Lab4
Neural Network

Chia-Suan Yu
Po-Chih Kuo

Introduction

In this lab, you will apply neural networks to three tasks:

- **Regression:** Approximate a mathematical function.
- **Binary Classification:** Classify OCT retinal images as either normal or affected by Choroidal Neovascularization (CNV), a condition that can cause vision loss.
- **Multi-Class Classification:** Classify OCT retinal images into four categories:
  - CNV: Abnormal blood vessel growth in the eye
  - DME: Swelling of the macula due to diabetes
  - Drusen: Yellow deposits under the retina, potential sign of age-related macular degeneration
  - Normal: Healthy retina

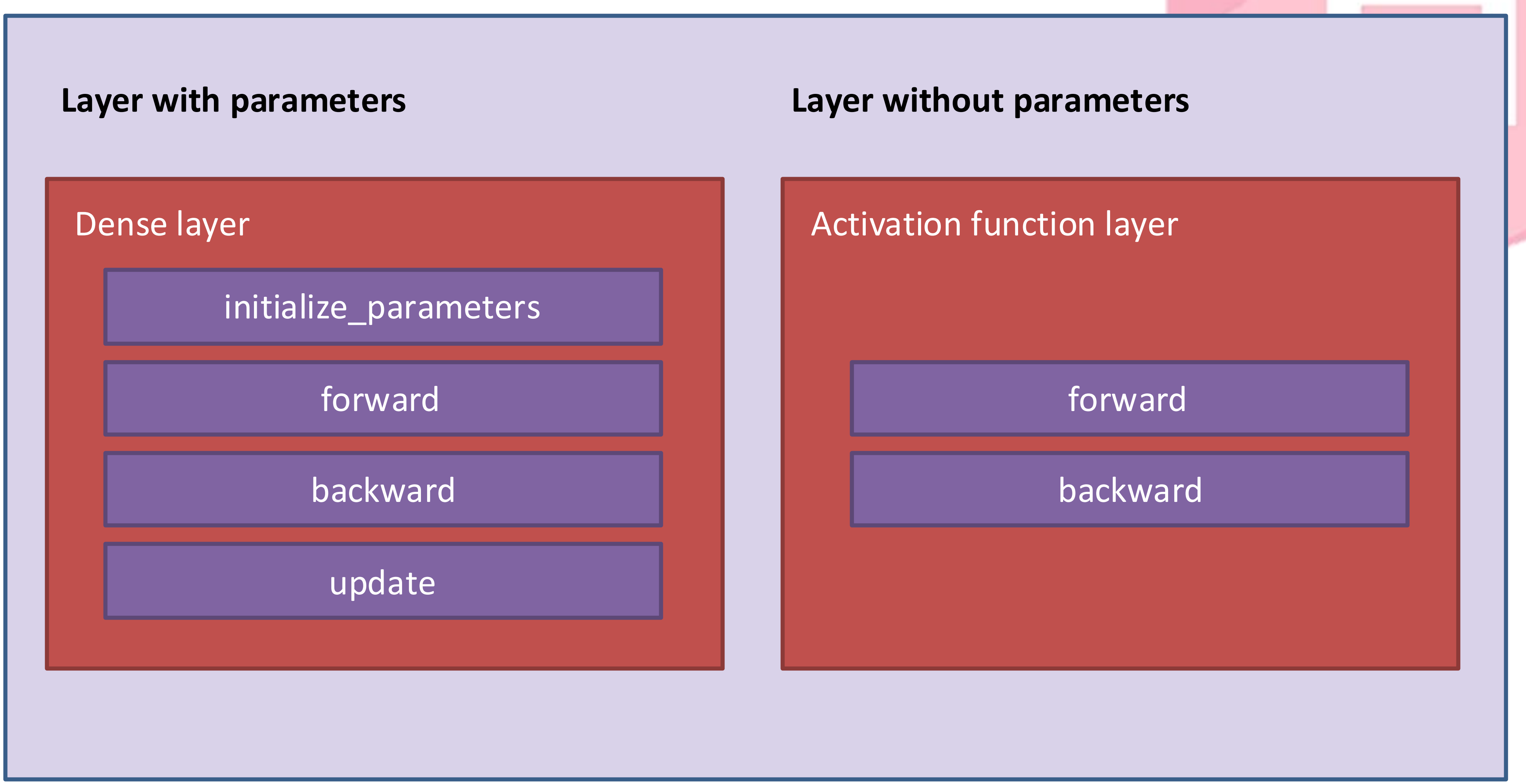
Goal

- Build your own deep neural network step by step
- Implement all the functions required to build a deep neural network
- Understanding forward propagation, backward propagation and update
- Implement Binary Cross-Entropy loss and Categorical Cross-Entropy loss
- Implement **regression** (basic part), **binary classifier** (basic part) and **multi-class classifier** (advanced part)

Grading Policy

Item	Score
Basic Implementation	65%
Advanced Implementation	30%
Basic & Advanced Report	5%

Overview



Layer with parameters

Dense layer

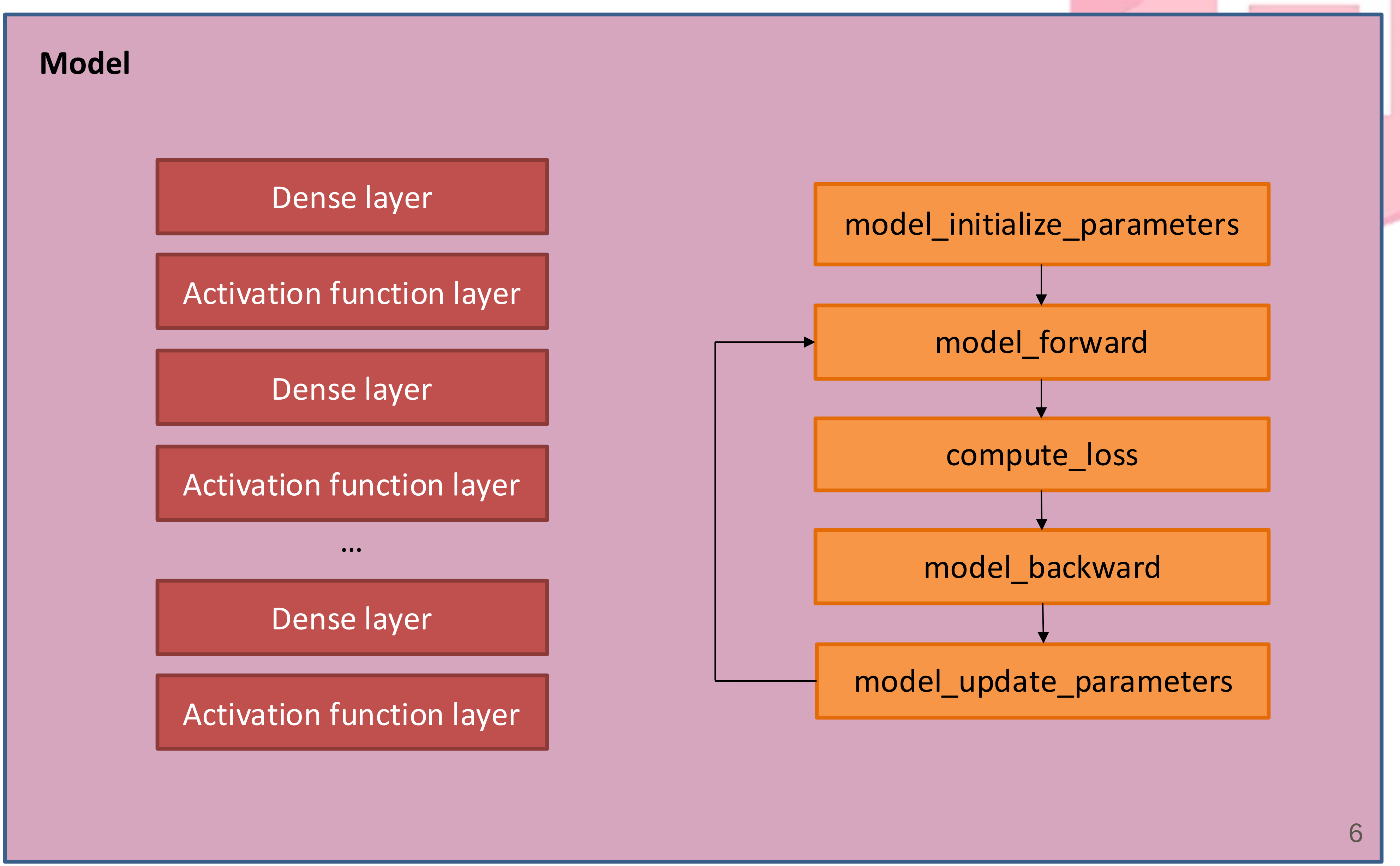
- initialize_parameters
- forward
- backward
- update

Layer without parameters

Activation function layer

- forward
- backward

Overview



Model

Dense layer

Activation function layer

Dense layer

Activation function layer

…

Dense layer

Activation function layer

model_initialize_parameters

model_forward

compute_loss

model_backward

model_update_parameters

Basic Implementation (65%)

Section 1: Neural network implementation (30%)

Part 1: Linear layer (10%)

- Step 1: Linear Initialize parameters (0%)
- Step 2: Linear forward (4%)
- Step 3: Linear backward (4%)
- Step 4: Linear update parameters (2%)

Part 2: Activation function layer (10%)

- Step 1: Activation forward (5%)
- Step 2: Activation backward (5%)

Part 3: Build model (10%)

- Step 1: Model Initialize parameters (0%)
- Step 2: Model forward (4%)
- Step 3: Model backward (4%)
- Step 4: Model update parameters (2%)

Basic Implementation (65%)

Section 2: Loss function (10%)

Part 1: Binary cross-entropy loss (5%)

Part 2: Categorical cross-entropy loss (5%)

Part 3: Mean square error (0%)

Section 3: Training and prediction (25%)

Part 1: Training function & batch function (5%)

Part 2: Regression (10%)

- Baseline : MAE <= 0.05 (5%)
- Submit "**Lab4_basic_regression.gif**" (5%)

Part 3: Binary classification (10%)

- Baseline : Public f1 score >= 0.8 (5%)
- Baseline : Private f1 score >= 0.8 (5%)

Advanced Implementation (30%)

Multi-class classification

- Baseline : Public f1 score >= 0.6 (5%)
- Baseline : Private f1 score >= 0.6 (10%)
- Private Ranking (15%)

Loss function and Activation function

Warning: only the following 3 combinations are allowed!

1. Regression : linear + mse
2. Binary classification: sigmoid + cross_entropy
3. Multi-class classification: softmax + cross_entropy

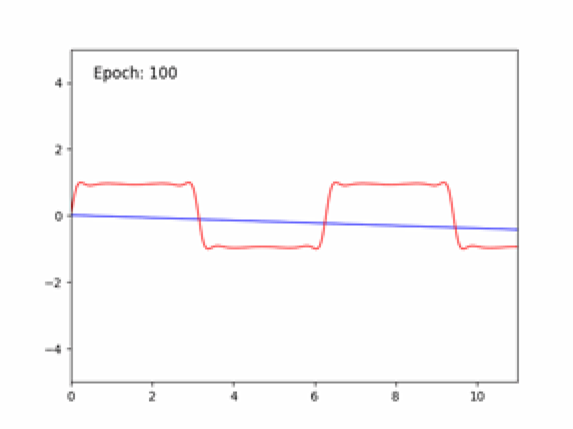
Data (Simulation data)

Regression: Math function approximation

The target function to approximate is:

$$y = \sin(2 * \sin(2 * \sin(2 * \sin(x))))$$

where x is in the range [0.01, 11]



Data (OCT scans)

Binary classification: OCT scan images of retina

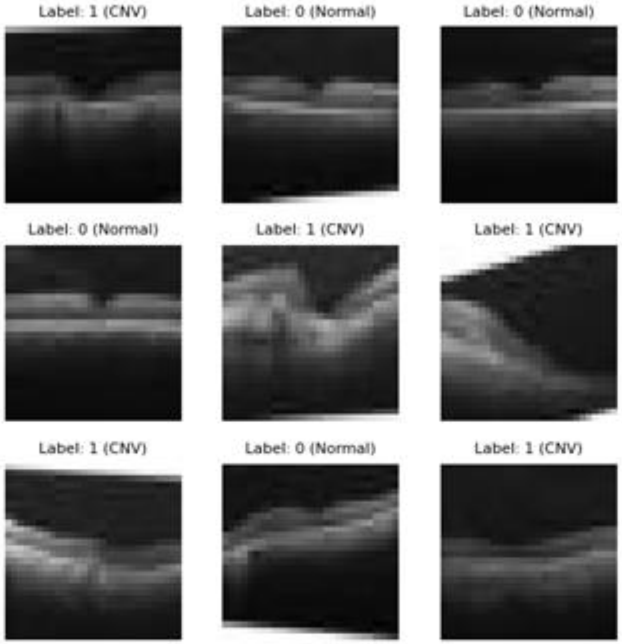
The dataset consists of 28x28 pixels grayscale OCT scan images of the retina, focusing on two classes: CNV (Choroidal Neovascularization) and Normal.

Details of the dataset:

- shape of x_train: (20000, 28, 28)
- shape of y_train: (20000, 1)
- shape of x_test: (5000, 28, 28)

Classes:

- CNV: label = 1
- Normal: label = 0



Data (OCT scans)

Multi-class classification: OCT scan images of retina

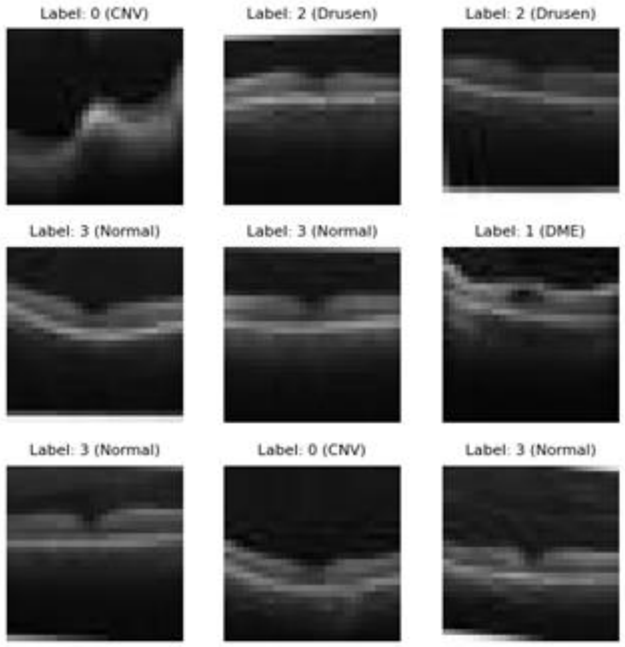
The dataset consists of 28x28 pixels grayscale OCT scan images of the retina, categorized into four classes: CNV (Choroidal Neovascularization), DME (Diabetic Macular Edema), Drusen, and Normal.

Details of the dataset:

- shape of x_train: (37754, 28, 28)
- shape of y_train: (37754,)
- shape of x_test: (3000, 28, 28)

Classes:

- CNV: label = 0
- DME: label = 1
- Drusen: label = 2
- Normal: label = 3



Output .csv file format

Basic Part : regression

There should be (1000+1) rows in your csv file

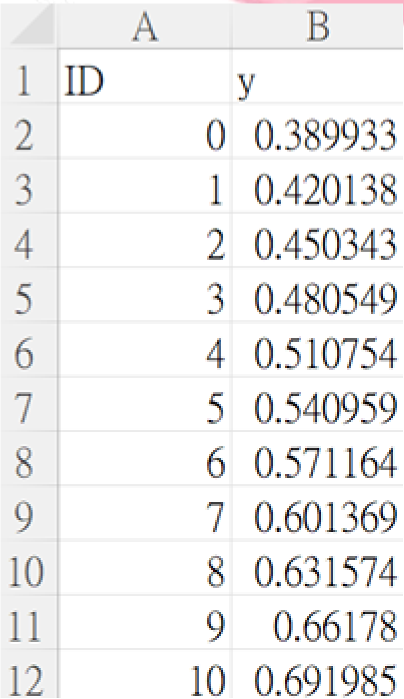
First row is the header [‘ID’, ’y’]

ID starts from 0, and **y** is the predicted y value

Please make sure that your output format is correct

Submit the answer (Lab4_basic_regression.csv) to Kaggle

ML2024-Lab4-BasicPart-Reg



	A	B
1	ID	y
2	0	0.389933
3	1	0.420138
4	2	0.450343
5	3	0.480549
6	4	0.510754
7	5	0.540959
8	6	0.571164
9	7	0.601369
10	8	0.631574
11	9	0.66178
12	10	0.691985

Output .csv file format

Basic Part : binary classifier

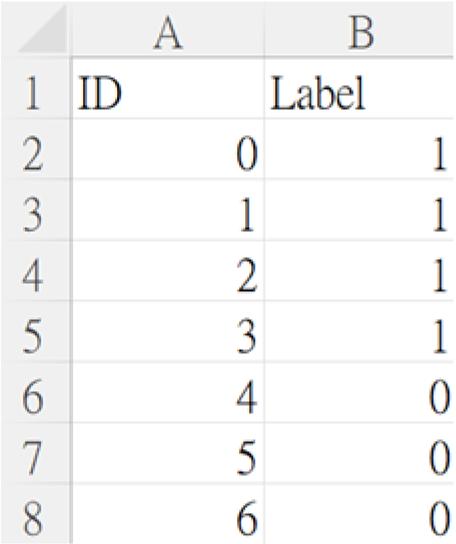
There should be (5000+1) rows in your csv file

- First row is the header [‘ID’, 'Label']
- Your prediction answer should be either 0 or 1
- **ID** starts from 0, and **Label** is the predicted answer

Please make sure that your output format is correct

Submit the answer (Lab4_basic.csv) to Kaggle

ML2024-Lab4-BasicPart-B.C.



	A	B
1	ID	Label
2	0	1
3	1	1
4	2	1
5	3	1
6	4	0
7	5	0
8	6	0

Output .csv file format

Advanced Part : multi-class classifier

There should be (3000+1) rows in your csv file

First row is the header [‘ID’, 'Label']

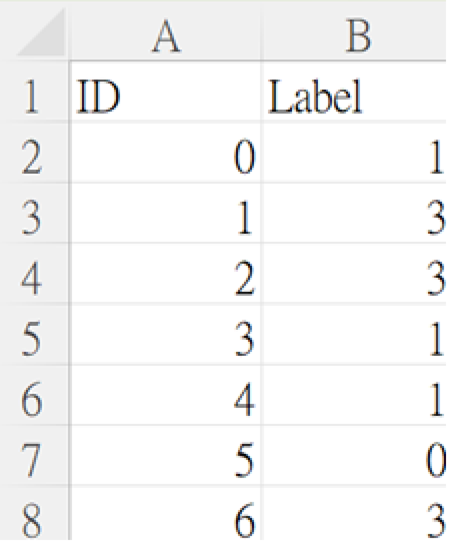
Your prediction answer should be (0~3)

ID starts from 0, and **Label** is the predicted answer

Please make sure that your output format is correct

Submit the answer (Lab4_advanced.csv) to Kaggle

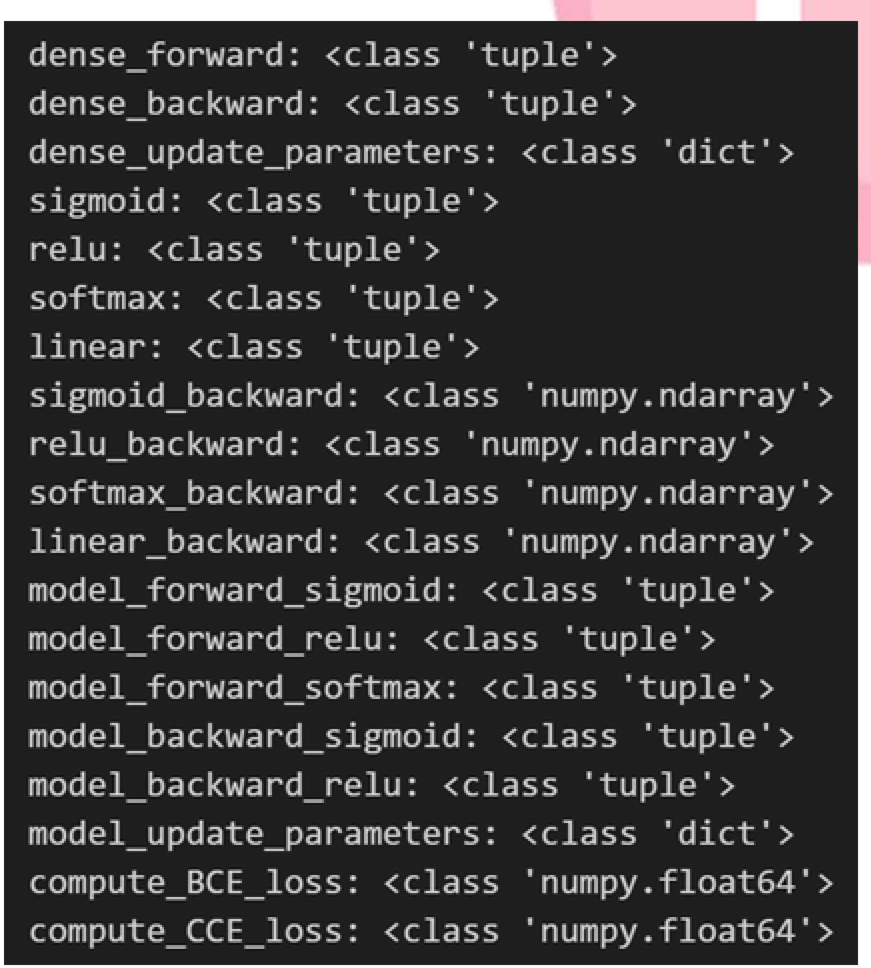
ML2024-Lab4-AdvancedPart



	A	B
1	ID	Label
2	0	1
3	1	3
4	2	3
5	3	1
6	4	1
7	5	0
8	6	3

Output .npy File Format

- Named as “**Lab4_output.npy**”
- This file is a dictionary that stores your output for each function.
- We will test your ”**Lab4_output.npy**” to verify the correctness of your neural networks.

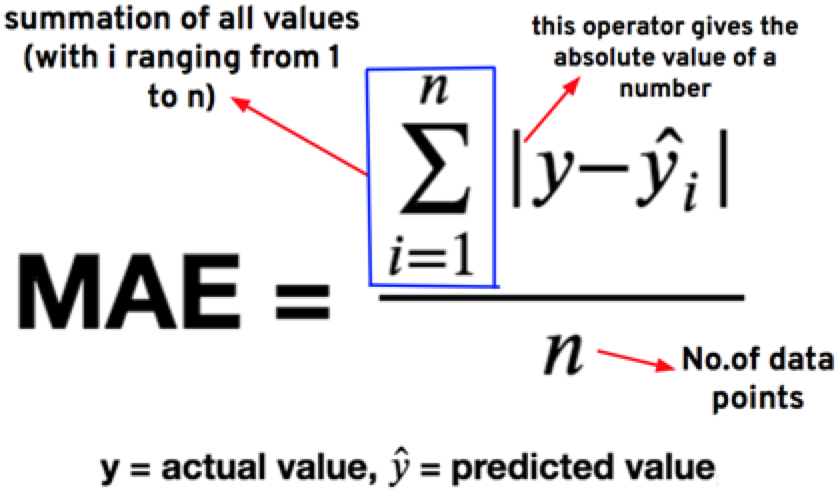


```
dense_forward: <class 'tuple'>
dense_backward: <class 'tuple'>
dense_update_parameters: <class 'dict'>
sigmoid: <class 'tuple'>
relu: <class 'tuple'>
softmax: <class 'tuple'>
linear: <class 'tuple'>
sigmoid_backward: <class 'numpy.ndarray'>
relu_backward: <class 'numpy.ndarray'>
softmax_backward: <class 'numpy.ndarray'>
linear_backward: <class 'numpy.ndarray'>
model_forward_sigmoid: <class 'tuple'>
model_forward_relu: <class 'tuple'>
model_forward_softmax: <class 'tuple'>
model_backward_sigmoid: <class 'tuple'>
model_backward_relu: <class 'tuple'>
model_update_parameters: <class 'dict'>
compute_BCE_loss: <class 'numpy.float64'>
compute_CCE_loss: <class 'numpy.float64'>
```

Evaluation Metric

Regression

- MAE (mean absolute error)



summation of all values (with i ranging from 1 to n)

this operator gives the absolute value of a number

$$\text{MAE} = \frac{\boxed{\sum_{i=1}^{n}} |y - \hat{y}_i|}{n}$$

No.of data points

y = actual value, $\hat{y}$ = predicted value

Evaluation Metric

Classification

- F1-score

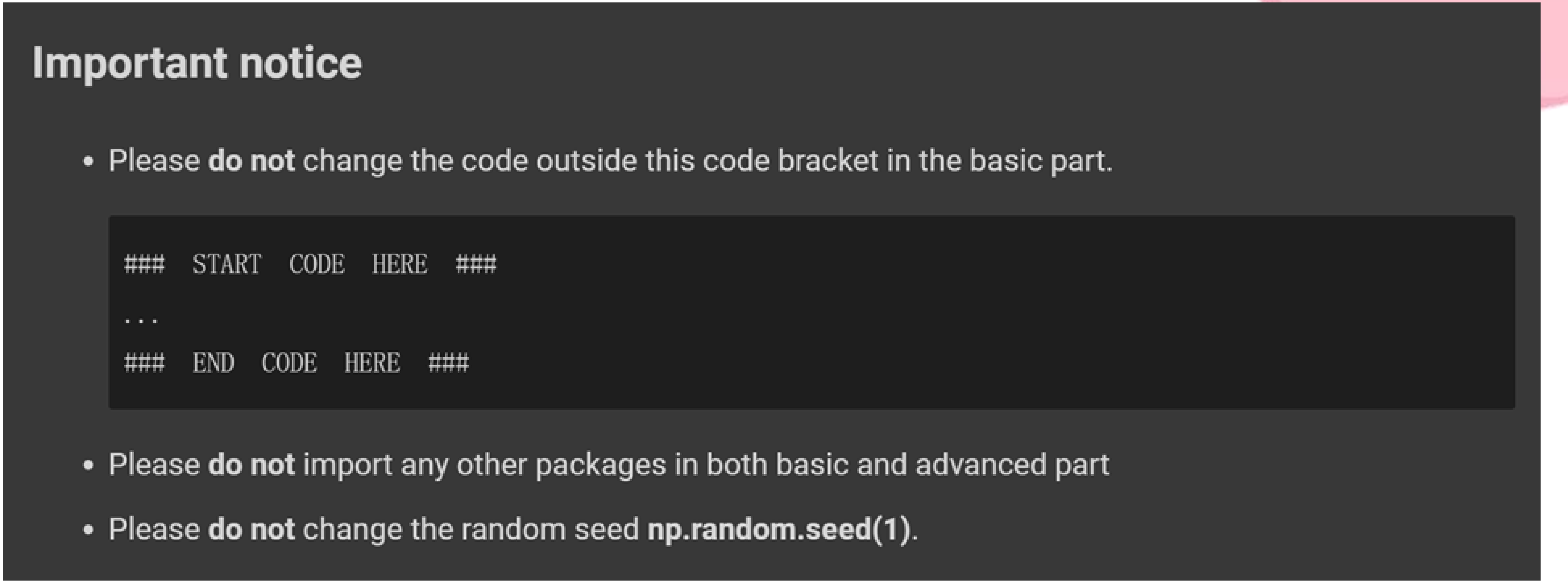
$$F1\text{-}score = 2 \times \frac{(\boldsymbol{Precision} \times \boldsymbol{Recall})}{(\boldsymbol{Precision} + \boldsymbol{Recall})}$$

		Actual/True value	
		positive	negative
Predicted value	positive	**TP**	**FP**
	negative	**FN**	**TN**

Given Items

- Template: Lab4_template.ipynb
- Basic data (binary classifier): basic_data.npz
- Advanced data: advanced_data.npz

Template



Important notice

- Please **do not** change the code outside this code bracket in the basic part.

```
### START CODE HERE ###
...
### END CODE HERE ###
```

- Please **do not** import any other packages in both basic and advanced part
- Please **do not** change the random seed **np.random.seed(1)**.

Remember to save the code file to **Lab4.ipynb**

Kaggle

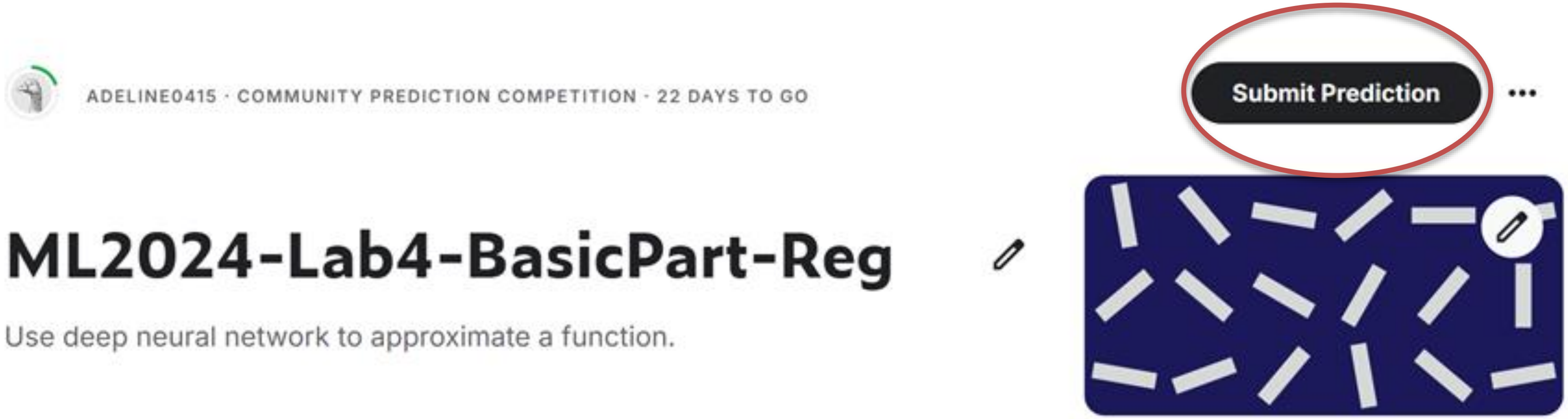
We've created competitions for 3 tasks respectively.

- Basic regression link:
  https://www.kaggle.com/competitions/ml-2024-lab-4-basic-part-fa
- Basic binary classification link:
  https://www.kaggle.com/competitions/ml-2024-lab-4-basic-part-bc
- Advanced link:
  https://www.kaggle.com/competitions/ml-2024-lab-4-advanced-part

- For regression, only **public** data are provided.
- For both binary and multi-class classification tasks, we split the testing data randomly into **public** (50%) and **private** (50%) parts, maintaining the same class distribution ratio.
- Only the public score will be visible on Kaggle.

Kaggle

- Please register your account.
- Click the ‘Join competition’ button to join.



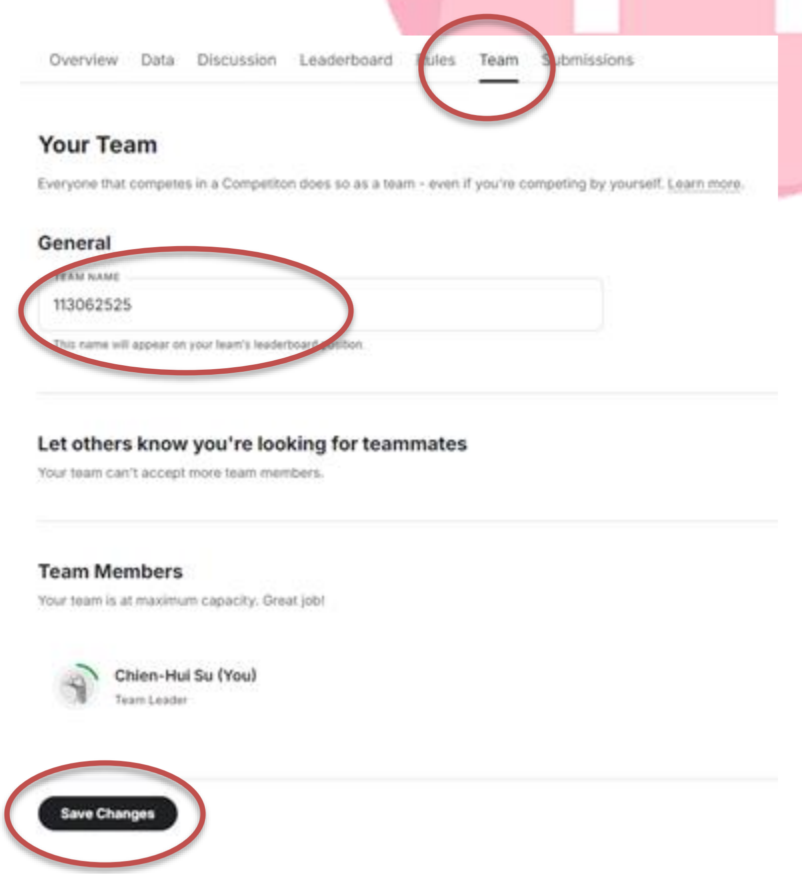
ADELINE0415 · COMMUNITY PREDICTION COMPETITION · 22 DAYS TO GO

ML2024-Lab4-BasicPart-Reg

Use deep neural network to approximate a function.

Kaggle

- After joining the competition, you should change your team name (each student is a team) to your **student ID**.
- **Please remember to SAVE CHANGES**
- You can submit 50 times per day.



Your Team

Everyone that competes in a Competiton does so as a team - even if you're competing by yourself. Learn more.

General

TEAM NAME: 113062525

This name will appear on your team's leaderboard position.

Let others know you're looking for teammates

Your team can't accept more team members.

Team Members

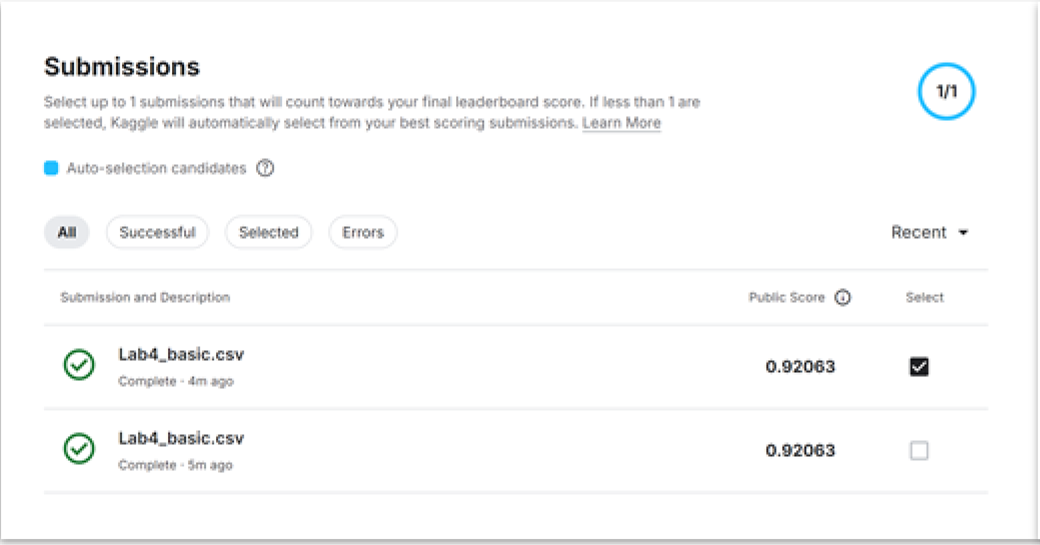
Your team is at maximum capacity. Great job!

Chien-Hui Su (You)
Team Leader

Notes: Please verify your team name on the leaderboard - changing profile name does not change team name.

Kaggle

You can manually select up to 1 submission that will count towards your final leaderboard score. If no submission is selected, Kaggle will automatically select your submission with the best public score.

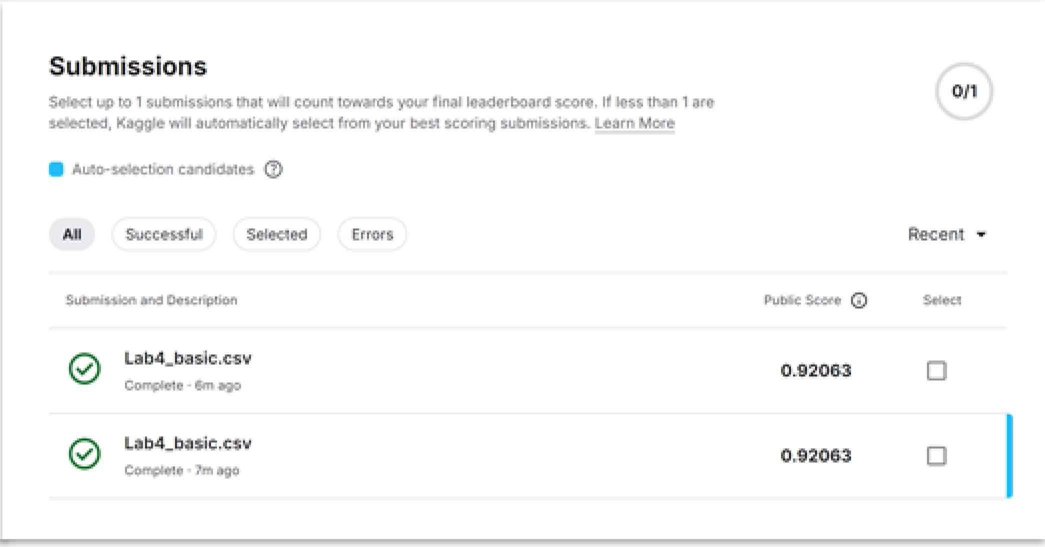


Submissions

Select up to 1 submissions that will count towards your final leaderboard score. If less than 1 are selected, Kaggle will automatically select from your best scoring submissions. Learn More

Submission and Description	Public Score ⓘ	Select
✓ Lab4_basic.csv Complete · 4m ago	0.92063	☑
✓ Lab4_basic.csv Complete · 5m ago	0.92063	□

Manual-selection



Submissions

Select up to 1 submissions that will count towards your final leaderboard score. If less than 1 are selected, Kaggle will automatically select from your best scoring submissions. Learn More

Submission and Description	Public Score ⓘ	Select
✓ Lab4_basic.csv Complete · 6m ago	0.92063	□
✓ Lab4_basic.csv Complete · 7m ago	0.92063	□

Auto-selection

Basic & Advanced Report (5%)

1. What are the key differences between sigmoid and softmax activation functions, and why did we choose them for binary and multi-class classification respectively? (1%)
2. Why does the loss oscillate during model training? (list at least 2 reasons) (2%)
3. How does changing the learning rate and batch size affect model training time? (1%)
4. Put your regression results (*lab4_basic_regression.jpg*) on report. (1%)

Notes:

1. Do not exceed 1 page!
2. Name your report file as "**Lab4_report.pdf**".

Requirement

- Do it individually! Not as a team! (team is for final project)
- Announce date: 2024/10/31
- Deadline: 2024/11/12 23:59 (Late submission is not allowed!)
- Submit the answers (csv) to corresponding Kaggle competition.
  - **ML2024-Lab4-BasicPart-Reg**
  - **ML2024-Lab4-BasicPart-B.C.**
  - **ML2024-Lab4-AdvancedPart**
- Hand in following files to eeclass in the following format (Do not compressed!)
  - Lab4.ipynb
  - Lab4_report.pdf
  - Lab4_basic_regression.gif
  - Lab4_output.npy

Penalty

0 points if any of the following conditions happened

- Plagiarism
- Late submission
- Not using a template or importing any other packages
- No submission record on Kaggle (we cannot identify who you are)
- Wrong team name on Kaggle (we cannot identify who you are)
- No code(“**Lab4.ipynb**”) submission on eeclass
- Your submission was not generated by your code

5 Points would be deducted if your submission format is incorrect

0 Points will be given in the Basic section 1&2 if you don’t submit **“Lab4_output.npy”**

Questions?

- TA: Chia-Suan Yu 余佳軒 (adeline041503@gmail.com)
- No debugging service