EURECOM UNIVERSITY

Energy efficient communication for task-oriented networks and intelligent communication systems

Research Project Report

Jerold Kingston GANASEKARAN,

Divya Teresa JOHNSON

Supervised by:

Derya MALAK, EURECOM University

June 28, 2022

Contents

Abstract

We initiate this project for the purpose of reducing the computation cost. The Energy required to transfer a single bit costs a lot in terms of computation and communication. In late 2018, Telecom operators accounted for around 5 percent of operating expenditures considered only for computation; they account for 2 to 3 percent of total global energy demand, often making them some of the most energy-intensive companies in their geographic markets. As an operator's energy consumption exceeds the limit due to an increasing amount of population, it increases their carbon footprint and tends to harm the environment unintentionally. Therefore, we figured out an optimal way to reduce the computation cost in an energy efficient way.

1 Introduction

Lets dive into the basics to understand the concept well

1.1 Communication

In ancient days communication between the people was established in the way of expressing their emotions by actions and later into word of mouth. In the late 19's, the invention of the radio signals changed the entire world. People start using the radio signals to communicate from one end to other end wirelessly. The improvement over the radio signals helps to combine more than one user to communicate per channel using an advanced multiple access technique known as TDMA, FDMA, CDMA and OFDMA. Nowadays, The name IOT pops up everywhere indicating the internet of things connected in the network. The thing represents the number of nodes (end devices) connected in the single network that wants to exploit the resources of the network. If the number of devices in the network increases then the amount of energy needed to transfer the packets also increases. There are many energy efficient ways to reduce the energy consumption of a

device using a high level network layer such as LoRa, LoRAwan and 6lowan referenced from the textbook fundamental of LTE by Jeffrey G. Andrews. These are the well known protocols used in order to reduce the power consumption in the network layer, these high level layers will not be helpful when the number of users increases, then the computational and communicational cost raise exponentially.
To overcome the problem we initiated a method to reduce the computation cost. The computation cost is the amount of energy required to compute the total number of bits to transfer in an encoder, and the communication cost is the energy needed to transfer the bits from an encoder to another device. The above protocols are defined to reduce the communication cost between the users, our focus is fully on the computation part in an encoder to reduce the energy consumption in an optimal way. Let's look at a simple example to get a basic overview of computational load minimization, consider two sources sending its data to the decoder both are uncorrelated but not fully uncorrelated only they don't have a communication contact between each other. By sending a single source data with the joint probability density function to the decoder side we can efficiently reduce the computation load of any one of the source by looking on to the function used, we can get a clear overview about this uncorrelated sources and minimization of computational cost in the following uncorrelated to destination section.

Figure 1: World of communication.

1.2 Source and destination

How the communication really established between the two entities?. Here let's take a simple example to understand the communication. A source is the single node or a single user wants to communicate with the other user or a destination node. There are two ways to communicate with the destination entity. One is making a connection as wired and another is connecting wirelessly. For the wireless transferring, both the entities must equipped with antennas to transmit and receive a radio signal. On the transmitter side the encoder is required; its job is to encode all the bits needed to communicate and at the receiver side a decoder is required to decode the captured radio signal from the noise. The wireless connection is more complex than the wired one, one needs to consider a channel model before sending any information to the

channel otherwise the information would get faded by the environment. A modulation technique at the encoder helps to carry the signal from the lower baseband to the higher passband to transmit the single to long distance through a space. This is how basic communication models work.

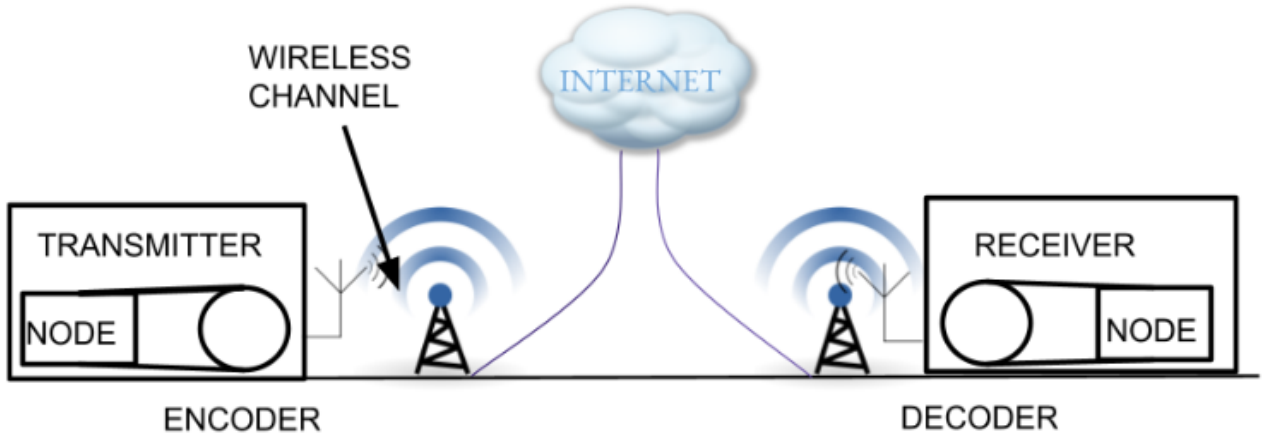

Figure 2: Communication between source and destination.

1.3 Uncorrelated sources to destination

Here is where the real problem arises. So far we have seen a point to point communication between the single source and the single destination. Now in this case, we are considering two or more sources at the transmitter that don't have any knowledge of each other in the network and want to communicate to the server or gateway. This is what we call an uncorrelated source of communication problems. However, to be more precise, they are not fully uncorrelated. sources are actually correlated just they cannot contact each other makes the two sources uncorrelated. Here in the below figure there are two users communicating with the network server having no intelligence of each other. Each user has to send their own info to the server individually. Apparently, if the two sources send the data separately to the network, then the energy used to encode all the information bits to transfer to the network is not the same for both the sources it varies arbitrarily. In this case, the energy spent to compute bits and to transfer the same info is doubled at the destination. In other words, the communication and computation load is doubled. A lot of energy wasted at the transmitter side to compute the same information.

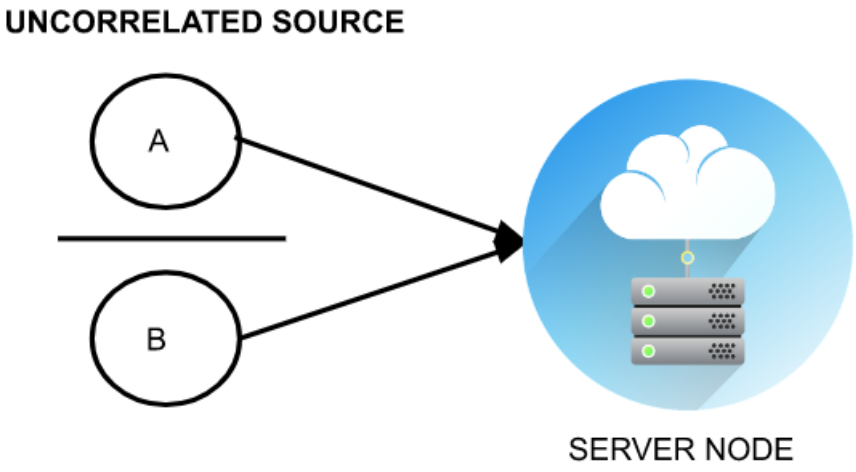

Figure 3: Communication between uncorrelated sources and destination.

2 Efficient communication for computation

Efficient communication refers to the ability to deliver a clear message with less computational cost.Through Neural Network Coding, the data-driven approach to joint source and network coding. It does not involve any long block-length source or channel coding techniques, and therefore is free of their associated latency penalty and computational cost[2]. A statistical method has been adopted for computation of channel capacity, showing that a better convergence per computation can be achieved [3]. Using MapReduce and Coded Distributed computing is proposed to demonstrate that increasing the computation load of the Map function can create novel coding opportunities that reduce the communication load by the same factor[1].

3 Theory of Shannon

From the book of "Elements of Information Theory" Shannon surprised the communication theory community by proving that the probability of error could be made nearly zero for all communication rates below channel capacity. The capacity can be computed simply from the noise characteristics of the channel. Shannon further argued that random processes such as music and speech have an irreducible complexity below which the signal cannot be compressed. This he named the entropy. If the entropy of the source is less than the capacity of the channel, error-free communication can be achieved. The Shannon limit or Shannon capacity of a communication channel refers to the maximum rate of error-free data that can theoretically be transferred over the channel if the link is subject to random data transmission errors, for a particular noise level. If the channel is noiseless and the speed is maximum possible in the given physics of the system, a channel can be used in a range from 0 to 100 percent. Let's say that is from 0 to 1 (inclusive). If a signal is perfect (ideal) it takes a small differential width of the full bandwidth. So, infinitely differentiable small parts fit in a whole small width between 0 and 1 and the speed is maximum with infinite capacity.
On the side of data compression, Shannon's entropy represents a lower limit for lossless data compression: the minimum amount of bits that can be used to encode a message without loss. Shannon's source coding theorem states that a lossless data compression scheme cannot compress messages, on average, to have more than one bit of Shannon's information per bit of encoded message. Elementary calculus shows that the expected description length must be greater than or equal to the entropy. Shannon's simple construction shows that the expected description length can achieve the bound asymptotically for repeated descriptions. Thus, the entropy is the data compression limit as well as the number of bits needed in random number generation, and codes achieving H turn out to be optimal from many points of view which is for lossless compression of data. The ratio of the entropy of a source to the maximum value it could have while still restricted to the same symbols is called its relative entropy. This is the maximum compression possible.

3.1 Entropy

The entropy is a measure of the average uncertainty in the random variable. It is the number of bits on average required to describe the random variable.

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log \mathrm{P}(x_i)$$

Which is defined as the negative sum of the probability of each possible outcome times the log (the inverse function to exponentiation) of the probability of each probability outcome.
Shannon's entropy is a measure of the potential reduction in uncertainty in the receiver's knowledge. Which is the process of gaining information as equivalent to the process of losing uncertainty. Hence, according to

Shannon's formula, a message's entropy is maximized when the occurrence of each of its individual parts is equally probable. What this means is that we will gain the largest amount of Shannon's information when dealing with systems whose individual possible outcomes are equally likely to occur.

3.2 Conditional Entropy

The conditional entropy quantifies the amount of information needed to describe the outcome of a random variable Y given that the value of another random variable X is known. Here, information is measured in shannons, nats, or hartleys. The entropy of Y conditioned on X is written as H (Y|X) which is given by the negative sum of the probability of x and y times the log (the inverse function to exponentiation) of the probability x and y by the probablity of x.
Let H (Y|X = x) be the entropy of the discrete random variable Y conditioned on the discrete random variable X taking a certain value x. H (Y|X) = 0 if and only if the value of Y is completely determined by the value of X.

The conditional entropy of Y given X is defined as

$$\mathrm{H}(Y|X) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)}$$

Where x and y denote the support sets of X and Y.

According to the definition, $H(Y|X) = \mathbb{E}[f(X,Y)]wheref : (x,y) \rightarrow -\log(\ p(y|x)\ ).fassociates(x,y)$

the information content of $(Y = y)given(X = x)$,

which is the amount of information needed to describe the event $(Y = y)given(X = x)$.

According to the law of large numbers, $H(Y|X)$

is the arithmetic mean of a large number of independent realizations of $f(X,Y)$.

3.3 Mathematical Expression

A given communication system is described (from the external point of view) by giving the conditional probability Px(y) that if message x is produced by the source the recovered message at the receiving point will be y. The system as a whole (including source and transmission system) is described by the probability function P(x;y) of having message x and final output y. If this function is known, the complete characteristics of the system from the point of view of fidelity are known. Any evaluation of fidelity must correspond mathematically to an operation applied to P(x;y). This operation must at least have the properties of a simple ordering of systems; i.e., it must be possible to say of two systems represented by P1(x;y) and P2(x;y) that, according to our fidelity criterion, either (1) the first has higher fidelity, (2) the second has higher fidelity, or (3) they have equal fidelity. This means that a criterion of fidelity can be represented by a numerically valued function: P(x;y) whose argument ranges over possible probability functions P(x;y)[4].

4 Problem Formulation

After going through Shannon's Mathematical Theory of Communication [4] and Amit Solomon's neural network coding [1]. Let's consider the example again from Figure 3, two uncorrelated sources. The uncorrelated source energy crisis can be minimized by considering one source information with the joint probability function at the destination side. Since the two sources are uncorrelated but not fully uncorrelated, we can make use of a function that makes the uncorrelated sources correlate with the joint probability density. The function represents the likeness of the two sources. For instance, node A collects the information of weather data (rainy or sunny) in some location that B doesn't know. the node B collects the same weather data at some location that A doesn't know. Since A and B collect the same weather data representing a function. To reduce the computational load on the encoder side any one of the two uncorrelated sources send its full data to the destination node with the function. The function used at the destination node calculates the possible probability of the same data occurring for another node using Shannon's conditional probability definition. For now, assuming B alone sends its information to the destination node, the node at the destination uses the given function to calculate the probability of the same data occurring for three different functions. For each function, the computational cost is calculated for A node.

Function assumed F= A*B, F= A+B, F= A = B

consider source data B in the format of matrix = $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

Let's make it fix for the rest of the functions Another source A which is unknown to the destination node. The destination node tries to find the possible occurrences of A using the functions.

Here we are calculating the computational load of A using three different joint probabilities

Pab (1) = $\begin{bmatrix} f(x) & 1/2-f(x) \\ 1/2-f(x) & f(x) \end{bmatrix}$, Pab(2) = $\begin{bmatrix} f(x) & 1/2 \\ 1/2-2f(x) & f(x) \end{bmatrix}$, the third joint probability distribution formed by the addition of 1st two joint distributions. Pab(3) = pab(1)+pab(2) and then the conditional entropy taken to ensure the likeness.

4.1 Computation Part

Computation cost of A for the function F=A*B

With the help of the B data, the function at the destination node can find an exact remaining amount of load that A needs to send to the destination. From the basic arithmetic calculation. we know the fact of multiplication that when 0 entry in B will also be 0 in A as well. Therefore the total load of A calculated for three different joint probability distributions given in the below figure

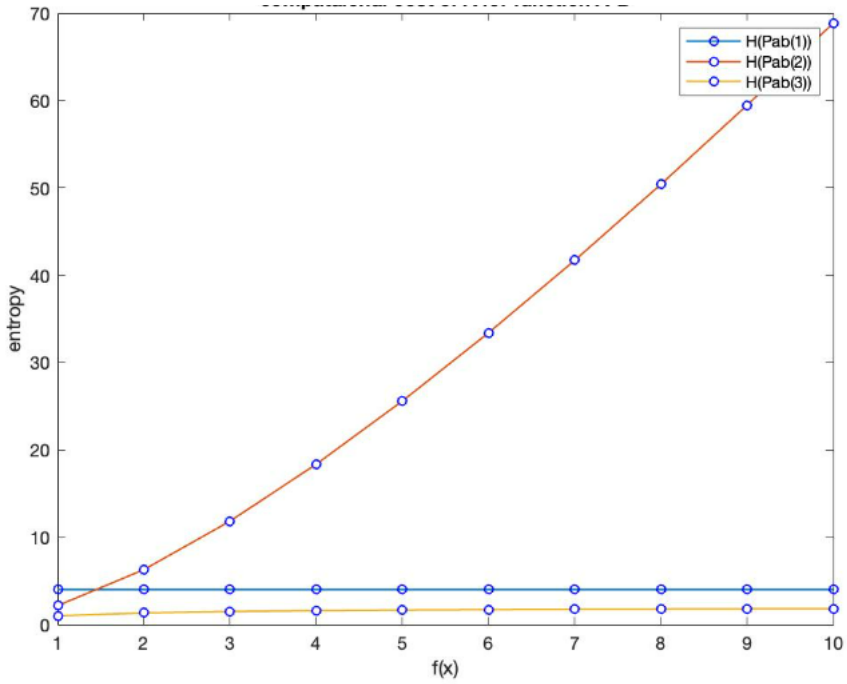

Figure 4: Computational cost of A for function A*B

In this case we considered the function F = (A = B) since B is equal to A, it don't need to send anything to the destination node, therefore, the computational cost for A is zero.

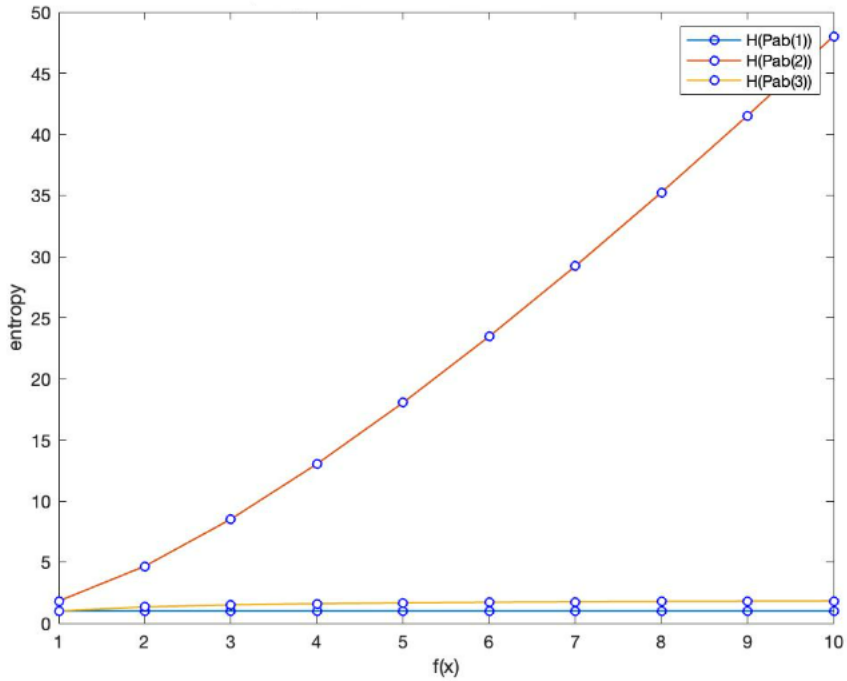

Figure 5: Computational cost of A for function A=B

In the third function A+B we don't know anything about the A data and its difficult to calculate the probability for this function therefore, A has to send everything to the destination node.

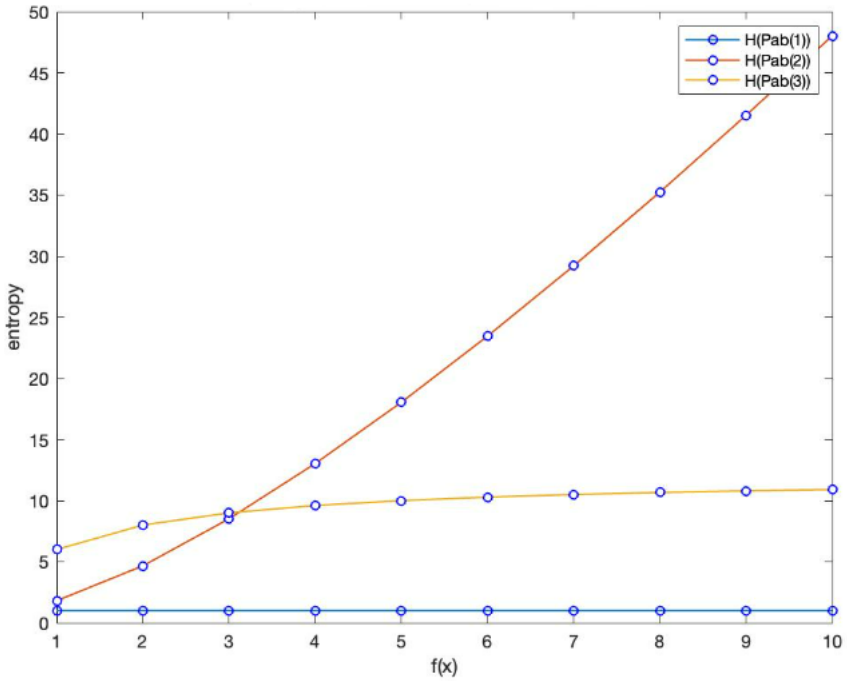

Figure 6: Computational cost of A for function A+B

5 Results

The above three figures show the computational cost for three different probabilities varies according to the function that we considered. The computational complexity increases when we increase the distribution's complexity range from 1 to 20. It clearly shows that the first two joint probability distributions remain constant in all the functions. This is because the entropy of these distributions represent the individual computation cost of A. Our focus is mainly on the behavior of the conditional entropy of the third joint distribution. The conditional entropy behavior changes according to the load computed. The result of the 1st figure shows that the 3rd conditional distribution needs to send a total of four entries to the destination. The four entries were calculated based on the function. Following the second shows that there is no need to send any entries from A to the destination for the function A= B, since the destination knows everything about A. At last in the 3rd figure the destination does not have any information about A because of the function A+B, therefore all the entries need to be sent to the destination so the computation cost increases slightly. The conditional entropy gives us a slight increase even though when we transfer all the entries this is what we need and this is how the computation cost can be reduced.

6 Conclusion

Compared to all the three joint distributions, when we increase the complexity, the conditional entropy of the third joint distribution gives us a lesser computation cost compared to the individual cost of, this is due to the condition on A given B. Thus the destination node requested only for the data that are not available in B to the transmitter node. From these aforementioned results we can efficiently reduce the energy required to compute the data to transfer. This energy efficient method would have also been used in the neural network based encoders to reduce the computation cost.

References

[1] Songze Li, Mohammad Ali Maddah-Ali, Qian Yu, and A. Salman Avestimehr. "A Fundamental Tradeoff between Computation and Communication in Distributed Computing". In: (2017). URL: https://arxiv.org/pdf/1604.07086.pdf.

[2] Litian Liu, Amit Solomon, Salman Salamatian, and Muriel M´edard. "Neural Network Coding". In: *Research Lab of Electronics, MIT, Cambridge, MA*. (2020).

[3] Mohammad Rezaeian. "Computation of Capacity for Gilbert-Elliott Channels, Using a Statistical Method". In: *Department of Electrical and Electronic Engineering, University of Melbourne, Victoria* (2005). URL: https://people.eng.unimelb.edu.au/rezaeian/AUSCTW05.pdf.

[4] C. E. Shannon. "A Mathematical Theory of Communication". In: *Reprinted with corrections from The Bell System Technical Journal* (1948). URL: http://www.cs.ox.ac.uk/activities/ieg/e-library/sources/shannon1948.pdf.

7 Appendix

Code of our project

```
clc;
clear all;
close all;
count = 0;
// for function F = A*B
B = [0,1,1;1,0,1]
A = ones(2,3)
f = @(x) x
X = 1:10
for i = 1:length(B)-1
for j = 1:length(B)
b1(i,j) = B(i,j)*A(i,j)
if b1(i,j) == 0
count = count+1;
end
end
end

// total entities transfer from A
total = numel(A)-count
for x = 1:length(X)

// Probability distribution table 1
pa_b1 = [f(x)1/2 − f(x); 1/2 − f(x)f(x)]
A1_0 = sum(pa_b1(1,:))
A1_1 = sum(pa_b1(2,:))
P1_A = A1_0 + A1_1; //totalprobabilityofA
B1_0 = sum(pa_b1(:,1))
```

$B1_1 = sum(pa_b1(:,2))$
$P1_B = B1_0 + B1_1 //totalprobabilityofB$

$//EntropyofH(X)$
$H1_A(x,:) = A1_0 * log2(A1_0) + 1 - (A1_1) * log2(1 - (A1_1))$
$H1_B(x,:) = B1_0 * log2(B1_0) + 1 - (B1_1) * log2(1 - (B1_1))$

$//Probabilitydistributiontable2$
$pa_b2 = [f(x)1/2; 1/2 - 2 * f(x)f(x)]$
$A2_0 = sum(pa_b2(1,:))$
$A2_1 = sum(pa_b2(2,:))$
$P2_A = A2_0 + A2_1; //totalprobabilityofX$
$B2_0 = sum(pa_b2(:,1))$
$B2_1 = sum(pa_b2(:,2))$
$P2_B = B2_0 + B2_1 //totalprobabilityofY$

$//EntropyofH(X)$
$H2_A(x,:) = (A2_0 * log2(A2_0) + 1 - (A2_1) * log2(1 - (A2_1)))$
$H2_B(x,:) = (B2_0 * log2(B2_0) + 1 - (B2_1) * log2(1 - (B2_1)))$

$//probabilitydistributiontable3$
$pa_b3 = 0.5 * pa_b1 + ((0.5) * pa_b2)$
$A3_0 = sum(pa_b3(1,:))$
$A3_1 = sum(pa_b3(2,:))$
$P3_A = A3_0 + A3_1; //totalprobabilityofX$
$B3_0 = sum(pa_b3(:,1))$
$B3_1 = sum(pa_b3(:,2))$
$P3_B = B3_0 + B3_1 //totalprobability$

$//EntropyofH(X)$
$H3_A(x,:) = (A3_0 * log2(A3_0) + 1 - (A3_1) * log2(1 - (A3_1)))$
$H3_B(x,:) = abs(-B3_0 * log(B3_0) - (B3_1) * log(B3_1))$

$//EntropyofH(A/B = 1)$
$H322(x,:) = pa_b3(2,2)$
$H_AB(x,:) = (H322(x,:))./(B3_1)$
$//H_AB3 = (H_AB).* log(H_AB)$
$//H_AB2(x,:) = (P_{a1b1}) * log(P_{a1b1})$
end

$plot(X, total * H1_A,'-$
$o','MarkerIndices',1:20:length(H1_A),'LineWidth',1,'MarkerSize',5,'Marker$
$EdgeColor','b')$
$holdon$
$plot(X, H2_A,'-$

```
o',' MarkerFaceColor',[1,1,1],' MarkerIndices',1:20:length(H2_A),' LineWidth
',1,' MarkerSize',5,' MarkerEdgeColor',' b')
plot(X,H_AB,' -
o',' MarkerFaceColor',[1,1,1],' MarkerIndices',1:20:length(H_AB),' LineWidth
',1,' MarkerSize',5,' MarkerEdgeColor',' b')
holdon
xlabel('numberofiteration')
ylabel('entropy')
title('communicationcostofAforfunctionA*B')
legend('H(Pab(1))',' H(Pab(2))',' H(Pab(3))')

//forfunctionF = A+B
B1 = [0,1,1;1,0,1]
A1 = ones(2,3)
f = @(x)x
count1 = 0
fori = 1:length(B1)-1
forj = 1:length(B1)
B1(i,j) = A1(i,j)
ifB1(i,j) == A1(i,j)
count1 = count1+1;
end
end
end

//totalentitiestransferfromA
total1 = numel(A)-count1
forx = 1:length(X)

//Probabilitydistributiontable1
pa_b1 = [f(x)1/2-f(x);1/2-f(x)f(x)]
A1_0 = sum(pa_b1(1,:))
A1_1 = sum(pa_b1(2,:))
P1_A = A1_0+A1_1;//totalprobabilityofA
B1_0 = sum(pa_b1(:,1))
B1_1 = sum(pa_b1(:,2))
P1_B = B1_0+B1_1//totalprobabilityofB

//EntropyofH(X)
H1_A(x,:) = A1_0*log(A1_0)+1-(A1_1)*log(1-(A1_1))
H1_B(x,:) = B1_0*log(B1_0)+1-(B1_1)*log(1-(B1_1))

//Probabilitydistributiontable2
pa_b2 = [f(x)1/2;1/2-2*f(x)f(x)]
A2_0 = sum(pa_b2(1,:))
```

$A2_1 = sum(pa_b2(2,:))$
$P2_A = A2_0 + A2_1; //totalprobabilityofX$
$B2_0 = sum(pa_b2(:,1))$
$B2_1 = sum(pa_b2(:,2))$
$P2_B = B2_0 + B2_1 //totalprobabilityofY$

$//EntropyofH(X)$
$H2_A(x,:) = (A2_0 * log(A2_0) + 1 - (A2_1) * log(1 - (A2_1)))$
$H2_B(x,:) = (B2_0 * log(B2_0) + 1 - (B2_1) * log(1 - (B2_1)))$

$//probabilitydistributiontable3$
$pa_b3 = 0.5 * pa_b1 + ((0.5) * pa_b2)$
$A3_0 = sum(pa_b3(1,:))$
$A3_1 = sum(pa_b3(2,:))$
$P3_A = A3_0 + A3_1; //totalprobabilityofX$
$B3_0 = sum(pa_b3(:,1))$
$B3_1 = sum(pa_b3(:,2))$
$P3_B = B3_0 + B3_1 //totalprobabilityofY$

$//EntropyofH(X)$
$H3_A(x,:) = (A3_0 * log(A3_0) + 1 - (A3_1) * log(1 - (A3_1)))$
$H3_B(x,:) = (B3_0 * log(B3_0) + 1 - (B3_1) * log(1 - (B3_1)))$
$//EntropyofH(A/B = 1)$
$H3B1(x,:) = pa_b3(2,2)$
$H_AB(x,:) = (H3B1(x,:))./(B3_1)$
end

$figure(2)$
$plot(X, H1_A,'-$
$o','MarkerIndices',1:20:length(H1_A),'LineWidth',1,'MarkerSize',5,'Marker$
$EdgeColor','b')$
$holdon$
$plot(X, H2_A,'-$
$o','MarkerFaceColor',[1,1,1],'MarkerIndices',1:20:length(H2_A),'LineWidth$
$',1,'MarkerSize',5,'MarkerEdgeColor','b')$
$plot(X, H_AB,'-$
$o','MarkerFaceColor',[1,1,1],'MarkerIndices',1:20:length(H_AB),'LineWidth$
$',1,'MarkerSize',5,'MarkerEdgeColor','b')$
$xlabel('numberofiteration')$
$ylabel('entropy')$
$title('communicationcostofAforfunctionA = B')$
$legend('H(Pab(1))','H(Pab(2))','H(Pab(3))')$

$//forfunctionF = A + B$
$B1 = [0,1,1;1,0,1]$

$A1 = ones(2,3)$
$f = @(x)x$
$count2 = 0$

$//for functionF = A + B$
$for i = 1 : length(B1) - 1$
$for j = 1 : length(B1)$
$b2(i,j) = B1(i,j) + A1(i,j)$
$if b2(i,j) == 0$
$count2 = count2 + 1;$
end
end
end

$//totalentitiestransferfromA$
$total2 = numel(A) - count2$
$for x = 1 : length(X)$

$//Probabilitydistributiontable1$
$pa_b1 = [f(x)1/2 - f(x); 1/2 - f(x)f(x)]$
$A1_0 = sum(pa_b1(1,:))$
$A1_1 = sum(pa_b1(2,:))$
$P1_A = A1_0 + A1_1; //totalprobabilityofA$
$B1_0 = sum(pa_b1(:,1))$
$B1_1 = sum(pa_b1(:,2))$
$P1_B = B1_0 + B1_1 //totalprobabilityofB$

$//EntropyofH(X)$
$H1_A(x,:) = A1_0 * log(A1_0) + 1 - (A1_1) * log(1 - (A1_1))$
$H1_B(x,:) = B1_0 * log(B1_0) + 1 - (B1_1) * log(1 - (B1_1))$

$//Probabilitydistributiontable2$
$pa_b2 = [f(x)1/2; 1/2 - 2 * f(x)f(x)]$
$A2_0 = sum(pa_b2(1,:))$
$A2_1 = sum(pa_b2(2,:))$
$P2_A = A2_0 + A2_1; //totalprobabilityofX$
$B2_0 = sum(pa_b2(:,1))$
$B2_1 = sum(pa_b2(:,2))$
$P2_B = B2_0 + B2_1 //totalprobabilityofY$

$//EntropyofH(X)$
$H2_A(x,:) = (A2_0 * log(A2_0) + 1 - (A2_1) * log(1 - (A2_1)))$
$H2_B(x,:) = (B2_0 * log(B2_0) + 1 - (B2_1) * log(1 - (B2_1)))$

$//probabilitydistributiontable3$

$pa_b3 = 0.5 * pa_b1 + ((0.5) * pa_b2)$
$A3_0 = sum(pa_b3(1,:))$
$A3_1 = sum(pa_b3(2,:))$
$P3_A = A3_0 + A3_1; //totalprobabilityofX$
$B3_0 = sum(pa_b3(:,1))$
$B3_1 = sum(pa_b3(:,2))$
$P3_B = B3_0 + B3_1 //totalprobabilityofY$

$//EntropyofH(X)$
$H3_A(x,:) = (A3_0 * log(A3_0) + 1 - (A3_1) * log(1 - (A3_1)))$
$H3_B(x,:) = (B3_0 * log(B3_0) + 1 - (B3_1) * log(1 - (B3_1)))$

$//EntropyofH(A/B = 1)$
$H3B1(x,:) = pa_b3(1,2)$
$H_AB(x,:) = (H322(x,:))./(B3_1)$
end
$figure(3)$
$plot(X, H1_A,'-$
$o','MarkerIndices',1:20:length(H1_A),'LineWidth',1,'MarkerSize',5,'Marker$
$EdgeColor','b')$
$holdon$
$plot(X, H2_A,'-$
$o','MarkerFaceColor',[1,1,1],'MarkerIndices',1:20:length(H2_A),'LineWidth$
$',1,'MarkerSize',5,'MarkerEdgeColor','b')$
$plot(X, total2 * H_AB,'-$
$o','MarkerFaceColor',[1,1,1],'MarkerIndices',1:20:length(H_AB),'LineWidth$
$',1,'MarkerSize',5,'MarkerEdgeColor','b')$
$xlabel('numberofiteration')$
$ylabel('entropy')$
$title('communicationcostofAforfunctionA+B')$
$legend('H(Pab(1))','H(Pab(2))','H(Pab(3))')$