STAT 433 – Midterm Part I

1. During backpropagation, when the gradient passes backward through a sigmoid activation function, the gradient will always decrease in magnitude. A
   A. True
   B. False
2. Suppose that you find that your model's training error looks so good (potential overfitting). What can you do to address this issue? (Check all that apply) a,b,c
   A. Data augmentation
   B. Dropout
   C. Batch Normalization
   D. RMSprop Optimizer
3. Which of the following is true? B
   A. Batch Normalization is an alternative method of dropout.
   B. Batch Normalization makes training faster.
   C. Batch Normalization is a non-linear transformation to give nonlinearity to the network.
   D. Batch Normalization is standardizing the data before training neural network.
4. You want to make the weights sparse and smaller. How can you do that? Why?

Impose l1-penalty. (visual interpretation is omitted in this solution).

5. $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ has a similar performance as sigmoid function except that it is zero-centered. Write down tanh(x) in terms of $\sigma(x)$ where $\sigma(x) = 1/(1 + e^{-x})$. Show your work to get the full credit.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} = \frac{2 - (1 + e^{-2x})}{1 + e^{-2x}} = \frac{2}{1 + e^{-2x}} - 1 = 2\sigma(2x) - 1$$

6. You have a single layer neural network for a binary classification with a sigmoid activation function as below. (X: nXm matrix, predicted y & true label y: 1 X m)

$$z = WX + b$$
$$h = \sigma(z)$$
$$\hat{y} = h$$
$$L = -\sum_{i}^{m} y_i \log \hat{y} + (1 - y_i) \log(1 - \hat{y}_i)$$

What is $\frac{\partial L}{\partial W}$? Write your answer as a matrix-matrix multiplication.

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial W}$$

$$\frac{\partial L}{\partial \hat{y}} = \left[\frac{\hat{y}_1 - y_1}{\hat{y}_1(1 - \hat{y}_1)}, \dots, \frac{\hat{y}_m - y_m}{\hat{y}_m(1 - \hat{y}_m)}\right]$$
$$\frac{\partial \hat{y}}{\partial z} = diag[\hat{y}_1(1 - \hat{y}_1), \dots, \hat{y}_m(1 - \hat{y}_m)]$$
$$\frac{\partial z}{\partial W} = X^T$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial W} = \begin{bmatrix} \hat{y}_1 - y_1 \\ \vdots \\ \hat{y}_m - y_m \end{bmatrix} X^T$$

Your answers can be written in as the transpose of the above equation. (numerator-layout convention)

7. (continued from the above question) suppose that you apply ReLU activation before sigmoid activation. i.e., $\hat{y} = \sigma(ReLU(z))$. Then you classify the object by checking if $\hat{y} \geq 0.5$ or $\hat{y} < 0.5$. What will happen? Why?

ReLU will give non-negative value, and then the sigmoid activation will give the numbers >=0.5 always. Thus, all predictions will be positive.

8. Suppose that your classmate finds an activation function that is similar to ReLU such that

$$f(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

Will you use this? Why?

No. The gradient is zero except the origin. Thus it would not pass any gradient back during backpropagation.

9. Provide two reasons why we are using convolutional layers instead of fully connected layers for image classification.

- ✓ Convolutional layers captures the spatial characteristic of the data.
- ✓ Less parameters compared to the fully connected layers since CNN's share weights.
- ✓ Translation invariance

10. Consider to build a CNN for an image classification problem in which the layers are defined by the left column below. Fill the table below. Assume that width & height of the kernels (for Conv, Pool) are the same. Stride 1 Pad 1 for convolving layers. Stride 2 Pad 0 for Pooling layers. FC: a fully-connected layer.

	Output Size		Layer		
Layer	C	H/W	filters	kernel	Number of parameters
Input	3	32	-	-	0
Conv	16	32	16	3	16*(3*3*3+1)=448
ReLU	16	32	-	-	0
Pool	16	16	-	2	0
BatchNorm	16	16	-	-	2*16=32
Conv	16	16	16	3	16*(3*3*16+1)=2320
ReLU	16	16	-	-	0
Pool	16	8	-	2	0
Flatten	16*8*8=1024	-	-	-	0
FC	10	-	-	-	(1024+1)*10=10250