MACHINE LEARNING, Section A Final Exam

TOTAL POINTS

92 / 100

QUESTION 1

1 SVM 14 / 14

✓ **- 0 pts Correct**

- 5 pts 1a incorrect

- 4 pts 1b incorrect

- 5 pts 1c incorrect

QUESTION 2

2 Neural Networks 14 / 14

✓ **+ 14 pts Correct**

- 14 pts Incorrect

- 3 pts Missing Plot

- 3 pts Incomplete plot

- 3 pts Missing Code for part (c)

QUESTION 3

3 Backpropagation 14 / 14

✓ **- 0 pts Correct**

- 2 pts a) Missing summations

- 2.5 pts c) du/dz is not shown

- 2.5 pts c) du/dz is incorrect

- 2.5 pts du/dz is not always 0

QUESTION 4

4 CNN dimensions 8 / 14

- 0 pts Everything is correct

- 1 pts a) wrong answer for maxpool output

- 2 pts (a) no answer for conv1 input and wrong answer for maxpool output

✓ **- 3 pts (a) wrong answers for conv1 output, maxpool1 input and maxpool1 output**

- 4 pts (a) wrong answers for conv1 and maxpool1 dimensions

- 2 pts (b) wrong answer for conv1 params

- 2 pts (b) wrong answer for maxpool1 params

- 4 pts (b) wrong answers for conv1 and maxpool1 layers

- 6 pts (c) wrong answers for conv1 and maxpool1 layers

✓ **- 3 pts (c) wrong answers for maxpool1 layer**

(a) wrong answers for conv1 output, maxpool1 input and maxpool1 output (c) wrong answers for maxpool1 layer

QUESTION 5

5 Transfer learning 15 / 15

✓ **- 0 pts Correct**

- 3 pts Part a incorrect

- 1 pts Minor Code Issues

- 2 pts Click here to replace this description.

- 5 pts Click here to replace this description.

- 15 pts Did not submit

QUESTION 6

6 PCA 12 / 14

+ 4 pts (a) Correct

✓ **+ 2 pts Part of (a) wrong**

+ 0 pts (a) is wrong.

✓ **+ 5 pts (b) correct**

+ 3 pts Should use the value of question.

+ 2.5 pts Part of (b) wrong

+ 0 pts (b) is wrong

✓ **+ 5 pts (c) correct**

+ 4.5 pts PCs are orthonormal, or norm of basis should be one

+ 4 pts No v1

+ 3 pts Should use the value of question

+ 2.5 pts Part of (c) is wrong

+ 0 pts (c) is wrong

+ 4 pts Not minus, should be add.

QUESTION 7

7 K-means 15 / 15

- ✓ + 15 pts **Correct**
- - 15 pts Incorrect
- - 4 pts Part a incorrect
- - 6 pts Part b incorrect
- - 5 pts Part c incorrect
- - 2 pts Part a partial credit
- - 3 pts Part b partial credit
- - 2 pts Part c partial credit
- - 1 pts Part a partial credit
- - 1 pts Part b partial credit
- - 1 pts Part c partial credit
- - 3 pts Part a partial credit
- - 4 pts Part b partial credit
- - 4 pts Part c partial credit

1.

(a)

$$d_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix}^2 \qquad d_2 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^2 \qquad d_3 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 0.5 & 0.5 & -1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix}^2$$

$$\| \quad \| \quad \|$$

$$(0.5)^2+(-1)^2+(0.5)^2+(1)^2 \qquad 1^2+1^2+0.5^2+(-1)^2+(-0.5)^2 \qquad 1^2+1^2+0.5^2+0.5^2+(-1)^2+(-1)^2+(-1)^2+(-1)^2$$

$$\| \quad \| \quad \|$$

$$2.5 \qquad 4.5 \qquad 7.5$$

$$d = \begin{bmatrix} 2.5 \\ 4.5 \\ 7.5 \end{bmatrix}$$

(b)

for $y_i = 1$ $\quad K(x, x_i) > 0$

$$1 - d_i\gamma > 0$$

$$\begin{matrix} 1 - d_1\gamma > 0 \\ 1 - d_2\gamma > 0 \end{matrix} \Rightarrow \begin{cases} 1 - 2.5\gamma > 0 \\ 1 - 4.5\gamma > 0 \end{cases} \Rightarrow \gamma < \frac{1}{4.5} = \frac{2}{9} \qquad \gamma < \frac{2}{9}$$

for $y_i = -1$ $\quad K(x, x_i) \leq 0$

$$1 - d_3\gamma \leq 0$$

$$\gamma \geq \frac{2}{15} \qquad \gamma \geq \frac{2}{15}$$

$$\frac{2}{15} \leq \gamma < \frac{2}{9}$$

(c)

```
def predict(X,Xtr,ytr,b,alpha,gamma):
    D = np.sum((X[:,None,:] - Xtr[None,:,:])**2, axis = 2) * gamma
    K = np.maximum((0,1-D))
    z = K.dot(ytr * alpha) + b
    yhat = 2*(z > 0)-1
    return yhat
```

1 SVM **14 / 14**

✓ **- 0 pts** **Correct**

- 5 pts 1a incorrect

- 4 pts 1b incorrect

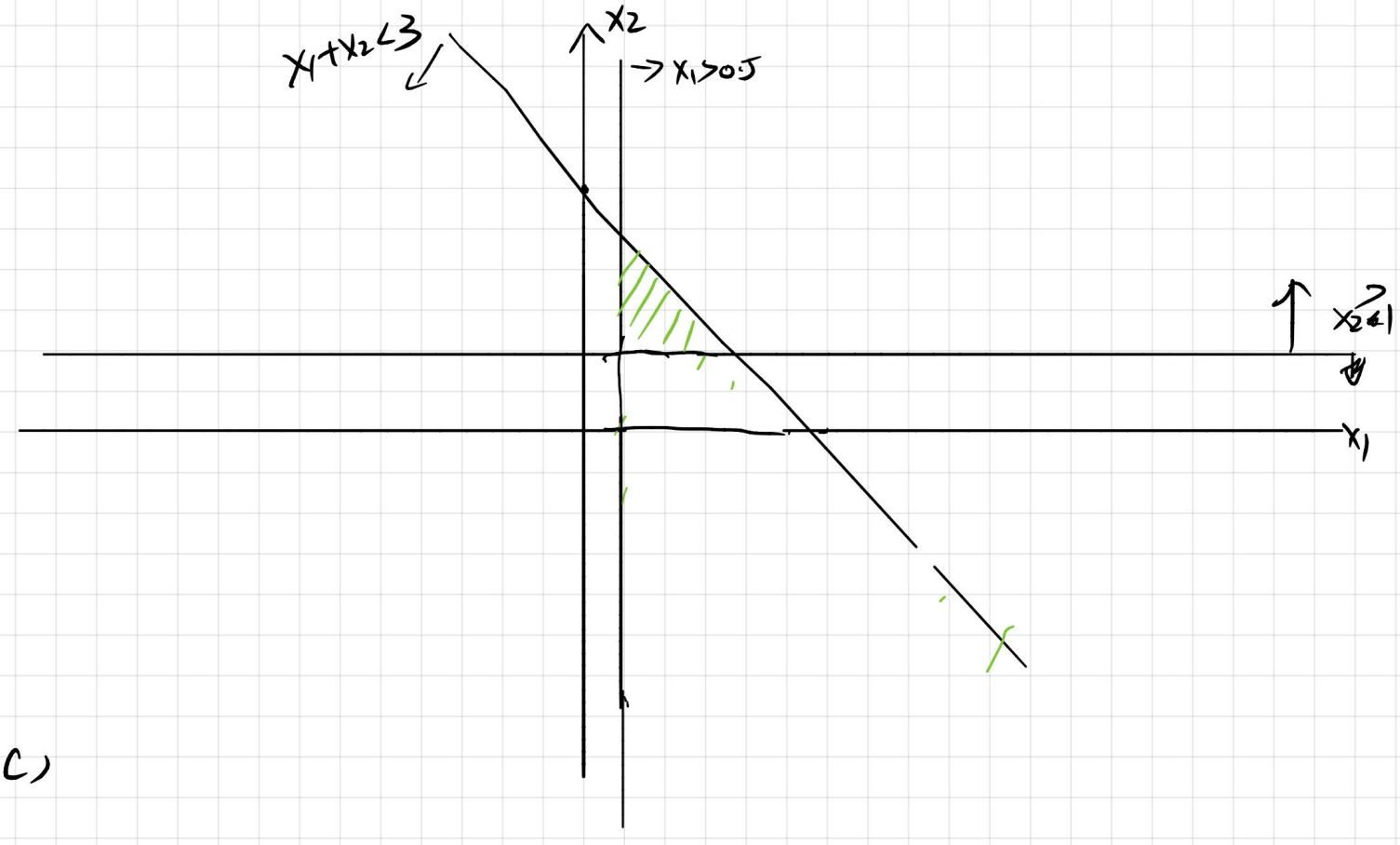
- 5 pts 1c incorrect

2.

(a) $z_j^H = \begin{bmatrix} x_{j1} - 0.5 \\ x_{j2} - 1 \\ -x_{j1} + x_{j2} + 3 \end{bmatrix}$

(b) input (x_1, x_2)

$$z^H = \begin{bmatrix} x_1 - 0.5 \\ x_2 - 1 \\ -x_1 - x_2 + 3 \end{bmatrix} \Rightarrow \text{if } u_j^H > 0 \begin{cases} x_1 - 0.5 > 0 \\ x_2 - 1 > 0 \\ -x - x_2 + 3 > 0 \end{cases} \Rightarrow \begin{array}{l} x_1 > 0.5 \\ x_2 > 1 \\ x_1 + x_2 < 3 \end{array}$$



(c)

```
zh = wh.dot(X) + bh[None,:]
zh[zh < 0] = 0
uh = np.maximum((0, zh))
beta = lstsq(uh,y)
b = beta[:,0]
w = [:,1:]
```

2 Neural Networks **14 / 14**

✓ **+ 14 pts Correct**

- 14 pts Incorrect

- 3 pts Missing Plot

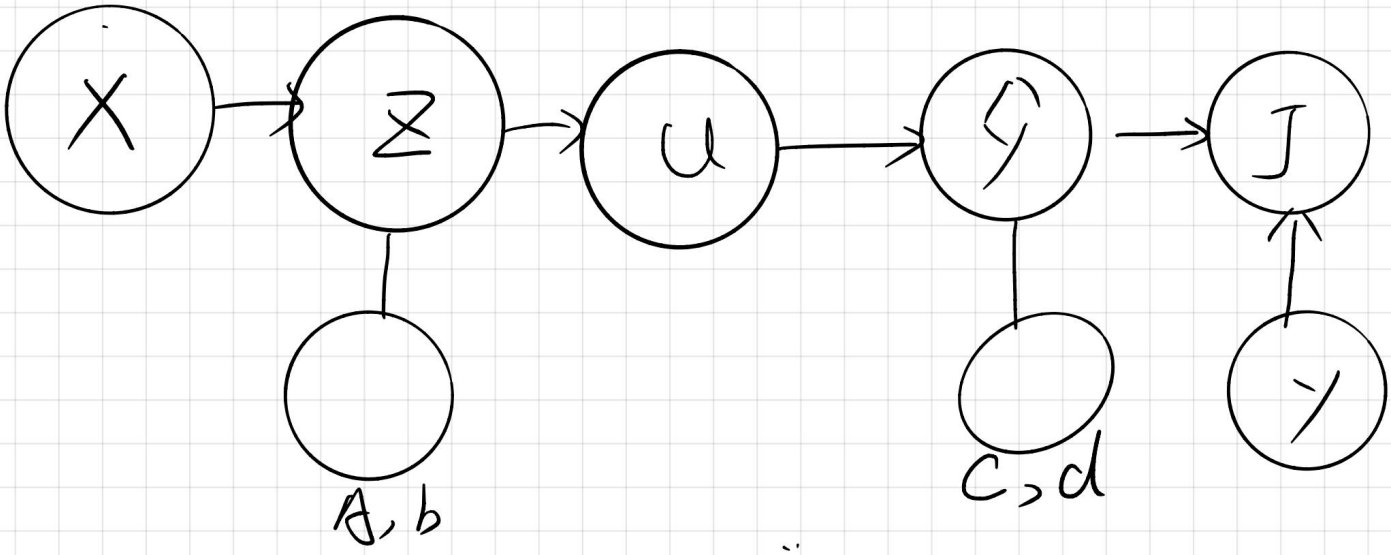
- 3 pts Incomplete plot

- 3 pts Missing Code for part (c)

3.

(a) $y_i = \sum_{k=1}^{K} \sum_{j=1}^{M} c_{kj} x_{ij} u_{ik} + \sum_{k=1}^{K} d_k u_{ik}$

(b)



(C)

$$z_{ik} = \sum_{j=1}^{M} A_{kj} x_{ij} + b_k \qquad u_{ik} = \frac{e^{z_{ik}}}{\sum_{l=1}^{K} e^{z_{il}}} \quad k = 1 \cdots K$$

$$\frac{\partial J}{\partial z_{ik}} = \frac{\partial J}{\partial u} \frac{\partial u_{ik}}{\partial z_{ik}}$$

$$\frac{\partial u_{ik}}{\partial z_{ik}} = \frac{e^{z_{ik}} \cdot \sum_{l=1}^{k} e^{z_{ik}} - e^{2z_{ik}}}{\left(\sum_{l=1}^{k} e^{z_{ik}}\right)^2}$$

$$\therefore \frac{\partial J}{\partial z_{ik}} = \frac{\partial J}{\partial u} \cdot \frac{\partial u_{ik}}{\partial z_{ik}} = \frac{\partial J}{\partial u} \cdot \frac{e^{z_{ik}} \cdot \sum_{l=1}^{k} e^{z_{ik}} - e^{2z_{ik}}}{\left(\sum_{l=1}^{k} e^{z_{ik}}\right)^2}$$

3 Backpropagation 14 / 14

✓ **- 0 pts** **Correct**

- 2 pts a) Missing summations

- 2.5 pts c) du/dz is not shown

- 2.5 pts c) du/dz is incorrect

- 2.5 pts du/dz is not always 0

4.

(a)

conv1 { input : $32 \times 128 \times 256 \times 3$

output : $32 \times 128 \times 256 \times 3$

maxpooling { input : $32 \times 128 \times 256 \times 3$

output : $32 \times 63 \times 127 \times 64$

(b)

conv1 $= 9 \times 9 \times 3 \times 64 + 64 = 15616$

maxpooling $= 0$

(c)

conv1 : $9 \times 9 \times 64 = 5184$

maxpooling : $6 \times 6 \times 64 = 2304$

4 CNN dimensions 8 / 14

- **0 pts** Everything is correct
- **1 pts** a) wrong answer for maxpool output
- **2 pts** (a) no answer for conv1 input and wrong answer for maxpool output

✓ **- 3 pts** **(a) wrong answers for conv1 output, maxpool1 input and maxpool1 output**

- **4 pts** (a) wrong answers for conv1 and maxpool1 dimensions
- **2 pts** (b) wrong answer for conv1 params
- **2 pts** (b) wrong answer for maxpool1 params
- **4 pts** (b) wrong answers for conv1 and maxpool1 layers
- **6 pts** (c) wrong answers for conv1 and maxpool1 layers

✓ **- 3 pts** **(c) wrong answers for maxpool1 layer**

(a) wrong answers for conv1 output, maxpool1 input and maxpool1 output (c) wrong answers for maxpool1 layer

5.

(a)

X (N, 5, 128, 128, 3)

y (N, 1)

(b) concat these 5×2000 output as an input for Xtr[i] to fit

```
def classify(X,pretrained):
    yhat = []
    for i in range(len(X)): #Traverse all samples
        index = 0
        for j in range(len(X[i])): #Traverse 5 pics in one sample X[i]
            Z = pretrained_base.predict(X[i][j])
            if(np.argmax(Z) == 49):
                index = 1
                break
        yhat.append[index]
```

(c)

```
def fit(Xtr,ytr):
    K = np.ones(len(Xtr),5*2000)
    for i in range(len(Xtr)):
        k = np.ones(5,2000)
        for j in range(5):
            z = pretrain.predict(Xtr[i][j])
            k[j] = z
        k.reshape(2000*5, 1)
        K[i] = k
    reg = LogisticRegression()
    reg.fit(K,ytr)

def predict(X):
    K = np.ones(len(X),5*2000)
    for i in range(len(X)):
        k = np.ones(5,2000)
        for j in range(5):
            z = pretrain.predict(X[i][j])
            k[j] = z
        k.reshape(2000*5, 1)
        K[i] = k
    reg.predict(K)
```

5 Transfer learning 15 / 15

✓ **- 0 pts Correct**

- 3 pts Part a incorrect

- 1 pts Minor Code Issues

- 2 pts Click here to replace this description.

- 5 pts Click here to replace this description.

- 15 pts Did not submit

6.

(a) U 100×3

V 3×200

(b) $X_{hat} = \sqrt{n}\, U \cdot \mathrm{diag}(s) \cdot V^T$

$$X_1 = 10\,[0.5, 0.3, 0.1] \begin{bmatrix} 10 & & \\ & 8 & \\ & & 2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \\ V_3^T \end{bmatrix} + \mu$$

$$X_1 = 50V_1^T + 24V_2^T + 2V_3^T + \mu$$

(c) $\hat{X}_1 - X_1 = \sum_{j=d+1}^{?} U[j:] \cdot S \cdot V^T = \| 24V_2^T + 2V_3^T \|^2$

$$V_i \cdot V_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = 1 \end{cases}$$

$$\| \hat{X} - X_i \|^2 = 24^2 + 2^2 = 580$$

6 PCA 12 / 14

+ **4 pts** (a) Correct

✓ **+ 2 pts** **Part of (a) wrong**

+ **0 pts** (a) is wrong.

✓ **+ 5 pts** **(b) correct**

+ **3 pts** Should use the value of question.

+ **2.5 pts** Part of (b) wrong

+ **0 pts** (b) is wrong

✓ **+ 5 pts** **(c) correct**

+ **4.5 pts** PCs are orthonormal, or norm of basis should be one

+ **4 pts** No v1

+ **3 pts** Should use the value of question

+ **2.5 pts** Part of (c) is wrong

+ **0 pts** (c) is wrong

+ **4 pts** Not minus, should be add.

7.

(a) because it's far from other centers

So $\|x-\mu_k\|^2$ is large So $e^{-\gamma\|x-\mu_k\|^2}$ is very small

So it can only have small influence on the result

we can just focus on the K we've near with

the equation becomes

$$\hat{y} \approx \frac{\bar{y}_k e^{-\gamma\|x-\mu_k\|^2}}{e^{-\gamma\|x-\mu_k\|^2}} \approx \bar{y}_k$$

$$\hat{y} \approx 2+4+5+6/4 = 4.75$$

(b)

```
def fit(Xtr,ytr,nc)
    yk = np.ones(nc)
    km = KMeans(n_cluster=nc)
    km.fit(X)
    pre = km.predict(X)
    centers = km.cluster centers_
    for i in range(nc):
        idx = np.where(pre == i)
        yk[i] = np.sum(ytr[idx])/len(idx)

    return centers, yk
```

(c)

```
def predict(X,centers, yk,gamma):
    dis = np.sum((X[:,None] - centers[None,:])**2, axis = 1)
    portion = np.sum(yk[None,:]*exp(-gamma*dis),axis = 1) #the upper
    sums = np.sum(exp(-gamma*dis),axis = 1)
    yhat = portion/sums
    return yhat
```

7 K-means 15 / 15

✓ **+ 15 pts Correct**

- 15 pts Incorrect

- 4 pts Part a incorrect

- 6 pts Part b incorrect

- 5 pts Part c incorrect

- 2 pts Part a partial credit

- 3 pts Part b partial credit

- 2 pts Part c partial credit

- 1 pts Part a partial credit

- 1 pts Part b partial credit

- 1 pts Part c partial credit

- 3 pts Part a partial credit

- 4 pts Part b partial credit

- 4 pts Part c partial credit