Mathematical Cardiology Lecture 10

Local image filters and learned classification

Lecture notes

An image classifier can be decomposed into operations whose inputs and outputs have definite shapes. We shall follow an image through a local weighted sum, a nonlinearity, a spatial average and a fitted decision rule. This decomposition lets us distinguish the computations that resemble a convolutional network from the quantities that are actually learned. It also makes several checks possible before any accuracy is reported.

1 An image, a mask, and a label

Slides for this section

Let $I \in \mathbb{R}^{16\times 16}$ contain grayscale intensities. In this experiment each image contains a noisy synthetic ellipse. Its label is zero for an ellipse elongated horizontally and one for an ellipse elongated vertically, relative to the image axes. The target is a property assigned by the generator, not an anatomical or disease label.

Classification maps the whole image to one label. Segmentation maps image locations to labels, as in the threshold mask of the preceding lesson. Correctly classifying an image as wide therefore does not locate its boundary. Conversely, a mask can provide a useful shape measurement without being a classifier. The output type should be specified before an algorithm is selected.

The companion uses two fixed directional filters to turn the image into two numbers, then fits two class centers from training images. There is no fitted image normalization in this pipeline, unlike the standardized two-feature example in the previous calibration lesson. The two filter features already have matching numerical units. All image intensities are dimensionless synthetic values; physical pixel spacing and acquisition calibration are absent.

2 A local weighted sum

Slides for this section

The first kernel is

$$K_x = \frac{1}{3}\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \qquad K_y = K_x^T. \tag{2.1}$$

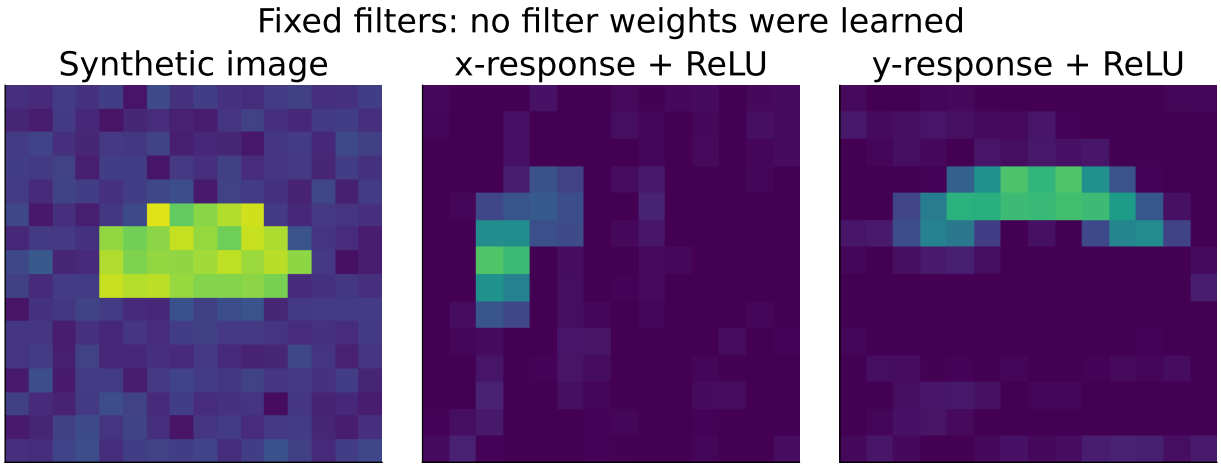
For a valid window beginning at (i, j), its response is

$$A^x_{ij} = \sum_{a=0}^{2}\sum_{b=0}^{2}(K_x)_{ab} I_{i+a,j+b}. \tag{2.2}$$

Here i, j run from 0 to 13. The kernel is not reversed, so the precise operation is a sliding dot product, often called convolution in neural-network software. Each response uses only the local 3×3 patch, and

the same nine weights are used at every valid location. For an $N \times N$ image and $k \times k$ kernel with stride one and no padding, there are $N - k + 1$ valid starts in either direction. Thus the 16×16 image produces a 14×14 map.

A patch with three rows $(0, 0, 1)$ gives response $3(1/3) = 1$ under K_x. Reversing the rows to $(1, 0, 0)$ gives -1. A constant patch gives zero because the kernel weights sum to zero. Positive response therefore identifies an increase in intensity in one direction; it is neither a diagnosis nor a probability. With values in $[0, 1]$, each column difference lies between -1 and 1, so the response also lies in $[-1, 1]$.



Original sixteen-by-sixteen ellipse image and two fourteen-by-fourteen nonnegative directional response maps on a shared 0–1 color scale. Filters are fixed; none was learned.

3 Nonlinearity, averaging, and information loss

Slides for this section

Apply the positive-part function, also called ReLU, to each response:

$$B^x_{ij} = \max(A^x_{ij}, 0), \qquad B^y_{ij} = \max(A^y_{ij}, 0). \tag{3.1}$$

Negative and positive responses no longer cancel. Global mean pooling then produces

$$f_x(I) = \frac{1}{196} \sum_{i,j} B^x_{ij}, \qquad f_y(I) = \frac{1}{196} \sum_{i,j} B^y_{ij}. \tag{3.2}$$

The result is a point $f(I) = (f_x, f_y)$ in a two-dimensional feature space. Its coordinates are nonnegative and at most one for the supplied intensity range. Averaging discards the locations of the responses. Distinct images can therefore share the same feature pair, so a classifier using only these two numbers must give them the same prediction.

The operations yield useful exact checks. Constant images have zero features. Transposing a square image exchanges the two directional response maps up to transposition; since the average is unchanged by transposing a map,

$$f(I^T) = (f_y(I), f_x(I)). \tag{3.3}$$

This matches the generator target: transposing a wide ellipse produces a tall one. A ninety-degree rotation also swaps wide and tall relative to the axes, so it is not a label-preserving augmentation. Pooling can reduce sensitivity to where a response occurs, but finite image boundaries, cropping and discarded negative responses prevent a claim of invariance to every translation or rotation.

4 Fitted centers and learnable convolutional weights

Slides for this section

The training sample contains eighty images generated with one seed; forty additional images generated with a different seed are held out. The class center is

$$c_k = \frac{1}{n_k} \sum_{i:\, y_i = k} f(I_i), \qquad k \in \{0, 1\}, \tag{4.1}$$

using training images only. A new image is assigned to the nearer center in ordinary squared Euclidean distance. Exact ties go to label zero in the implementation. The filters, ReLU and mean-pooling operations are fixed; only the centers are fitted. This is consequently a fixed-filter classifier, not a trained convolutional neural network.

A learned convolutional layer would adjust its kernel weights using training examples and a specified loss. Weight sharing matters even before considering how the weights are learned. One 3×3 filter has nine distinct weights, reused at 196 valid image locations. A locally connected operation with separate weights at each of those locations would have $196 \cdot 9 = 1764$ weights for the same single output map, before adding biases. Our two kernels contain eighteen fixed coefficients; the fact that the coefficients are reused does not imply that they were trained.

A possible learned extension could use a scalar score $s_i = w^T f_K(I_i) + b$ and minimize a loss such as the training mean of $(s_i - y_i)^2$, updating K, w, b. This merely specifies one possible optimization problem; it is not the companion algorithm, and the score need not be a calibrated probability. Demonstrating such an extension would require actual parameter updates, checks of the computation and independent evaluation.

5 Evaluate the frozen rule

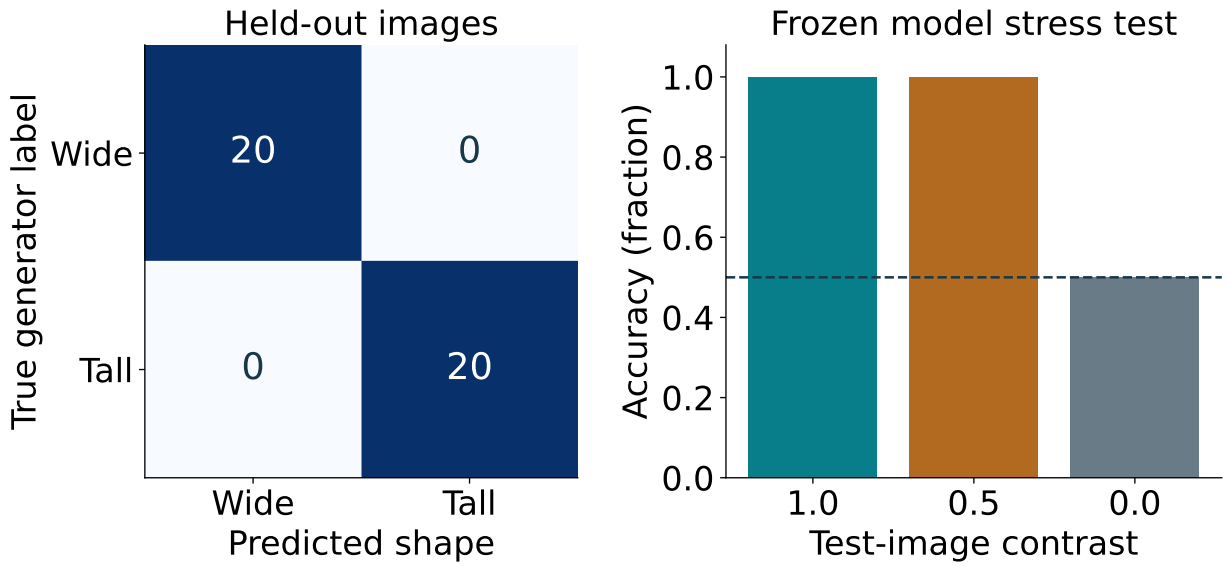
Slides for this section

The default held-out confusion table, with true labels as rows, is

	predicted wide	predicted tall
true wide	20	0
true tall	0	20

All forty synthetic images are classified correctly. The result reflects a generator with a strong orientation difference that the chosen filters can represent. It is not evidence about medical-image variation or clinical targets. A constant wide prediction would correctly label twenty of these forty balanced cases.

The second comparison freezes the centers and changes the contrast parameter used to construct the same test masks. The tested contrast levels are 1, 0.5 and 0, with observed accuracies 1, 1 and 0.5. The noise realization and shape choices are reproducible under the fixed seed; the model is not refitted. At zero contrast the intended shape signal disappears, leaving background noise. The score at that level describes this finite realization; the absence of label information in the generator is the stronger reason no classifier can systematically recover orientation from that input distribution alone.



Confusion counts on forty independent synthetic test images and a frozen-model contrast stress test. Labels describe image orientation. Perfect default performance is not clinical validation.

6 Contrast, target transformations, and a defensible conclusion

Slides for this section

A simple algebraic contrast calculation clarifies the stress test. If an image is replaced by $I' = aI + b\mathbf{1}$ with $a \geq 0$, and there is no clipping or other alteration, zero-sum kernels give $A(I') = aA(I)$. Positive homogeneity of ReLU then gives $f(I') = af(I)$. Fixed class centers do not scale along with the test image, so this operation does not in general preserve nearest-center predictions. The actual generator changes the shape contrast while retaining noise and clipping values to $[0, 1]$, so its stress images are not exactly this affine transformation. The algebra is a check on an idealized operation, not an assertion about every generated pixel.

For real data, the held-out unit must match the intended new case. Slices or repeated visits from one person may share information; a new-patient claim calls for separating people. A more complex network does not remove the need for appropriate labels, a frozen evaluation procedure and relevant failure tests. In a final model argument, give the target, the observations available at prediction time, the fitted quantities and the comparison that was actually performed. Here that conclusion is narrow and testable: two fixed directional features and training centers separate the specified ellipse orientations in the default synthetic test, and the result deteriorates when the orientation signal is removed.

Exercises

MC10-E1. Slide a small filter

A 4 by 4 image has four identical rows (0, 0, 0, 1). Use Kx from the notes, stride 1 and no padding.
a Find the output shape.

b Calculate all four responses.

c Apply ReLU and global mean pooling.

MC10-E2. What was learned?

A pipeline uses two hand-set filters, ReLU, mean pooling and class centers estimated from training examples.

a Classify each component as fixed or fitted.

b Why is calling this a trained CNN misleading?

c What would a genuine convolutional-training extension need?

MC10-E3. Evaluate new cases

On twenty balanced synthetic images, true-wide/pred-wide=9, true-wide/pred-tall=1, true-tall/pred-wide=3, true-tall/pred-tall=7.

a Calculate accuracy and the fraction of true-tall images recognized as tall.

b Compare with a classifier that always says wide.

c Explain why these results do not show reliable medical-image analysis.

MC10-E4. A valid project stress test

A group labels images wide or tall, then rotates them 90 degrees as a supposed label-preserving augmentation.

a Identify and repair the label problem.

b Propose one predeclared stress test that does not intentionally change that target. State what stays frozen and which tested levels you will report.

c State three elements of the final project evidence chain.

References

Zan Ahmad, Mathematical Cardiology syllabus, Fall 2023. Public PDF pages 1–2, Course Topics and Course Objectives.
https://zanahmad.com/files/Fall_2023_HEART_Syllabus.pdf

Zan Ahmad, original editable Mathematical Cardiology syllabus. Rendered export pages 2–3, Assignments & Grading and Projects.

Stanford CS231n, Convolutional Neural Networks. Architecture Overview; Layers used to build ConvNets; Convolutional Layer; Pooling Layer; read 2026-11-14.
https://cs231n.github.io/convolutional-networks/

scikit-learn, Common pitfalls and recommended practices. Section 12.2, Data leakage; read 2026-11-14.
https://scikit-learn.org/stable/common_pitfalls.html