Read the data from the hands_aligned_train.txt.new file

```
with open('./data/hands_aligned_train.txt.new') as newData:
    lines = [line.rstrip('\n') for line in newData]
    data = [np.fromstring(line, dtype=int, sep=' ') for line in lines[1:]]
# data = np.loadtxt('data/hands_aligned_train.txt.new', dtype=int, comments='#', delimiter=None)
```

Added the data in x

```
x = np.array(np.array(data))
```

taking x transpose

```
x = x.transpose()
#print(x)
```

Calculating the mean

```
x_mean = np.sum(x,axis=0)/x.shape[0]
#print(x_mean)
X = x[:] - x_mean
#print(X)
```

Calculating the WWT from slide 49

```
XXT =  X @ np.transpose(X)
#print(XXT)
```

$WW^T = U\Sigma V^T V\Sigma U^T = UL^2U^T$

```
u, e, v = np.linalg.svd(XXT)
print(e)
#e.sort()
eCollection = 0
```

Since we need to take the N which is the minimum number of principal components preserving 90% of the energy.

```
N = 1
for i in range (len(e)):
    eCollection += e[i]
    minimum_energy = eCollection / e.sum()
    if (minimum_energy > 0.9):
        N = i
        break
```

print("N------------",N)

#print(u.shape)

$$\mathbf{W}\mathbf{W}^T = \mathbf{U}\mathbf{L}^2\mathbf{U}^T$$

$$\hat{\sigma}^2 = \frac{1}{D-K}\sum_{j=K+1}^{D} L_{jj}^2$$

$$\hat{\boldsymbol{\Phi}} = \mathbf{U}_K(\mathbf{L}_K^2 - \hat{\sigma}^2\mathbf{I})^{1/2}$$

```
l2jj = e[N:].sum()
sigma_sq = l2jj / (e.shape[0] - N)
uk =  u[:, :N]
l2k = e[:N]
# print(sigma_sq)
# print(l2k)
phi = uk * np.sqrt(l2k-sigma_sq)
phi = uk * np.sqrt(l2k-sigma_sq)
#print(x_mean)
#print (phi)
hi = np.array([-0.4,-0.2,0.0,0.2,0.4])
#print(x_mean.shape)
#print(phi.shape)
```

Wi calculation

$$\mathbf{w}_i \approx \boldsymbol{\mu} + \sum_{k=1}^{K} \phi_k h_{ik}$$

```
wi = x_mean + (phi * hi.reshape(1,N)).sum(axis=1)
```

PLOT

```
fig = plt.figure()

ax = fig.add_subplot()

wi = np.transpose(wi.reshape(wi.shape[0] // 2, 2, order='F'))

ax.plot(wi[0,:],wi[1,:])

plt.show()
```