VIR 2021 **Name:** ______________________

Midterm test

Variant: B Points ______________________

---

1. **MLE loss**: You are given Laplace probability distribution model

$$p(y|x, w) = \frac{1}{2} \exp(-|wx - b - y|),$$

which models probability of variable $y \in \mathbb{R}^+$, given measurement $x \in \mathbb{R}$ and unknown model parameters $w, b \in \mathbb{R}$. You are given a training set $\mathcal{D} = \{(x_1, y_1) \dots (x_N, y_N)\}$. Write down the optimization problem, which corresponds to the maximum likelihood estimate of the model parameters w? Simplify resulting optimization problem if possible.

2. **Backprop and computational graph:** Consider the following network

$$y = \Big(\sin(\mathbf{w})\Big)^\top \mathbf{x},$$

where $\sin(\mathbf{w})$ denotes element-wise function $[\sin(\mathbf{w}_1), \sin(\mathbf{w}_2)]^\top$. Consider an input $\mathbf{x} = [2, 1]\mathbf{w} = [\pi/2, \pi]$ and label $l = 1$.

- Draw the computational graph of the forward pass of this network. Preserve vectorized form, so that the vector $\mathbf{w}$ correspond to a single edge in the graph.

- Compute the forward pass of the network in the computational graph.

- Use logistic loss $\mathcal{L}(y, l) = ...$ to compute the loss value between the forward prediction y and label l. Add this loss to the computational graph.

- Populate the computation graph by local gradients and use the chain rule to compute the gradient $\frac{\partial \mathcal{L}(y,l)}{\partial \mathbf{w}}$ and estimate an update of parameters $\mathbf{w}$ with learning rate $\alpha = 0.5$.

3. **Convolution feedforward pass:** You are given an input volume X of dimension $[batch \times channel \times width \times height] = [4 \times 2 \times 13 \times 13]$
Consider a one 2D convolutional filter F of size $[width \times height] = [5 \times 5]$

- Assuming a stride of 2, What is the size of padding, which ensures that the feature map is size 7x7 of the input map? Note: A padding size of 1 for a $[30 \times 30]$ image gives it a resulting size of $[32 \times 32]$, in other words, zeros are added on both sides.

- Calculate the total memory in bytes of the learnable parameters of the filter, assuming that each weight is a dual-precision float (FP64).

- Calculate the amount of operations performed by a single application of the filter (just one "stamp"). Each addition or multiplication counts as a single operation.
  For example: $\alpha x + \beta y + c$ amounts to 2 multiplication and 2 addition operations, totaling 4 operations.

- Considering the entire input dimensions of X, given a stride of 2, no padding, calculate the amount of filter applications ("stamps") that you have to perform to process the entire input.

4. **Convolution in SGD:** You are given convolutional network $y = f(\mathbf{x}, \mathbf{w}, \mathbf{v})$ consisting of two layers

   1. convolutional layer with one 3×3 kernel (stride 1, padding=0), with weights denoted $\mathbf{w}$ (bias is completely ignored for simplicity)
   2. convolutional layer with one 3×3 kernel (stride 1, padding=0), with weights denoted $\mathbf{v}$ (bias is completely ignored for simplicity)

   - What is the dimensionality of input $\mathbf{x}$ if output y is a scalar value?

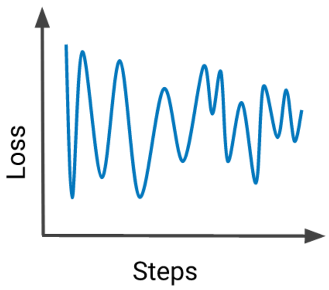
   - Let us assume that you initialized values of all kernels by *zeros*, such that $\mathbf{w} = \mathbf{0}$ and $\mathbf{v} = \mathbf{0}$. You are given a training set consisting of pairs of real-valued, finite inputs $\mathbf{x}_i$ and corresponding real-valued scalar outputs y_i. You trained the network on the training set to minimize L_2-loss using Stochastic Gradient Descent (SGD). What relations (if any) will hold among trained weights after the training (assuming that the SGD converged to a finite values)? Proof your claims if possible.
     Hint: Recall that, if you have convolutional layer $z = \text{conv}(\mathbf{x}, \mathbf{w})$ followed by another layer $u = p(z)$, than gradient $\frac{\partial p(\text{conv}(\mathbf{x},\mathbf{w}))}{\partial \mathbf{w}} = \text{conv}(\mathbf{x}, \frac{\partial p(\mathbf{z})}{\partial \mathbf{z}})$, where $\frac{\partial p(\mathbf{z})}{\partial \mathbf{z}}$ denotes the upstream gradient. Similarly $\frac{\partial p(\text{conv}(\mathbf{x},\mathbf{w}))}{\partial \mathbf{x}} = \text{conv}(\frac{\partial p(\mathbf{z})}{\partial \mathbf{z}}, \mathbf{w})$ with padding corresponding to a desired gridient size. Look at backprop in computational graph.

- Let us assume, that we create new network $g(\mathbf{x}, \mathbf{w}_1, \dots \mathbf{w}_4, \mathbf{v}_1, \dots \mathbf{v}_4)$ by introducing additional kernels into convolutional layers of $f(\mathbf{x}, \mathbf{w}, \mathbf{v})$. What kind of function is g? Can you replace g by a simplier function $\hat{g}(\mathbf{x}, \theta)$, that preserves *expressing power* of g: i.e. given any parameters $\mathbf{w}_1, \dots \mathbf{w}_4, \mathbf{v}_1, \dots \mathbf{v}_4$, there exists lower dimensional parameter θ such that
$$g(\mathbf{x}, \mathbf{w}_i, \mathbf{v}_j) = \hat{g}(\mathbf{x}, \theta)$$
for any possible $\mathbf{x}$. What lowest possible dimensionality of θ?

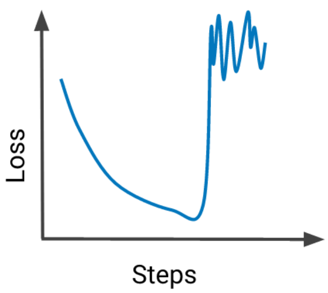
5. **Training:** You are given a following figures of loss function over the training iterations.

- Explain for each of the figures, what might be happening to the model during the training based on these curves
- Propose at least a one way how to solve each of these issues.

1)



2)



3)