

Introduction to Machine Learning

Homework 3

Released on April 3, 2019
Due at 20:00, April 17, 2019

1 [15pts] Decision Tree I

(1) [5pts] Assume there is a space contains three binary features X , Y , Z and the objective function is $f(x, y, z) = \neg(x \text{ XOR } y)$. Let H denotes the decision tree constructed by these three features. Please answer the following question: Is function f realizable? If the answer is yes, please draw the decision tree H otherwise please give the reason.

(2) [10pts] Now we have a dataset show by Table.1:

Table 1:example dataset

X	Y	Z	f
1	0	1	1
1	1	0	0
0	0	0	0
0	1	1	1
1	0	1	1
0	0	1	0
0	1	1	1
1	1	1	0

Please use Gini value as partition criterion to draw the decision tree from the dataset. When Gini value is same for two features, please follow the alphabetical order.

2 [25pts] Decision Tree

Consider the following matrix:

$$\begin{bmatrix} 24 & 53 & 23 & 25 & 32 & 52 & 22 & 43 & 52 & 48 \\ 40 & 52 & 25 & 77 & 48 & 110 & 38 & 44 & 27 & 65 \end{bmatrix}$$

which contains 10 examples and each example contains two features x_1 and x_2 . The corresponding label of these 10 examples as follows:

$$[1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1]$$

In this problem, we want to build a decision tree to do the classification task.

- (1) [5pts] Calculate the entropy of the root node.
- (2) [10pts] Building your decision tree. What is your split rule and the classification error?
- (3) [10pts] A multivariate decision tree is a generalization of univariate decision trees, where more than one attribute can be used in the decision for each split. That is, the split need not be orthogonal to a feature's axis.

Building a multivariate decision tree where each decision rule is a linear classifier that makes decisions based on the sign of $\alpha x_1 + \beta x_2 - 1$. What is the depth of your tree, as well as α and β ?

3 [25pts] Convolutional Neural Networks

Using Fig. 1 as an example. Assuming that the loss function of the convolutional neural network is cross-entropy:

- (1) [5 pts] Briefly describe the forward propagation process;
- (2) [5 pts] What is the difference between Relu and Sigmoid activation functions;
- (3) [5 pts] Derivation of the fully connected layer;
- (4) [5 pts] Derivation of the pooling layer with average pooling;
- (5) [5 pts] Derivation of the convolutional layer with Relu;

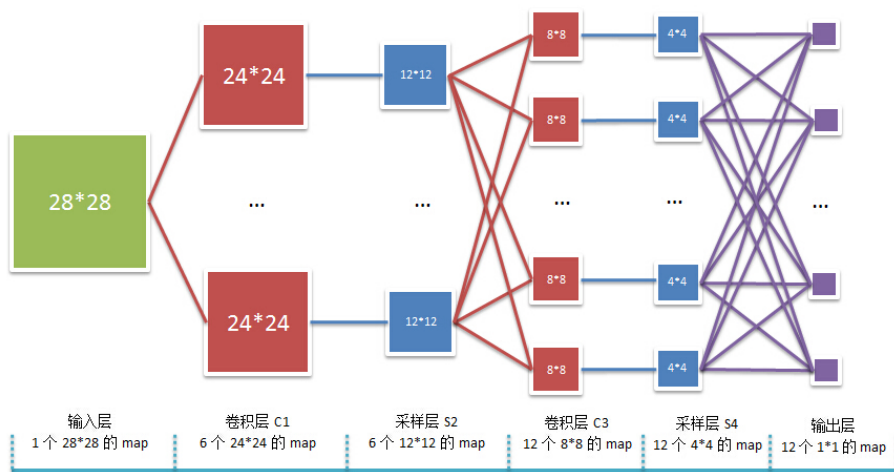


Figure 1: CNN.

4 [35 pts] Neural Network in Practice

In this task, you are asked to build a Convolutional Neural Networks (CNNs) from scratch and examine performance of the network you just build on **MNIST** dataset. Fortunately, there are some out-of-the-box deep learning tools that can help you get started very quickly. For this task, we would like to ask you to work with the **Pytorch** deep learning framework. Additionally, Pytorch comes with a built-in dataset class for MNIST digit classification task in the **torchvision** package, including a training set and a validation set. You may find a pytorch introduction at [here](#). Note that, you can use CPU or GPU for training at your choice.

Please find the detailed requirements below.

- (1) [5 pts] You are encouraged to implement the code using *Python3*, implementations in any other programming language will not be judged. Please name the source file (which contains the main function) as *CNN_main.py*. Finally, your code needs to print the performance on the provided validation set once executed.
- (2) [10 pts] Use any type of CNNs as you want and draw graphs to show your network architecture in the submitted report. You are encouraged to try more architectures.
- (3) [15 pts] During training, you may want to try some different optimization algorithms, such as SGD, Adam. Also, you need to study the effect of learning rate and the number of epoch, on the performance (accuracy).
- (4) [5 pts] Plot graphs (learning curves) to demonstrate the change of training loss as well as the validation loss during training.