Université abdelmalek essaadi
Faculté des Sciences
Département de Mathématiques

Année 2023-2024
MLAI(S5)
Prof. Naji Yebari
ناجي الجباري

Control of module Mathematical Fundations for Machine Learning : M352
(Duration:2H two hours)

Exercice1: 10 pts
(1.1) Provide the commands that calculate the following results (i) and (ii) , then provide (iii) and (iv) from a results perspective
(*i*) tensor([5,4,3,2,1]) (*ii*) tensor([5,4,3,2,1,0])
(*iii*) a=torch.arange(6,0,-1) (*iv*) b=torch.arange(5,-2,-1)
(1.2) We set $a = torch.arange(16).reshape(4, 4)$, $b = torch.arange(4, 0, -1)$. Calculate the following

(*i*) • A=torch.arange(16).reshape(4,4) •• A[-1] • • • A[:3,:]
(*ii*) • B=torch.arange(16,0,-1).reshape(4,4) •• B.sum() • • • a+b
(*iii*) • S= A+B •• A==B
• • • Provide the commands which add Y to X in the same tensor, where Y is stacked below X

$$X = tensor([[0, \quad 1, \quad 2, \quad 3] , \\ [4, \quad 5, \quad 6, \quad 7] , \\ [[8, \quad 9, \quad 10, \quad 11]])$$

$$Y = tensor([[12, \quad 11, \quad 10, \quad 9] , \\ [8, \quad 7, \quad 6, \quad 5] , \\ [4, \quad 3, \quad 2, \quad 1]])$$

(1.3) Execute the following code and provide its result.

$$\begin{array}{l} def \quad ltochadd(a, b, c) : \\ \quad c = torch.zeros(a.shape) \\ for \quad i \quad in \quad range(a.shape[0]) : \\ \quad for \quad j \quad in \quad range(a.shape[1]) : \\ \qquad c[i, j] = a[i, j] + b[i, j] \\ \qquad return \quad c \end{array}$$

Exercice2: (10pts)
$X_1 = torch.tensor([-0.92, -0.46, -0.72])$, $W_1 = torch.tensor([1.28, -0.99, 1.82])$,

$$B_1 = torch.tensor([-0.60])$$

$$Y = sigmoid(W_{11}X_{11} + W_{12}X_{12} + W_{13}X_{13} + B_{11}), \qquad sigmoid(s) = \frac{1}{1 + exp(-s)}$$

(2.1) (*i*) Calculate the output Y with input X_1,W_1 and B_1. Similar to Numpy, PyTorch has a $torch.sum()$ function, as well as $.sum()$ method on tensors, for taking sums. Use the function $sigmoid$ defined above.
(*ii*) Calculate Y using matrix multiplication.

(2.2) (i)

$$X^T = \begin{pmatrix} -0.92 \\ -0.46 \\ -0.72 \end{pmatrix} \in \mathbb{R}^3;$$

$$W_1 = \begin{pmatrix} -0.19 & 0.28 \\ 0.13 & 0.33 \\ -0.078 & -0.92 \end{pmatrix} \in \mathbb{R}^{3\times 2}$$

$$W_2 = (1.28, 1.82)^T \in \mathbb{R}^2, \quad B_1^T = \begin{pmatrix} -1.68 \\ -0.72 \end{pmatrix} \in \mathbb{R}^2; \quad B_2 = -0.60$$

We put $h = sigmoid(XW_1 + B_1)$. Write the code that calculates the output function defined as follows

$$output = sigmoid(hW_2 + B_2)$$

(ii) Conclusion ?

$$Le \quad corrigé \quad du \quad cntrôle: \quad Correction \quad of \quad the \quad test$$

$$Control \quad of \quad module \quad Mathematical \quad Fundations \quad for \quad Machine \quad Learning: \quad M352$$

Exercice1: 10 pts

(1.1) Provide the commands that calculate the following results (i) and (ii) , then provide (iii) and (iv) from a results perspective

(i) tensor([5,4,3,2,1]) $\quad Answer: \quad torch.arange(5, 0, -1)$

(ii) tensor([5,4,3,2,1,0]) $\quad Answer: \quad torch.arange(5, -1, -1)$

(iii) a=torch.arange(6,0,-1) $\quad Answer: \quad a = tensor([6, 5, 4, 3, 2, 1])$

(iv) b=torch.arange(5,-2,-1) $\quad Answer: \quad b = tensor([5, 4, 3, 2, 1, 0, -1])$

(1.2) We set $a = torch.arange(16).reshape(4, 4)$, $b = torch.arange(4, 0, -1)$.

Calculate the following

(i)

$$\bullet \quad A = torch.arange(16).reshape(4, 4) \quad \bullet\bullet \quad A[-1] \quad \bullet\bullet\bullet \quad A[:3, :]$$

(i) • A is a tensor given as follows

$$\begin{aligned} tensor([&[0, \quad 1, \quad 2, \quad 3], \\ &[4, \quad 5, \quad 6, \quad 7], \\ &[8, \quad 9, \quad 10, \quad 11], \\ &[12, \quad 13, \quad 14, \quad 15]]) \end{aligned}$$

•• A[-1] provide this tensor $\quad tensor([12, 13, 14, 15])$

••• A[:3,:] provide this tensor

$$\begin{aligned} tensor([&[0, \quad 1, \quad 2, \quad 3], \\ &[4, \quad 5, \quad 6, \quad 7], \\ &[8, \quad 9, \quad 10, \quad 11]]) \end{aligned}$$

(1.2)

(ii)

$$\bullet \quad B = torch.arange(16, 0, -1).reshape(4, 4) \quad \bullet\bullet \quad B.sum() \quad \bullet\bullet\bullet \quad a + b$$

• B=torch.arange(16,0,-1).reshape(4,4) provide this tensor

$$
\begin{aligned}
tensor([&[16, \quad 15, \quad 14, \quad 13],\\
&[12, \quad 11, \quad 10, \quad 9],\\
&[8, \quad 7, \quad 6, \quad 5],\\
&[4, \quad 3, \quad 2, \quad 1]])
\end{aligned}
$$

•• B.sum() provide this tensor tensor(136)
••• a=torch.arange(16).reshape(4,4)
b=torch.arange(4,0,-1)
a+b provide this tensor

$$
\begin{aligned}
tensor([&[4, \quad 4, \quad 4, \quad 4],\\
&[[8, \quad 8, \quad 8, \quad 8],\\
&[12, \quad 12, \quad 12, \quad 12],\\
&[16, \quad 16, \quad 16, \quad 16]])
\end{aligned}
$$

(iii) • S= A+B •• A==B
••• Provide the commands which add Y to X in the same tensor, where Y is stacked below X

$$
\begin{aligned}
X = tensor \quad ([&[0, \quad 1, \quad 2, \quad 3],\\
&[4, \quad 5, \quad 6, \quad 7],\\
&[8, \quad 9, \quad 10, \quad 11]])
\end{aligned}
$$

$$
\begin{aligned}
Y = tensor \quad ([&[12, \quad 11, \quad 10, \quad 9],\\
&[8, \quad 7, \quad 6, \quad 5],\\
&[4, \quad 3, \quad 2, \quad 1]])
\end{aligned}
$$

• S= A+B provide this tensor

$$
\begin{aligned}
tensor \quad ([&[16, \quad 16, \quad 16, \quad 16],\\
&[16, \quad 16, \quad 16, \quad 16],\\
&[16, \quad 16, \quad 16, \quad 16],\\
&[16, \quad 16, \quad 16, \quad 16]])
\end{aligned}
$$

•• A==B provide this tensor

$$
\begin{aligned}
tensor \quad ([&[False, \quad False, \quad False, \quad False],\\
&[False, \quad False, \quad False, \quad False],\\
&[True, \quad False, \quad False, \quad False],\\
&[False, \quad False, \quad False, \quad False]])
\end{aligned}
$$

••• the answer for this question yields as follows:
X=torch.tensor([[0,1,2,3],[4,5,6,7],[8,9,10,11]])
Y=torch.tensor([[12,11,10,9],[8,7,6,5],[4,3,2,1]])
Z=torch.cat((X,Y),dim=0)
Z

$$
tensor \quad ([[0, \quad 1, \quad 2, \quad 3],
$$

$$[4, \quad 5, \quad 6, \quad 7],$$
$$[8, \quad 9, \quad 10, \quad 11],$$
$$[12, \quad 11, \quad 10, \quad 9]$$
$$[[8, \quad 7, \quad 6, \quad 5]$$
$$[4, \quad 3, \quad 2, \quad 1]])$$

(1.3) Execute the following code and provide its result.

$$def \quad ltochadd(a, b, c):$$
$$c = torch.zeros(a.shape)$$
$$for \quad i \quad in \quad range(a.shape[0]):$$
$$for \quad j \quad in \quad range(a.shape[1]):$$
$$c[i, j] = a[i, j] + b[i, j]$$
$$return \quad c$$

Below is the answer
a = torch.arange(16).reshape(4, 4)
b = torch.arange(16, 0, -1).reshape(4, 4)
result = ltochadd(a, b)
print(result)
Which give this result

$$tensor \quad ([[16., \quad 16., \quad 16., \quad 16.],$$
$$[[16., \quad 16., \quad 16., \quad 16.],$$
$$[16., \quad 16., \quad 16., \quad 16.],$$
$$[16., \quad 16., \quad 16., \quad 16.]])$$

Exercice2: 10 pts
$X_1 = torch.tensor([-0.92, -0.46, -0.72])$, $W_1 = torch.tensor([1.28, -0.99, 1.82])$,

$$B_1 = torch.tensor([-0.60])$$

$$Y = sigmoid(W_{11}X_{11} + W_{12}X_{12} + W_{13}X_{13} + B_{11}), \qquad sigmoid(s) = \frac{1}{1 + exp(-s)}$$

(2.1) (i) Calculate the output Y with input X_1,W_1 and B_1. Similar to Numpy, PyTorch has a $torch.sum()$ function, as well as $.sum()$ method on tensors, for taking sums. Use the function $sigmoid$ defined above.
(ii) Calculate Y using matrix multiplication.
The answer for this question is as follows:
(i) def sigmoid(x): return 1/(1+torch.exp(-x))
X1=torch.tensor([-0.92,-0.46,-0.72])
W1=torch.tensor([1.28,-0.99,1.82])
B1=torch.tensor([-0.60])
Y=sigmoid(torch.sum(X1*W1)+B1)
Y
the result is like this: tensor([0.0671])
The method with .sum() is writting as the following
Z=sigmoid((X1*W1).sum()+B1)
Z
this gives the same tensor as above
(ii)

Youtput=sigmoid(torch.mm(X1.reshape(1,3),W1.reshape(3,1))+B1)
Youtput
Youtput yield: tensor([0.0671])
(2.2) (i)

$$X^T = \begin{pmatrix} -0.92 \\ -0.46 \\ -0.72 \end{pmatrix} \in \mathbb{R}^3; W_1 = \begin{pmatrix} -0.19 & 0.28 \\ 0.13 & 0.33 \\ -0.078 & -0.92 \end{pmatrix} \in \mathbb{R}^{3\times 2}$$

$$W_2 = (1.28, 1.82)^T \in \mathbb{R}^2, \quad B_1^T = \begin{pmatrix} -1.68 \\ -0.72 \end{pmatrix} \in \mathbb{R}^2; \quad B_2 = -0.60$$

We put $h = sigmoid(XW_1 + B_1)$. Write the code that calculates the output function defined as follows

$$output = sigmoid(hW_2 + B_2)$$

(ii) Conclusion ?
the answer for (i) provide the comands as the following:
X=torch.tensor([-0.92, -0.46, -0.72]).reshape(1,3)
W1=torch.tensor([-0.19, 0.28, 0.13,0.33,-0.078,-0.92]).reshape(3,2)
W2=torch.tensor([1.28, 1.82]).reshape(2,1)
b1=torch.tensor([-1.68, -0.72]).reshape(1,2)
b2=torch.tensor([-0.60])
h=sigmoid(torch.mm(X,W1)+b1)
output=sigmoid(torch.mm(h,W2)+b2)
h.shape
h,output
h $\cdots$ tensor([[0.1811, 0.3853]])
output $\cdots$ tensor([[0.5825]])
(ii) Conclusion ?
(2.1)
X=torch.tensor([-0.92,-0.46,-0.72])
W=torch.tensor([1.28,-0.99,1.82])
b=torch.tensor([-0.60])
Output=sigmoid(torch.sum(X*W)+b)
Output
the result is like this: tensor([0.0671])
This result can be interpreted as follows:
Input 1 : -0.92 w_1: 1.28

Input 2 : -0.46 w_2: -0.99 (Neuron Layer, sigmoid(torch.sum(X*W)+b)$\rightarrow$ *Output* : 0.0671

Input 3 : -0.72 w_3: 1.82
Bias (b) : -0.60
Here, w_1, w_2, and w_3 represent the weights associated with the connections between the inputs and the neuron. We adjust the diagram based on the details of our neural network.
(ii) Conclusion ?
(2.1)
X=torch.tensor([-0.92,-0.46,-0.72])
W=torch.tensor([1.28,-0.99,1.82])
b=torch.tensor([-0.60])
Output=sigmoid(torch.sum(X*W)+b)
Output
the result is like this: tensor([0.0671])
This result can be interpreted as follows:
Input 1 :-0.92 w_1:1.28

Input 2 : -0.46 w_2: -0.99 (Neuron Layer, sigmoid(torch.sum(X*W)+b))$\rightarrow$ *Output* : 0.0671

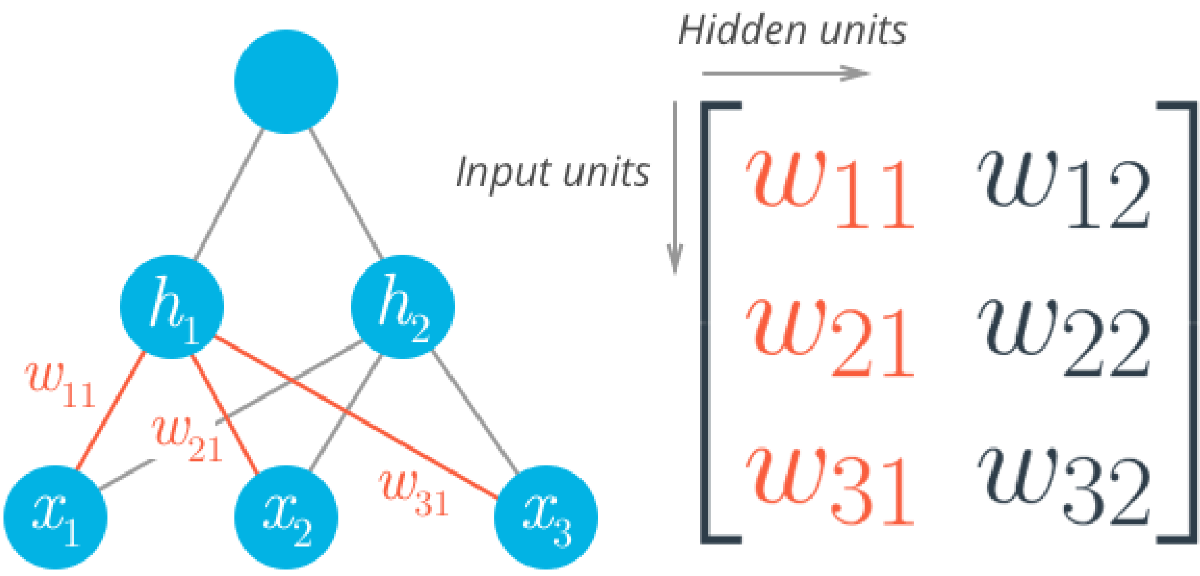

Figure 1: graph of our exercice

Input 3 :-0.72 w_3: 1.82
Bias (b) : -0.60
Here, w_1, w_2, and w_3 represent the weights associated with the connections between the inputs and the neuron. We adjust the diagram based on the details of our neural network.

the answer for (2.2)(i) provide the comands as the following:
X=torch.tensor([-0.92, -0.46, -0.72]).reshape(1,3)
W1=torch.tensor([-0.19, 0.28, 0.13,0.33,-0.078,-0.92]).reshape(3,2)
W2=torch.tensor([1.28, 1.82]).reshape(2,1)
b1=torch.tensor([-1.68, -0.72]).reshape(1,2)
b2=torch.tensor([-0.60])
h=sigmoid(torch.mm(X,W1)+b1)
output=sigmoid(torch.mm(h,W2)+b2)
h.shape
h,output
(tensor([[0.1811, 0.3853]]), tensor([[0.5825]]))
In order word, This result can be interpreted as follows:
Input 1:-0.92 h_1 Neuron (Layer 1, Sigmoid) 0.1811
$W_{11} = -0.19$ $W_{21} = 0.13$ $W_{31} = -0.078$
Bias (Layer 1): -1.68,
Input 2 : -0.46
h_2 Neuron (Layer 2, Sigmoid) 0.3853
$W_{12} = 0.28$ $W_{22} = 0.33$ $W_{32} = -0.92$
Bias (Layer 2): -0.72

$$Output = 0.5825$$

$$Bias(Input) = -0.60$$

Input 3:-0.72
We can conclude that here we have presented an example of a neural network, which justifies

the necessity of understanding basic mathematical foundations to comprehend and delve deeper into the field of machine learning and artificial intelligence.

، يمكننا أن نستنتج أننا هنا قدمنا مثالاً للشبكة العصبية Neural network مما يبرر ضرورة فهم الأسس الرياضية الأساسية لفهم مجال التعلم الآلي و الذكاء الاصطناعي و التعمق فيه