Polarr CV/DL Engineer Assignment

This assignment is meant to emulate a day's work as a CV developer here at Polarr. Expect some tasks to be very trivial, while others a bit more challenging and open-ended. Feel free to stack-overflow, Google, and research any questions/issues you have. You are also welcome to use any programming language. The only thing we ask is that this work is done independently by you without the help of your family/friends. Recommended Time Limit: 8 hours (you should not spent more time than this, but if you do, that's okay as well).

When you have completed the assignment, please email your completed assignment to **hiringassignments@polarr.co** with the subject line **YOUR NAME-ASSIGNMENT NAME (e.g. Clement Tam-Polarr Full Stack Generalist Assignment A).**

- All Technical Assignments need to be sent as either an archive or private Github repo.
- All Non Technical Assignments should be sent as a PDF, Word, Pages or Google Doc.

Question 1: Is this Photo Blurry? (30mins)

1. Download the following 3 photos, where one each has been applied certain amount of Gaussian blur:

2. Write a function that calculates a `bluriness score` for each of the 3 photos and rank them in terms of the score. Confirm visually that your ranking of the photos make sense. For instance, a photo that is very sharp could have a score of 5, while a photo that is very blurry has a score of 1.

Question 2: Simple Neural Networks (1 hour)

Assume we have a fully-connected neural network that does image recognition with one hidden layer where the input is an image with channels R,G,B, and dimensions 224x224.

```
The input is X, W1 is weight 1, B1 is bias 1, W2 is weight 2, B2 is bias 2, and y is the output. The output is y = choose_max_index(softmax(W2 x ReLU(W1 x X + B1) + B2)) X: N x D, N is the number of samples, D is 3x224x224 W1: D x H, D is 3x224x224, H the hidden layer size B1: 1 x H, a vector of size H ReLU: np.maximum(value, 0) W2: H x C, H is the hidden layer size, C the number of output labels B2: 1 x C, vector of size C softmax: e^z_j/Sum(e^x_i) for i = 1, ... C; for j in 1 to C choose_max_index: for all values of softmax function, choose the index with the max output
```

JavaScript

1. What would be an appropriate loss function for this neural network? How would you ensure that the network doesn't overfit?

2. Explain how you would train the network to find the appropriate weights and biases. Be sure to include details on what the gradients are for W1, W2, B1, and B2 if you are using back-propagation. Write the pseudo code for your training process.

Question 3: Slightly More Complicated Neural Networks (1-1.5 hour)

1. Scan through the paper on depthwise-separable factorization using MobileNets: https://arxiv.org/abs/1704.04861
2. Fork the TF implementation of MobileNets and download the pre-trained weights: https://github.com/Zehaos/MobileNet
3. Given the weights, write a test script to restore the TF session and run image classification on this image:

4. Find the top 5 labels and their respective probabilities.

5. Using TensorFlow, dump out the name (including BN parameters) and shape of the first 3 layers. I've completed part of it for you, stop at

```
conv_ds_2/pw_batch_norm/moving_variance: u'global_step:0', ()
u'MobileNet/conv_1/weights:0', (3, 3, 3, 32), u'MobileNet/conv_1/biases:0',
(32,), ... u'MobileNet/conv_ds_2/pw_batch_norm/moving_variance:0', (64,)
```

JavaScript

6. Instead of doing separate convolution and batch normalization step for each layer, we would like to combine the two steps so that convolution takes care of the normalization. Combine the batch normalization parameters and weights to calculate the batch-normalized weights and biases for only the layer conv_1 of shape 3x3x3x32. You can ignore the biases provided in the model. The formula for calculating BN weights and bias is here: https://forums.developer.apple.com/thread/65821. Assume gamma = 1, calculate the weights modifiers and biases for each output channel (32). Do it only for the first convolution layer, conv_1. Please, save your modified model as a separate frozen graph file. Test your saved frozen graph and compare the results. You need to provide it along with the solution to this task.
7. The current model contains 1000 labels. Using the same model, reduce it to only the following 10 labels: `["pickup", "pier", "piggy bank", "pill bottle", "pillow", "ping-pong ball", "pinwheel", "pirate", "pitcher", "plane"]` Run the test program again with the photo from question 3, what are the top 5 labels and their probabilities this time?

Question 4: CPU level parallelism and GPU (1 hour)

1. As a CV engineer, you should have had experience doing image manipulations using multi-threading in the CPU or using GPU kernels. First, study the single threaded program below that converts an RGB input that's loaded in an array to a grayscale image. The input array size is 1920x1080x3 flattened to an 1D array.

```
int *get_greyscale(int* input) // this is the input image with a size of
[image_width*image_height*3] { int output_greyscale[image_width*image_height] =
{0}; // output range from 0 - 255 for (int i=0; i<image_height; i++) { for (int
j = 0; j<image_width; j++) { output_grey_scale = 0.2989 *
input[i*image_width*3+j*3] + 0.5870 * input[i*image_width*3+j*3 + 1] + 0.1140 *
input[i*image_width*3+j*3 + 2]; } } return output_greyscale; }
```

C++

2. Rewrite the code above using CPU thread-level parallelism to utilize all the available cores on the CPU to accomplish the same output as above.

3. Write the pseudo-code in either OpenCL or CUDA for the kernel that performs the same function. Remember to specify what your threadgroup id and local thread id is. Write both the instantiation code in C++ as well as the kernel code in CUDA/OpenCL (pseudo code is okay). Assume a max thread count of 256 per thread group (or per block depending on your terminology), and thread group count (or block count) it has no limit.

4. Explain what you expect as the speed up is for each implementation. You do not need to be precise. Assume the following:
    - input image is 1080p (1920x1080) and 3 channels
    - GPU has a max 256 local cores, 8GB of local memory
    - CPU has 8 cores, runs at 2GHz, 8GB of RAM

Question 5: Research Paper (1 hour)

Please review this paper which won the best paper for 2018 at CVPR: https://arxiv.org/pdf/1804.08328.pdf Answer the following questions:

1. What are the key findings for this paper?
2. What ground-breaking applications can you think of based on this paper's findings?
3. The paper only showed a first-order task affinity matrix for all of the tasks, assuming we would like to extend the findings of the paper and try to compute a 2-D affinity matrix for all of the tasks, how would you go about designing a procedure for developing such a flow?

Question 6: Deep Learning is King (5 minutes)

Explain in your own words why Deep Learning approaches has become the state of art approach in solving traditional CV and PR problems. Why does image recognition and object detection work the best using neural networks as opposed to some of the other older traditional CV approaches?

Question 7: Fun Pattern Question (30 minutes)

You are given a sequence of bit of size 5 where 4 bits are 0 and one bit is 1. So here are all the valid sequences 10000, 01000, 00100, 00010, 00001. At every second, the 1 bit will have a shift either to the left or to the right. If the current sequence is 10000, the 1 bit can only shift to the right, so the next sequence must be 01000. Also for the sequence 00001, the next sequence can only be 00010. Assume at every second, you are only allowed to sample at one index, would you be able to come up with an algorithm that finds the index of the value 1 in a finite amount of time?

For example, see the following attempt to locate the index of the 1 bit:

```
time 0, actual sequence is 01000, sample index 0, returned 0; time 1, actual
sequence is 10000, sample index 1, returned 0; ...
```

C++

Thanks again for attempting our assignment.