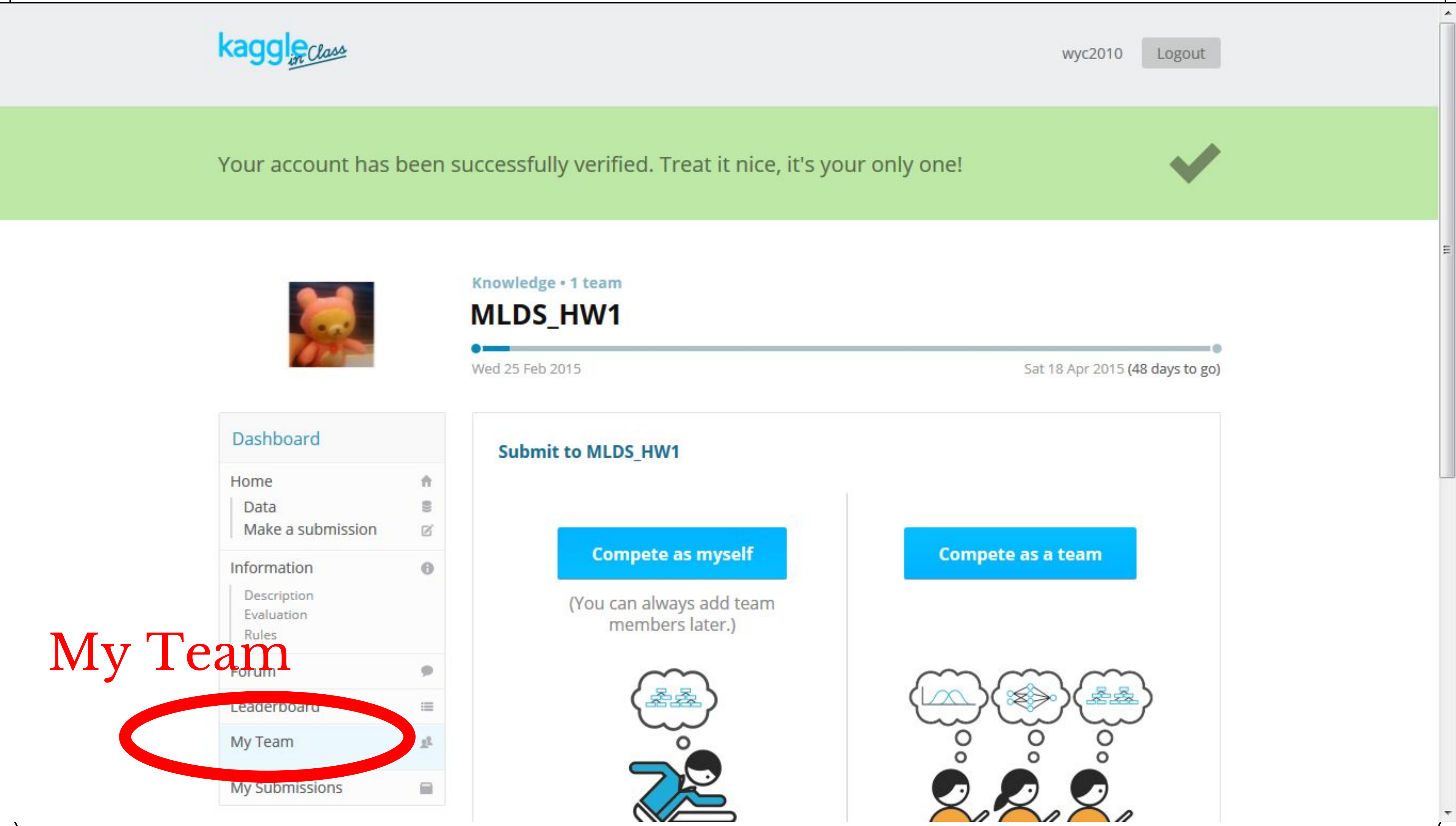
You should have received your verification code. Please enter it below.

123456

Enter your code here

Didn't receive a code?
Try again with the same or a different number.

Join the Competition



kaggle in Class

wyc2010 Logout

Your account has been successfully verified. Treat it nice, it's your only one! ✓

Knowledge • 1 team

MLDS_HW1

Wed 25 Feb 2015

Sat 18 Apr 2015 (48 days to go)

Dashboard

- Home
  - Data
  - Make a submission
- Information
  - Description
  - Evaluation
  - Rules
- Forum
- Leaderboard
- My Team
- My Submissions

Submit to MLDS_HW1

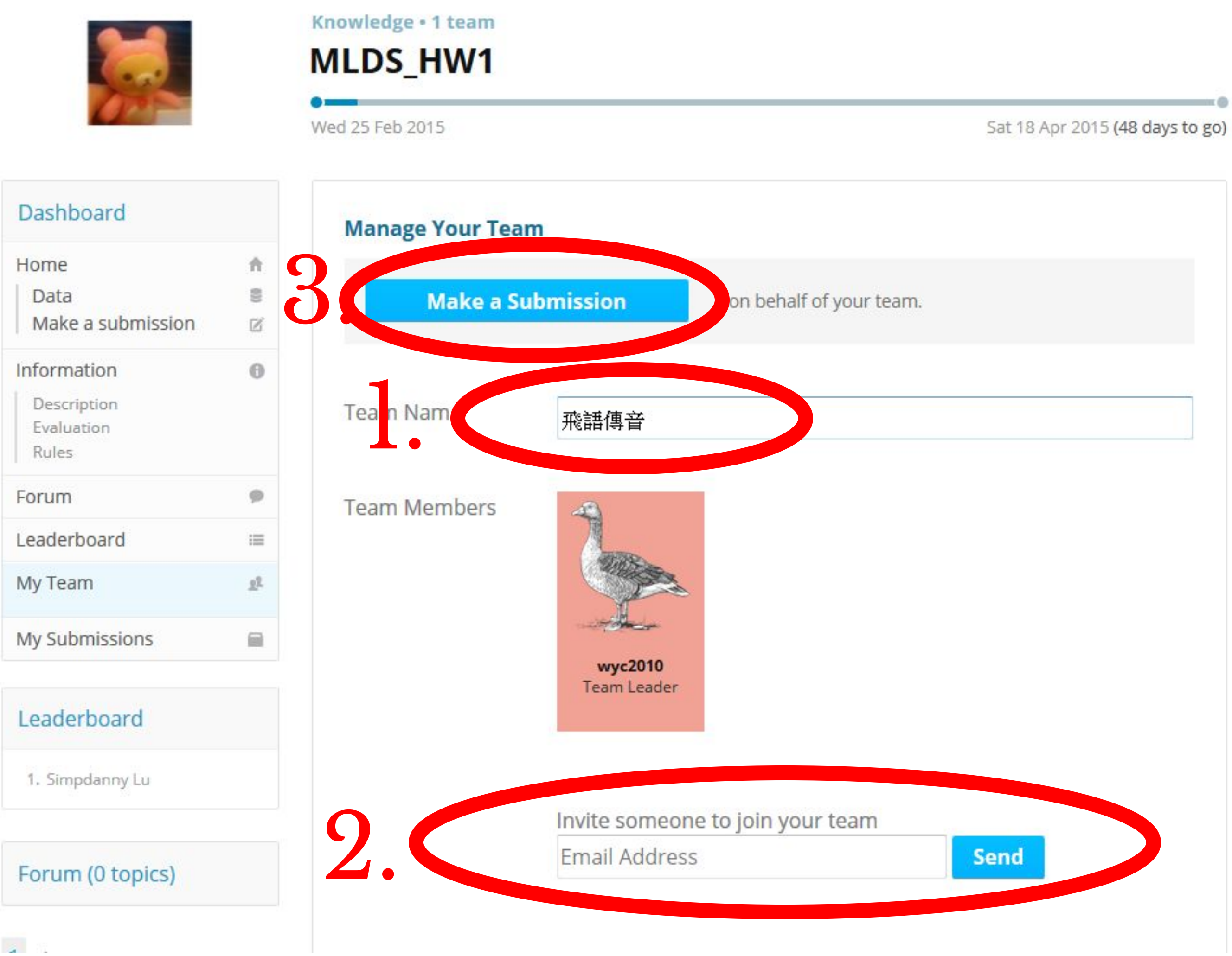
Compete as myself

(You can always add team members later.)

Compete as a team

My Team

Manage Your Team



Knowledge • 1 team

MLDS_HW1

Wed 25 Feb 2015 — Sat 18 Apr 2015 (48 days to go)

Dashboard

- Home
  - Data
  - Make a submission
- Information
  - Description
  - Evaluation
  - Rules
- Forum
- Leaderboard
- My Team
- My Submissions

Leaderboard

1. Simpdanny Lu

Forum (0 topics)

Manage Your Team

3. **Make a Submission** on behalf of your team.

1. Tea n Nam 飛語傳音

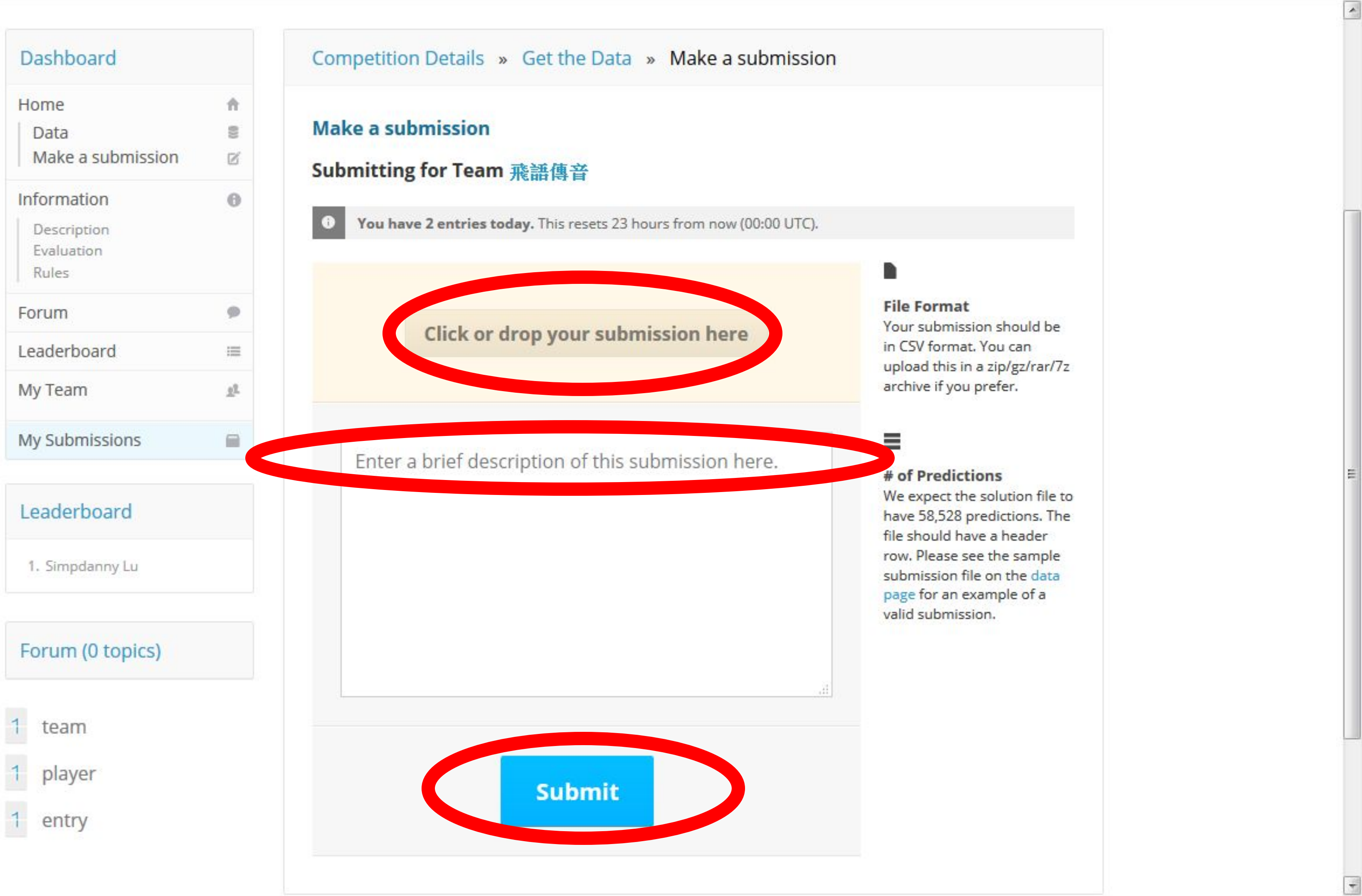
Team Members

wyc2010
Team Leader

2. Invite someone to join your team

Email Address **Send**

Submitting for Team



Make a submission

Submitting for Team 飛語傳音

You have 2 entries today. This resets 23 hours from now (00:00 UTC).

Enter a brief description of this submission here.

File Format
Your submission should be in CSV format. You can upload this in a zip/gz/rar/7z archive if you prefer.

# of Predictions
We expect the solution file to have 58,528 predictions. The file should have a header row. Please see the sample submission file on the data page for an example of a valid submission.

Final Rules

- Please team-up with a group name in Kaggle **in 1 week.**
- **Any submission by individuals would be neglected and team mergers forbidden.**
- **Any submission by individuals would be neglected and team mergers forbidden.**
- **Any submission by individuals would be neglected and team mergers forbidden.**
- **Any submission by individuals would be neglected and team mergers forbidden.**
- **Any submission by individuals would be neglected and team mergers forbidden.**

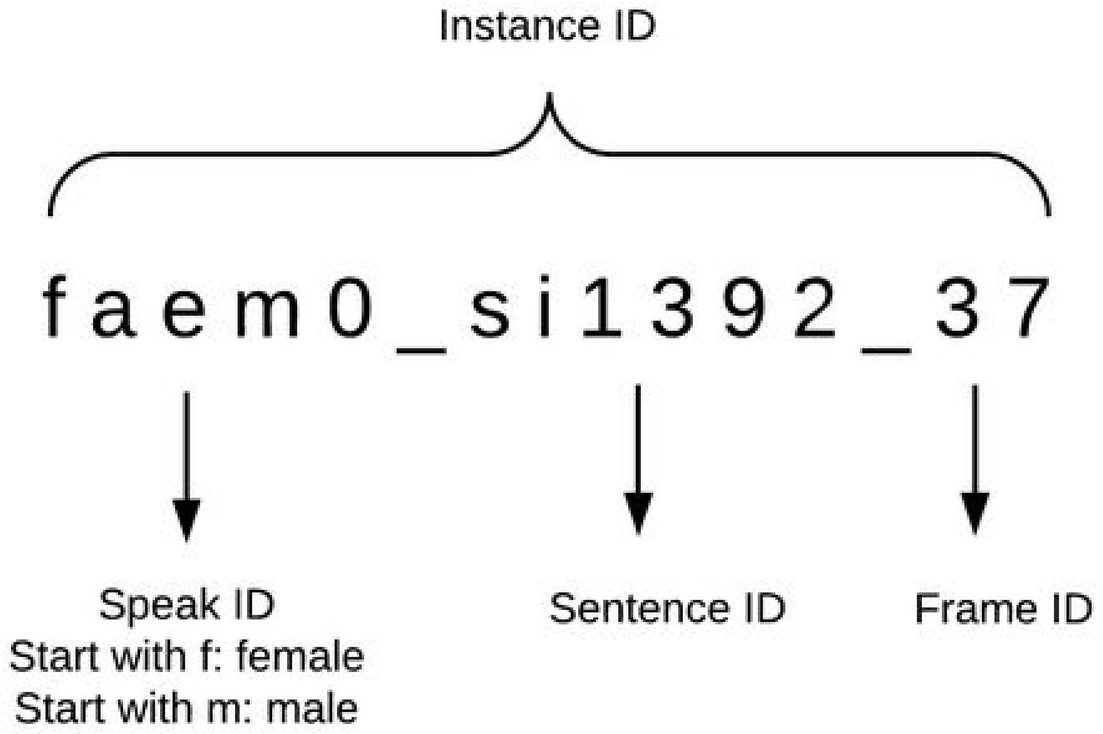
Dataset and Format

Dataset

- **TIMIT**(**T**exas **I**nstrument and **M**assachusetts **I**nstitute of **T**echnology)
- Well-transcribed speech of American English speakers of different sexes and dialects.
- Designed for the development and evaluation of ASR systems.

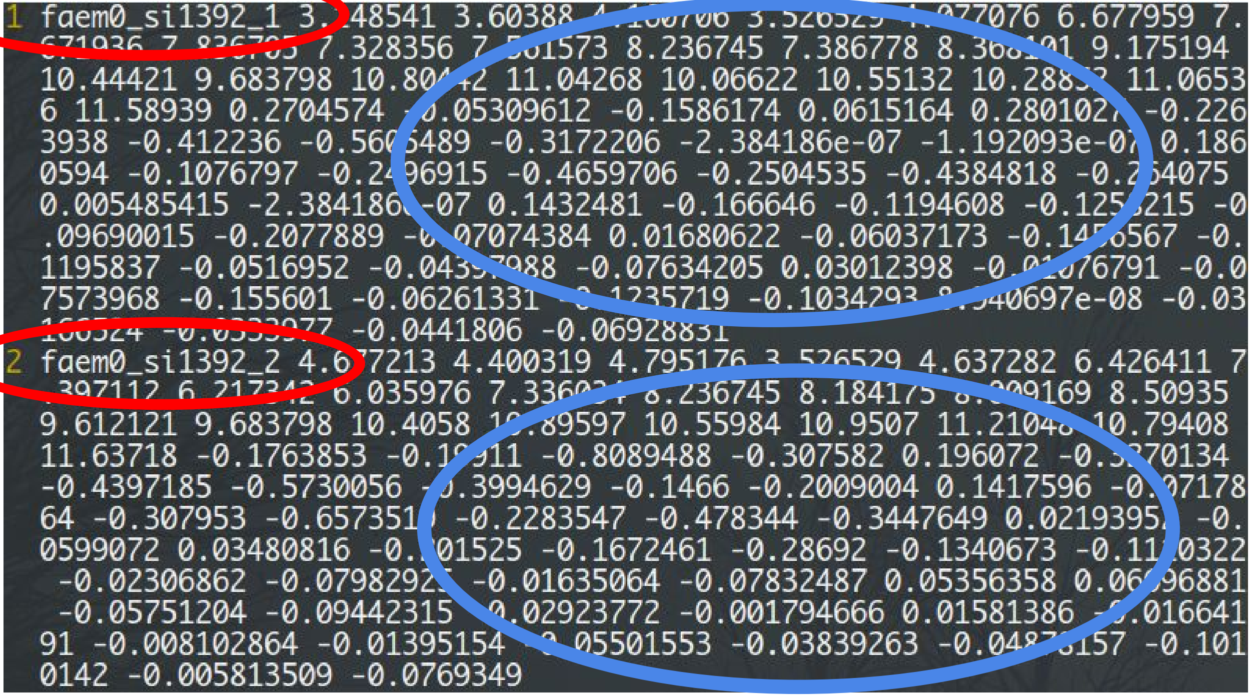
Dataset

- Each instance consists of 3 parts:
  - speaker **faem0**, sentence **si1392**, the **37th frame**



Data Format

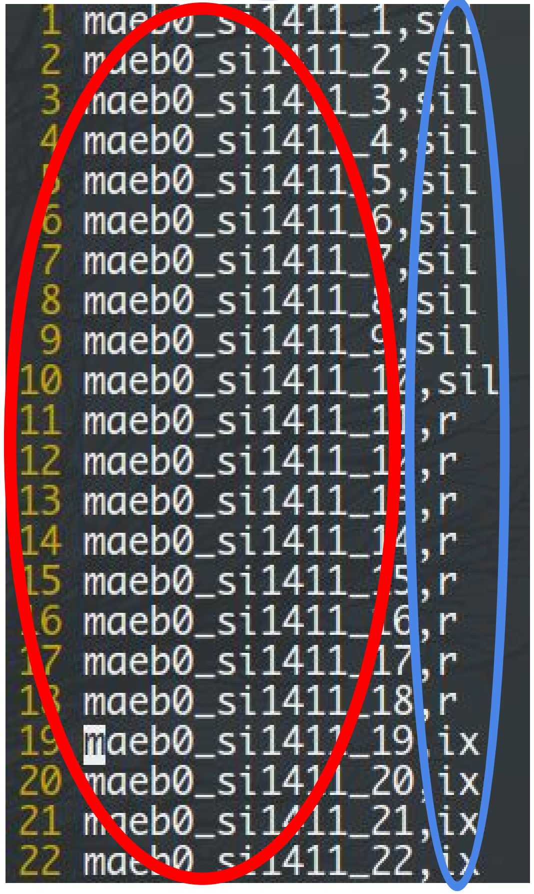
- **WAV file: Speak-Sentence ID + .wav**
  - Check by your ear(s)
- **ARK file: Instance ID + features**



```
faem0_si1392_1 3.[illegible]48541 3.60388 4.[illegible]00706 3.526529 4.077076 6.677959 7.
671936 7.836705 7.328356 7.[illegible]01573 8.236745 7.386778 8.368101 9.175194
10.44421 9.683798 10.80442 11.04268 10.06622 10.55132 10.28852 11.0653
6 11.58939 0.2704574 -0.05309612 -0.1586174 0.0615164 0.2801027 -0.226
3938 -0.412236 -0.5606489 -0.3172206 -2.384186e-07 -1.192093e-07 0.186
0594 -0.1076797 -0.2496915 -0.4659706 -0.2504535 -0.4384818 -0.2[illegible]54075
0.005485415 -2.384186e-07 0.1432481 -0.166646 -0.1194608 -0.125[illegible]215 -0
.09690015 -0.2077889 -0.07074384 0.01680622 -0.06037173 -0.14[illegible]6567 -0.
1195837 -0.0516952 -0.043[illegible]7988 -0.07634205 0.03012398 -0.0[illegible]076791 -0.0
7573968 -0.155601 -0.06261331 -0.1235719 -0.1034293 8.940697e-08 -0.03
1[illegible]0324 -0.0[illegible]3077 -0.0441806 -0.06928831
faem0_si1392_2 4.677213 4.400319 4.795176 3.526529 4.637282 6.426411 7
.397112 6.217342 6.035976 7.336034 8.236745 8.184175 8.[illegible]09169 8.50935
9.612121 9.683798 10.4058 10.89597 10.55984 10.9507 11.21048 10.79408
11.63718 -0.1763853 -0.1[illegible]911 -0.8089488 -0.307582 0.196072 -0.3[illegible]70134
-0.4397185 -0.5730056 -0.3994629 -0.1466 -0.2009004 0.1417596 -0.[illegible]7178
64 -0.307953 -0.6573519 -0.2283547 -0.478344 -0.3447649 0.0219395[illegible] -0.
0599072 0.03480816 -0.[illegible]01525 -0.1672461 -0.28692 -0.1340673 -0.11[illegible]0322
 -0.02306862 -0.0798292[illegible] -0.01635064 -0.07832487 0.05356358 0.06[illegible]96881
 -0.05751204 -0.09442315 [illegible]0.02923772 -0.001794666 0.01581386 -0.016641
91 -0.008102864 -0.01395154 -0.05501553 -0.03839263 -0.048[illegible]8157 -0.101
0142 -0.005813509 -0.0769349
```

Data Format

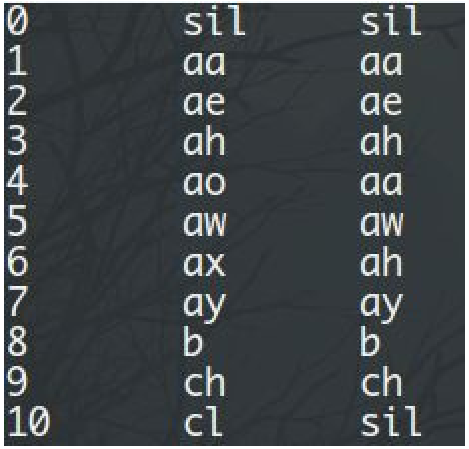
- **LAB file: Instance ID + comma + label**
- state label and 48 phones
- you have to map them to 39 phonemes **yourselves.**



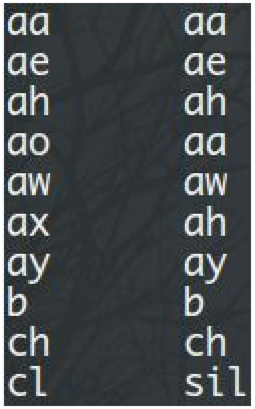
```
maeb0_si1411_1,sil
maeb0_si1411_2,sil
maeb0_si1411_3,sil
maeb0_si1411_4,sil
maeb0_si1411_5,sil
maeb0_si1411_6,sil
maeb0_si1411_7,sil
maeb0_si1411_8,sil
maeb0_si1411_9,sil
maeb0_si1411_10,sil
maeb0_si1411_11,r
maeb0_si1411_12,r
maeb0_si1411_13,r
maeb0_si1411_14,r
maeb0_si1411_15,r
maeb0_si1411_16,r
maeb0_si1411_17,r
maeb0_si1411_18,r
maeb0_si1411_19,ix
maeb0_si1411_20,ix
maeb0_si1411_21,ix
maeb0_si1411_22,ix
```

Data Format

- **MAP file: 2 mapping**
  - State ID - 48 phones - 39 phones
  - 48 phones - 39 phones
  - Delimiter: tab '\t'



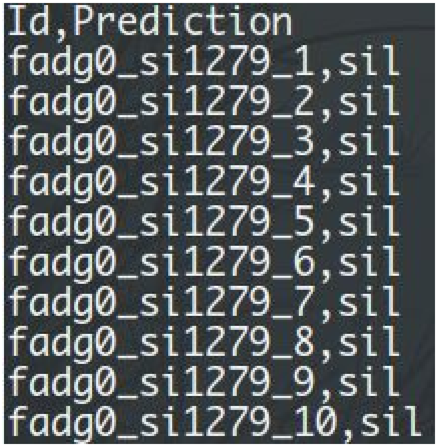
```
0	sil	sil
1	aa	aa
2	ae	ae
3	ah	ah
4	ao	aa
5	aw	aw
6	ax	ah
7	ay	ay
8	b	b
9	ch	ch
10	cl	sil
```



```
aa	aa
ae	ae
ah	ah
ao	aa
aw	aw
ax	ah
ay	ay
b	b
ch	ch
cl	sil
```

Data Format

- **CSV file**: Your prediction when submission.
- **Must mapped to 39-phonemes**
- **With header row: “Id,Prediction”**
- **Instance ID + comma + 39-phonemes**

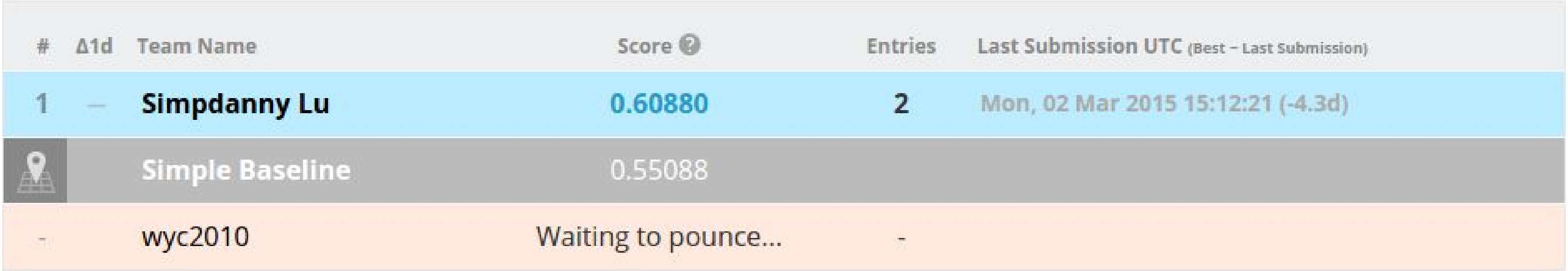


```
Id,Prediction
fadg0_si1279_1,sil
fadg0_si1279_2,sil
fadg0_si1279_3,sil
fadg0_si1279_4,sil
fadg0_si1279_5,sil
fadg0_si1279_6,sil
fadg0_si1279_7,sil
fadg0_si1279_8,sil
fadg0_si1279_9,sil
fadg0_si1279_10,sil
```

SUBMISSION REQUIREMENT

Submission(Kaggle)

- Your predict.csv.
- Twice a day.
- Public score: validation set of TIMIT
- Private score: test set of TIMIT



#	Δ1d	Team Name	Score	Entries	Last Submission UTC (Best – Last Submission)
1	—	Simpdanny Lu	0.60880	2	Mon, 02 Mar 2015 15:12:21 (-4.3d)
		Simple Baseline	0.55088		
-		wyc2010	Waiting to pounce...	-	

Submission(Ceiba)

- **Your source code and documentation**
  - Usage and environment Setting
  - Package dependency
  - **NO NN-based toolkit is allowed.**
    - Only matrix manipulation acceleration package can be utilized.
- **Report**
- Upload 1 copy per group.

Report

- **Group Information 10%**
  - Group Name, Student Name, Student ID
  - Member Contribution
- **What have you done? 40%**
  - Data structure and algorithm design
  - Data preprocessing
  - Implementation tips and obstacles
  - Bug(s) and how to solve it(them).

Report

- **Experiment Setting and Results 50%**
  - How you design your experiments
  - Compare different models/techniques
- **No more than 4 pages with font 12, A4 size.**

Grading Policy

- **Accuracy 60%**
  - Simple Baseline in Kaggle (released day 7)
  - Once achieve the baseline, you can get the full credit.
  - 1 % less absolute accuracy = 1% absolute credit loss
- **Report 40%**
- **Implementation 20%.**
- **Bonus**
  - First Place 15%
  - First Runner-up 10%
  - Second Runner-up 5%

Grading Policy

- Once you achieve simple baseline, you get 60% full credit **no matter how well your model performs**.
- Please do not get stuck in tuning learning rates or other basic parameters. However, they matter indeed.
- Your team members could be in charge of different methods to improve the results.

Grading Policy

- 40% score is based on how many methods or extensions you have tried, with the experimental **results** in comparison.
- Momentum, dropout, CNN, RBM, learning rate policies, optimization algorithms or…
- Even if your score is not improved by some specific method, **the trial is worthy and you will get credits thereby**.

Penalty

- **Latency**
  - Half-life = 24 hrs
  - The excess time will round up to hours.
  - The time is determined by ceiba upload.

- **Usage on NN-based toolkit**
  - **You shall get 0 credit in both implementation and accuracy parts.**
  - **Please ask TAs whether you could use the package or not.**

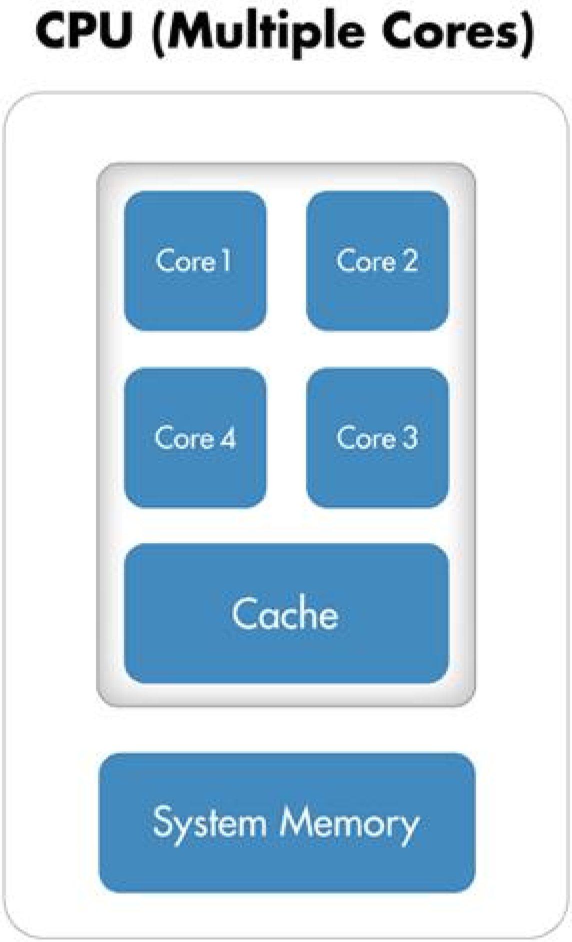
GPU Acceleration

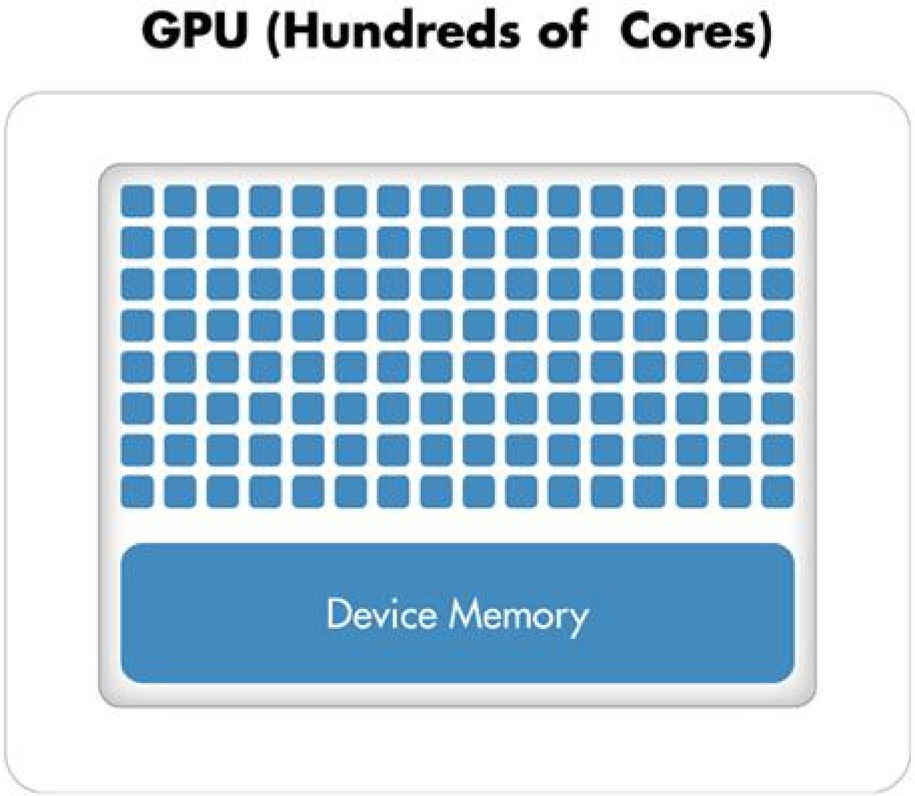
Scalability Issue

- **GPU Acceleration**
  - C/C++: CUDA library
  - Python: Theano
  - Matlab: GPU toolbox

- **Typically, acceleration lies about 5-20 times.**
  - 20 – 80 hrs -> 4 hrs
  - You got 3 weeks to have your HW1 done.

GPU Acceleration





PRO: Highly parallel with many micro floating point processors.
CON: Memory transportation bottleneck.

GPU Acceleration - C/C++

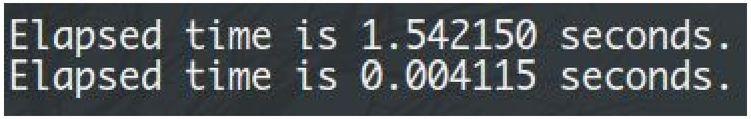
- **CUDA**
- **A good toolkit**
- **https://github.com/botonchou/libcumatrix**
- **Or write your toolkit! (Hard)**
- **https://developer.nvidia.com/cublas**

GPU Acceleration - Python

- Theano is recommended
- Follow the instructions of Theano configuration
- http://deeplearning.net/software/theano/library/config.html

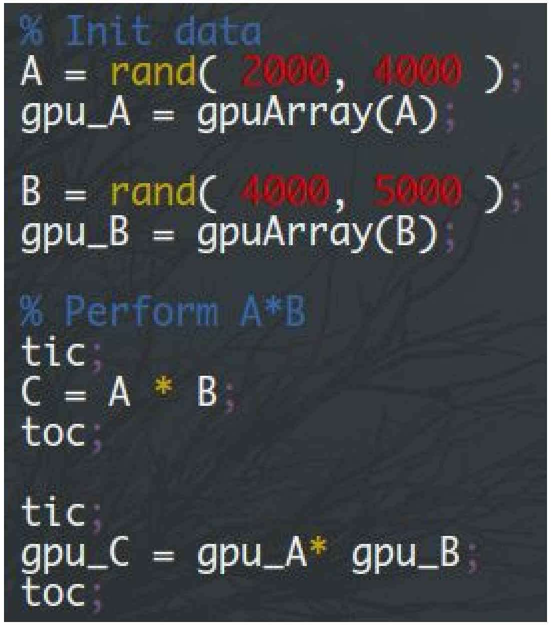
GPU Acceleration - Matlab

- **gpuArray**
- **Usage**



```
Elapsed time is 1.542150 seconds.
Elapsed time is 0.004115 seconds.
```

- Reference



```
% Init data
A = rand( 2000, 4000 );
gpu_A = gpuArray(A);

B = rand( 4000, 5000 );
gpu_B = gpuArray(B);

% Perform A*B
tic;
C = A * B;
toc;

tic;
gpu_C = gpu_A* gpu_B;
toc;
```

About Toolkits

- **Theano is recommended for you to better comprehend GPU programming with easy usage.**
- **However, theano.nnet is forbidden**
- Any package that wraps DNN together is not allowed.
- Ask if you’re unsure about the usage of some specific package.

Recommendation

- **Start as early as possible.**
  - In TA's **poor** machine(Nvidia GTX 660) , 1 epoch costs about 200 seconds.
  - 10 epochs are sufficient to achieve baseline if correctly implemented.
  - Bug is everywhere.

Tips

Tip 1

- Start-ups!
  - Group discussion and task assignments
  - Derive all formulas. Everyone.
  - Get familiar with Kaggle.
  - Which language?

Tip 2

- Understand what happened....
  - Look over training data and testing data
  - Many decision
  - Data preprocessing.
  - Basic structure and core methods implementation.
  - Test on tiny set(make it yourselves!).

Tip 3

- **Everything settled down**
  - **Start training and report testing results.**
  - **Debugging...**
  - **Can you overfit the training data?**
  - **Precision overflow? (exponential)**
  - **Matrix multiplication?**
  - **Error backpropagation?**

Tip 4

- **Basic learning techniques**
  - **Parameter selection**
    - batch size, learning rate, or...
  - **Frame shuffle?**
  - **Splice features in context window?**
  - **Regularization(weight decay)?**
  - **Learning rate decay?**

Baselines

Simple Baseline **- 0.5310**

- Once achieve this score, **you will get full credit in the grading part of accuracy**.
- However, only pure DNN is considered. Your mix learning techniques would help you better win the competition in Kaggle, but **your DNN alone should be powerful enough**.
- You must present your experiment results with pure DNN **in the report**.

Simple Baseline - **0.5310**

Input	69	fbank features
Hidden	128	only one hidden layer
Output	48	-
Batchsize	128	-
Initial Learning Rate	0.0001	-
Learning Rate Decay	0.9999	decay with each batch updating
Initialization Variance	0.1	Model initial weights gaussian variance
Momentum	0.9	-
Training Time	about 5.5 mins	-

Real Baseline - **0.6179**

- Due to the structure of the speech, **splicing contextual frames** (say previous 4 and subsequent 4) would usually helps
- Increase the model complexity with usage of **state level** pronunciation unit.
- A **real 2-3 days** training time required.

Real Baseline **- 0.6179**

Input	621	previous 4 frames + subsequent 4 frames + center 1 frame = 9 frames fbank features
Hidden	1024-1024-1024-1024	4 layers
Output	1943	state label
Batchsize	256	-
Learning Rate	0.001	-
Dropout	0.1	-
Initialization	random [-0.1. 0.1]	uniform
Momentum	0.5	-
Training Time	2-3 days	600-700 seconds per epoch