Homework 03 – Probability

Arthur J. Redfern
arthur.redfern@utdallas.edu

---

0 Outline

1 Reading
2 Theory
3 Practice

1 Reading

1. Probability
   Motivation: a xNN related probability refresher
   https://github.com/arthurredfern/UT-Dallas-CS-6301-CNNs/blob/master/Lectures/xNNs_030_Probability.pdf

   Complete

2. Visual information theory
   Motivation: an alternative presentation of information theory
   http://colah.github.io/posts/2015-09-Visual-Information/

   Complete

2 Theory

3. Consider the standard game of 20 questions as described in the following pseudo code:

   *# standard game of 20 questions (**interleaved** questions and answers)*
   Winner = Person A
   Person A chooses an object X but doesn't tell Person B what object X they chose
   For q = 1 to 20
       Person B asks a yes or no question to help determine what the object is
       Person A responds truthfully with yes or no

If the question asked by Person B is "Is the object X" then
Winner = Person B
Break

See https://en.wikipedia.org/wiki/Twenty_Questions or other web sites if you're unfamiliar with the game.

Now consider a modified version of 20 questions where

 # modified game of 20 questions (all questions ***then*** *all answers* ***then*** *1 guess)*
Winner = Person A
Person A chooses an object X but doesn't tell Person B what object X they chose
For q = 1 to 19
Person B asks a yes or no question to help determine what the object is
For q = 1 to 19
Person A responds truthfully with yes or no to each question q from Person B
Person B asks "Is the object Y?"
If X == Y
Winner = Person B

3.A. You're Person B and want to win. Which version of the game do you play, the standard or the modified? Why?

The standard game:

- The standard game allows questions to be optimized based on previous answers
- The standard game allows for multiple object guesses
- The modified game is a strict subset of the standard game from the perspective of Person B (i.e., in the standard game they could predetermine the 1st 19 questions and that would make it equivalent to the modified game from their perspective)

3.B. Consider a classification network where multiple CNN or RNN layers transform a data tensor to a feature vector and a dense layer transforms the feature vector to a class pmf vector:

Data tensor → [multiple layers] → feature vector → [dense layer] → class pmf vector

Is this more similar to the standard or modified version of 20 questions? Why?

A classification network is more similar to the modified game. The multiple layers of feature extraction are determined ahead of time and not optimized / modified sequentially based on the output of the previous layer for a specific input (note that attention based layers sort of do optimize for a specific input). After feature extraction is complete, the dense layer linearly combines all of the features and makes a guess with respect to the class. This is similar to the modified game:

- An object is chosen -> a random input is selected
- All questions are asked -> the network body is fixed
- All answers are given -> features are generated from the input
- A final guess is made -> network head is applied

3.C. Change the standard game such that with some probability Person A will answer each question incorrectly. How does this change the strategy of Person B? After answering this question, look at the chain rule of probability with the correct given values replaced by estimates that may or may not be correct (i.e., instead of the typical $p(x_{n-1} \mid x_{n-2}, \ldots, x_0)$ consider $p(x_{n-1} \mid x_{n-2}^{hat}, \ldots, x_0^{hat})$ where hat indicates estimate).

Each answer can be viewed as a Bernoulli random variable with a maximum of 1 bit of information per answer (if Person A always answers truthfully or always answers incorrectly) and a minimum of 0 bits of information per answer (if Person A answers incorrectly 1/2 of the time).

Assuming that Person A does not answer incorrectly 1/2 of the time, Person B needs to:

- Look for non sequentially dependent questions such that an incorrect answer for 1 does not lead to multiple poorly chosen follow up questions
- Look to construct sequential questions with information overlap to detect incorrect answers
- Make object guesses that are possibly inconsistent with some of the answers

3.D. Now change the modified game such that with some probability Person A will answer each question incorrectly. How does this change the strategy of Person B? After answering this question, think about the number of features vs the number of classes that are being estimated.

Similar to the previous question, Person B will now need to construct questions with redundancy, the ideal amount of redundancy being dependent on the bits of information per answer: the closer the probability of an incorrect answer is to 1/2, the more redundancy that is needed (reducing the effective number of questions).

4. A multiple choice test can be viewed as a test with an implicit curve that is noisy and grade dependent. Consider a 100 question ABCD multiple choice test and assume that for each question on the test a person either knows the answer or has no idea and makes a totally random guess. Define the implicit curve as the number of questions answered correctly via knowledge or guess minus the number of questions answered correctly via knowledge. For example, a person knows the answer to 73 questions, guesses on 27 questions and gets a total of 79 questions correct; the implicit curve for this case would be 79 – 73 = 6. Plot the mean and standard deviation of the implicit curve vs the number of questions that a person answered

correctly via knowledge. Use Python to simulate many trials for each point and Matplotlib to plot.

Each guess can be viewed as a Bernoulli random variable with probability p = 1/4 of being correct

- μ = p = 1/4
- σ^2 = p(1 – p) = 3/16

Let C = the number of questions that a person answered correctly via knowledge, then

- Implicit curve mean = (100 – C) (1/4) = 0.2500 (100 – C)
- Implicit curve standard deviation = $((100 - C)(3/16))^{1/2}$ ≈ $0.4330 (100 - C)^{1/2}$

A nice plot would have C on the x axis, the implicit curve mean as a line from (0, 25) to (100, 0) and +/- the implicit curve standard deviation shown relative to the mean.

5. Assume ImageNet has 1.28 million images of size 3 x 256 x 256 with 1280 images each in 1000 different classes. How many bits of information are in the ImageNet labels?

The label indicates class membership. There are a uniform number of images for each class so each class is equally likely, p(k) = 1/1000, and the information in a single label is

$-\Sigma_k\, p(k) \log_2(p(k)) \approx 9.97$ bits

The information in all 1.28M labels is

$-1.28e6\, \Sigma_k\, p(k) \log_2(p(k)) \approx 12.76$M bits

Note that a moderately sized ImageNet optimized network can easily have more than 12.76M weights

3 Practice

6. Starting from the previous example of a simple sequential CNN for CIFAR-10 image classification, do the following:

- Add data augmentation (see the training slides or search online for some options)
- Add batch normalization (replace CNN style 2D convolution layers which include bias and ReLU with CNN style 2D convolution, batch normalization and ReLU layers)
- Add L2 regularization (see the training slides for motivation)

What is the best accuracy value the network achieves on the test set before and after these modifications? List the corresponding training hyperparameters for both cases.

> List best accuracy for the original network and the modified network here. Also list the associated training hyperparameters for both cases.

7. [**Optional**] Take the above trained network and save the intermediate feature maps $\mathbf{x}_d$ after ReLU for layer d for a batch of images (this should be a 4D tensor of batch x channel x rows x cols). Quantize these feature maps to unsigned 8 bits ([0, 255]) via $\mathbf{x}_{q,d} = \text{round}(255*\mathbf{x}_d/\max(\mathbf{x}_d))$. How many bytes (elements) are there in the quantized feature maps? Now create a Huffman code for this set of quantized feature maps and use it to compress the quantized feature maps. How many bytes are in the Huffman encoded quantized feature maps? Which value in [0, 255] is assigned the shortest code word? How does this result change for different layers d after different ReLUs in the network?

> Let B_d = batch*channels*rows*cols be the number of bytes in the original batch of quantized feature maps for layer d. The number of bytes in the Huffman encoded quantized feature maps should be < B_d (because ~ 1/2 the values are 0 and there's a non uniform distribution of the remaining values). 0 should be assigned the shortest code. Different layers will have different compression values depending on the distribution of quantized values.