CS 5787 Deep Learning, Spring 2020
Homework 0
Ananya Shivaditya (aks298)

Due: See Canvas

Instructions

Your homework solution must be typed. We urge you to prepare it in LaTeX. It must be output to PDF format. To use LaTeX, we suggest using `http://overleaf.com`, which is free and can be accessed online.

Your submission must cite any references used (including articles, books, code, websites, and personal communications). All solutions must be written in your own words, and you must program the algorithms yourself. **If you do work with others, you must list the people you worked with.** Submit your solutions as a PDF to Canvas.

Your programs must be written in python. The relevant code should be in the PDF you turn in. If a problem involves programming, then the code should be shown as part of the solution. One easy way to do this in LaTeX is to use the verbatim environment, i.e., \begin{verbatim} YOUR CODE \end{verbatim}.

You may wish to use the program *MathType*, which can easily export equations to AMS LaTeX so that you don't have to write the equations in LaTeX directly: `http://www.dessci.com/en/products/mathtype/`

If told to implement an algorithm, don't call a function that implements that algorithm in a toolbox, or you will receive no credit.

About Homework 0: Homework 0 is intended to review prerequisite skills, help you become familiar with LaTeX, and ensure you have your programming environment prepared. Later assignments will be more challenging and take far longer. This assignment should take no longer than four hours. Copy and paste this template into an editor, e.g., `www.overleaf.com`, and then just type the answers in. You can use a math editor to make this easier, e.g., CodeCogs Equation Editor or MathType.

CodeCogs: `https://www.codecogs.com/latex/eqneditor.php`

MathType: `http://www.dessci.com/en/products/mathtype/`

Problem 1 - Probability Review

Recall these rules from probability:

Bayes' rule is

$$P(B|A) = \frac{P(A|B)\,P(B)}{P(A)}.$$

Joint probability's relationship with conditional probability is

$$P(A \cap B) = P(A|B)\,P(B) = P(B|A)\,P(A).$$

If two events A and B are independent then their joint probability is

$$P(A \cap B) = P(A)\,P(B).$$

If two events A and B are not mutually exclusive then,

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

If two events A and B are mutually exclusive then $P(A \cap B) = 0$.

Hint: None of the questions in problem 1 have the same answer.

In this question, assume that the kangaroos have equal probability of having children that are male or female. Kangaroos almost always give birth to a single offspring.

Let $P(O = F)$ be the probability of the older offspring being female and let $P(Y = F)$ be the probability of the younger offspring being female.

Part 1 (3 points)

Suppose a kangaroo has two children. What is the probability that both are female? Show your derivation.

Answer:

$$P(O = F \cap Y = F)$$
$$= P(O = F) * P(Y = F)$$
$$= 0.5 * 0.5 = 0.25$$

Part 2 (3 points)

Suppose a kangaroo has two children and the oldest is female. What is the probability that both are female? Show your derivation.

Answer:

$$P(Y = F|O = F)$$

But these two events are independent.

$$= P(Y = F)$$

$$= 0.5$$

Part 3 (3 points)

Suppose a kangaroo has two children and at least one of them is female. What is the probability that both are female? Show your derivation.

Answer:

$$P[(O = F \cap Y = F)|(O = F \cup Y = F)]$$
$$= \frac{P(O = F \cap Y = F)}{P(O = F) + P(Y = F) - P(O = F \cap Y = F)}$$
$$= \frac{0.25}{0.5 + 0.5 - 0.25} = \frac{1}{3} = 0.667$$

Problem 2 - Linear Algebra Review #1

Let $\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 5 & -2 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 2 \end{pmatrix}$

Part 1 (3 points)

Compute $Trace(\mathbf{A})$.

Answer: $Trace(\mathbf{A}) = \Sigma a_{ii} = 1 + (-2) = -1$.

Part 2 (4 points)

Compute $Trace(\mathbf{B}\mathbf{B}^T)$.

Answer:
$Trace(\mathbf{B}\mathbf{B}^T) = \Sigma b_{ij}.b_{ij} = 1.1 + 2.2 + 2.2 = 9$

Problem 2 - Linear Algebra Review #2 (4 points)

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{D} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, and $\mathbf{x}^T(\mathbf{A} + \mathbf{D}) = \mathbf{b}^T$. Use the matrix inverse to solve for $\mathbf{x}$ and simplify.

Answer:
$\mathbf{x} = ?$
Given, $\mathbf{x}^T(\mathbf{A} + \mathbf{D}) = \mathbf{b}^T$
Mulitplying on both sides by $(\mathbf{A} + \mathbf{D})^{-\mathbf{1}}$,
$\mathbf{x}^T = \mathbf{b}^T(\mathbf{A} + \mathbf{D})^{-\mathbf{1}}$
Transposing both sides,
$\mathbf{x} = ((\mathbf{A} + \mathbf{D})^{-\mathbf{1}})^{\mathbf{T}}.\mathbf{b}$
So, $\mathbf{x} = (\mathbf{A}^{\mathbf{T}} + \mathbf{D}^{\mathbf{T}})^{-\mathbf{1}}.\mathbf{b}$

Problem 3 - Linear Algebra Review #3

Let matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$, where $n \neq m$.

Part 1 (1 point)

If it is possible to compute the matrix product $\mathbf{AB}$, give the size of the matrix produced. Otherwise, write, 'Not possible.'

Answer:
$\mathbf{A}.\mathbf{B} \in \mathbb{R}^{n\times m}$

Part 2 (1 point)

If it is possible to compute the matrix product **BA**, give the size of the matrix produced. Otherwise, write, 'Not possible.'

Answer: Not possible.

Problem 4 - Setting Up Python

Install Anaconda (or Whatever Works for You) and PyTorch

Python 3.6+, numpy, scipy, and matplotlib are necessary for the homeworks in this class. You can obtain these along with a nice IDE called Spyder by installing Anaconda (`http://www.anaconda.com/download`). If you know what you are doing and have experience already, you may use whatever you would like.

After installing your Python 3.6+ environment along with the other toolboxes, you may optionally install PyTorch, but it will not be required until Homework 2: `https://pytorch.org/`

Part 1 - Visualizing CIFAR-10 (4 points)

The CIFAR-10 dataset contains 60,000 RGB images from 10 categories. Download it from here: `https://www.cs.toronto.edu/~kriz/cifar.html`
Read the documentation.

Using the first CIFAR-10 training batch file, display the first three images from each of the 10 categories as a 3×10 image array. The images are stored as rows, and you will need to reshape them into $32 \times 32 \times 3$ images if you load up the raw data yourself. It is okay to use the PyTorch toolbox for loading them or you can roll your own.

Image and Code:

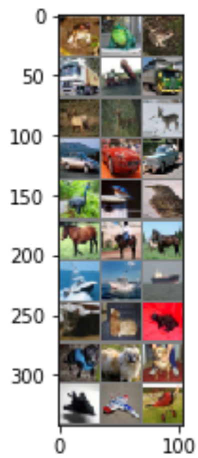
```
%matplotlib inline

import torch
import torchvision
import torchvision.transforms as transforms
import matplotlib.pyplot as plt
```

```
import numpy as np

transform = transforms.Compose(
    [transforms.ToTensor(),
     transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])

trainset = torchvision.datasets.CIFAR10(root='./data', train=True,
                                        download=True, transform=transform)

trainloader = torch.utils.data.DataLoader(trainset, batch_size=1000,
                                        shuffle=False, num_workers=2)

classes = ('plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog',
                                        'horse', 'ship', 'truck')

def imshow(img):
    img = img / 2 + 0.5 # denormalize
    npimg = img.numpy()
    plt.imshow(np.transpose(npimg, (1, 2, 0)))

dataiter = iter(trainloader)
images, labels = dataiter.next()

ct = {}
for i in range(len(images)):
    if int(labels[i]) not in ct:
        ct[int(labels[i])] = [i]
    elif len(ct[int(labels[i])]) < 3:
        ct[int(labels[i])].append(i)
    if len(list(ct)) == 30:
        break

lbls = []
imgs = []
for i in ct:
    lbls.append(i)
    c = torch.stack((images[ct[i][0]], images[ct[i][1]],images[ct[i][2]]))
    imgs.append(c)

i = torch.cat((imgs[0],imgs[1],imgs[2],imgs[3],imgs[4],imgs[5],
                                        imgs[6],imgs[7],imgs[8],imgs[9]))
```

```
imshow(torchvision.utils.make_grid(i, nrow=3))
print(' '.join('%5s' % classes[lbls[j]] for j in range(10)))
```

Output of images shown below

```
frog truck  deer   car  bird horse  ship   cat   dog plane
```



Part 2 - Playing with NumPy (4 points)

Write a function called gaussian(n, m) that returns an $n \times m$ NumPy array, and each entry of that array is a random number drawn from the standard normal distribution (Gaussian with mean 0 variance 1). Generate 100 2-D points using this function and then make a scatter plot of the points using Matplotlib.

Scatter Plot and Code:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

def gaussian(n,m):
    return np.random.normal(size=(n,m))

a = []
for i in range(100):
    a.append(gaussian(1,2))
```

```
a = np.array(a)
a = a.transpose()

plt.scatter(a[0],a[1],color='b',edgecolors='k')
plt.ylim(-6,10)
plt.xlim(-6,10)

plt.show()
```

Plot of 100 2D points shown below: