Introduction to Machine Learning in Engineering Science

National Cheng Kung University Department of Engineering Science

Instructor: Chi-Hua Yu

Lab 4

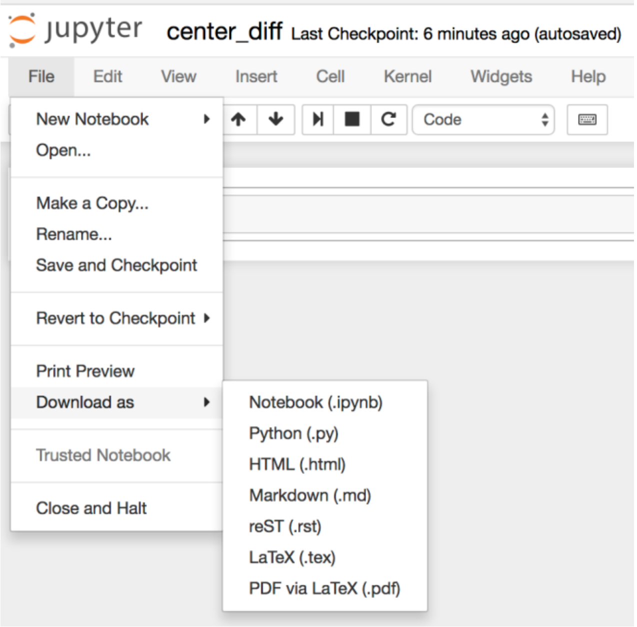
Programming, Due 12:00, Saturday, November 13rd, 2021

Late submission before post of solution: score*0.8 (the solution will usually be posted within a week); no late submission after the post of solution

---

Lab Submission Procedure (請仔細閱讀)

1. You should submit your Jupyter notebook and Python script (`*.py`, in Jupyter, click File, Download as, Python (`*.py`)).



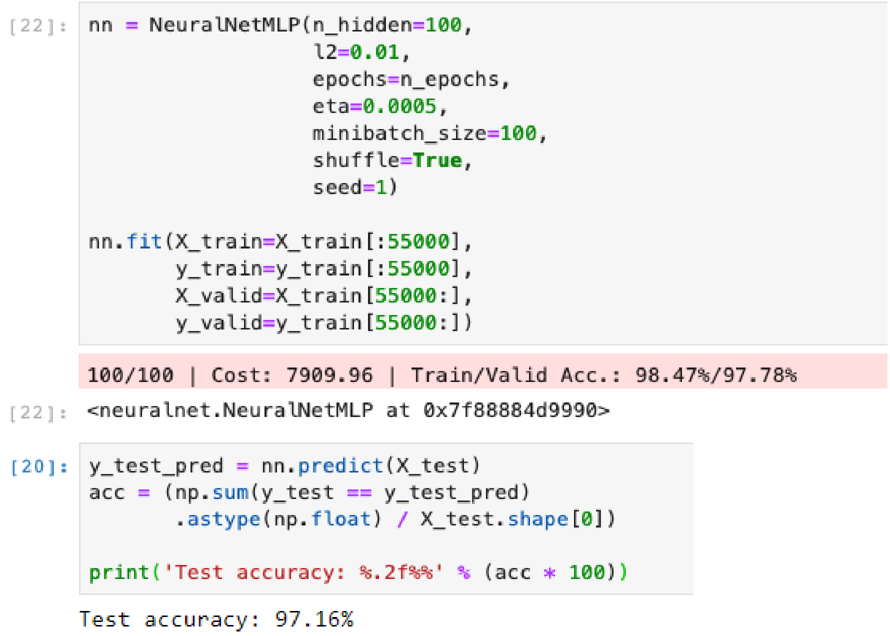
2. Name a folder using your student id and lab number (e.g., n96081494_lab1), put all the python scripts into the folder and zip the folder (e.g., n96081494_ lab 1.zip).

3. Submit your lab directly through the course website.

Total 120%

1. **(120%) Please download the zip file lab4.zip from Moodle.** `Lab4.ipynb` is the main program for you to test your code. In the previous course, we introduced the most important algorithm in deep learning: backpropagation. And you are expected to be able to derive the formulation, compute the gradient and update variable for a simple ANN.

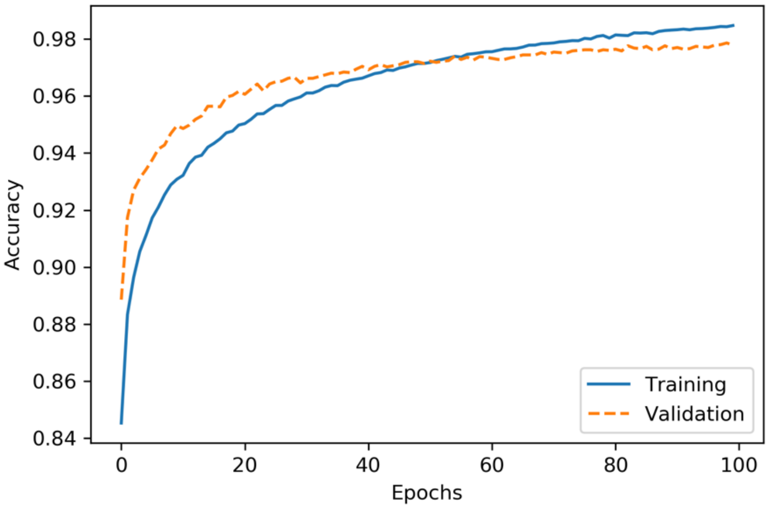
   (a) (60%) Please complete the backpropagation part in `fit` function in `neuralnet.py`. You need to compute sensitivity, implement backward feeding and update weights. Please use `Lab4.ipynb` to test the code



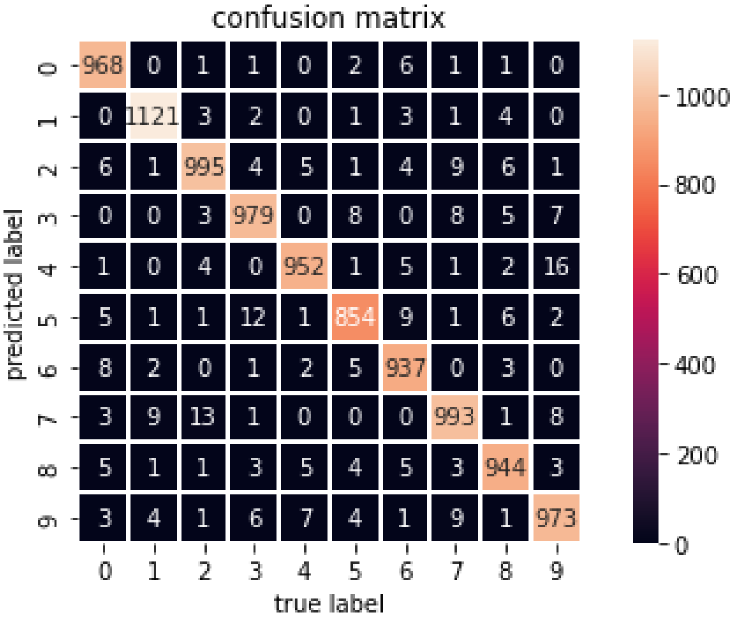
```
[22]: nn = NeuralNetMLP(n_hidden=100,
                        l2=0.01,
                        epochs=n_epochs,
                        eta=0.0005,
                        minibatch_size=100,
                        shuffle=True,
                        seed=1)

      nn.fit(X_train=X_train[:55000],
             y_train=y_train[:55000],
             X_valid=X_train[55000:],
             y_valid=y_train[55000:])

      100/100 | Cost: 7909.96 | Train/Valid Acc.: 98.47%/97.78%

[22]: <neuralnet.NeuralNetMLP at 0x7f88884d9990>

[20]: y_test_pred = nn.predict(X_test)
      acc = (np.sum(y_test == y_test_pred)
             .astype(np.float) / X_test.shape[0])

      print('Test accuracy: %.2f%%' % (acc * 100))

      Test accuracy: 97.16%
```

(b) (40%) We have implemented several useful methods in the NeuralNetMLP class. For instance, we can monitor the value of cost function using `_compute_cost`. Please implement a function to plot the training history for both training and validation.
Hint: You can use the attribute `eval_` to retrieve the accuracy of training set and validation set.

Below is the running example:



(c) (20%) The confusion matrix is a common way to evaluate the performance of our classifier as we did in the previous lecture. Please plot a confusion matrix as blow:



	0	1	2	3	4	5	6	7	8	9
0	968	0	1	1	0	2	6	1	1	0
1	0	1121	3	2	0	1	3	1	4	0
2	6	1	995	4	5	1	4	9	6	1
3	0	0	3	979	0	8	0	8	5	7
4	1	0	4	0	952	1	5	1	2	16
5	5	1	1	12	1	854	9	1	6	2
6	8	2	0	1	2	5	937	0	3	0
7	3	9	13	1	0	0	0	993	1	8
8	5	1	1	3	5	4	5	3	944	3
9	3	4	1	6	7	4	1	9	1	973