Department of Engineering Physics
Tsinghua University

Introduction to Machine Learning (HW1)

Author: 姓名
Class: 班级
Student ID: Your ID
E-mail: xxx@mails.tsinghua.edu.cn
Supervisor: 教师姓名
Course Code: 1111111

June 11, 2020

Homework 1

1 Exercise Solution

This part provides you with two methods for homework solution. One could simply use \section and the result is exactly the same as what this article shows. Another method would be Verbatim. We could start without any content of exercises like,

```
Exercise 1
```

Or, we could start with a full description of the exercise.

```
Exercise 1:
Consider the optimization problem of soft margin support vector machine, derive the dual form
of this optimization problem and write the corresponding Karush-Kuhn-Tucker (KKT) optimality
conditions.
```

One could adjust the margin between these boxes and main text. Note that verbatim could not wrap automatically which should be done by hand.

2 Formula

The formula can be written with $\cdots$ interline or $$\cdots$$ for solitary ones. For a auto numbering formula, one should utilize \begin{equation} $\cdots$ \end{equation}.

For example, this shows the interline formula $\mathfrak{R}_n(g) = \mathbb{E}_{S_n \sim D^n} \hat{\mathfrak{R}}_{S_n}(g)$. And this is an example of individual line formula.

$$\begin{cases} \nabla \times \boldsymbol{H} = \boldsymbol{J} + \dfrac{\partial \boldsymbol{D}}{\partial t} \\ \nabla \times \boldsymbol{E} = -\dfrac{\partial \boldsymbol{B}}{\partial t} \\ \nabla \cdot \boldsymbol{B} = 0 \\ \nabla \cdot \boldsymbol{D} = \rho \end{cases}$$

For formula derivation, one could use \begin{aligned} $\cdots$ \end{aligned}. For example,

$$\begin{aligned} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j &= a_{11} x_1^2 + a_{12} x_1 x_2 + \cdots + a_{1n} x_1 x_n + \cdots + a_{n1} x_n x_1 + a_{n2} x_n x_2 + \cdots + a_{nn} x_n^2 \\ &= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \end{aligned}$$

Here is the example of auto numbering formula.

$$i\hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \nabla^2 \psi + V \psi \tag{1}$$

If one wants to cross reference formulas, one could use this way: \ref{label} which would induce the following result, as is seen in Equ (1).

3 Figure and Table

Here I do not want to bother introducing different ways of inserting figures or constructing tables, one should refer to corresponding documentions for further exploration.

As a simple example of table, one could construct it this way:

Table 1: Table example

col 1	col 2
A	some value or data displayed

Or, other kinds of tables like Tab 2.

Table 2: Another example of Table

Model	#parameter	accuracy
Faster-RNN	1M	0.78
GRU	2M	0.72

As to figures, we could change its size with `scale` parameter. The Fig 1 and Fig 2 shows the different ways of showing figures.

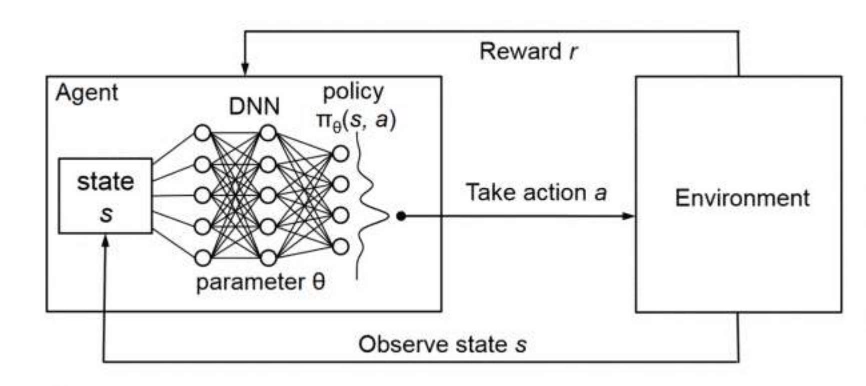
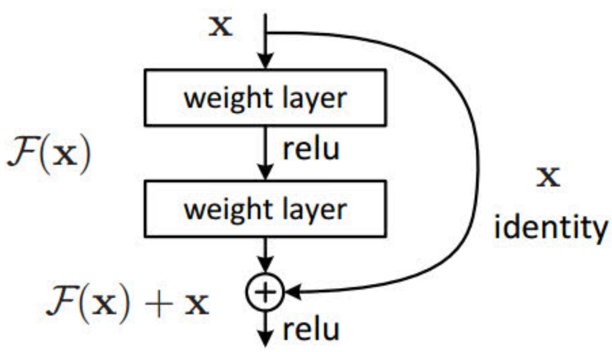
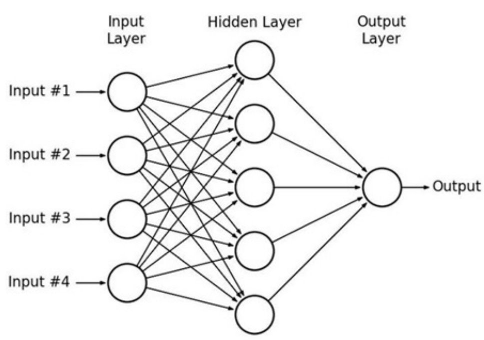


Figure 1: Figure example



(a) Resnet



(b) MLP

Figure 2: subfigure example

4 Algorithms and Code Display

One could write algorithms like the following form:

Algorithm 1 algorithm caption

Input: input parameters A, B, C
Output: output result

```
some description
for condition do
    ...
    if condition then
        ...
    else
        ...
while condition do
    ...
return result
```

Also, if the homework requires one to add codes as an appendix or attach codes in the corresponding exercise, one could utilize `\begin{lstlisting}` $\cdots$ `\end{lstlisting}`. The following gives a simple example of codes display.

```python
import numpy as np
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_breast_cancer
from sklearn.externals.six import StringIO
import graphviz
import pydotplus

# Load data
bc = load_breast_cancer()
X, y = bc.data, bc.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3, random_state=1)

# train a classification tree and evaluate it
# set max_depth = 3
dtc = DecisionTreeClassifier(max_depth = 3)
dtc.fit(X_train, y_train)
print("train␣score:", dtc.score(X_train, y_train))
print("test␣score:", dtc.score(X_test, y_test))
# visualize the result
dot_data = StringIO()
export_graphviz(dtc, feature_names= bc.feature_names,
             class_names = bc.target_names,
             out_file = dot_data)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_pdf("cancer_tree.pdf")
print('Visible␣tree␣plot␣saved␣as␣pdf.')
```

5 Instruction

This template is more suitable for math/physics/statistics/computer science. If this template does not meet some of the requirements your homework, you could supplement it yourself.