VIR 2019 **Name:** ______________________

Exam test

Variant: A Points ______________________

1. Consider the following network.

$$y = \sin(\mathbf{w}^\top \mathbf{x}) - b \tag{1}$$

- 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 $\mathbf{x} = [2, 1]\mathbf{w} = [\frac{\pi}{2}, \pi], b = 0$ 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 $\frac{\partial L(y,l)}{\partial \mathbf{w}}$ and estimate an update of parameters $\mathbf{w}$ with learning rate $\alpha = 0.5$.

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 spatial dimensions as the image. The kernel size is 3×3 and uses a stride 1.

   - What is the size of padding, which ensure the same spatial resolution of the output feature map?

   - 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).

3. Activation function maps single input $\mathbf{x}$ on a single output value $\mathbf{y}$.

   - Define a Leaky Rectified Linear Unit $\mathbf{lrelu}(\mathbf{x})$ activation function in pseudocode, with $\alpha = 0.1$. The function has a single argument $\mathbf{x}$ and output $\mathbf{y} = \mathbf{lrelu}(\mathbf{x})$.

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

4. You are given batch of three one-dimensional training examples $x_1 = 5, x_2 = 2, x_3 = 1$.

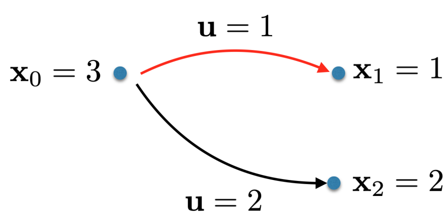
   - Compute output of the batch-norm layer with learnable parameters $\gamma = 6$, $\beta = -1$.

   - Compute gradient of the batch-norm layer with respect to the parameter β.
     Hint: output of the batch-norm layer for this batch is three-dimensional.

5. Consider MDP consisting of three states $\mathbf{x}_0 = 3, \mathbf{x}_1 = 1, \mathbf{x}_2 = 2$ and two types of actions $\mathbf{u} = 1$ and $\mathbf{u} = 2$, see image below. Agent selects action $\mathbf{u}$ in the state $\mathbf{x}$ according the following stochastic policy

$$\pi_\theta(\mathbf{u}|\mathbf{x}) = \begin{cases} \sigma(\theta\mathbf{x}) & \text{if } \mathbf{u} = 1 \\ 1 - \sigma(\theta\mathbf{x}) & \text{if } \mathbf{u} = 2 \end{cases}$$

with scalar parameter $\theta = 2$. This policy maps one-dimensional state $\mathbf{x}$ on the probability distribution of two possible actions $\mathbf{u} = 1$ or $\mathbf{u} = 2$.



Consider trajectory-reward function defined as follows:

$$r(\tau) = \sum_{\mathbf{x}_i \in \tau} \frac{1}{\mathbf{x}_i}$$

Given training trajectory $\tau = [(\mathbf{x}_0 = 3), (\mathbf{u} = 1), (\mathbf{x}_1 = 1)]$, which consists of the single transition (outlined by red color), estimate:

- Policy gradient

$$\frac{\partial \log \pi_\theta(\mathbf{u}|\mathbf{x})}{\partial \theta}\Big|_{\substack{\mathbf{x} = \mathbf{x}_0 \\ \mathbf{u} = \mathbf{u}_0}} \cdot r(\tau) =$$

- Updated weights with learning rate $\alpha = 1$

$\theta^{\text{updated}} =$