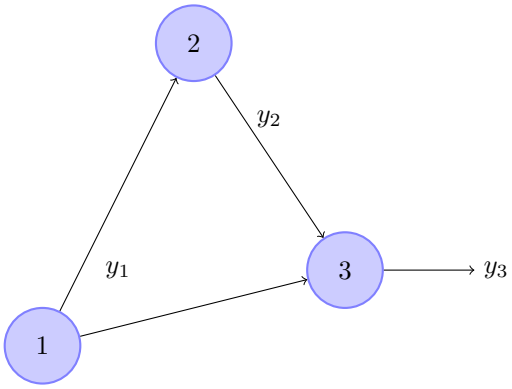
CS6910: Tutorial 3

VERSION I

1. Show that for a binary classification problem, minimising the cross entropy loss is the same as minimising the KL divergence between the true and predicted distributions.

2. You want your neural network based classification model to be highly confident in addition to being accurate. One way of achieving this is to ensure that the probability predicted for the correct class y_i should be larger than the probabilities predicted for the other classes by a significant margin Δ (say, ≥ 0.3). How would you design a loss function to ensure this? For example, if we have 3 classes and if the correct class label is 0, and the probabilities predicted by the model are $[y_0 = 0.58, y_1 = 0.37, y_2 = 0.05]$ then the model should incur: (i) no loss for the correct class, (ii) a loss for assigning a probability of 0.37 for the 2nd class since the difference in the probabilities for the correct class and incorrect class is just 0.21 (that is lesser than Δ), (iii) no loss for assigning a probability of 0.05 for the 3rd class since the difference in the probability is greater than Δ.
   Loss function = _____

3. An ordered network is a network where the state variables can be computed one at a time in a specified order. Given the ordered network below, give a formula for calculating the ordered derivative $\frac{\partial y_3}{\partial y_1}$ in terms of partial derivatives w.r.t. y_1 and y_2 where y_1, y_2 and y_3 are the outputs of nodes 1, 2 and 3 respectively.



(a)
$$\frac{dy_3}{dy_1} = \frac{\partial y_3}{\partial y_2}\frac{dy_2}{dy_1} + \frac{\partial y_3}{\partial y_1}$$

(b)
$$\frac{dy_3}{dy_1} = \frac{\partial y_3}{\partial y_2}\frac{dy_2}{dy_1}\frac{\partial y_3}{\partial y_1}$$

(c)
$$\frac{dy_3}{dy_1} = \frac{\partial y_3}{\partial y_2}\frac{dy_2}{dy_1} - \frac{\partial y_3}{\partial y_1}$$

(d)
None of the above.

4. Let $\phi_1(.)$ and $\phi_2(.)$ denote the sigmoid and the tanh functions respectively. Tick the correct options.

(a) $\phi_1(\text{-}\nu) = \phi_1(\nu)$ and $\phi_2(\text{-}\nu) = 1$ - $\phi_2(\nu)$
(b) $\phi_1(\text{-}\nu) = \text{-}\phi_1(\nu)$ and $\phi_2(\text{-}\nu) = 1$ - $\phi_2(\nu)$
(c) $\phi_1(\text{-}\nu) = 1$ - $\phi_1(\nu)$ and $\phi_2(\text{-}\nu) = \text{-}\phi_2(\nu)$
(d) None of the above.

5. Consider vectors $\mathbf{u}, \mathbf{x} \in R^d$, and matrix $\mathbf{A} \in R^{n \times n}$. The derivative of a scalar f w.r.t. a vector $\mathbf{x}$ is a vector by itself, given by

$$\nabla_{\mathbf{x}} f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

Derive the expressions for the following derivatives (gradients).

$$\nabla_{\mathbf{x}} \mathbf{u^T x} \quad , \quad \nabla_{\mathbf{x}} \mathbf{x^T x} \quad \text{and} \quad \nabla_{\mathbf{x}} \mathbf{x^T A x}$$

(a) u^T, x^T and Ax^T
(b) u^T, $2x^T$ and $2Ax^T$
(c) u, $2x$ and $2Ax$
(d) u, $2x$ and Ax

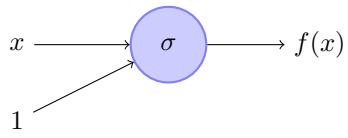
6. A fair coin results in either Head (1) or Tail (0) with equal probability. What is the entropy of the random variable indicating the outcome of the toss? If instead, we had a biased coin with P(H) = 0.7, does the entropy increase or decrease?

(a) With fair coin, entropy is 1. With biased coin, it is 0.88.
(b) With fair coin, entropy is 0.88. With biased coin, it is 1.
(c) With fair coin, entropy is 0.88. With biased coin, it is 0.66.
(d) With fair coin, entropy is 0.25. With biased coin, it is 0.90.

7. Recall the gradient descent update rule that comes from the Taylor series when at each step we ensure $\mathrm{L}(\theta_{K+1}) < \mathrm{L}(\theta_K)$ where L is the loss function. Suppose we are dealing with a quadratic loss function. Can you come up with a better update rule such that we reach the minima quickly.
Bonus question: Think about why this is not a widely used update rule even when this is much faster then gradient descent.

8. Consider a fully connected network with 3 inputs x_1, x_2, x_3. Suppose there are two hidden layers with 4 neurons having sigmoid activation functions. Further, the output layer is a softmax layer. Assume that all the weights in the network are set to 1 and all biases are set to 0. Write down the output of the network as a function of $\mathbf{x} = [x_1, x_2, x_3]$.
y = _____

9. Consider the following computation,



$$f(x) = tanh(w \cdot x + b)$$

The value L is given by,

$$L = \frac{1}{2}(y - f(x))^4$$

Here, x and y are constants and w and b are parameters that can be modified. In other words, L is a function of w and b.
Derive the partial derivatives, $\frac{\partial L}{\partial w}$ and $\frac{\partial L}{\partial b}$.

(a)

$$\frac{\partial L}{\partial w} = 2(y - f(x))^3 f(x)(1 - f(x))^2 x$$
$$\frac{\partial L}{\partial b} = 2(y - f(x))^3 f(x)(1 - f(x))^2$$

(b)

$$\frac{\partial L}{\partial w} = 2(y - f(x))^3 (f(x)^2 - 1)x$$
$$\frac{\partial L}{\partial b} = 2(y - f(x))^3 (f(x)^2 - 1)$$

(c)

$$\frac{\partial L}{\partial w} = 2(f(x) - y)^3 f(x)^2$$
$$\frac{\partial L}{\partial b} = 2(f(x) - y)^3 f(x)^2 x$$

(d)

$$\frac{\partial L}{\partial w} = 2(y - f(x))^3 f(x)(1 - f(x))yx$$
$$\frac{\partial L}{\partial b} = 2(y - f(x))^3 f(x)(1 - f(x))y$$

10. Let f(x, y) = $x^2 + \frac{y^2}{100}$. Gradient descent with a fixed step size η is run for finding the minimum value of f from an initial point (x_o, y_o).

    1. Give an expression for (x_t, y_t) in terms of η and (x_o, y_o).
       ---
    2. Let (x_o, y_o) = (10, 0), give the range of η for which convergence to the solution is guaranteed.
       ---
    3. Let (x_o, y_o) = (2, 5), give the range of η for which convergence to the solution is guaranteed.
       ---

11. Consider a multivariate linear regression problem where the output $\hat{Y} = XW$ where X $\in R^{m \times n}$, m is the number of training samples, n is the number of features and Y is the true labels. The objective is to minimize the squared error function where $Y, \hat{Y} \in R^m$.

$$L(W) = \frac{1}{M} \sum_{i=1}^{M} (Y_i - \hat{Y}_i)^2$$

    Derive the gradient $\frac{\partial L}{\partial W}$ for the gradient descent update rule.
    Answer:
    $\frac{\partial L}{\partial W}$ = ---

12. Consider a binary classification problem. Which of the following loss functions when used with a deep neural network ($\geq$ 1 hidden layer) with **non-linear** activations is a convex loss function? Provide a proof for your answer.

    (a) Cross Entropy
    (b) Mean Squared error
    (c) All of the above
    (d) None of the above

13. Consider a multivariate linear regression problem where the output $\hat{Y} = \mathrm{X}W$ where $\mathrm{X} \in R^{m \times n}$, m is the number of training samples, n is the number of features and Y is the true labels. The objective is to minimize the squared error function where $Y, \hat{Y} \in R^m$.

$$L(W) = \frac{1}{M} \sum_{i=1}^{M} (Y_i - \hat{Y}_i)^2$$

Find a closed form solution to this problem if it exists. Think about why do we use gradient descent (an iterative approach) in practice over this.

(a) W = $X^T X X^T Y$
(b) W = $X^T (X X^T)^{-1} Y$
(c) W = $(X^T X)^T X Y$
(d) W = $(X^T X)^{-1} X^T Y$

14. Suppose we train a deep neural network using the cross entropy loss for classification. Now, instead of minimizing the cross-entropy loss, suppose we change our objective function (J(θ)) to maximize the probability of the correct class. What changes will have to be made in our training setup?

(a) We cannot use backpropagation since it is applicable only in scenarios where we are minimizing an objective function, not maximizing it.
(b) We will have to change the update rule to $\theta_j : \theta_j + \alpha \frac{\partial J(\theta)}{\partial \theta_j}$.
(c) We do not need to change anything and the network will still get trained properly without any modification.

15. Which of the following loss functions when used with logistic regression is a convex loss function? Provide a proof for your answer.

(a) Cross Entropy
(b) Mean Squared error
(c) All of the above
(d) None of the above

16. Suppose we have the following four points: x_1 = (1,1), x_2 = (-1, 3), x_3 = (2, 4) and (y_1, y_2, y_3) = (5, 11, 18). Find $\min_w \sum_{i=1}^{3} ({x_i}^T w - y_i)^2$ and also the value of w that leads to this minimum value.

    (a) min value = 0, w = [1,4]
    (b) min value = 0, w = [4,1]
    (c) min value = 1, w = [2,5]
    (d) min value = 1, w = [5,2]

17. Which of the following metrics can be used to measure the similarity between two probability distributions?

    (a) Jensen-Shannon divergence
    (b) Kullback–Leibler(KL) divergence
    (c) Cross-Entropy
    (d) Mahalanobis divergence

18. Consider this function: $x^2y^2 + y^2z^2 + z^2x^2 = 0$. Compute $\frac{\partial x}{\partial y}$.

    (a) $$-\frac{y(x+z)}{x(y+z)}$$

    (b) $$-\frac{y^2(x+z)}{x^2(y+z)}$$

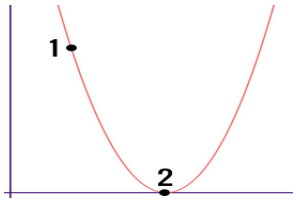
    (c) $$\frac{y(x^2-z^2)}{x(y^2-z^2)}$$

    (d) $$-\frac{y(x^2+z^2)}{x(y^2+z^2)}$$

19. Consider a binary classification problem. Which of the following loss functions when used with a deep neural network ($\geq$ 1 hidden layer) with **linear** activations is a convex loss function? Provide a proof for your answer.

    (a) Cross Entropy
    (b) Mean Squared error
    (c) All of the above
    (d) None of the above

20. Consider this quadratic loss function J(θ). Which of the following update equations will take minimum steps to reach from point 1 to point 2?

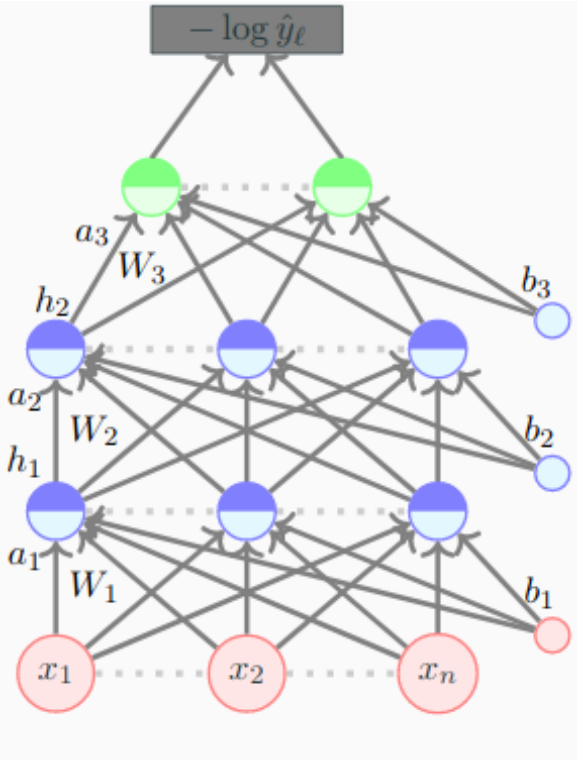


(a) $\theta^* = \theta_o$ - 0.5 $\nabla_\theta J(\theta_o)$
(b) $\theta^* = \theta_o$ - H^{-1} $\nabla_\theta J(\theta_o)$ where H is hessian of J at θ_o
(c) $\theta^* = \theta_o$ - 4 $\nabla_\theta J(\theta_o)$
(d) $\theta^* = \theta_o$ - 2 $\nabla_\theta J(\theta_o)$

21. We are given astronomical data for star classification. Stars can be classified into seven main types (O, B, A, F, G, K, M) based on their surface temperatures. Additionally, there are sub-classes identified based on their sizes: supergiants, giants, main-sequence stars, and subdwarfs. Hence, a given training sample can be a supergiant star of O type. What changes should be done to the standard feed forward neural network to handle such cases where the classes are not mutually exclusive?

(a) No changes are needed and we can model this problem with the standard setup.
(b) Use sigmoid instead of softmax as the output activation function.
(c) Use Swish activation function instead of ReLU.
(d) Use binary cross-entropy loss for each class instead of categorical cross-entropy loss.

22. An e-commerce company builds a feed forward neural network that predicts how similar are two products. The network has 2 hidden layers and an output layer.



Instead of a linear module, the company decides to have a quadratic module in the first layer with $b_1 = 0$ where $x \in R^n$. In addition to this, they use Swish activation function(g(x)) instead of ReLU. Loss is cross-entropy.

$$a_1 = x^T W_1 x$$

$$g(x) = x.\text{sigmoid}(x)$$

Compute the backpropagation updates of this network, specifically derive: $\frac{\partial L}{\partial a_3}$, $\frac{\partial L}{\partial a_2}$, $\frac{\partial L}{\partial a_1}$ and $\frac{\partial L}{\partial W_{111}}$

END