EECS 4422/5323 Computer Vision

Image Understanding 5

Calden Wloka

23 October, 2019

Announcements

- Site Visit in 2 weeks
- Site Visit rubric posted to the course website
- If you are willing to present Nov. 4th instead of Nov. 2nd, please email me
- Reminder: Midterm next class

Outline

- Continuation from last lecture: VQA
- A Closer Look at Gradient Descent
- Analysis of Computer Vision Models

Visual Question Answering

An emerging research area for which the types of questions raised in this lecture are of paramount importance is *Visual Question Answering (VQA)*.

Who is wearing glasses?

man

woman

Where is the child sitting?

fridge

arms

Is the umbrella upside down?

yes

no

How many children are in the bed?

2

1

VQA Involves Other Modalities

A number of modalities involved in VQA research fall outside of the domain of this course, such as linguistic understanding. Nevertheless, the visual side of the problem area are highly emblematic of the open challenges which remain in visual understanding.

As we can see, the type of question being asked greatly impacts the accuracies which can be achieved.

Real Open-Ended

Standard Dev Challenge

Results as of 05/10/2019 (deadline for VQA Challenge 2019).
For information about each test split, please see the challenge page.

	By Answer Type			Overall
	Yes/No	Number	Other	
MIL@HDU[11]	90.33	58.91	65.91	75.26
MSM@MSRA[15]	89.74	59.01	65.89	75.01
LXRT[13]	89.33	57.29	65.32	74.38
XFZ[23]	87.86	57.87	64.3	73.35
AIOZ[4]	87.99	56.16	63.93	73.04
ks_vqa[32]	87.97	55.17	63.97	72.94

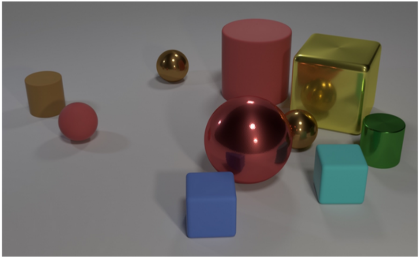
Image source: VQA Challenge 2019

CLEVR: An Alternative Approach to VQA Data

CLEVR is a generational framework for rendering block-world stimuli and corresponding questions.

How does this approach compare to the VQA Challenge from the previous slide?

Questions in CLEVR test various aspects of visual reasoning including **attribute identification**, **counting**, **comparison**, **spatial relationships**, and **logical operations**.



Q: Are there an **equal number** of **large things** and **metal spheres**?
Q: **What size** is the **cylinder that is left of** the **brown metal** thing **that is left of** the **big sphere**?
Q: There is a **sphere** with the **same size as** the **metal cube**; is it **made of the same material as** the **small red sphere**?
Q: **How many** objects are **either small cylinders** or **red** things?

Image source: Johnson *et al.*, 2017

Recap of Gradient Descent

Given a surface which we can probe but not analytically minimize, gradient descent provides a way to iteratively attempt to find a minimum.

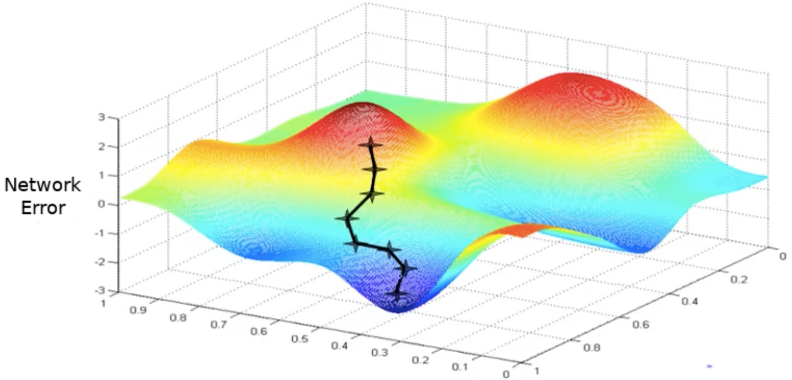


Image source: Original source unknown

Gradient Descent is a General Concept

Although we introduced gradient descent as an integral component of neural network training, it is actually a more general concept in scientific computing. We saw an additional example using gradient descent to compute an adversarial transform of one image class into another.

Image source: Brendel *et al.*, 2018

A Example

Let's look at some toy examples.

Evaluating Models

A computer vision model is typically an encapsulated algorithm which takes an image (or image stream) as input, and performs some operation over this input.

Often we want to characterize the behaviour of a given model, or compare a set of models which are all attempting to accomplish the same task.

Qualitative vs. Quantitative Evaluation

Evaluation is *qualitative* if it is non-standardized from example to example, and is more narrative based (*i.e.*, "This output *looks* better").

Evaluation is *quantitative* if it applies some standardized method of evaluation (usually a *metric*, *i.e.* a distance function) over a test dataset.

The Role of Qualitative Tests

Qualitative tests are helpful for experimental development and hypothesis generation, as well as form a useful way of checking the validity of a given evaluation metric.

The Role of Qualitative Tests

Qualitative tests are helpful for experimental development and hypothesis generation, as well as form a useful way of checking the validity of a given evaluation metric.

However, because qualitative analysis essentially requires visual inspection of every instance of comparison, it is not well suited to large scale testing.

Quantitative Evaluation

Why is quantitative evaluation important?

- Using qualitative testing only can be misleading; it is often possible to pick specific examples which make any given model look good

Quantitative Evaluation

Why is quantitative evaluation important?

- Using qualitative testing only can be misleading; it is often possible to pick specific examples which make any given model look good
- Testing over large datasets is more likely to capture a fuller range of possible stimulus variation

Quantitative Evaluation

Why is quantitative evaluation important?

- Using qualitative testing only can be misleading; it is often possible to pick specific examples which make any given model look good
- Testing over large datasets is more likely to capture a fuller range of possible stimulus variation
- Small scale case testing is often very difficult to design for vision models, since we don't always know how they are likely to behave or fail

Established vs. Custom Data and Metrics

Figuring out how to evaluate your project is **not** a trivial aspect of the work.

Established vs. Custom Data and Metrics

Figuring out how to evaluate your project is **not** a trivial aspect of the work.

Picking an established dataset or metric has the following benefits:

- Provides prior context to your choice and evaluation

Established vs. Custom Data and Metrics

Figuring out how to evaluate your project is **not** a trivial aspect of the work.

Picking an established dataset or metric has the following benefits:

- Provides prior context to your choice and evaluation
- Work has already gone into the development of the tool, reducing your burden

Established vs. Custom Data and Metrics

Figuring out how to evaluate your project is **not** a trivial aspect of the work.

Picking an established dataset or metric has the following benefits:

- Provides prior context to your choice and evaluation
- Work has already gone into the development of the tool, reducing your burden

Nevertheless, there are reasons you might need to develop your own. Any given metric or dataset carries with it assumptions or biases. If you think this obscures or prevents analysis of a useful aspect of your study, you need to mitigate that with something new.

Example Metric: Error Rates

One of the most straightforward ways that model performance can be quanitified is by defining a binary notion of error (*e.g*. classification error). Note: this is not suitable to all model evaluations, but covers many common computer vision situations.

.

Example Metric: Error Rates

One of the most straightforward ways that model performance can be quanitified is by defining a binary notion of error (*e.g.* classification error). Note: this is not suitable to all model evaluations, but covers many common computer vision situations.

There is more than one type of error: *false positives* and *false negatives*.

Truth Conditions

		True Condition	
		Target Present	Target Absent
Predicted Condition	Target Present	True Positive	False Positive
	Target Absent	False Negative	True Negative

Truth Conditions

		True Condition	
		Target Present	Target Absent
Predicted Condition	Target Present	True Positive	False Positive
	Target Absent	False Negative	True Negative

For example, if we are detecting images with dogs, then any image which contains a dog that our model labels with the *dog* tag is a true positive, tp. Any image which contains a dog which is not given the *dog* tag is a false negative, fn. Any image without a dog which is given the *dog* tag is a false positive, fp. Any image without a dog which is not given the *dog* tag is a true negative, tn.

Precision and Recall

Two common methods of quantifying algorithm performance are *precision* and *recall*.

Precision and Recall

Two common methods of quantifying algorithm performance are *precision* and *recall*.

$$\text{Precision} = \frac{tp}{tp + fp}$$

Precision and Recall

Two common methods of quantifying algorithm performance are *precision* and *recall*.

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Precision-Recall Curve Example

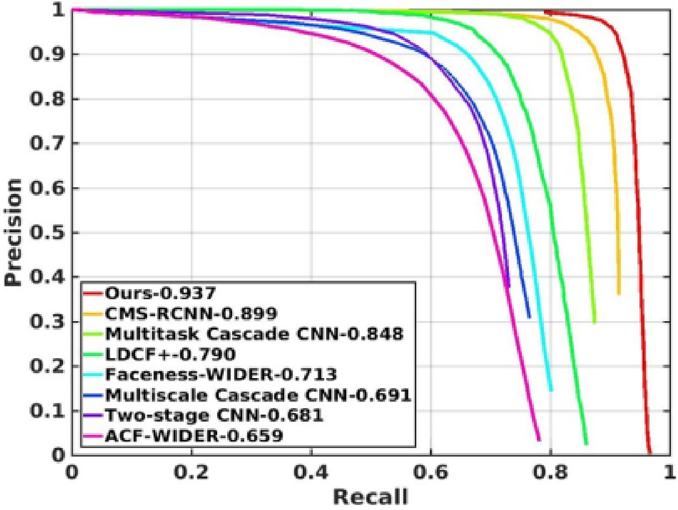


Image source: Zhang *et al.*, 2017