CS 231N
Convolutional Neural Networks for Visual Recognition
Spring 2017 Sample Midterm Exam

May 9, 2017

Full Name: _____

Question	Score
Multiple Choice (20 pts)	
True/False (20 pts)	
Short Answer (60 pts)	
Total (100 pts)	

Welcome to the CS231N Midterm Exam!

- The exam is 1 hour 15 minutes.
- No notes or electronic devices are allowed.

I understand and agree to uphold the Stanford Honor Code during this exam.

Signature: _____ Date: _____

Good luck!

1 Multiple Choice (20 points)

Circle the letters of your choice.

Each question is worth 2 points. Each one of the four individual choices is 0.5 points for a correct answer, or 0 points otherwise.

1. You start training your Neural Network but the loss is almost completely flat. What could be the cause?
   (a) The learning rate could be too low
   (b) The regularization strength could be too high
   (c) The class distribution could be very uneven in the dataset
   (d) The weight initialization scale could be incorrectly set

2. A VGGNet only uses a sequence of 3x3 CONV with stride 1 pad 1 and 2x2 POOL stride 2 pad 0 layers. It eventually transitions to Fully Connected layers and the classifier. There are 5 POOL layers in total. On ImageNet, the VGGNet takes 224x224 images. If we tried to run the VGGNet on a 32x32 input (e.g. CIFAR-10 image):
   (a) The code would crash on the very first CONV layer because 3x3 filters with stride 1 pad 1 wouldn't "fit" across 32x32 input
   (b) The amount of memory needed to store the forward activations in the first CONV layer would be reduced by a factor of 7 (since $224/32 = 7$)
   (c) The network would run fine until the very first Fully Connected layer, where it would crash
   (d) The network would run forward just fine but its predictions would, of course, be ImageNet class predictions

3. A max pooling layer in a ConvNet:
   (a) Is approximately as fast to compute in both forward and backward pass as a CONV layer (with the same filter size and strides).
   (b) Is similar to batch normalization in that it will keep all of your neuron activities in a similar range.
   (c) Could contribute to difficulties during gradient checking.
   (d) Could contribute to the vanishing gradient problem (recall: this is a problem where by the end of a backward pass the gradients are very small)

2 True / False (20 points)

For these problems, no explanation is required; simply circle True, False, or do not give an answer.

Scoring: There are 20 questions, each worth 1 point. To discourage guessing, incorrect answers are worth -1 point. Leaving a question blank will give 0 points.

1. **True or False:** Your Neural Network is not gradient checking. It could be that it's because you're using the AdaGrad update instead of Vanilla SGD.

2. **True or False:** If the input to a ConvNet is a zero image (all zeros), then the class probabilities will come out uniform.

3. **True or False:** Turning off L2 weight regularization will likely lead to higher accuracy on the training set.

4. **True or False:** It's sufficient for symmetry breaking in a Neural Network to initialize all weights to 0, provided that the biases are random

5. **True or False:** The derivative of the loss with respect to some weight in your network is -3. That means that decreasing this weight (by a tiny amount) would decrease the loss.

6. **True or False:** Regularizing the biases in a neural network is not as important because they do not interact multiplicatively with the inputs.

7. **True or False:** During backpropagation, as the gradient flows backwards through a tanh non-linearity, it will always become smaller or equal in magnitude (Recall: if $z = tanh(x)$ then $\frac{\partial z}{\partial x} = 1 - z^2$).

8. **True or False:** During backpropagation, as the gradient flows backwards through any of sigmoid/tanh/ReLU non-linearities, it cannot change sign.

9. **True or False:** If a neuron with the ReLU activation function receives input that is all zero, then the final (not local!) gradient on its weights and biases will also be zero (i.e. none of its parameters will update at all).

10. **True or False:** The loss function will always decrease after a parameter update when performing Vanilla Gradient Descent on the full objective (no mini-batches).

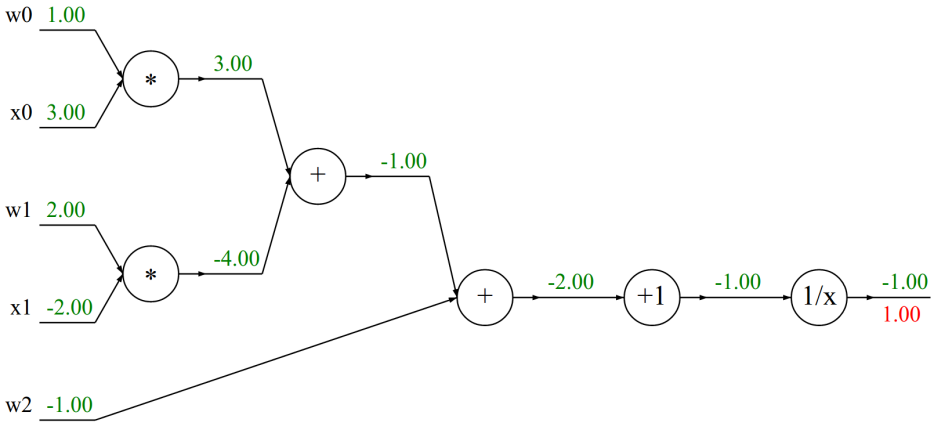
11. **True or False:** One reason that the centered difference formula for the finite difference approximation of the gradient is preferable to the uncentered alternative is because it is better at avoiding kinks in the objective. Recall: the centered formula is $f'(x) = (f(x+h) - f(x-h))/2h$ instead of $f'(x) = (f(x+h) - f(x))/h$.

3 Short Answer (60 points)

Answer each question in provided space.

3.1 Backpropagation

Fill in the missing gradients underneath the forward pass activations in each circuit diagram. The gradient of the output with respect to the loss is one (1.00) for every circut, and has already been filled in.



3.2 Convolutional Architectures

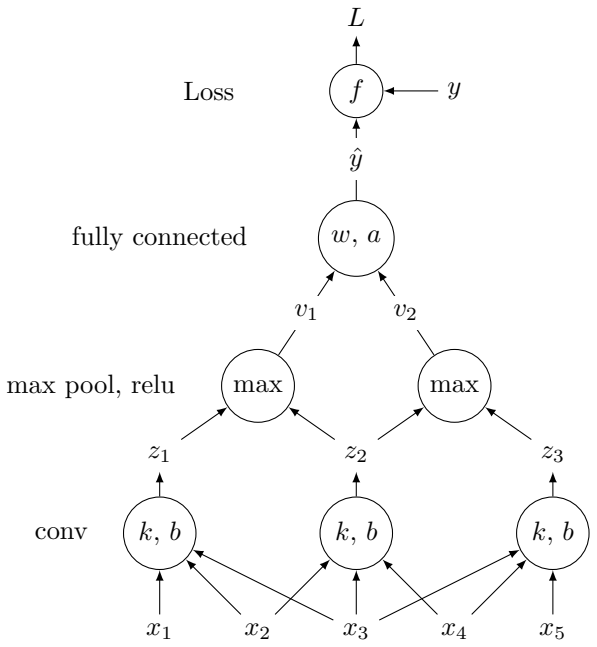
Consider the convolutional network defined by the layers in the left column below. Fill in the size of the activation volumes at each layer, and the number of parameters at each layer. You can write your answer as a multiplication (e.g. 128x128x3).

- CONV5-N denotes a convolutional layer with N neurons, each having 5x5xD filters, where D is the depth of the activation volume at the previous layer. Padding is 2, and stride is 1.
- POOL2 denotes a 2x2 max-pooling layer with stride 2 (pad 0)
- FC-N denotes a fully-connected layer with N neurons.

Layer	Activation Volume Dimensions (memory)	Number of parameters
INPUT	32x32x1	0
CONV5-10		
POOL2		
CONV5-10		
POOL2		
FC-10		

3.3 Simple ConvNet (12 points)

Consider the following 1-dimensional ConvNet, where all variables are scalars:



$$L = \frac{1}{2}(y - \hat{y})^2$$

$$\hat{y} = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + a$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \max\{z_1, z_2, 0\} \\ \max\{z_2, z_3, 0\} \end{bmatrix}$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 & 0 & 0 \\ 0 & k_1 & k_2 & k_3 & 0 \\ 0 & 0 & k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix}$$

(a) **(1 point)** List the parameters in this network.

(b) **(3 points)** Determine the following

$$\frac{\partial L}{\partial w_1} =$$

$$\frac{\partial L}{\partial w_2} =$$

$$\frac{\partial L}{\partial a} =$$

(c) **(3 points)** Given the gradients of the loss L with respect to the second layer activations v, derive the gradient of the loss with respect to the first layer activations z. More precisely, given

$$\frac{\partial L}{\partial v_1} = \delta_1 \qquad \frac{\partial L}{\partial v_2} = \delta_2$$

Determine the following

$$\frac{\partial L}{\partial z_1} =$$

$$\frac{\partial L}{\partial z_2} =$$

$$\frac{\partial L}{\partial z_3} =$$

(d) **(3 points)** Given the gradients of the loss L with respect to the first layer activations z, derive the gradient of the loss with respect to the convolution filter k. More precisely, given

$$\frac{\partial L}{\partial z_1} = \delta_1 \qquad \frac{\partial L}{\partial z_2} = \delta_2 \qquad \frac{\partial L}{\partial z_3} = \delta_3$$

Determine the following

$$\frac{\partial L}{\partial k_1} =$$

$$\frac{\partial L}{\partial k_2} =$$

$$\frac{\partial L}{\partial k_3} =$$

$$\frac{\partial L}{\partial b} =$$

(e) **(2 points)** Suppose we have a general 1D convolution layer

$$\begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} k_1 & \cdots & k_d & & & \\ & k_1 & \cdots & k_d & & \\ & & & \ddots & & \\ & & & k_1 & \cdots & k_d \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b \\ \vdots \\ b \end{bmatrix}$$

And we know that

$$\frac{\partial L}{\partial z_i} = \delta_i$$

Determine

$$\frac{\partial L}{\partial k_j} =$$

$$\frac{\partial L}{\partial b} =$$