Result

Question 1 ¶

p = 3 :

Train Error: 0.012672176308539946

Test Error: 0.04221635883905013

p = 4 :

Train Error: 0.007988980716253443

Test Error: 0.030343007915567283

p = 5 :

Train Error: 0.006887052341597796

Test Error: 0.051451187335092345

Question 2

p = 3 :

Train Error: 0.012396694214876033

Test Error: 0.052770448548812667

p = 4 :

Train Error: 0.009090909090909

Test Error: 0.0316622691292876

p = 5 :

Train Error: 0.007713498622589532

Test Error: 0.04617414248021108

Question 3

Corresponding strings are 'WDTAG' and 'LFLNK'

Code

Read Files

In [1]:

```
amino_acid = ["A","R","N","D","C","Q","E","G","H","I","L","K","M","F","P","S","T","W","Y","V"]
```

In [2]:

```
train = open("pa4train.txt")
train = [l.strip().split() for l in train]
train = [["".join([w if w in amino_acid else "X" for w in x])]+[int(y)] for x,y in train]
```

In [3]:

```
test = open("pa4test.txt")
test = [l.strip().split() for l in test]
test = [["".join([w if w in amino_acid else "X" for w in x])]+[int(y)] for x,y in test]
```

Functions

In [4]:

```
def kernel(data, length):
    d = []
    for each in data:
        d_each = dict()
        a = each[0]
        b = each[1]
        for i in range(len(a) - length + 1):
            if not a[i:i+length] in d_each:
                d_each[a[i:i+length]] = 0
            d_each[a[i:i+length]] += 1
        d = d + [[d_each] + [b]]
    return d
```

In [5]:

```
def modified_kernel(data, length):
    d = []
    for each in data:
        d_each = dict()
        a = each[0]
        b = each[1]
        for i in range(len(a) - length + 1):
            d_each[a[i:i+length]] = 1
        d = d + [[d_each] + [b]]
    return d
```

In [6]:

```
def modified_add(x, y):
    d = dict()
    for i in x:
        d[i] = x[i]
    for i in y:
        if not i in d:
            d[i] = 0
        d[i] = d[i] + y[i]
    return d
```

In [7]:

```
def modified_dot(x, y):
    result = 0
    for i in x:
        if i in y:
            result = result + x[i] * y[i]
    return result
```

In [8]:

```
def modified_mul(x,y):
    d = dict()
    for i in y:
        d[i] = x * y[i]
    return d
```

In [9]:

```
def modified_perception(data):
    d = dict()
    for each in data:
        x = each[0]
        y = each[1]
        thresh = y * modified_dot(d,x)
        if thresh <= 0:
            d = modified_add(d, modified_mul(y,x))
    return d
```

In [10]:

```
import random
def get_error(data, s, p):
    c = 0
    for i in range(len(data)):
        thresh = modified_dot(s[i][0], p)
        if thresh > 0:
            sign = 1
        elif thresh < 0:
            sign = -1
        else:
            sign = random.choice([-1, 1])
        if(sign != data[i][-1]):
            c = c + 1
    return c / len(data)
```

Question 1

In [11]:

```
print("Errors: ")
print()
for i in range(2, 6, 1):
    s_train = kernel(train, i)
    p = modified_perception(s_train)
    train_error = get_error(train, s_train, p)

    s_test = kernel(test, i)
    test_error = get_error(test, s_test, p)

    print("p = ", i, ":")
    print("Train Error:",train_error)
    print("Test Error:",test_error)
```

```
Errors: 

p =  2 :
Train Error: 0.07107438016528926
Test Error: 0.08179419525065963
p =  3 :
Train Error: 0.012672176308539946
Test Error: 0.04221635883905013
p =  4 :
Train Error: 0.007988980716253443
Test Error: 0.030343007915567283
p =  5 :
Train Error: 0.006887052341597796
Test Error: 0.051451187335092345
```

Question 2

In [12]:

```
print("Errors: ")
print()
for i in range(2, 6, 1):
    s_train = modified_kernel(train, i)
    p = modified_perception(s_train)
    train_error = get_error(train, s_train, p)

    s_test = modified_kernel(test, i)
    test_error = get_error(test, s_test, p)

    print("p = ", i, ":")
    print("Train Error:",train_error)
    print("Test Error:",test_error)
```

```
Errors:

p =  2 :
Train Error: 0.08181818181818182
Test Error: 0.09630606860158311
p =  3 :
Train Error: 0.012396694214876033
Test Error: 0.052770448548812667
p =  4 :
Train Error: 0.009090909090909
Test Error: 0.0316622691292876
p =  5 :
Train Error: 0.007713498622589532
Test Error: 0.04617414248021108
```

Question 3

In [13]:

```
s = kernel(train, 5)
p = modified_perception(s)
```

In [14]:

```
if len(p) < 21**5:
    print("Less than 21**5")
elif len(p) == 21**5:
    print("Equal to 21**5")
else:
    print("More than 21**5")
```

```
Less than 21**5
```

In [15]:

```
two_max = sorted([(p[i], i) for i in p], reverse = True)[:2]
two_max
```

Out[15]:

```
[(3, 'WDTAG'), (3, 'LFLNK')]
```