Introduction to Deep Learning M2177.0043
Seoul National University

Homework #(**HW1**)
Seo Junwon

INSTRUCTIONS

- Anything that is received after the deadline will be considered to be late and we do not receive late homeworks. We do however ignore your lowest homework grade.
- Answers to every theory questions need to be submitted electronically on ETL. Only PDF generated from LaTex is accepted.
- Make sure you prepare the answers to each question separately. This helps us dispatch the problems to different graders.
- Collaboration on solving the homework is allowed. Discussions are encouraged but you should think about the problems on your own.
- If you do collaborate with someone or use a book or website, you are expected to write up your solution independently. That is, close the book and all of your notes before starting to write up your solution.

1 Q1

1.1

Junwon	Seo
Computer Science & Engineering	2017-19428

1.2

2 Q2

2.1 Probability integral transform

$F_Y(y) = Pr(F_X(X) \leq y) = Pr(X \leq F_X^{-1}(y)) = F_X(F_X^{-1}(y))$ = y
As F_X is a continuous function,
$0 \leq F_X \leq 1$.
$F_Y(y) = y$ for $0 \leq y \leq 1$.
$\therefore$ Y is uniformly distributed in [0,1]

Introduction to Deep Learning M2177.0043
Seoul National University

Homework #(**HW1**)
Seo Junwon

2.2 Inverse transform sampling

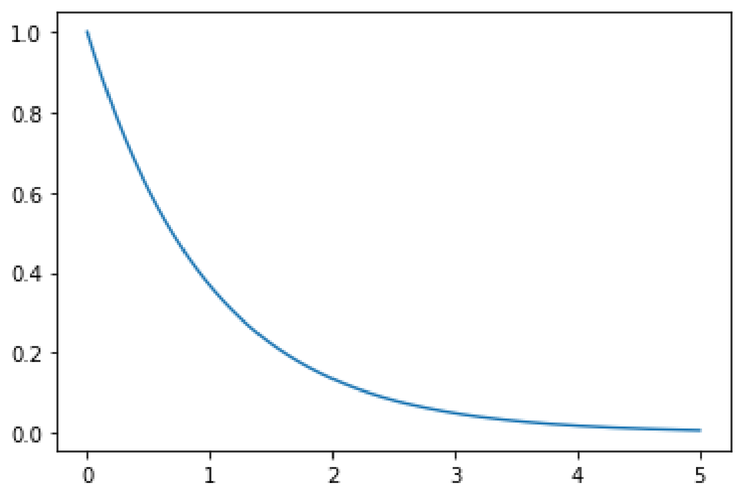
Let $Y = F_X^{-1}(U)$
Then the cdf of Y is as follow

$$F_Y(x) = Pr(F_X^{-1}(U) \leq x)$$
$$\iff F_Y(y) = Pr(U \leq F_X(x))$$

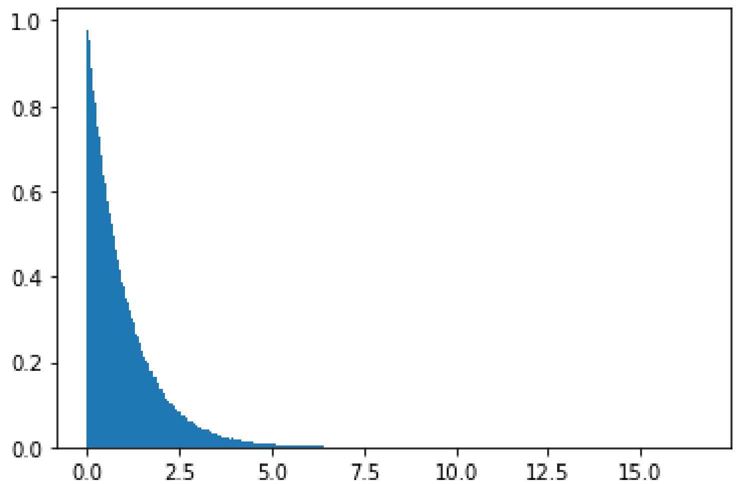
$Pr(U \leq F_X(x)) = F_X(x)$
because, if we constitute $F_X(x)$ for simply k
$Pr(U \leq k)$ is simply k.
Therefore, $F_X^{-1}(U)$ has $F_X(x)$ for its CDF.

2.3 Simulation

```
# (1) analytical PDF of exponential distribution.
import numpy as np
import matplotlib.pyplot as plt
    x = np.linspace(0, 5, 10000)
    lamb = 1
    y = lamb * np.exp(-lamb*x)
    plt.plot(x,y)
```




```
#  (2) normalized histogram of your samples (use 500 bins)
import numpy as np
import matplotlib.pyplot as plt
n = 500
a = np.random.exponential(1., size=1000000)

plt.hist(a, bins=500, density = True)
plt.show()
```



3 Q3

Let's assume I have owned K shares of stock B initially and sold x shares of it.
Then the variance of the shares "V" is

$$Var((k-x)X_B + X_A)$$
$$\iff (k-x)^2\sigma_B^2 + \sigma_A^2 + 2(k-x)\rho\sigma_B\sigma_A$$

If we differentiate this with respect to x,

$$\frac{\partial(V)}{\partial(x)} = -2(k-x)\sigma_B^2 - 2\rho\sigma_B\sigma_A$$

to Minimize V by $\frac{\partial(V)}{\partial(x)} = 0$, we obtain

$$\therefore x = k + \frac{\rho\sigma_A}{\sigma_B}$$

So if $\rho \geq 0$ we should sell all k shares of B
Else if $\rho < 0$ and $x \geq 0$, we should sell $k + \frac{\rho\sigma_A}{\sigma_B}$ shares of B

4 Q4

$Xv = \lambda v$
if v = $(1, 1, 1, \ldots, 1)^T$,
$Xv_i = 1 + c(p-1), \forall\lambda$
$\lambda = 1 + c(p-1)$

Then the traces of matrix X is p.
$(n-1) * \lambda + (1 + c(p-1)) = p$
$\lambda = -c + 1$ for v_i = $(1, \ldots, -p+1, \ldots, 1)$

Introduction to Deep Learning M2177.0043
Seoul National University

Homework #(**HW1**)
Seo Junwon

---

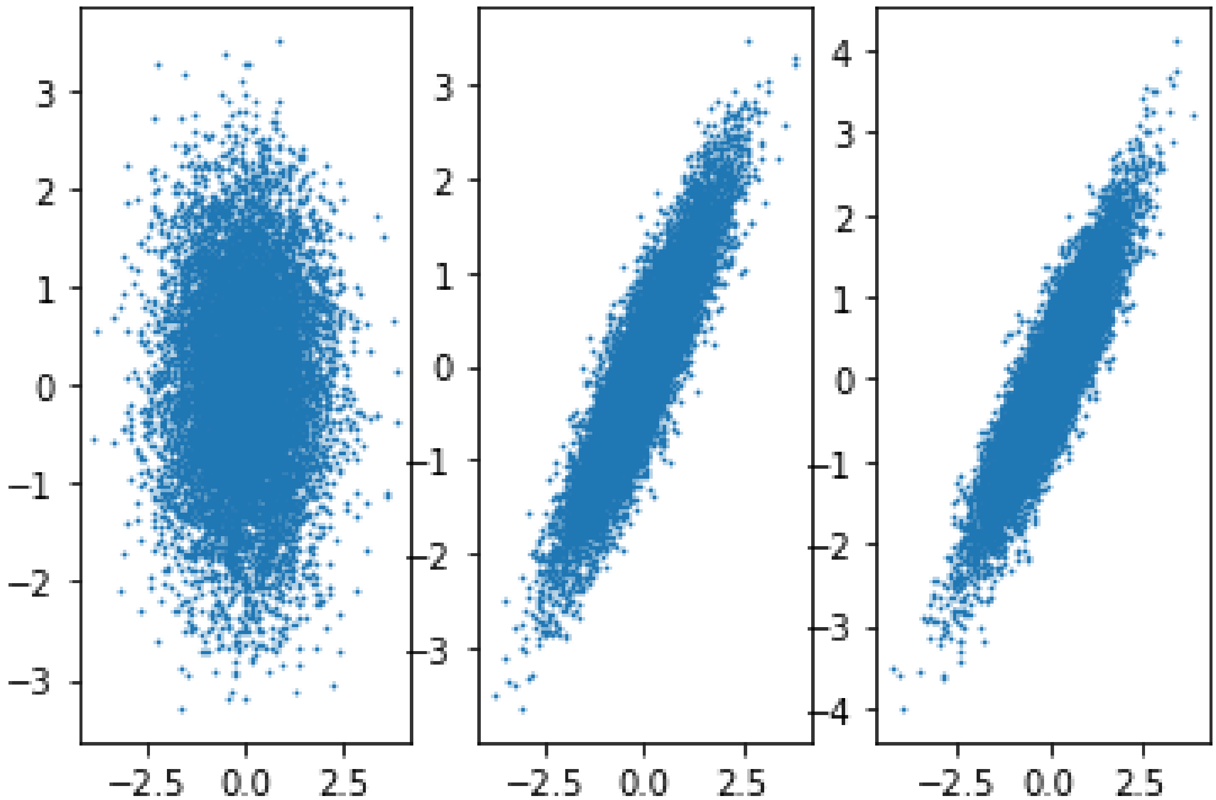
5 Q5

5.1

$E[Z] = \vec{0}$
$TZ = (TZ_1, \ldots, TZ_n)$
$\Sigma = Cov(TZ)$ and Σ is symmetric.
$\Sigma = QDQ^T = SS^T$. S is generated by eigen decomposition of the covariance matrix, $S = (QD^{1/2})$
Let $T = \mu + SZ$
$E[TZ] = \mu$ and, $Cov(TZ) = E[(TZ)(TZ)^T] = E[(SZ)(SZ)^T] = SE[ZZ^T]S^T = \Sigma$
$\therefore X = \mu + SZ$ takes Z to $N(\mu, \Sigma)$

5.2

```
# 1) the std normal, 2) your transformed multivariate samples, and 3) multivariate samples

import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg
sigma = np.array([ [1.0, 0.9], [0.9, 1.0]])
mean = np.zeros(2)
S = scipy.linalg.sqrtm(sigma)

np.random.seed(1337)

z = np.random.normal( size = (2, 10000) ) #std normal

X = np.matmul(S, z) # transformation

y = np.random.multivariate_normal(mean, sigma, 10000) # numpy

plt.subplot(1,3,1)
plt.scatter(z[0], z[1], s=0.5)

plt.subplot(1,3,2)
plt.scatter(X[0], X[1], s=0.5)

plt.subplot(1,3,3)
plt.scatter(y[...,0], y[...,1], s = 0.5)
```



The result of my own quite matches that of the Numpy's, in my humble opinion.

6 Q6

$\triangledown f(x^*)^T(x - x^*) \geq 0, \forall x \in R^+$
Let $x = \vec{0}$, we obtain

$$\triangledown f(x^*)^T(x^*) \leq 0 \tag{1}$$

As $x^* \in R^+$, if let $x = x^* - x_i^*$, s.t $x_i^* = (0, \ldots, x_i^*, \ldots, 0)$, which means x_i^* has only i^th component of x_i^*, otherwise 0.
$\forall i, \triangledown f(x^*)$'s ith component $\triangledown f(x^*)_i$ satisfies following

$$f(x^*)_i * x_i^* \leq 0 \tag{2}$$

which implies $\triangledown f(x^*) \in R^+$
By equation (1), (2), we obtain

$$\triangledown f(x^*) = \vec{0}$$

$$or$$

$$\exists\ i\ s.t\ x_i^* = 0$$

as $x \in R^+$