VIR 2018 **Name:** ____________________

Exam 1

Time Limit: Teaching Assistant ____________________

1. Consider the following network.

$$y = sin(w_1 \cdot x_1 + w_2 \cdot x_2) - b \tag{1}$$

- Draw the computational graph of the forward pass of this network. Note that **every** operator is a node with a given arity and output. For example, the + operator is a node which has two input arguments and a single output argument, etc...

- Consider an input $x_1 = 2, x_2 = 1, w_1 = \frac{\pi}{2}, w_2 = \pi, b = 0$ and label $l = 2$.
  - Compute the forward pass of the network.

  - Use an L_2 loss (Mean square error) to compute the loss value between the forward prediction y and label l. Add this loss to the computation graph.

  - Use the chain rule to compute the gradient $\frac{\partial L(y,l)}{\partial \mathbf{w_1}}$.

2. Consider a convolutional neural network layer l_1 which maps an RGB image of size 128×128 to 16 feature maps having the same spacial dimensions as the image. The kernel size is 3×3 and uses a stride of 1.

   - How much memory (in bytes) do kernel weights in layer l_1 take up, assuming *float32* weights? Ignore the *bias* weights.

   - How many mathematical operations does this layer perform for a single forward pass. A multiplication or addition of two numbers can be considered as one operation.

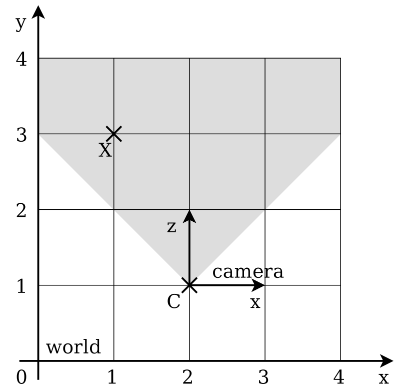
   - Name one way that we can regularize (prevent overfitting) a large parametric model (for example a neural network).

   - Define a Rectified Linear Unit $\mathbf{relu}(\mathbf{x})$ activation function in pseudocode. The function has a single argument $\mathbf{x}$ and outputs $\mathbf{relu}(\mathbf{x})$.

   - Define the gradient of the $\mathbf{relu}(\mathbf{x})$ activation function in pseudocode. The function has a single argument $\mathbf{x}$ and outputs $\frac{\partial relu(x)}{\partial \mathbf{x}}$. Hint: Break up the function into two separate cases (if-else).

   - Name an technique that improves the basic gradient descent update step which helps prevent getting stuck in shallow local minima and allows better training of deeper models.

3. Consider a perspective camera with center of projection **C** and point **X** from the following illustration:



(Field of view of the camera is outlined in gray.)

Assume we are given matrix of intrinsic camera parameters **K**, camera rotation matrix **R**, and translation vector **t**:

$$\mathbf{K} = \begin{bmatrix} 500 & 0 & 500 \\ 0 & 500 & 250 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \mathbf{t} = \begin{bmatrix} -2 \\ 0 \\ -1 \end{bmatrix},$$

where $\mathbf{X}_c = \mathbf{R}\mathbf{X}_w + \mathbf{t}$ converts world coordinates $\mathbf{X}_w$ to camera coordinates $\mathbf{X}_c$.

a) Construct camera projection matrix $\mathbf{P} \in \mathbb{R}^{3\times 4}$:

$\mathbf{P} =$

b) Project point **X** into the camera ($\bar{\mathbf{X}}_w$ denotes homogeneous coordinates):

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{P}\bar{\mathbf{X}}_w = \mathbf{P} \begin{bmatrix} 1 \\ 3 \\ 0 \\ 1 \end{bmatrix} =$$

c) What are pixel coordinates x, y of the projection?

$x =$

$y =$