1. **Multiple choice (Shreyas)**
   Please pick the correct answers for each questions, note that each question can have one or more than one correct.

   (a) Consider Figure 1 plotting loss values as a function of the number of epochs, select the option that best describe the shaded regions in the plot, and the point where you would stop training to achieve the best generalization.

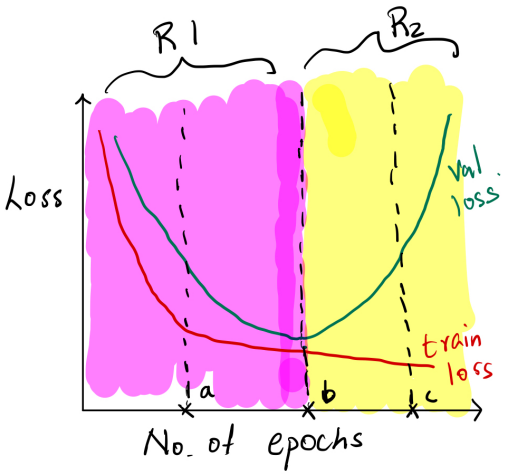

Figure 1

   i. R1: Overfitting, R2: Underfitting, stop at a.
   ii. R1: Overfitting, R2: Underfitting, stop at b.
   iii. R1: Underfitting, R2: Overfitting, stop at b.
   iv. R1: Underfitting, R2: Overfitting, stop at c.

   (b) When we minimize the negative log likelihood for a classification problem with c classes, which of the following are we inherently performing?

   i. Maximizing the likelihood of observing the training data.
   ii. Minimizing the Mean Squared Error.
   iii. Minimizing the Cross Entropy loss.

   (c) Mark all the correct choices regarding cross validation.

   i. A 5-fold cross-validation approach results in 5-different model instances being fitted.

ii. A 5-fold cross-validation approach results in 1 model instance being fitted over and over again 5 times.

iii. A 5-fold cross-validation approach results in 5-different model instances being fitted over and over again 5 times.

iv. None of the above.

(d) Which of the following are considered as hyperparameter choices while training a neural network.

i. Loss Function.

ii. Learning Rate.

iii. Number of Layers.

iv. Batch Size.

v. All of the above.

(e) Assuming Stochastic Gradient Descent (SGD) computes gradient using a single sample from the training data, which of the following statements are true.

i. Gradient computed using SGD will be noisier than gradient computed using Batch Gradient Descent.

ii. Empirically, SGD takes longer (in terms of clock time) to converge than Batch Gradient Descent.

iii. SGD usually avoids the trap of poor local minima.

iv. SGD is computationally more expensive than Batch Gradient Descent.

2. **Short answer (Kaifeng)**

(a) Please explain the difference between batchnormalization during training and testing.

(b) Your friend designed a novel activation function:

$$f(x) = x^3 \tag{1}$$

Please discuss if this is a good idea to use this activation in a neural network.

(c) Your friend is utilizing a Multi-layer Perceptron (MLP) for a deep learning task and is trying to increase the number of units within each layer to enhance the model's complexity. Please explain potential effect of this action on the model performance.

(d) Please explain the role of ℓ_1 regularization.

(e) Please explain the role of the bias correction step in the Adam optimizer.

3. **Backpropagation in parallel neural network (Tonmoy)**

A parallel neural network consists of twin networks which accept distinct inputs but share the same weights. The outputs of the twin networks are later processed by more hidden layers. Let's assume we have a parallel neural network with the following architecture:

$$
\begin{aligned}
h_p &= W_1 x_p^{(i)} + b_1 \\
z_1 &= ReLU(h_p) \\
h_q &= W_1 x_q^{(i)} + b_1 \\
z_2 &= ReLU(h_q) \\
z &= z_1 - z_2 \\
z_3 &= W_2 z + b_2 \\
\hat{y}^{(i)} &= \sigma(z_3) \\
L^{(i)} &= L_{CE}(y^{(i)}, \hat{y}^{(i)}) \\
L &= -\frac{1}{m} \sum_{i=1}^{m} L^{(i)}
\end{aligned}
$$

In the above architecture, ($x_p^{(i)}$,$x_q^{(i)}$) represent the pair of i^{th} input example and are each of shape D_x. $y^{(i)}$ represent the label of the i^{th} input example and is a scalar. We also assume z_1 and z_2 have shape of D_z.

(a) Draw the computational graph for the parallel neural network described above. You can start from $L^{(i)}$ as your output variable and then backtrack to the input variables $x_p^{(i)}$ and $x_q^{(i)}$.

(b) Compute $\nabla_{\hat{y}^{(i)}} L^{(i)}$ and denote it as $\delta_{\hat{y}^{(i)}}$. For all the following parts, you can refer to this computed gradient as $\delta_{\hat{y}^{(i)}}$.

(c) Compute $\nabla_{z_3} L^{(i)}$ and denote it as δ_{z_3}. For all the following parts, you can refer to this computed gradient as δ_{z_3}.

(d) Compute $\nabla_{b_2} L^{(i)}$ and denote it as δ_{b_2}. For all the following parts, you can refer to this computed gradient as δ_{b_2}.

(e) Compute $\nabla_{W_2} L^{(i)}$ and denote it as δ_{W_2}. For all the following parts, you can refer to this computed gradient as δ_{W_2}.

(f) Compute $\nabla_z L^{(i)}$ and denote it as δ_z. For all the following parts, you can refer to this computed gradient as δ_z.

(g) Compute $\nabla_{z_1} L^{(i)}$ and denote it as δ_{z_1}. For all the following parts, you can refer to this computed gradient as δ_{z_1}.

(h) Compute $\nabla_{z_2} L^{(i)}$ and denote it as δ_{z_2}. For all the following parts, you can refer to this computed gradient as δ_{z_2}.

(i) Compute $\nabla_{h_q} L^{(i)}$ and denote it as δ_{h_q}. For all the following parts, you can refer to this computed gradient as δ_{h_q}.

(j) Compute $\nabla_{h_p} L^{(i)}$ and denote it as δ_{h_p}. For all the following parts, you can refer to this computed gradient as δ_{h_p}.

(k) Compute $\nabla_{b_1} L^{(i)}$.

(l) Compute $\nabla_{W_1} L^{(i)}$.

4. **Regularization techniques (Yang)**

   (a) True or False: Regularization is intended to reduce training error but not validation error.

   (b) Consider a model $\tilde{\mathcal{L}}(\theta) = \mathcal{L}(\theta; \mathbf{X}, \mathbf{y}) + \alpha\Omega(\theta)$ where $\mathcal{L}(\theta; \mathbf{X}, \mathbf{y})$ is some loss function and $\Omega(\theta)$ is some norm penalty. What are the effects on the model when $\alpha = 0$ and $\alpha \to \infty$?

   (c) Mathematically show that ℓ_2 regularization shrinks the weight in gradient descent. Hint: start with $\tilde{\mathcal{L}}(\theta; \mathbf{X}, \mathbf{y}) = \mathcal{L}(\theta; \mathbf{X}, \mathbf{y}) + \frac{\alpha}{2}||\theta||_2^2$ and derive the gradient descent step for θ.

   (d) List two dataset augmentation techniques for image classification.

   (e) How did you implement dropout in homework 4? Please comment on both training and testing.

5. **Optimization techniques (Lahari)**

   (a) In lecture, we have learnt about Nesterov Momentum and it's update rule for parameters. The update rule for parameter θ is given by:

$$\begin{aligned} v_t &= \alpha v_{t-1} - \epsilon \nabla_\theta L(\theta_{t-1} + \alpha v_{t-1}) \\ \theta_t &= \theta_{t-1} + v_t \end{aligned} \tag{2}$$

   Prove that the update rule in (3) is equivalent to the update rule in (2)

$$\begin{aligned} v_{new} &= \alpha v_{old} - \epsilon \nabla_{\widetilde{\theta}_{old}} L(\widetilde{\theta}_{old}) \\ \widetilde{\theta}_{new} &= \widetilde{\theta}_{old} + v_{new} + \alpha(v_{new} - v_{old}) \end{aligned} \tag{3}$$

   Explain one advantage of using the update update rule in (3) over the update rule in (2).

   (b) Consider the two loss curves $L_1(x)$ and $L_2(x)$ shown in Figure 2. Which loss curve has a saddle point? Which loss curve has a poor local minima? In which of the loss curves, is an optimizer more likely to escape the trap of a saddle point or a poor local minima? And what property does the optimizer require for it to escape these traps in this case?

   (c) Consider the contour plot shown in Figure 3 , where the loss surface is plotted with respect to just 2 weights w1 and w2, where $w1, w2 \in R$ (scalars). Assume you are given a hypothetical scenario, where you start from point A and use vanilla gradient descent in many iterations to get to point B. During this process, we have started accumulating momentum based on the following equation,

$$\begin{aligned} g_t &= \nabla_{\theta_t} L(\theta_t) \\ v_t &= v_{t-1} - \epsilon g_t \end{aligned}$$

   Comment on which direction does the weight update occur if we use the following optimizers : vanilla gradient descent, gradient descent with momentum, gradient descent with Nesterov momentum, Adagrad.

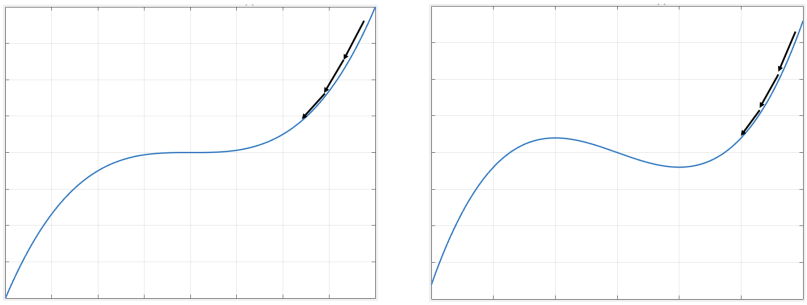

Figure 2: Loss curves $L_1(x)$ (Left), $L_2(x)$ (Right)

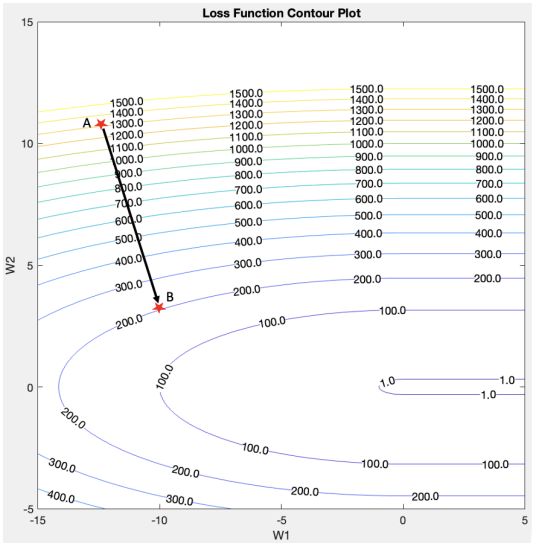

Figure 3: Contour plot of a Loss function $L(w1, w2)$

(d) In the Gradient descent + momentum scheme, find a general expression of v_t in terms of gradients $g_1, g_2, ..., g_t$ and ϵ(learning rate), considering an initial value of momentum $v_0 = 0$.

(e) Consider that the gradients $g_1, g_2, ..., g_t$ in part (d) are i.i.d. random variables with mean μ and variance σ. Find the expected value of weights θ_t at t = 3.

6. ℓ_∞ **regularization (Tonmoy)**

Let $x \in R^n$, then we define the ℓ_∞ norm and the Log-Sum-Exponent (LSE) of it as follows:

$$\|x\|_\infty = \max_i \ |x_i|$$

$$LSE(x) = \ln\left(\sum_{i=1}^{n} e^{x_i}\right)$$

(a) Show that the following inequality holds for $n \geq 1$,

$$\|x\|_\infty \leq LSE(x) \leq \|x\|_\infty + \ln(n) \tag{4}$$

(b) Is the lower bound in (4) strict for $n > 1$?

(c) Under what condition on x, will the upper bound in (4) be satisfied with equality.

(d) Use the result from (4) to show that the following inequality holds,

$$\|x\|_\infty \leq \frac{1}{t} LSE(tx) \leq \|x\|_\infty + \frac{\ln(n)}{t} \tag{5}$$

for some scaling constant $t > 0$.