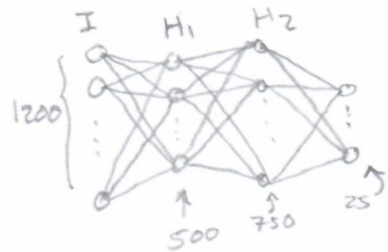
Name:

Student Number:

Graduate students must answer the bonus question.

1. (15 points) We have a densely connected neural network with the following specifications:

   - 1200 nodes in the input layer (I)
   - 25 nodes in the output layer (O)
   - 2 hidden layers
   - 500 nodes in hidden layer 1 (H-1)
   - 750 nodes in hidden layer 2 (H-2)



   Answer the following questions:

   (a) (2 points) What is the size of the feature space? Explain why. 1200. Input size = feature Space

   (b) (2 points) What is the number of parameters in the 1st hidden layer H-1 assuming no bias terms? (why).

   (c) (2 points) What is the number of parameters in the 2nd hidden layer H-2 assuming no bias terms? (why).

   (d) (2 points) What is the number of parameters in the output layer O assuming no bias terms? (why).

   (e) (2 points) What is the overall size of the network in number of parameters? (why).

   (f) (5 points) Write the equation of the network assuming that each layer (H-1, H-2, and O) utilizes the tanh activation functions.

a) 1200. Input size = No. of features

b) $W_{H_1} = W^1_{500\times1200}$ → No. Params $500\times1200 = 600{,}000$

c) $W_{H_2} \rightarrow W^2_{750\times500}$ → No. of Params = 375,000

d) $W_0 \rightarrow W^0_{25\times750} = 18{,}750$

e) Net Size = 600,000 + 375,000 + 18,750 = 453,750

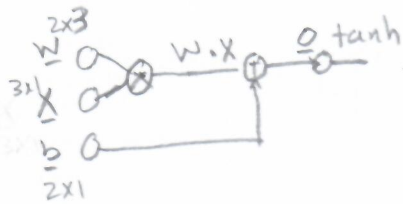
f) $\tanh(W^0 \cdot \tanh(W_2 \cdot \tanh(W_1 \cdot I)))$

2. (10 points) We have a perceptron, with one linear layer with the input size of 3×1 and one bias vector and output size of 2×1. Suppose the input is $\mathbf{x}$, the weights matrix is $\mathbf{w}$, the bias term is $\mathbf{b}$, and the activation map is $\tanh(\cdot)$.

   (a) (2 points) Write down the equation of this perceptron in terms of one matrix multiplication.

   (b) (4 points) Draw the computational graph (with only one matrix multiplication node), and for each edge write the size of the matrix representing the computation.

   (c) (4 points) Write down the full backprop equation for the derivative of the activation output with respect to weights+biases and inputs.

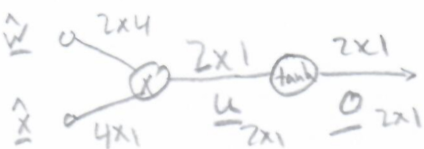


$$= \quad \underset{2\times 4}{\hat{w}} = (w \,|\, b) \qquad \hat{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}_{4\times 1}$$

$$o = \tanh(\hat{w} \cdot \hat{x}) \qquad \text{(b)}$$

$$\boxed{\tanh\Big(\underbrace{[\,w \,;\, b\,]}_{2\times 4} \cdot \begin{bmatrix} x \\ 1 \end{bmatrix}_{4\times 1}\Big)} \qquad \text{(a)}$$

b)



c) $$\frac{\partial o}{\partial o} = 1 \quad \rightarrow \quad \frac{\partial o}{\partial u} = 1 - \tanh^2(u)$$

$$\frac{\partial o}{\partial \hat{w}} = \frac{\partial o}{\partial u} \cdot \frac{\partial u}{\partial \hat{w}} \,, \quad \frac{\partial u}{\partial \hat{w}} = \frac{\partial}{\partial \hat{w}}(\hat{w} \cdot \hat{x}) = \hat{x}^T \quad \Rightarrow \quad \frac{\partial o}{\partial \hat{w}} = \underbrace{[1 - \tanh^2(u)]_{2\times 1} \cdot \hat{x}^T_{1\times 4}}_{2\times 4 \,\checkmark}$$

$$\frac{\partial o}{\partial \hat{x}} = \frac{\partial o}{\partial u} \cdot \frac{\partial u}{\partial \hat{x}} \,, \quad \frac{\partial u}{\partial \hat{x}} = \frac{\partial}{\partial \hat{x}}(\hat{w} \cdot \hat{x}) = \hat{w}^T \quad \Rightarrow \quad \frac{\partial o}{\partial \hat{x}} = \underbrace{\hat{w}^T_{4\times 2} \cdot [1 - \tanh^2(u)]_{2\times 1}}_{4\times 1 \,\checkmark}$$

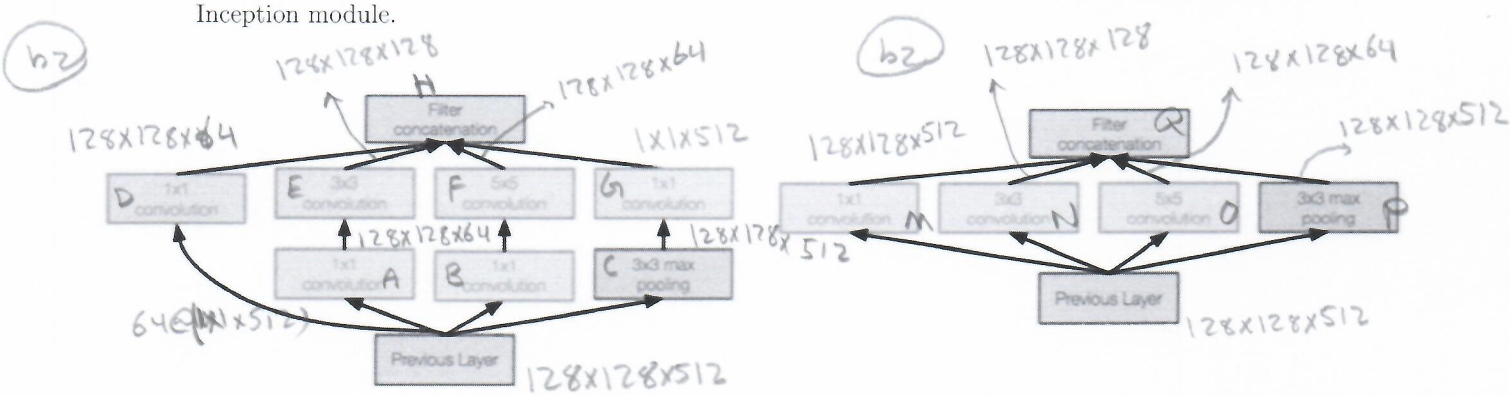
3. (15 points) We have designed a Convolutional Neural Network (CNN) to classify 10 classes, with the following specifications:

   - Input Layer $256 \times 256 \times 3$
   - Conv-1 Layer 128 filters of size 3×3, stride 1, padding 1.
   - Conv-2 Layer 64 filters of size 5×5, stride 2, padding 2. $\rightarrow m = \frac{N-f+2P}{S}+1 = \frac{\cancel{256-5+4}}{2} = 128$ $\quad 255 = 127+1 \quad 121$
   - Conv-3 Layer 128 filters of size 7×7, stride 2, no padding. $\rightarrow m = \frac{N-f+2P}{S}+1 = \frac{\cancel{128-7}}{2}+1 = 61$
   - Pool-1 Layer max-pooling of size 2×2, stride 2, no padding.
   - Conv-4 Layer 64 filters of size 7×7, stride 1, no padding. $\rightarrow m = \frac{N-f+2P}{S}+1 = \frac{30-7}{1}+1 = 30-6 = 24$
   - Conv-5 Layer 128 filters of size 7×7, stride 1, no padding. $\rightarrow m = 24-7+1 = 24-6 = 18$
   - FC-1 (fully connected) Layer of size $N \times 1$ same size as unwrapped previous layer.
   - FC-2 (fully connected) Layer of size $M \times 1$.

   Fill in the following table:

Layer	Input Size	Kernel Size	Output Size	No. of Params
Conv-1	256 × 256 × 3	3 × 3 × 3	256 × 256 × 128	3×3×3×128 = 3,456
Conv-2	256 × 256 × 128	5 × 5 × 128	128 × 128 × 64	5×5×128×64 = 204,800
Conv-3	128 × 128 × 64	7 × 7 × 64	61 × 61 × 128	7×7×64×128 = 401,408
Pool-1	61 × 61 × 128	~~× ×~~	30 × 30 × 128	
Conv-4	30 × 30 × 128	7 × 7 × 128	24 × 24 × 64	7×7×128×64 = 401,408
Conv-5	24 × 24 × 64	7 × 7 × 64	18 × 18 × 128	7×7×64×128 = 401,408
FC-1	41,472×1	NA	41,472×1	$(41,472)^2$ = 1,719,926,784
FC-2	41,472×1	NA	10×1	41,472×10 = 414,720

4. (25 points) Consider the architecture of GoogLeNet and answer the following questions:
   (a) (5 points) Describe the Inception module of the GoogLeNet shown in Figure 1(a).
   (b) (10 points) Assume the input from the previous layer of the inception module in Figure 1(a) is $128 \times 128 \times 512$, the depth of each of the bottom 1×1 conv-modules is 64, the depth of the top 1×1 module is 512, the depth of the 3×3 convolution is 128, and the depth of 5×5 convolution is 64.
      1. Calculate the padding used for the 3×3 and 5×5 convolutions to keep outputs the same as the inputs.
      2. Calculate the output size of each module and show on graph.
      3. Calculate the number of operations in each module.
      4. Calculate and number of parameters in each module.
      5. Calculate the total number of operations of the whole module.
      6. Calculate the total number of parameters of the whole module.
   (c) (5 points) Calculate the number of parameters and number of operations in the naive inception module shown in Figure 1(b). Discuss why the GoogLeNet module is more efficient than the naive module.
   (d) (5 points) Discuss how the Inception module differs in terms of the number of parameters from the naive Inception module.



(a)

(b)

a) See slides. (b1) for 3x3 ~> P=1 , for 5x5 ~> P=2

(b3,5) Inception OPS from figure labels:

A = 128x128x64x512 = 536,870,912
B = 128x128x64x512 = 536,870,912
C = 128x128x512x3x3 = 75,497,472 ~> Negligible
D = 128x128x64x512 = 536,870x912
E = 128x128x128x3x3x64 = 1,207,959,552
F = 128x128x64x5x5x64 = 3,355,443,200
G = 128x128x512x512 = 4,294,967,296

H = 10,544,480,256

(c)

ops
M = 128x128x64x512 = 536,870,912
N = 128x128x128x3x3x512 = 9,663,676,416
O = 128x128x64x5x5x512 = 26,843,545,600
P = 128x128x512x3x3 = 75,497,472

Q = 26,843,545,600
+ 9,663,676,416
+ 536,870,912 + 75,497,472

37,119,590,400

(b4,6) Params:
A = 64x1x1x512 = B, C = 0
D = 64x1x1x512
E = 3x3x64x128
F = 5x5x64x64
G = 1x1x512x512
H = 569,344

M = 512x1x1x512
N = 128x3x3x512
O = 64x5x5x512
P = 0
Q = total
Q = 1,671,168