Data C182 Designing, Visualizing & Understanding DNN
Fall 2024 Eric Kim, Naveen Ashish

Discussion 10

This discussion covers evaluation metrics.

1. Classification Question Statement

Let's consider a standard binary classification problem. Let's say you've trained a neural network to take in pictures of cats and dogs and perform binary classification to say which of those two kinds of animals they contain. Let's also say that their two outputs look like a probability distribution: the numbers corresponding to the two categories are non-negative and sum to one.

Problem: Review - Neural Network Architectures

What would an appropriate neural network architecture be for this problem, i.e., it takes in images and outputs something that looks like a probability distribution over the categories "cat" and "dog"?

Solution:

Solution: Review - Neural Network Architectures

Any architecture that takes in an image and outputs a fixed size vector is reasonable here. So, for instance, you could have a convolutional neural network (maybe with skip connections), followed by a flatten operation to convert the filter maps to a single long vector, followed by one or more fully connected layers mapping to a pair of numbers.
Alternatively, you could replace the convolution layers with a vision transformer. The flatten operation will need to be replaced by one of the aggregation methods discussed last time.
In either case, to make the output look like a probability distibution, you will need to softmax them.
In the case of binary classification, you can actually get away with outputting just a single number, with negative values corresponding to one class and positive to the other. In that case, to make it look like a probability distribution, you will need to pass it through a sigmoid nonlinearity (which is effectively an implicit 2-class softmax). This maps values from the real number line to $(0, 1)$ (with 0 mapping to exactly 0.5), which you can treat as the probability for one of the classes (subtract that amount from 1 to get the probability for the other class).

Now, let's say you pass an image of a dog or cat through the network. It gives you two scores, one corresponding to each category. Based on this, how do you choose which category you think the image came from?

2. Calibration

A pretty natural idea is to pick a **decision threshold** value: if the score for dog exceeds that amount, then you say the image contains a dog. Otherwise, you say that the image contains a cat. But then the question remains: how do we pick that threshold value?

Let's consider a simple case first: when your model outputs are **calibrated**. This means that, when your model assigns a probability score to a category (e.g., "This image is 60% likely to contain a dog and 40%

likely to contain a cat"), then those probabilities are correct: the image *actually* has a 60% of containing a dog.

In practice, to measure whether your model is calibrated or not, you pass in a bunch of images and put them into bins based on the model's assigned probability scores: e.g., bin all images that have $0-10\%$ probability score for dog, then $10-20\%$, etc. If your model is actually calibrated, the images in that first bin should actually be dogs with the probability assigned to them by the model (e.g., $0-10\%$ of the images in the first bin should be dogs).

Problem: Thresholding Calibrated Networks

In the case where your model is calibrated, what should your threshold be?

Solution:

Solution: Thresholding Calibrated Networks

Your threshold should be exactly 0.5 – in that case, if the assigned dog score is over that amount, it's more likely to be a dog than a cat.

A lot of the time the "standard" threshold is not necessarily the best one to use though. For example, suppose that in the dataset of dog and cat pictures that you want to classify, your model assigns pictures of dogs only 40% probability of being dogs, while it assigns cats 10% of being dogs (maybe because the dogs in this dataset look particularly like cats). In this case, your model isn't calibrated (or it might be calibrated on a different distribution of dog and cat pictures), and using the 0.5 threshold will classify none of the dogs correctly. How else can we choose the score then?

Let's first go over some terminology – for a given class that we care about in a binary classification problem, we have:

3. (a) *True positives* - The number of datapoints correctly classified as that class

(b) *True negatives* - The number of datapoints correctly classified as *not* being that class

(c) *False positives* - The number of datapoints incorrectly classified as the class we care about (when they should be the other class)

(d) *False negatives* - The number of datapoints incorrectly classified as the other class (when they should be the class we care about)

One idea is to find some **evaluation metric** that measures how well your model is classifying stuff, akin to the loss you used for training, then pick a threshold that empirically maximizes that score. We will discuss two metrics that we could care about, which are defined in terms of the above four definitions (see also Figure 1).

The first is **precision**, defined as the portion of datapoints correctly classified a particular way (true positives) divided by the total number of datapoints classified that way (true positives plus false positives). In this case, that would be the number of dog pictures correctly classified as dogs divided by the total number of pictures that the model classified as dogs *correctly or incorrectly*.

Problem: Precision Intuition

When would you want to maximize precision? What would a safe strategy for maximizing precision look like?

Solution:

Solution: Precision Intuition

Maximizing precision means being very "conservative" with predictions – you only predict a certain class when you're very sure that you're correct. Intuitively, this means you're not predicting the class that often (so the denominator – the total number of times the model predicted that class – is small) and most of your predictions are correct (the numerator is high). Note that precision does not penalize misclassifying the class of interest: saying that a picture of a dog is actually a cat does not affect either the numerator or denominator!

The second is **recall**, defined as the portion of datapoints correctly classified a particular way (true positives) divided by the total number of instances of that class (true positives plus false negatives). In this case, that would be the number of dog pictures correctly classified as dogs divided by the total number of pictures of dogs in your dataset.

Problem: Recall Intuition

When would you want to maximize recall? What would a safe strategy for maximizing precision look like?

Solution:

Solution: Recall Intuition

Maximizing recall means you want to correctly classify all instances of the class of interest. This can be done by being "generous" or "risky" with the classification – in particular, by always predicting the class, you can maximize recall (since all instances of the class will definitely be correctly classified!).

Problem: Precision and Recall Calculation

Suppose you have a dataset of 10 dogs and 12 cats. Your model classifies 8 images as dogs, 7 of which were actually dogs. What are the precision and recall of your model, where the category you care about retrieving is dogs?

Solution:

Solution: Precision and Recall Calculation

Precision is number of dogs correctly classified divided by total number of images classified that way, so $7/8$. For recall, that's number of correctly classified dogs over total number of dogs, so $7/10$.

Usually though, you'd want some in-between metric that balances between these two. The metric balancing the two is **F1 score**, the harmonic mean between the precision and recall:

$$F_1 = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}} \tag{1}$$

Each of these metrics is computed for a given threshold (or whatever other class decision strategy you have). So, to find a threshold that maximizes the metric, simply test out a wide range of thresholds and see which one achieves the highest. A lot of the time, people visualize this by plotting a **precision-recall curve**: a 2D

plot with precision on one axis and recall on the other (both from 0 to 1), with all points on the curve being empirical precision and recall from some threshold value you tried out.

Problem: Which Metric?

Suppose your classification task is to figure out whether or not medical patients have some disease based on some measured biomedical information. Consider two cases.

1. The disease is not very serious, though patients with the disease would still definitely prefer to be cured of it. If the model says the patient has the disease, they will undergo an expensive treatment (though patients with the disease do think the expense is worthwhile).
2. The disease is potentially serious, but the medication is both inexpensive (maybe it's covered by most insurances or something!) and non-harmful to patients without the disease.

In each case, which metric should you intuitively try to maximize?

Solution:

Solution: Which Metric?

In the first case, you'd presumably want to maximize precision – you want the classifications to be conservative since it's not very harmful if you incorrectly tell a patient they do not have the disease, but incorrectly telling them they *do* have the disease would cost them a lot.

In the second case, you'd presumably want to maximize recall – since the disease is serious, it's more important to make sure that patients that do have it get properly medicated and it doesn't cost much to accidentally prescribe the medicine when they don't have the disease.

Of course, in either case, you still probably want a balance between precision and recall, albeit with higher weighing towards one or the other. You can achieve this with the generalized F-β score, which is like F1 but with a relative weighing for the two terms.

Note that this is a very simplistic example, and the real world is usually much more complex (with unclear trade-offs and preferences) – however, the above just provides some of the examples of the considerations needed for picking evaluation metrics for choosing thresholds.

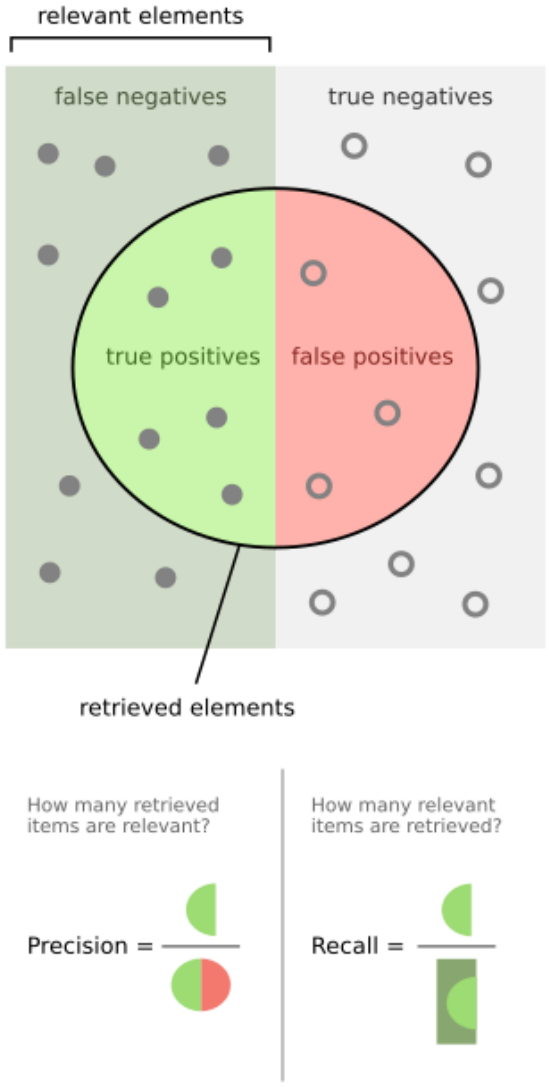

Figure 1: Visual diagram showing precision and recall

