Time: 45 minutes Max marks: ~~10~~ 8

1. (2+1=3 points) The sigmoid function is given as $\sigma(z) = \frac{1}{1+e^{-z}}$. Logistic regression can be implemented by a single neuron neural network that has a sigmoid activation function. (i) Compute the gradient of the output of this neural network with respect to the weight parameters $w_i, i = 0, 1, 2, \ldots, d$, where w_0 is the bias term and the input is $\mathbf{x}$ with a 1 appended. The *logit* is computed as $z = \mathbf{w}^\top \mathbf{x}$. (ii) Write this derivative expression in terms of $\sigma(z)$.

$x \in \mathbb{R}^d \quad P(y=1|x) = a = \dfrac{1}{1+e^{-w^T x}}$

$w^T x = w_0 + w_1 x_1 + \cdots + w_d x_d$

(i) $\dfrac{\partial a}{\partial w_i} = \dfrac{\partial a}{\partial z} \cdot \dfrac{\partial z}{\partial w_i}$

$\dfrac{\partial a}{\partial z} = \dfrac{\partial}{\partial z}\left(\dfrac{1}{1+e^{-z}}\right) = \dfrac{-1}{(1+e^{-z})^2} \cdot \dfrac{\partial}{\partial z}(1+e^{-z}) = \dfrac{-(-e^{-z})}{(1+e^{-z})^2}$

$= \dfrac{e^{-z}}{1+e^{-z}} \cdot \dfrac{1}{1+e^{-z}} = \left(\dfrac{1+e^{-z}-1}{1+e^{-z}}\right)\left(\dfrac{1}{1+e^{-z}}\right)$

$= \left(1 - \dfrac{1}{1+e^{-z}}\right)\left(\dfrac{1}{1+e^{-z}}\right) = (1-\sigma(z))(\sigma(z))$

$\dfrac{\partial z}{\partial w_i} = \dfrac{\partial}{\partial w_i}(w_0 + w_1 x_1 + w_2 x_2 + \cdots w_i x_i + \cdots + w_d x_d)$

$\therefore$ The gradient vector

$\nabla_w a = (1-\sigma(w^T x))\,\sigma(w^T x) \cdot \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

(ii) See derivation above,

$\sigma(w^T x)(1-\sigma(w^T x)) \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$

2. (1+1=2 points) One of my new students has trained a binary classifier and is achieving a (i) training accuracy of around 95% and a validation accuracy of around 64% (ii) Exactly 99.9% accuracy on both training and validation datasets. The classifier was trained for a `weapon` / `no-weapon` classification at a building security screening facility, and the the class distribution was 999000 samples of `no-weapon` category and 1000 samples of the `weapon category` in the training set and 49950 samples of `no-weapon` category and 50 samples of the `weapon category` in the validation set. I claim that the classification model isn't trained correctly in both cases. Can you suggest to my student what might be the problems in the two cases above?

(i) The model seems to be overfitting the training set. More diverse training samples or regularization may help improve the model performance

(ii) It appears that the classifier might be trivially classifying all samples as the "no-weapon" category. It may help to look at the False Negatives (FN) or the confusion matrix instead of the accuracy metric. Reporting the Precision & Recall or f1 score may also be more useful.

3. (2+1=3 points) I have a CNN model that works with RGB images of size 512×512. The first layer uses 5×5 kernels with a stride of 2 and generates 100 feature maps, and the second layer uses a 3×3 kernel with a stride of 4 and generates 200 feature maps. The activations in both layers are sigmoid. (i) What are the number of learnable parameters in layer 1 and layer 2? (ii) Would the number of parameters change if we change the stride? Would the feature map size change if we change th stride?

(i) Layer 1 :

input channels = 3 , # params for each kernel & bias $= 3 \times 5 \times 5 + 1$

kernels = # o/p feature maps = 100

learnable parameters $= (3 \times 25 + 1) \times 100 = 7600$

Layer 2 :

input channels = 100 , # params for each kernel + bias $= 100 \times 3 \times 3 + 1 = 901$

kernels = # o/p feature maps = 200

learnable params = $901 \times 200 = 180200$

(ii) No, the # parameters won't change with change in stride

Yes, the size of feature maps will change with a change in the stride.

4. (*Extra credit* - 2 points) Let x be a discrete random variable that can take integer values 0 through 9. Let $P(x)$ and $Q(x)$ be two distributions over x. The self-entropy is given as $H(P) = -\mathbf{E}_{x\sim P} \log P(x)$ and the cross-entropy is given as $H(P,Q) = -\mathbf{E}_{x\sim P} \log Q(x)$, where $\mathbf{E}[\cdot]$ is the expectation operator. Given that the KL Divergence is defined as $D_{KL}(P||Q) = -\sum_x P(x) \log \frac{P(x)}{Q(x)}$[1], show that

$$H(P,Q) = H(P) + D_{KL}(P||Q)$$

$D_{KL}(P||Q) = \sum_x P(x)[\log P(x) - \log Q(x)]$

Adding $H(P)$ on both sides, we get

$D_{KL}(P||Q) + H(P) = \sum_x P(x)[\log P(x) - \log Q(x)]$

$- \sum_x P(x) \log P(x)$

$= \sum_x P(x) \log P(x) - \sum_x P(x) \log Q(x)$

$- \sum_x P(x) \log P(x)$

$= - \sum_x P(x) \log (Q(x))$

$= - E_{x\sim P} \log (Q(x)) = H(P,Q)$

[1]The question paper had a mistake with the $D_{KL}(P||Q)$ defined as $D_{KL}(P||Q) = -\sum_x P(x) \frac{\log P(x)}{\log Q(x)}$. This question will be treated as extra-credit