For each of the following questions, circle the letter of your choice. Each question has AT LEAST one correct option unless explicitly mentioned. No explanation is required.

(a) **(2 points)** Your model for classifying different dog species is getting a high training set error. Which of the followings are promising things to try to improve your classifier?

(i) Use a bigger neural network

(ii) Get more training data

(iii) Increase the regularization parameter lambda

(iv) Increase the parameter keep_prob in dropout layer (assume the classifier has dropout layers)

(b) **(2 points)** Which of the followings are true about Batch Normalization?

(i) Batch Norm layers are skipped at test time because a single test example cannot be normalized.

(ii) Its learnable parameters can only be learned using gradient descent or mini-batch gradient descent, but not other optimization algorithms.

(iii) It helps speed up learning in the network.

(c) **(2 points)** If your input image is 64x64x16, how many parameters are there in a single 1x1 convolution filter, including bias?

(i) 2

(ii) 17

(iii) 4097

(iv) 1

(2 points) The shape of your input image is (n_h, n_w, n_c); the convolution layer uses a 1-by-1 filter with stride = 1 and padding = 0. Which of the following statements are correct?

(i) You can reduce n_c by using 1x1 convolution. However, you cannot change n_h, n_w.

(ii) You can use a standard maxpooling to reduce n_h, n_w, but not n_c.

(iii) You can use a 1x1 convolution to reduce n_h, n_w, n_c.

(iv) You can use maxpooling to reduce n_h, n_w, n_c.

(2 points) What is the benefit of using Momentum optimization?

(i) Simple update rule with minimal hyperparameters

(ii) Helps get weights out of local minima

(iii) Effectively scales the learning rate to act the same amount across all dimensions

(iv) Combines the benefits of multiple optimization methods

(2 points) Which of the below can you implement to solve the exploding gradient problem?

(i) Use SGD optimization

(ii) Oversample minority classes

(iii) Increase the batch size

(iv) Impose gradient clipping

Short answer

(2 points) A convolutional neural network has 4 consecutive layers as follows:

3x3 conv (stride 2) - 2x2 Pool - 3x3 conv (stride 2) - 2x2 Pool

How large is the support (the set of image pixels which activate) of a neuron in the 4^{th} non-image layer of this network?

(2 points) Is it always a good strategy to train with large batch size? Why or why not?

(2 points) Let **p** be the **probability of keeping** neurons in a dropout layer. We have seen that in forward passes, we often scale activations by dividing them by p during training time.

You accidentally train a model with dropout layers *without* dividing the activations by p. How would you resolve this issue at test time? Please justify your answer mathematically.