CS5242 August 2023, Quiz 1

Week 7, 03 October 2023, 18:30hours

Total marks 15% and bonus marks 4%

Instructions

Data can be find in 'Quizz_data.zip'. Please make sure to submit your responses in a single notebook, which should encompass both your code and explanations. The quality of your code will be evaluated, so it's important to clearly highlight the required output. Your code should be self-contained and work without any modifications (data path, outputs, ...). For example, if the task involves providing the number of parameters in your model, be sure to include a print statement in your code that displays that value.

Furthermore, your folder structure should adhere to the following format:

- AXXXXXXX.zip
  - Q_1
  - Q_4
  - Q_5
  - Q_6
  - AXXXXXXX.ipynb

For any answers requiring symbolic expressions or explanations, please use Markdown with LaTeX environment. This will ensure clarity and proper formatting in your explanations.

Question 1 [3 marks]

In this problem, we work on 100-class classification problem. With $M = 100$ logits for each data point $z \in \mathbb{R}^M$. Each point also comes with a one-hot label $y \in \mathbb{Z}^M$. The task is to calculate softmax and then cross entropy loss **from scratch**, you are not allowed use torch.nn.softmax(), tensorflow library or any other libraries. You can use numpy.exp to calculate the exponential.

Given logits z, the softmax p and the Cross Entropy Loss $\mathcal{L}$:

$$p_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{1}$$

$$\mathcal{L} = -\sum_i y_i \ln p_i \tag{2}$$

Read the logits from the file 'data.txt' and the labels from 'labels.txt'. Afterward, you can use Python code to perform these calculations. Please save your code and provide the results on your notebook.

Question 2 [3 marks]

Make the code that generates the compute tree given in Figure 1.

For checking, you may use the following code

```
from torchviz import make_dot
def print_compute_tree(name,node):
    dot = make_dot(node)
    dot.render(name)
```

For the operations that require a constant, use the value of 5. Additionally, when all the inputs are $[1, 2, 3]$, the output is $[7, 11, 17]$. Compute the output of the operations when all the inputs are $[4, 5, 6]$.

Please save your code and provide the results on your notebook.

Question 3 [3 marks]

Derive a symbolic expression for the gradient of cross entropy loss (CE) on sigmoid $\sigma(.)$ with respect to one single input $X \in \mathbb{R}^3$ and its corresponding features s.t $X = (x_1, x_2, x_3)^T$ for the following network (Fig.2). As a reminder, the sigmoid is defined as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \tag{3}$$

For CE, assume labels given by y.

Question 4 [3 marks]

For the following data points, find the function that can exactly encode them.

1. Original data in 2D, encode into 1D. Read from data file Q1Q4I1.txt. If you write down the correct equations for encoding then you solved this problem exactly. If you use a neural network, your objective is to use as few parameters as possible. Restrict the network to less than 10,000 fitting parameters. Report the number of network parameters.

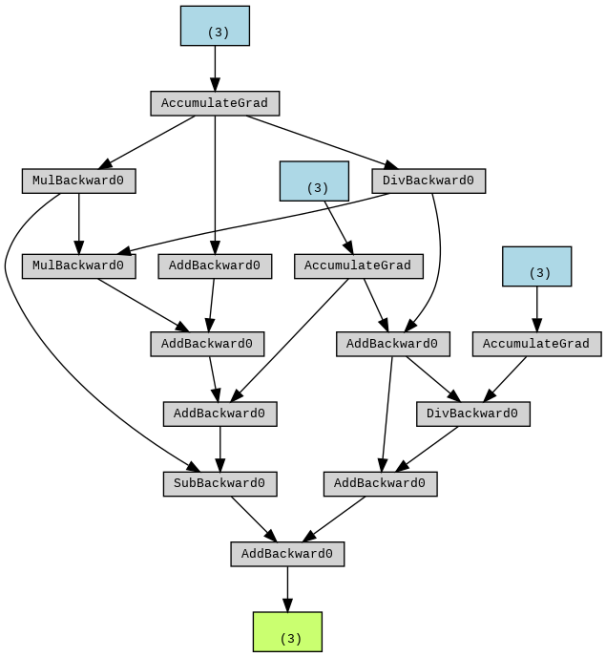

Figure 1: Compute tree for Question 2

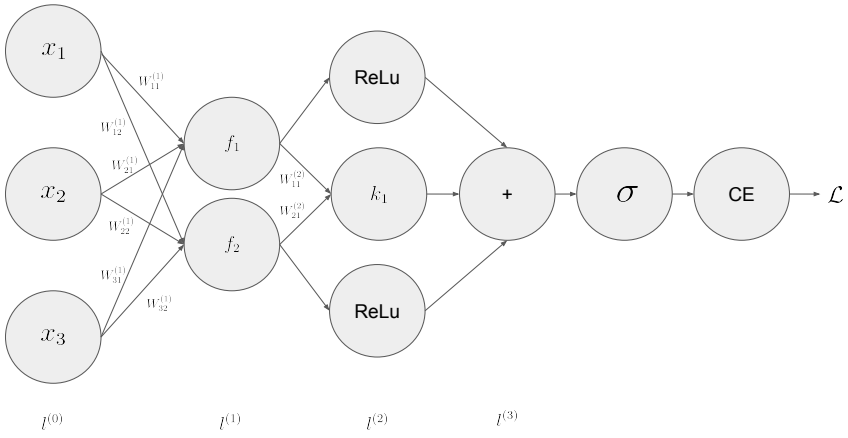

Figure 2: Network

2. Original data in 3D, encode into 2D. Read from data file Q1Q4I2.npz. If you write down the correct equations for encoding then you solved this problem exactly. If you use a neural network, your objective is to use as few parameters as possible. Restrict the network to less than 10,000 fitting parameters. Report the number of network parameters.

3. Original data in 4D, encode into 3D. Read file from Q1Q4I3.npz. If you us a neural network, your objective is to use as few parameters as possible. Restrict the network to less than 100,000 fitting parameters. Report the number of network parameters.

Some hints and instructions

1. If you can find the function symbolically, you get full marks.

2. If you cannot find the symbolic function and you use a neural network. Scoring is based on first make the loss as small as possible. After which we look at how you use as few number of network parameters as possible.

3. Hint: plot the latent variable with respect to original variable after you train your encoder/decoder and then guess the function form

Question 5 [3 marks]

Given the 2D figures below (Fig.3), build a multi-layer perceptron with two hidden layers to map one figure to the other. The two figures are related by a function f with input (x_i) and output (α_i, β_i) paired up by

$$(\alpha_i, \beta_i) = f(x_i) \tag{4}$$

You are given two sets of data points (x_i, y_i) files 'xy.npz' and (α_i, β_i) files 'ab.npz' but with the point randomly permuted so that you cannot match which (x) with which (α, β). You need to learn the function using a neural network.

1. 1D input x

2. 2 hidden layers of multilayer perceptrons with activation function

3. 2D output (α, β)

Plot (x_i) on the x-axis and (α_i, β_i) to reconstruct the plots shown in figure 3. Please save the result and code on your notebook.

Question 6 [2 extra marks]

Given a model M trained on MNIST ten classes digit which classifies the below images correctly. For instance, digit 0 is label as one-hot label $(1, 0, 0, ..., 0)$ and digit 1 is label as one-hot label $(0, 1, 0, ...)$. Now we take one image $I0$ (which

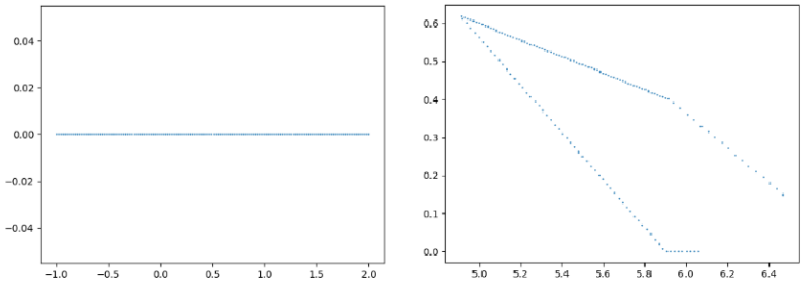

Figure 3: Left: indicates input, Right: output

is I0.png) and $I1$ (which is I1.png) and modify their pixel intensity a little bit to $I0'$ and $I1'$ such that the model M classifies $I0$ as digit 1 and classifies $I1$ as digit 7. Note that the model M does not change, only the data change. Impose the amount of change with L2-norm on 8bit grayscale images,

$$||I0 - I0'||_2 < \epsilon \tag{5}$$

$$||I1 - I1'||_2 < \epsilon \tag{6}$$

with $\epsilon = 0.1$. The way to find $I0'$ and $I1'$ is to optimize

$$I0' = \arg\min_{I0} ||p_1 - 1||_2^2 \tag{7}$$

where p_1 is the softmax of model output for digit 1. Similary, for $I1$,

$$I1' = \arg\min_{I1} ||p_0 - 1||_2^2 \tag{8}$$

You can use any optimizer and pass in image pixels as variables for gradient calculation. Please save the result and code on your notebook. Your code should clearly highlight (minimum): the classification of I0' and I1', images I0' and I1'. A notebook with a trained ResNet is given with weights 'model.pth'.

Question 7 [2 extra marks]

The following code cannot run. There is a problem of "Trying to backward through the graph a second time". Explain why you get this error and edit your code to fix the problem without changing the original functionality of the code. (e.g. if you delete all codes then the error message also disappears, however in this case, you will also not get any grades).

```
import torch
```

```
class G:

  def __init__(self):
    self.g=torch.ones([2])
  def __call__(self,x):
    self.g[0]=x*x
    return return torch.sum(self.g)

if __name__=='__main__':
  g=G()
  xinit=torch.ones([1],requires_grad=True)
  x = [xinit]
  x.append(x[-1]+g(x[-1]))
  loss=x[-1]**2
  loss.backward()
  print('------------ first time --------- ')
  x=[xinit]
  x.append(x[-1]+g(x[-1]))
  loss = x[-1]**2
  loss.backward()
  print('------------ well done --------- ')
```