Title of the Assignment Template
Assignment 1

Your Name | Student No.: 7355608 | Email: example@email.com

November 18, 2025

1 Common Commands

`\couriertext \texttt` $\left\|\operatorname*{argmin}_x \boldsymbol{M}^{-1}\right\|$

1.1 Questions

Question (a)

(i) This is sub item one in `sub` environment.

(ii) This is sub item two.

(iii) This is sub item three.

> This is a `quote` environment inside the question box.

This is a question box without title.

$$\begin{cases} \alpha = \dfrac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\ \beta = \bar{y} - \alpha\bar{x} \end{cases}$$

Problem VI)

- `\qst{blabla}[Problem ][][Roman][2]`
- blabla

1.2 Answers

Solution. This is the solution using `\sln`.

Solution.

This is the solution using `\sln*`. There is a new line after the prefix.

Answer: This is the answer using `\ans`.

Answer:

This is the answer using `\ans*`. There is a new line after the prefix.

Proof. This is the proof environment content using `\prf{}`.

There is a square at the end of the proof. □

2 Figures, Tables, and Code Listings

2.1 Figures

Use custom `\img` command to insert image (shown as Figure 1):

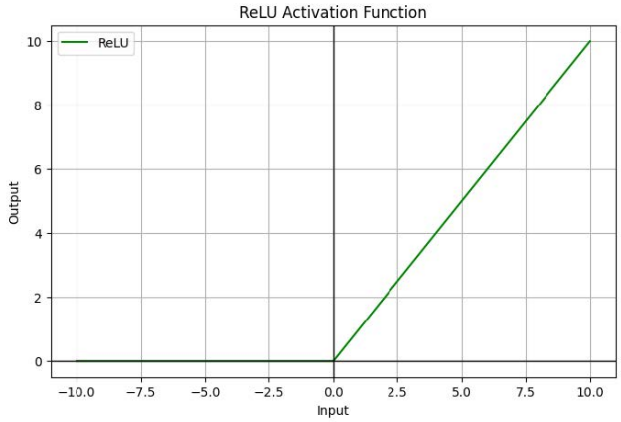
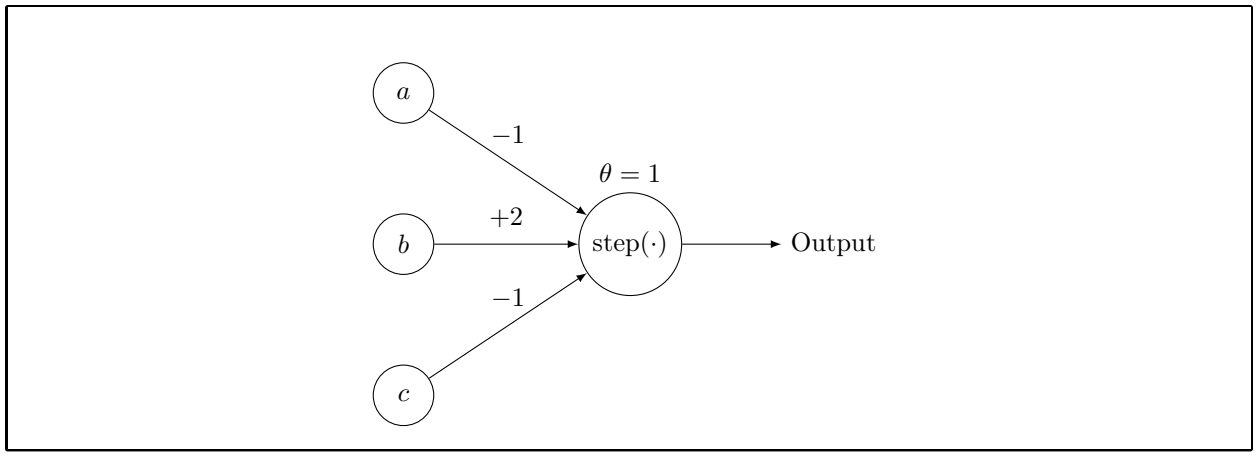


Figure 1: Example Image. This image displays the ReLU (Rectified Linear Unit) activation function, which is defined as $\text{ReLU}(x) = \max(0, x)$. It is widely used in neural networks due to its simplicity and effectiveness in introducing non-linearity.

```
\img
[0.5]              % Scale (as a fraction of text width)
[H]                % Placement
{img/exp-img.jpg} % Image file
[Example Image]    % Caption
[fig:img_example] % Label
[2cm]              % Margin
```

Use custom `\tikzimg` command to insert TikZ image (in a question box below):



2.2 Tables

Use `\tbl` command to insert tables (shown as Table 1):

Table 1: Truth table for the boolean expression $\neg a \vee b \vee \neg c$

a	b	c	$\neg a$	$\neg c$	$\neg a \vee b \vee \neg c$
0	0	0	1	1	1
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	0	0	1

```
\tbl
[H]                         % Placement
{                           % Table content
  \begin{tabular}{cols} % You can change the env you need
    ...
  \end{tabular}
}
[Example Table]             % Caption
[tab:example]               % Label
```

2.3 Code Listings

Inline code:

```
print("Hello, World!")
```

Block code (imported from external files):

```
import torch
import torch.nn as nn

# Define the neural network architecture
class SimpleFCNN(nn.Module):
    def __init__(self, input_size=784, hidden_size=500, output_size=10):
        super(SimpleFCNN, self).__init__()
```

```
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(hidden_size, output_size)

    def forward(self, x):
        """Forward pass of the network"""
        out = self.fc1(x)
        out = self.relu(out)
        out = self.fc2(out)
        return out
```

```
#!/bin/bash
# This is a sample bash script for testing code listing in LaTeX.
echo "Hello, World!"
for i in {1..5}
do
   echo "Iteration $i"
done
echo "Script completed."
```

Block code:

```
\begin{lstlisting}[
  language=tex,
  caption={Example LaTeX Code},
  label={lst:example_latex},
  % style=framestyle  % Custom style as follows
  basicstyle=\small\ttfamily,
  columns=flexible,
  numbers=none,
  backgroundcolor=\color{white},
  frame=single
]
  ...
\end{...}
```

Algorithm 1: Example LaTeX Code

2.4 Code Listings Using Custom Commands

Inline code using `\code` command: `print("Hello, World!")`

Block code that imports from external files using `\codef` command (shown as Listing 2):

```
import torch
import torch.nn as nn

# Define the neural network architecture
class SimpleFCNN(nn.Module):
    def __init__(self, input_size=784, hidden_size=500, output_size=10):
        super(SimpleFCNN, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(hidden_size, output_size)

    def forward(self, x):
```

```
        """Forward pass of the network"""
        out = self.fc1(x)
        out = self.relu(out)
        out = self.fc2(out)
        return out
```

Algorithm 2: Example Python Code