

Lab 9

Artificial Neural Network

Nov 9, 2023

Department of Computing
The Hong Kong Polytechnic University

Today's Arrangement

- Assignment 5 Check
- Neural Network Playground
- Exercise 1
- Introduction to Pytorch
- Exercise 2
- Assignment 6

Assignment 5 Check

Assignment 5

Online news is crucial in providing people with diverse, multifaceted perspectives on political and public issues. As an expert in artificial intelligence, an online news website seeks your assistance in predicting the popularity of online news based on its features (as shown in the table below).

The dataset contains a set of features describing published online news. The goal is to forecast their popularity on social networks. The articles with more than 1400 shares can be as popular. Your task is to predict whether a piece of online news will be popular.

Variable Name	Role	Type	Description
Feature 1	Feature	Continuous	Number of words in the title
Feature 2	Feature	Continuous	Number of words in the content
Feature 3	Feature	Continuous	Rate of unique words in the content
Feature 4	Feature	Continuous	Rate of non-stop words in the content
Feature 5	Feature	Continuous	Rate of unique non-stop words in the content
Feature 6	Feature	Continuous	Number of links
Feature 7	Feature	Continuous	Number of links to other articles
Feature 8	Feature	Continuous	Number of images
Feature 9	Feature	Continuous	Number of videos
Feature 10	Feature	Continuous	Average length of the words in the content
Feature 11	Feature	Continuous	Number of keywords in the metadata
Feature 12	Feature	Categorical	Is the article from the Lifestyle topic?
Feature 13	Feature	Categorical	Is the article from the Entertainment topic?
Feature 14	Feature	Categorical	Is the article from the Business topic?
Feature 15	Feature	Categorical	Is the article from the Social Media topic?
Feature 16	Feature	Categorical	Is the article from the Tech topic?
Feature 17	Feature	Categorical	Is the article from the World topic?
Feature 18	Feature	Continuous	Min. shares of worst keyword
Feature 19	Feature	Continuous	Max. shares of worst keyword
Feature 20	Feature	Continuous	Avg. shares of worst keyword
Feature 21	Feature	Continuous	Min. shares of best keyword
Feature 22	Feature	Continuous	Max. shares of best keyword
Feature 23	Feature	Continuous	Avg. shares of best keyword
Feature 24	Feature	Continuous	Min. shares of avg. keyword
Feature 25	Feature	Continuous	Max. shares of avg. keyword
Feature 26	Feature	Continuous	Avg. shares of avg. keyword

Assignment 5

Online news is crucial in providing people with diverse, multifaceted perspectives on political and public issues. As an expert in artificial intelligence, an online news website seeks your assistance in predicting the popularity of online news based on its features (as shown in the table below).

The dataset contains a set of features describing published online news. The goal is to forecast their popularity on social networks. The articles with more than 1400 shares can be as popular. Your task is to predict whether a piece of online news will be popular.

Variable Name	Role	Type	Description
Feature 27	Feature	Continuous	Min. shares of referenced articles
Feature 28	Feature	Continuous	Max. shares of referenced articles
Feature 29	Feature	Continuous	Avg. shares of referenced articles
Feature 30	Feature	Categorical	Was the article published on a Monday?
Feature 31	Feature	Categorical	Was the article published on a Tuesday?
Feature 32	Feature	Categorical	Was the article published on a Wednesday?
Feature 33	Feature	Categorical	Was the article published on a Thursday?
Feature 34	Feature	Categorical	Was the article published on a Friday?
Feature 35	Feature	Categorical	Was the article published on a Saturday?
Feature 36	Feature	Categorical	Was the article published on a Sunday?
Feature 37	Feature	Categorical	Was the article published on the weekend?
Feature 38	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 0
Feature 39	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 1
Feature 40	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 2
Feature 41	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 3
Feature 42	Feature	Continuous	Closeness to Latent Dirichlet Allocation (LDA) topic 4
Feature 43	Feature	Continuous	Text subjectivity
Feature 44	Feature	Continuous	Text sentiment polarity
Feature 45	Feature	Continuous	Rate of positive words in the content
Feature 46	Feature	Continuous	Rate of negative words in the content
Feature 47	Feature	Continuous	Rate of positive words among non-neutral tokens
Feature 48	Feature	Continuous	Rate of negative words among non-neutral tokens
Feature 49	Feature	Continuous	Avg. polarity of positive words
Feature 50	Feature	Continuous	Min. polarity of positive words
Feature 51	Feature	Continuous	Max. polarity of positive words
Feature 52	Feature	Continuous	Avg. polarity of negative words

Assignment 5

Online news is crucial in providing people with diverse, multifaceted perspectives on political and public issues. As an expert in artificial intelligence, an online news website seeks your assistance in predicting the popularity of online news based on its features (as shown in the table below).

The dataset contains a set of features describing published online news. The goal is to forecast their popularity on social networks. The articles with more than 1400 shares can be as popular. Your task is to predict whether a piece of online news will be popular.

Variable Name	Role	Type	Description
Feature 53	Feature	Continuous	Min. polarity of negative words
Feature 54	Feature	Continuous	Max. polarity of negative words
Feature 55	Feature	Continuous	Subjectivity of the title
Feature 56	Feature	Continuous	Sentiment polarity of the title
Feature 57	Feature	Continuous	Absolute level of subjectivity in the title
Feature 58	Feature	Continuous	Absolute level of sentiment polarity in the title
Label	Label	Categorical	0,1 (0 for not popular and 1 for popular)

Assignment 5

In the public dataset, you can train and validate your model on 30,000 samples. Then, you need to predict the labels for 5,000 samples in the private dataset, and your performance on the private dataset will determine your final score.

Hint 1: Cross-validation is important.

Hint 2: Consider preprocessing and feature engineering if it benefits your model.

Hint 3: Optimize hyperparameters for improved performance.

Hint 4: Utilize any algorithms you have learned, including the decision tree, K-nearest neighbor, support vector machine, etc. You may ensemble their predictions to achieve better performance.

Assignment 5

- Please download the public dataset from
https://drive.google.com/file/d/1FoXCntlw_0DFI3VzUVXL AoSwUHYlzhZ5/view?usp=sharing
- Please download the private dataset from
<https://drive.google.com/file/d/1SxEYOYIdSPbAlzCcGp-jGjAkXzgC6FZ6/view?usp=sharing>
- Please use the following Python template for submission. (You can copy the code below from lab8-Exercise.ipynb)
- Your results will be evaluated on 5000 samples in the private dataset, using classification accuracy (The labels will released in Lab 9).

Assignment 5

```
def read_data_from_csv(path):
    """Load datasets from CSV files.
Args:
    path (str): Path to the CSV file.
Returns:
    X (np.ndarray): Features of samples.
    y (np.ndarray): Labels of samples, only provided in the public
datasets.
"""
assert os.path.exists(path), f'File not found: {path}!'
assert os.path.splitext(path)[-1] == '.csv', f'Unsupported file type {os.path.splitext(path)[-1]}!'

data = pd.read_csv(path)
column_list = data.columns.values.tolist()

if 'Label' in column_list:
    # for the public dataset, label column is provided.
    column_list.remove('Label')
    X = data[column_list].values
    y = data['Label'].astype('int').values
    return X, y
else:
    # for the private dataset, label column is not provided.
    X = data[column_list].values
    return X
```

```
X_public, y_public = read_data_from_csv('assignment_5_public.csv')
print('Shape of X_public:', X_public.shape) # n_sample, m_feature
(30000, 58)
print('Shape of y_public:', y_public.shape) # n_sample (30000,)

"""
CODE HERE!
"""

X_private = read_data_from_csv('assignment_5_private.csv')
print('Shape of X_private:', X_private.shape) # k_sample, m_feature
(5000, 58)

import numpy as np

# remove and make your own predictions.
preds = np.full(len(X_private), -1,
                dtype=int)
"""
CODE HERE!
e.g.,
preds = np.full(len(X_private), -1, dtype=int)
"""

submission = pd.DataFrame({'Label': preds})
submission.to_csv('assignment_5.csv', index=True, index_label='Id')
```

Assignment 5 Check

- Check your answer:
 - **Assignment5_Answer.ipynb:** available on Blackboard or
<https://drive.google.com/file/d/1001Y-74tT4UT0vbQAigN9JRSdishEir/view?usp=sharing>
 - **Label of private dataset:** available on Blackboard or
https://drive.google.com/file/d/1u0Gk_UBNug13WM-XIYAhyzY65TnRDtjm/view?usp=sharing
- Some suggestions/comments:
 - Preprocessing: for discrete data, apply one-hot encoding, and for continuous data, apply normalization.
 - Grid search can be utilized to find the optimal combination of hyper-parameters. Use cross-validation to verify the performance of the model.
 - Analyze the performance of different models and perform ensemble. Models with better performance should have higher weights, while intersecting models should have lower weights (instead of discarding them).
 - When dealing with large datasets, models may run slowly. You will need tricks to achieve acceptable performance within a limited time. For instance, you can use Linear SVM and set max_iter, etc.



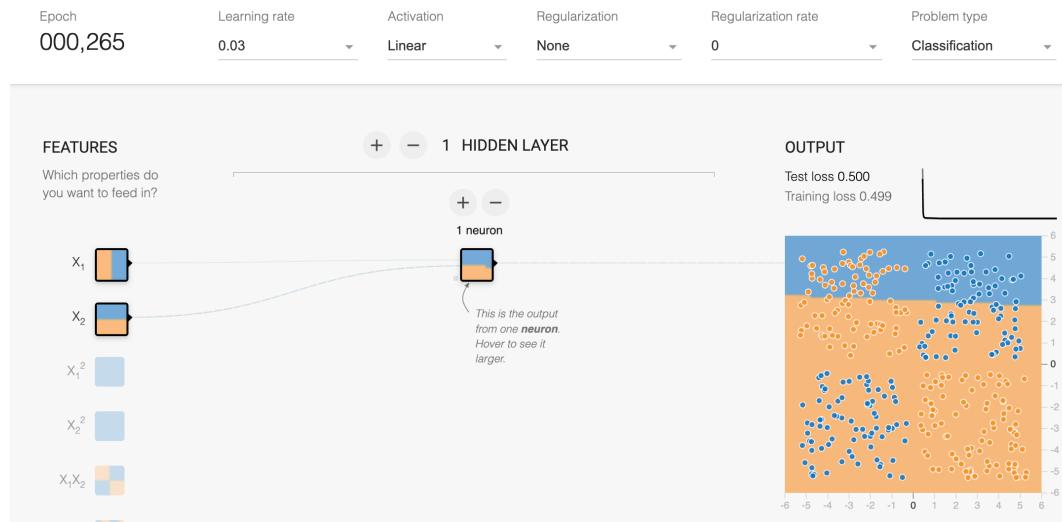
Neural Network Playground

Reference: <https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/playground-exercises>

Neural Network Playground

Play with the online demo “Neural Network Playground” via [this link](#):

- Example Model 1: Two features, one hidden layer, one neuron, and “Linear activation”. (“Linear activation” means that the activation is proportional to the input.) **Will this model learn any nonlinearities? Run it to confirm your guess.**

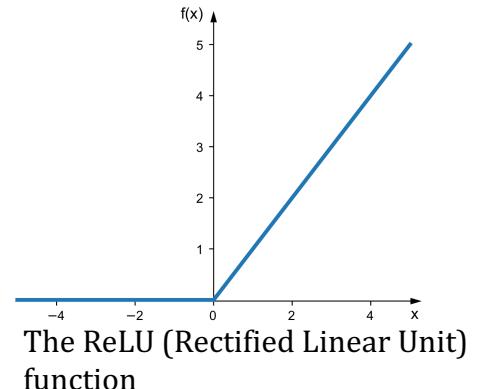
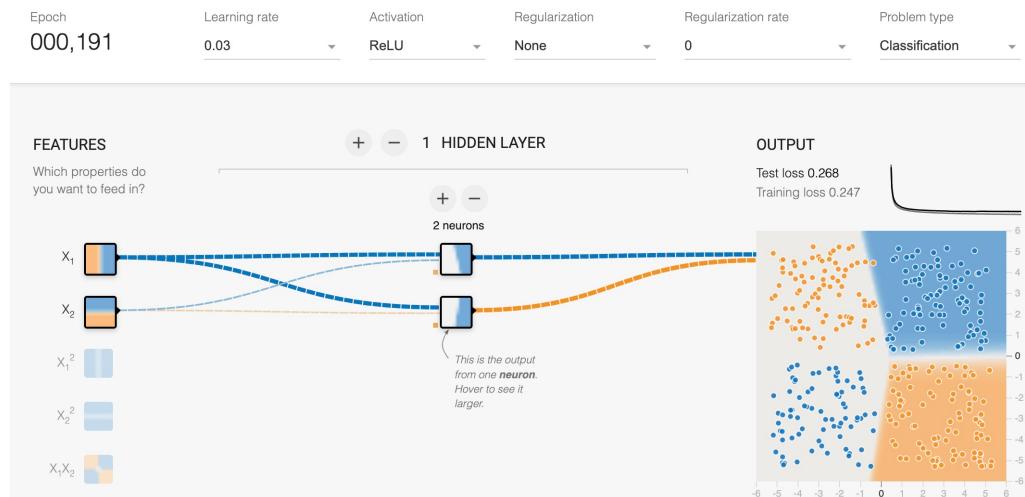


Findings:

- This model cannot learn any nonlinearity.
- For the dataset not linearly separable, nonlinear activation is needed.

Neural Network Playground

- Example Model 2: Now increase the number of neurons in the hidden layer from 1 to 2. Then, change Linear activation to a nonlinear activation like ReLU. **Can this model learn nonlinearities? Can this one model the data effectively?**

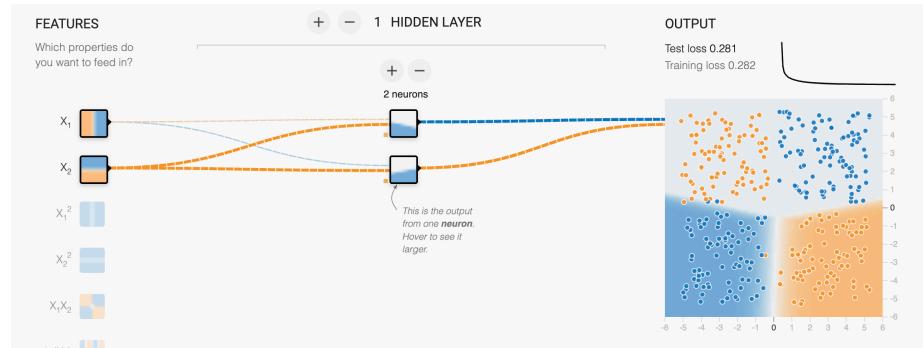
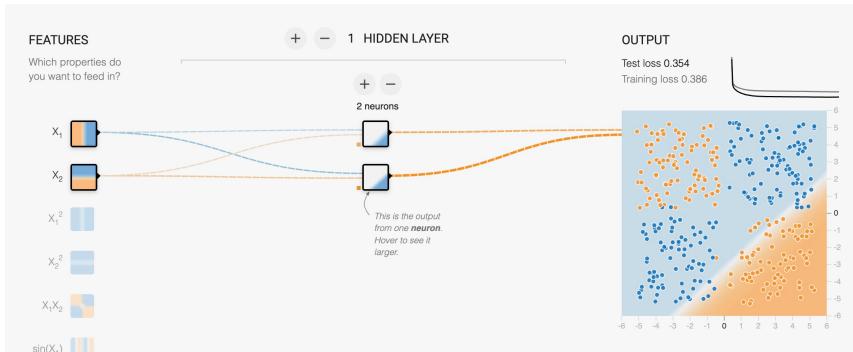


Findings:

- This model can learn nonlinear data. However, it cannot reflect all the nonlinearities in this data set. (2 neurons are not enough.)
- The outputs are nondeterministic (due to different initial weights).

Neural Network Playground

- Example Model 2: Now increase the number of neurons in the hidden layer from 1 to 2. Then, change Linear activation to a nonlinear activation like ReLU. **Can this model learn nonlinearities? Can this one model the data effectively?**



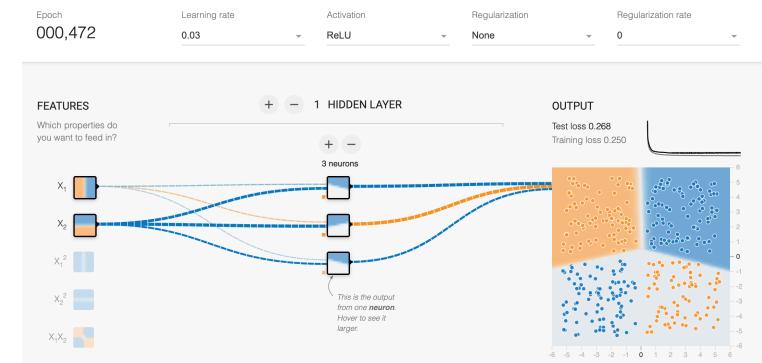
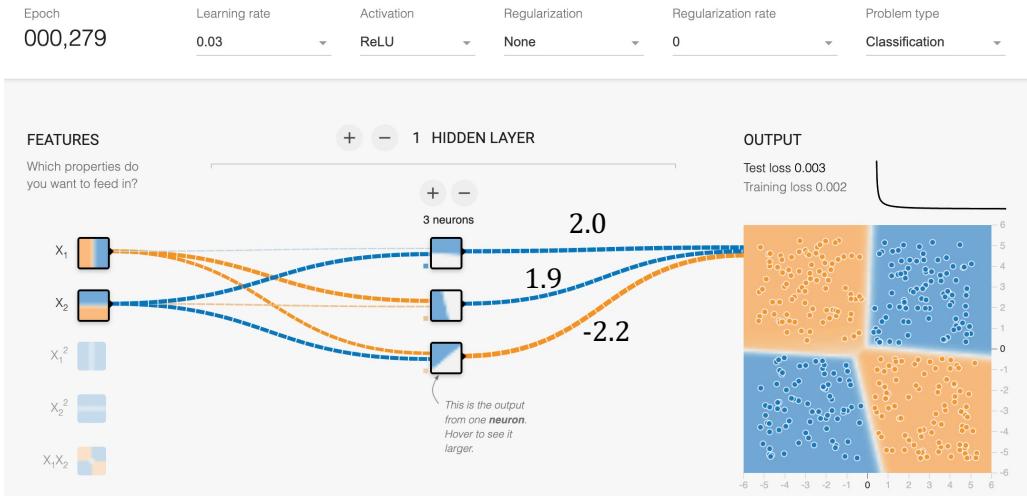
The outputs are nondeterministic.

Findings:

- This model can learn nonlinear data. However, it cannot reflect all the nonlinearities in this data set. (2 neurons are not enough.)
- The outputs are nondeterministic (due to different initial weights).

Neural Network Playground

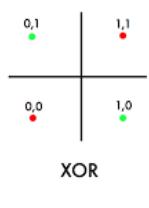
- Example Model 3: Now increase the number of neurons in the hidden layer from 2 to 3. **Can this model learn all the nonlinearities in this data set?**



Not all runs will converge to a good model.

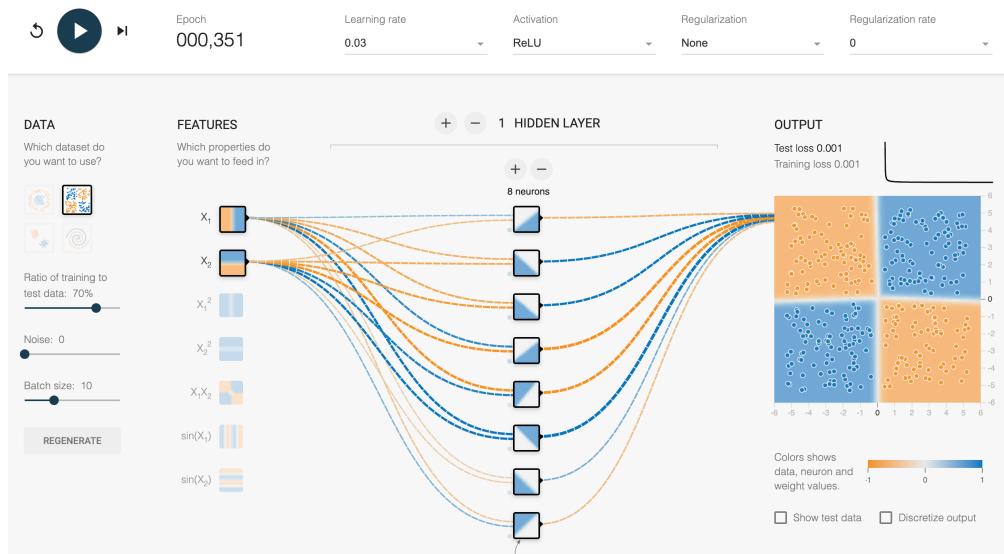
Findings:

- A single hidden layer with 3 neurons is enough to model the data set. (3 neurons are enough because the XOR function can be expressed as a combination of 3 half-planes (ReLU activation).)
- Not all runs will converge to a good model. (Some neurons may have similar output.) More neurons are needed to provide more redundancy, making the model more likely to converge to a good mode.



Neural Network Playground

- Example Model 4: Now increase the number of neurons in the hidden layer to 8.



By now, we have learned:

- With only linear activation functions, the approximation ability of the network is very limited. (In fact, stacking many linear layers is equivalent to just a single linear layer.)
- The nonlinear activation functions provide the ability to learn nonlinearities.
- By adding enough neurons in the first hidden layer (and the other layers), we create enough space for the model to fit the data.

Exercise 1

Exercise 1: Fine-tune the network

Continue experimenting by changing the dataset and network freely:

- Try to fit all four datasets.
- Add or remove input features.
- Add or remove hidden layers and neurons per layer.
- Change learning rates, regularization, batch size, and other learning settings.

You can work on the below tasks:

1. On each dataset, start with a simple network structure (such as one hidden layer and three neurons) and observe the repeatability of your model.
 2. What is the smallest number of neurons and layers you can use to perform well on each dataset? Which dataset is the most difficult one to fit? What input features are needed?
 3. Add some noise to your dataset and try again. Also, test the influences of the hyperparameters such as learning rate, batch size, regularization, and so on.
-

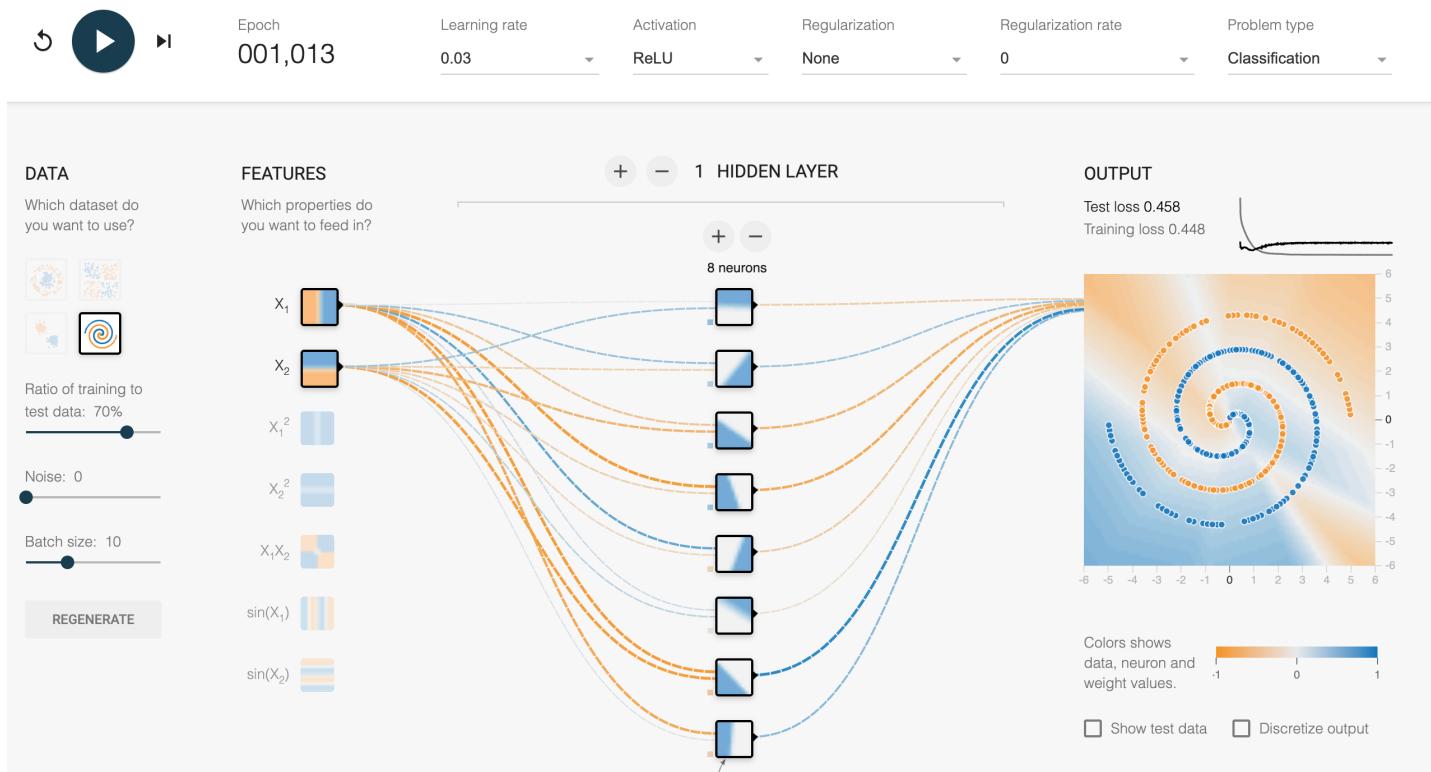
Exercise 1: Fine-tune the network

- Learning rates: The step size at each iteration while optimizing the network.
 - Regularization: A technique to prevent overfitting by keeping network weights small.
 - Batch size: The number of training samples used in one forward/backward pass of the neural network.
-

Example 1: The Spiral Data

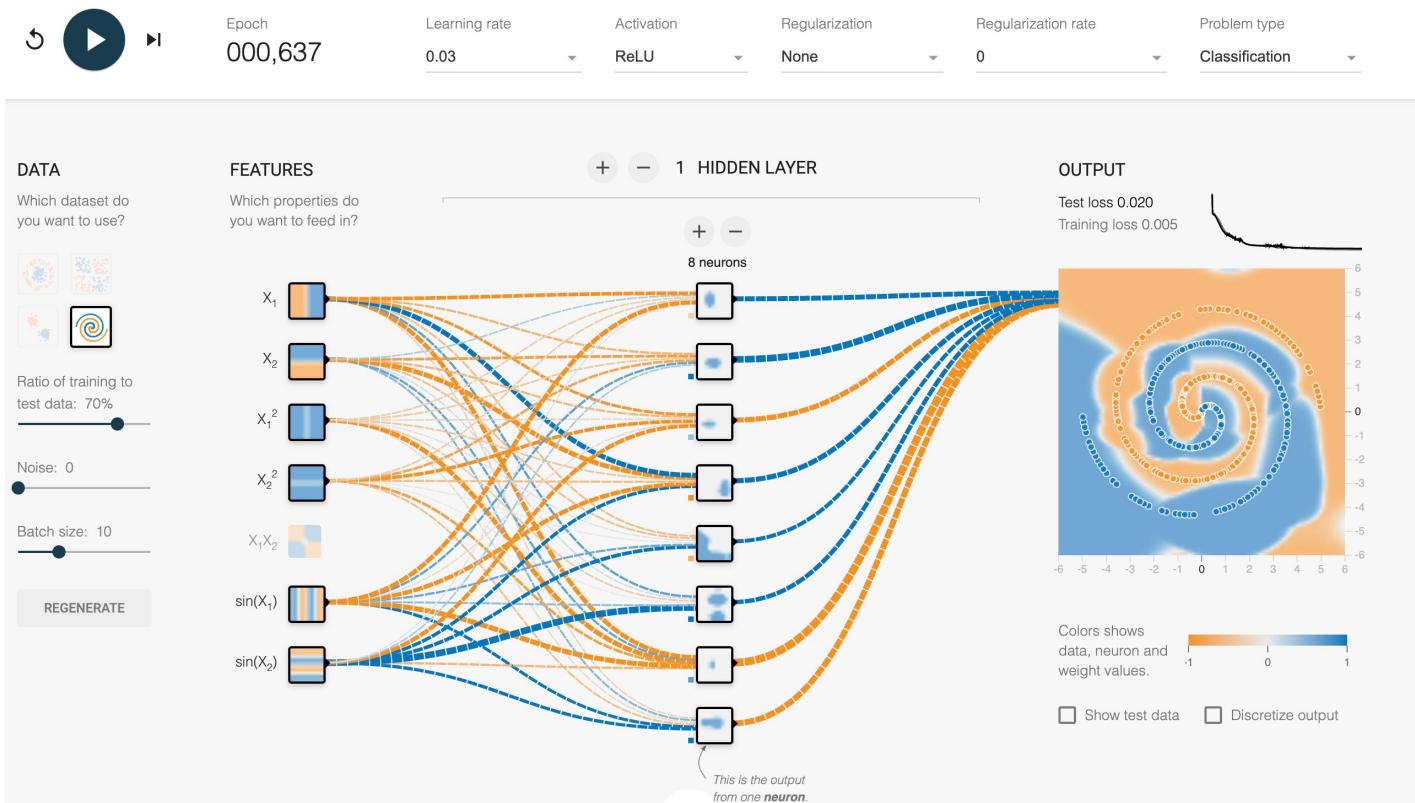
First, let's set the Noise to 0 to fit a simple version of the Spiral Data.

- What if we just use two default features and one hidden layer? **Bad output, high loss.**



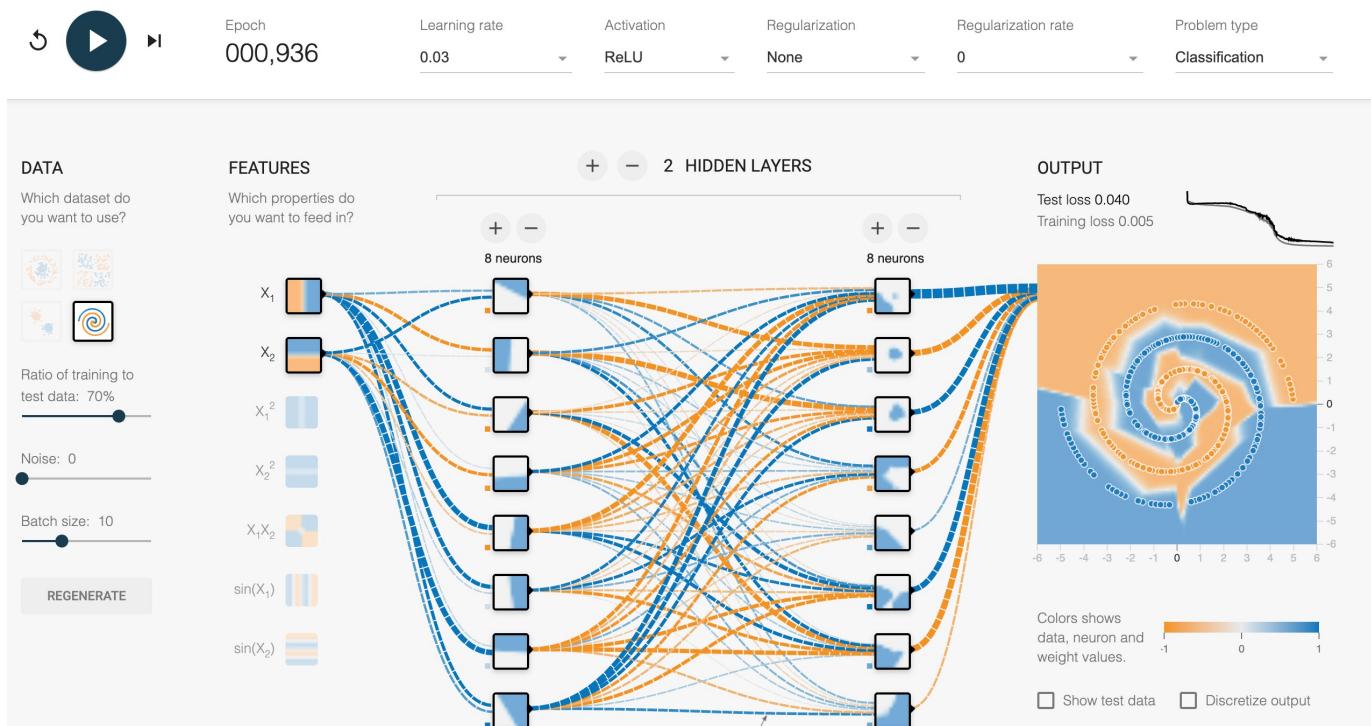
Example 1: The Spiral Data

- What if we apply some feature engineering? It results in a good enough model.



Example 1: The Spiral Data

- We can also add more layers and neurons while using the two default features. **It also results in a good model.**



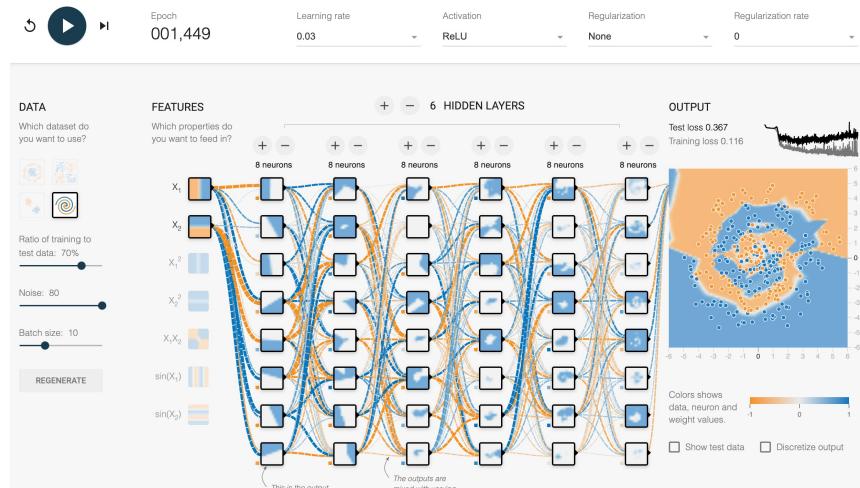
Example 2: The Spiral Data with noise

- Click this [link](#) where the noise is set to 80.
 - 2 features, 2 layers, and 8 neurons in each layer: Bad output (test loss), oscillating loss curve. Question: Is this due to the lack of approximation ability?



Example 2: The Spiral Data with noise

- 2 features, 8 layers, and 8 neurons in each layer: Bad output, oscillating loss curve, longer training time.

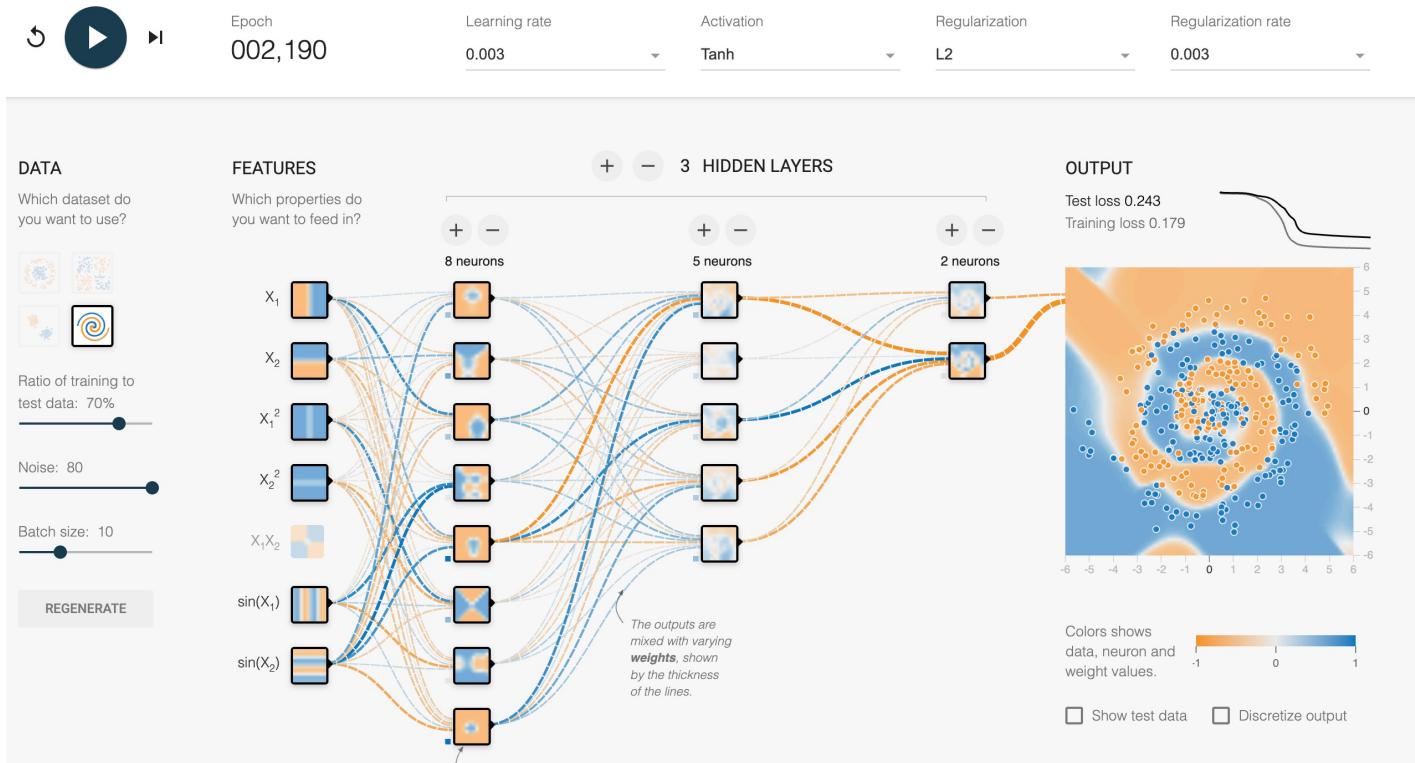


We can try:

- Adding more features to increase the ability to fit the data;
- Reducing the layers and neurons to reduce model complexity and avoid long training time;
- Adding regularizations to avoid overfitting;
- Lowering the learning rate to avoid the jumpy loss curve;
- ...

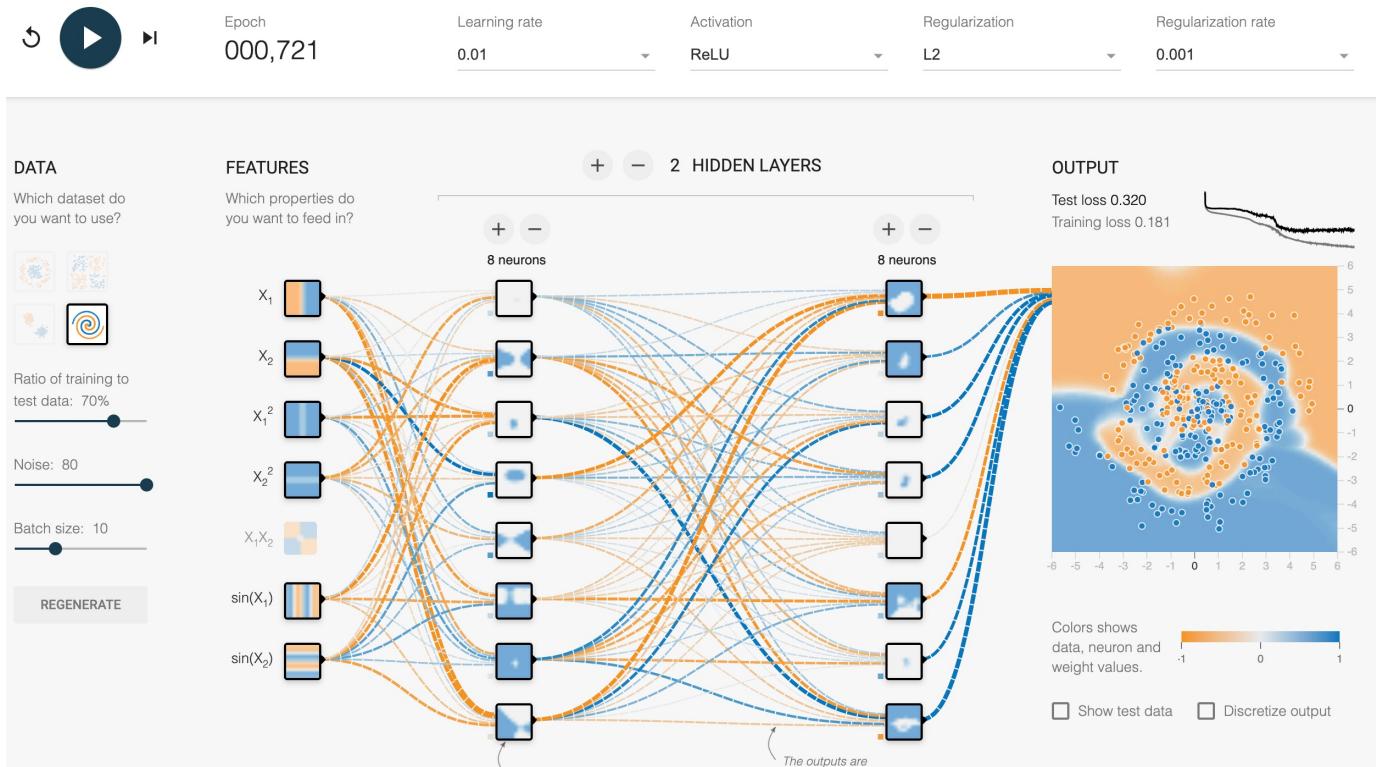
Example 2: The Spiral Data with noise

After trying multiple combinations of the input features, learning rate, activation function, regularization, number of layers, and number of neurons. We get several models with better performance:



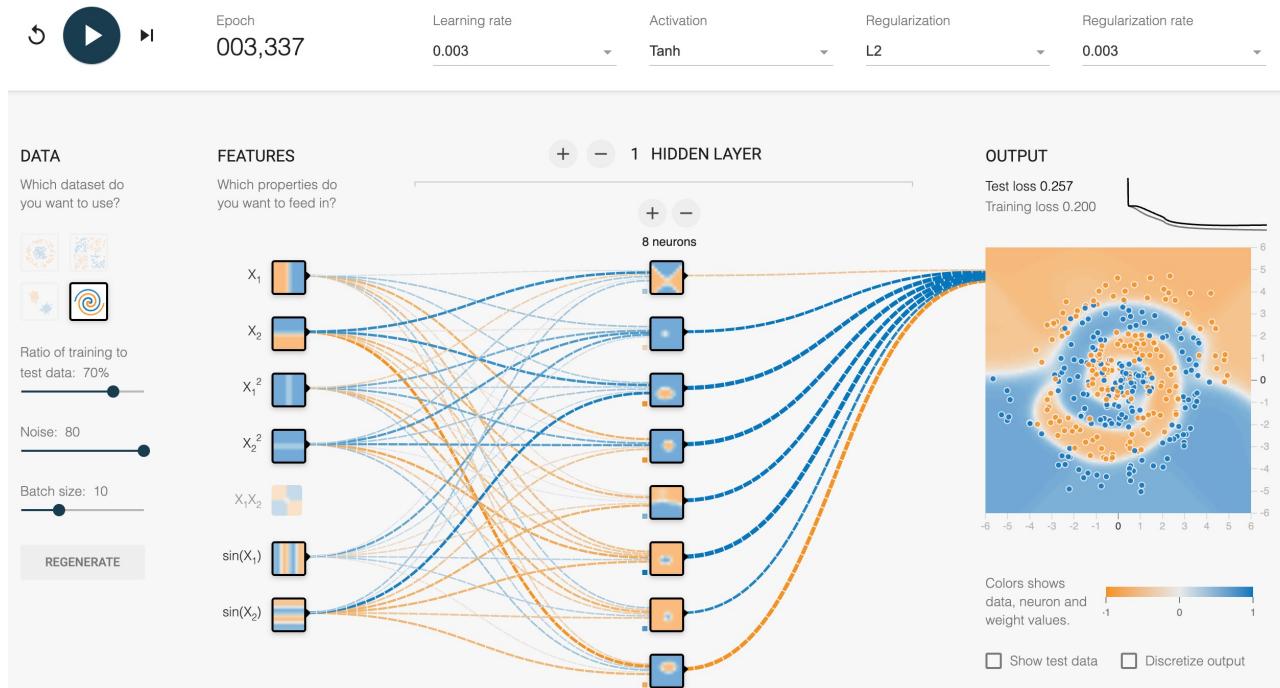
Example 2: The Spiral Data with noise

After trying multiple combinations of the input features, learning rate, activation function, regularization, number of layers, and number of neurons. We get several models with better performance:



Example 2: The Spiral Data with noise

After trying multiple combinations of the input features, learning rate, activation function, regularization, number of layers, and number of neurons. We get several models with better performance:



Even with neural networks, it usually requires some feature engineering to achieve the best performance.

The more, the better?

Now we know that increasing the number of input features, neurons, or layers in an Artificial Neural Network (ANN) can improve its ability to fit (or learn from) the data. **Does that mean the more, the better?**

Answered by ChatGPT:

While it's true that increasing the number of input features, neurons, or layers in an Artificial Neural Network (ANN) can improve its ability to fit or learn from data, it doesn't necessarily mean that more is always better. There are a few reasons for this:

1. **Overfitting:** If a model is too complex, it might start to learn from the noise in the data rather than the actual patterns. This is known as overfitting, and it can lead to poor performance when the model is tested on new, unseen data.
2. **Computational Complexity:** More features, neurons, and layers mean more parameters for the model to learn. This increases the computational complexity and the time it takes for the model to train.
3. **Diminishing Returns:** After a certain point, adding more features, neurons, or layers may not lead to a significant improvement in performance. This is due to the phenomenon of diminishing returns.
4. **Data Availability:** The number of features, neurons, and layers that can be effectively used also depends on the amount of available data. If there isn't enough data to train a complex model, it may fail to generalize well.

Introduction to Pytorch

Learn it via Part 1 of Lab9-Exercises.

Exercise 2

Remember the MNIST dataset? Now, let's train an ANN model on the MNIST dataset.

- Define the network architecture, the loss function and optimizer.
- Train the model and record the train loss, test loss and test accuracy.
- Try to achieve at least 90% accuracy.

Assignment 6

Assignment 6

It is an important problem for credit card companies to predict potential defaults to assess the risk probability of payments in advance. As an expert in artificial intelligence, a credit card company is seeking your assistance to predict default payments based on historical data.

The dataset you are provided contains a set of 23 features. The goal is to forecast the default of payment (yes or no). In the public dataset, you can train and validate your model on 20,000 samples. Then, you need to predict the labels for 5,000 samples in the private dataset, and **the area under the Receiver Operating Characteristic curve (AUC-ROC) on the private dataset will determine your final score.**

Variable Name	Role	Type	Description
Feature 1	Feature	Continuous	Amount of the given credit
Feature 2	Feature	Categorical	Gender (1 = male, 2 = female)
Feature 3	Feature	Categorical	Education (1 = graduate school, 2 = university, 3 = high school, 4 = others)
Feature 4	Feature	Categorical	Marital status (1 = married, 2 = single, 3 = others)
Feature 5	Feature	Continuous	Age (year)
Feature 6	Feature	Categorical	Repayment status in September (-1 = pay duly, 1 = payment delay for one month; . . . , 8 = payment delay for eight months, 9 = payment delay for nine months and above)
Feature 7	Feature	Categorical	Repayment status in August (-1 = pay duly, 1 = payment delay for one month; . . . , 8 = payment delay for eight months, 9 = payment delay for nine months and above)
Feature 8	Feature	Categorical	Repayment status in July (-1 = pay duly, 1 = payment delay for one month; . . . , 8 = payment delay for eight months, 9 = payment delay for nine months and above)
Feature 9	Feature	Categorical	Repayment status in June (-1 = pay duly, 1 = payment delay for one month; . . . , 8 = payment delay for eight months, 9 = payment delay for nine months and above)
Feature 10	Feature	Categorical	Repayment status in May (-1 = pay duly, 1 = payment delay for one month; . . . , 8 = payment delay for eight months, 9 = payment delay for nine months and above)

Assignment 6

It is an important problem for credit card companies to predict potential defaults to assess the risk probability of payments in advance. As an expert in artificial intelligence, a credit card company is seeking your assistance to predict default payments based on historical data.

The dataset you are provided contains a set of 23 features. The goal is to forecast the default of payment (yes or no). In the public dataset, you can train and validate your model on 20,000 samples. Then, you need to predict the labels for 5,000 samples in the private dataset, and **the area under the Receiver Operating Characteristic curve (AUC-ROC) on the private dataset will determine your final score.**

Variable Name	Role	Type	Description
Feature 11	Feature	Categorical	Repayment status in April (-1 = pay duly, 1 = payment delay for one month; . . . , 8 = payment delay for eight months, 9 = payment delay for nine months and above)
Feature 12	Feature	Continuous	Amount of bill statement in September
Feature 13	Feature	Continuous	Amount of bill statement in August
Feature 14	Feature	Continuous	Amount of bill statement in July
Feature 15	Feature	Continuous	Amount of bill statement in June
Feature 16	Feature	Continuous	Amount of bill statement in May
Feature 17	Feature	Continuous	Amount of bill statement in April
Feature 18	Feature	Continuous	Amount paid in September
Feature 19	Feature	Continuous	Amount paid in August
Feature 20	Feature	Continuous	Amount paid in July
Feature 21	Feature	Continuous	Amount paid in June
Feature 22	Feature	Continuous	Amount paid in May
Feature 23	Feature	Continuous	Amount paid in April
Label	Label	Categorical	Default payment (1=yes, 0=no)

Assignment 6

Hint 1: Consider preprocessing and feature engineering if it benefits your model.

Hint 2: A well-designed network architecture is crucial.

Hint 3: Optimize hyperparameters, including learning rate and weight decay, to enhance model performance.

Hint 4: Advanced neural network-related techniques, e.g., LeakyReLU, Dropout, and Batch Normalization, can help to improve the performance. You can use ChatGPT to implement them.

Hint 5: Loss functions and optimization algorithms, e.g., Adam and SGD for model optimization, also play an important role. You can use ChatGPT to implement them.

Hint 6: [OPTIONAL] Class imbalance is a common problem in machine learning. Can you tackle this bottleneck?

Note: External data and pre-trained models are not allowed.

Assignment 6

- Please download the public dataset from
<https://drive.google.com/file/d/1rVtGUBpsWWd0z2YWk806xyf5Ho0BwJwa/view?usp=sharing>
- Please download the private dataset from <https://drive.google.com/file/d/1pGJCLPlPHkPJX2xmnyEX-2aZhsbpi6Or/view?usp=sharing>
- Please use the following Python template for submission. (You can copy the code below from lab9-Exercise.ipynb)
- Your results will be evaluated on 5000 samples in the private dataset, using **AUC-ROC** (The labels will be released in Lab 10).

Assignment 6

```
def read_data_from_csv(path):
    """Load datasets from CSV files.
Args:
    path (str): Path to the CSV file.
Returns:
    X (np.ndarray): Features of samples.
    y (np.ndarray): Labels of samples, only provided in the public
datasets.
"""
assert os.path.exists(path), f'File not found: {path}!'
assert os.path.splitext(path)[-1] == '.csv', f'Unsupported file type {os.path.splitext(path)[-1]}!'

data = pd.read_csv(path)
column_list = data.columns.values.tolist()

if 'Label' in column_list:
    # for the public dataset, label column is provided.
    column_list.remove('Label')
    X = data[column_list].values
    y = data['Label'].astype('int').values
    return X, y
else:
    # for the private dataset, label column is not provided.
    X = data[column_list].values
    return X
```

```
X_public, y_public = read_data_from_csv('assignment_6_public.csv')
print('Shape of X_public:', X_public.shape) # n_sample, m_feature (20000, 23)
print('Shape of y_public:', y_public.shape) # n_sample (20000,)

"""
CODE HERE!
"""

X_private = read_data_from_csv('assignment_6_private.csv')
print('Shape of X_private:', X_private.shape) # k_sample, m_feature (5000, 23)

import numpy as np

# remove and make your own predictions.
preds = np.full(len(X_private), -1,
                dtype=int)
"""
CODE HERE!
e.g.,
preds = np.full(len(X_private), -1, dtype=int)
"""

submission = pd.DataFrame({'Label': preds})
submission.to_csv('assignment_6.csv', index=True, index_label='Id')
```