knn.ipynb

Inline Question 1:
Notice the structured patterns in the distance matrix, where some rows or columns are visibly brighter. (Note that with the default color scheme black indicates low distances while white indicates high distances.)
What in the data is the cause behind the distinctly bright rows? What causes the columns?
A: Bright rows are test images that are significantly different from all training images; bright columns are training images that are significantly different from all test images. In both cases, the differences may be caused by variations in viewpoint, illumination, deformation etc.

Inline Question 2: We can also use other distance metrics such as L1 distance.
For pixel values $p_{ij}^{(k)}$ at location (i, j) of some image I_k,
the mean μ across all pixels over all images is

$$\mu = \frac{1}{nhw} \sum_{k=1}^{n} \sum_{i=1}^{h} \sum_{j=1}^{w} p_{ij}^{(k)}$$

And the pixel-wise mean μ_{ij} across all images is

$$\mu_{ij} = \frac{1}{n} \sum_{k=1}^{n} p_{ij}^{(k)}.$$

The general standard deviation σ and pixel-wise standard deviation σ_{ij} is defined similarly.
Which of the following preprocessing steps will not change the performance of a Nearest Neighbor classifier that uses L1 distance? Select all that apply.

1. Subtracting the mean μ ($\tilde{p}_{ij}^{(k)} = p_{ij}^{(k)} - \mu$.)

2. Subtracting the per pixel mean μ_{ij} ($\tilde{p}_{ij}^{(k)} = p_{ij}^{(k)} - \mu_{ij}$.)

3. Subtracting the mean μ and dividing by the standard deviation σ.

4. Subtracting the pixel-wise mean μ_{ij} and dividing by the pixel-wise standard deviation σ_{ij}.

5. Rotating the coordinate axes of the data, which means rotating all the images by the same angle. Empty regions in the image caused by rotation are padded with a same pixel value and no interpolation is performed.

A: 1,2,3,5
1.2. unchanged under mean subtraction: (x1 - c) - (x2 - c) = x1 - x2
3. unchanged after dividing by STD: the value of L1 distance may change, but the relative order amongst distances remains unchanged (e.g. if x1 > x2 then x1/c > x2/c), thus the performance is the same.
4. the performance may change since each pixel is nonuniformly scaled.
5. the performance won't change since the relative transformation between images doesn't change when they're all rotated by the same angle.

Inline Question 3:
Which of the following statements about k-Nearest Neighbor (k-NN) are true in a classification setting, and for all k? Select all that apply.

1. The decision boundary of the k-NN classifier is linear.
2. The training error of a 1-NN will always be lower than or equal to that of 5-NN.
3. The test error of a 1-NN will always be lower than that of a 5-NN.
4. The time needed to classify a test example with the k-NN classifier grows with the size of the training set.
5. None of the above.

A: 2,4

1. The decision boundary of the k-NN classifier can be (highly, depending on the value of k) non-linear.
2. Correct since 1-NN always gives 100
3. There's no such guarantee.
4. True since finding the nearest neighbors of a testing point requires calculating distances to all training points.

svm.ipynb

Inline Question 1:
It is possible that once in a while a dimension in the gradcheck will not match exactly. What could such a discrepancy be caused by? Is it a reason for concern? What is a simple example in one dimension where a gradient check could fail? How

would change the margin affect of the frequency of this happening? Hint: the SVM loss function is not strictly speaking differentiable
A: The discrepancy happens around the ”hinge”, e.g. when the approximation range ‘x +/- h‘ (‘h‘ is the step size used for numerical gradients) covers the hinge point, where the numeric gradient will give different results than the analytical gradient. This is not a concern since the loss is not differentiable at the hinge.
An example could be, X = [1,1,1], and W = [1,2,3], delta (aka. margin) = 1, and h = 0.01 (so that W is at the hinge point):
The analytical gradient:

$$-(1 * (1 - 3 + 1 > 0) + 1 * (2 - 3 + 1 > 0)) * 1 = 0$$

The Numerical gradient:

$$(L(w+h)-L(w-h))/2h = ((max(0, 1*1-3.01*1+1)+max(0, 2*1-3.01*1+1))- (max(0, 1*1-2.99*1+1)+max(0, 2*1-2.99*1+1))/0.02 = -0.5$$

Changing the delta will not affect the frequency of this happening.

Inline Question 2:
Describe what your visualized SVM weights look like, and offer a brief explanation for why they look they way that they do.
A: The SVM weights for a class look like an ”average image” of the images of that class. This is reasonable, since during training we are trying to maximize the probability of an image belonging to class c being classified as c, which is given by the dot product of the weights of class c, thus making the weights resemble the images. Students may also include specific examples to explain the shape and color.

softmax.ipynb

Inline Question 1:
Why do we expect our loss to be close to -log(0.1)? Explain briefly.
A: Given a random weight matrix, on average we expect roughly equal scores across all classes for all training examples. Since CIFAR has 10 classes, we expect the loss to be roughly -log(0.1) for each training example.

Inline Question 2:
True or False: Suppose the overall training loss is defined as the sum of the per-datapoint loss over all training examples. It is possible to add a new datapoint to a training set that would leave the SVM loss unchanged, but this is not the case with the Softmax classifier loss.
A: True. Since the loss from a single data point may be 0 for SVM, but could never be 0 for softmax.

two_layer_net.ipynb

Inline Question 1:
We've only asked you to implement ReLU, but there are a number of different activation functions that one could use in neural networks, each with its pros and cons. In particular, an issue commonly seen with activation functions is getting zero (or close to zero) gradient flow during backpropagation. Which of the following activation functions have this problem? If you consider these functions in the one dimensional case, what types of input would lead to this behaviour?

1. Sigmoid
2. ReLU
3. Leaky ReLU

A: (1) Sigmoid (for large magnitude values) and (2) Relu (for negative values). Could also give points for (3) LeakyReLU for very very small slopes + close to zero values

Inline Question 2:
Now that you have trained a Neural Network classifier, you may find that your testing accuracy is much lower than the training accuracy. In what ways can we decrease this gap? Select all that apply.

1. Train on a larger dataset.
2. Add more hidden units.
3. Increase the regularization strength.
4. None of the above.

A: 1, 3
Problem here is overfitting, which may be mitigated by using a larger / more diverse training set (option 1) or regularization (option 3). Option 2 is not correct since a more complex model may make the overfitting worse.

features.ipynb

Inline Question 1:
Describe the misclassification results that you see. Do they make sense?
A: The explanation is qualitative and can get full marks as long as it's reasonable, e.g. misclassification happens between visually similar examples.