Scientific Computing in Python / 2027 Lecture 5

Neural-network gradients and spatial filters

Zan Ahmad

Proposed 2027; calendar dates pending

Images as arrays

$$X \in \mathbb{R}^{n \times 8 \times 8 \times 1}, \qquad X_{\text{flat}} \in \mathbb{R}^{n \times 64}.$$

Flattening preserves the values and fixes a coordinate convention.

An affine hidden layer

$$U = X_{\text{flat}} W_1 + b_1, \quad H = \tanh U,$$

$$z = H w_2 + b_2, \qquad p = \sigma(z).$$

For h hidden units, W_1 has shape $64 \times h$.

The binary loss

$$L = \frac{1}{n} \sum_i [\log(1 + e^{z_i}) - y_i z_i].$$

```
loss = np.mean(np.logaddexp(0, z) - y*z)
```

Derivative with respect to the logits

$$\frac{\partial L}{\partial z_i} = \frac{p_i - y_i}{n}.$$

The derivative is bounded even when a logit has large magnitude. Stable evaluation remains necessary.

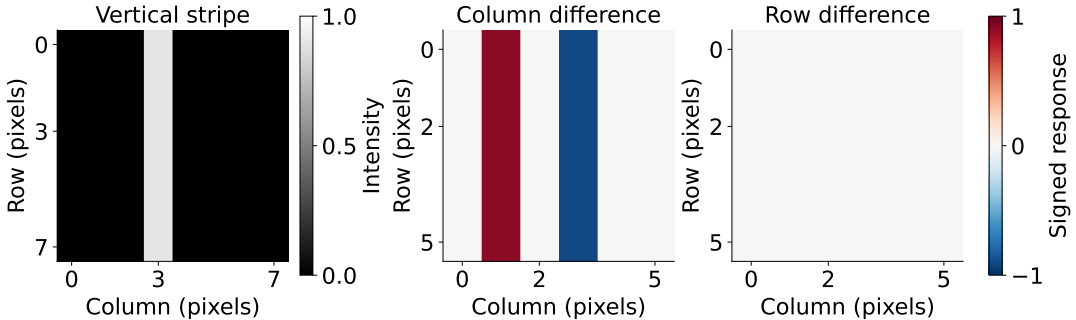
Backpropagation

Let $D = (p - y)/n$. Then

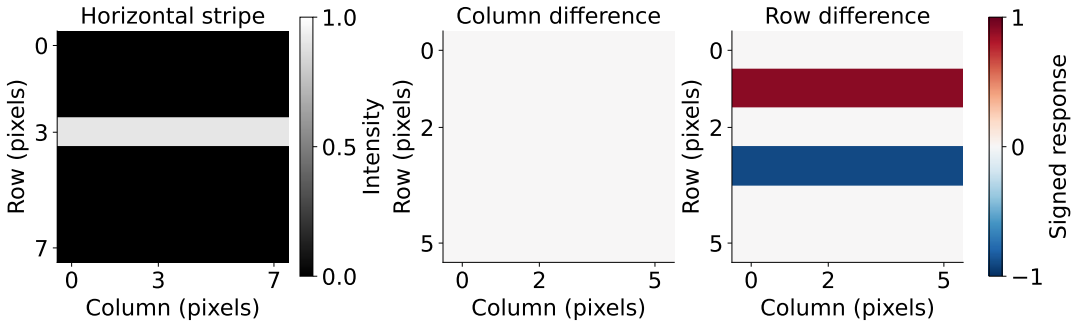
$$\nabla_{w_2} L = H^{\mathsf{T}} D, \qquad D_1 = (D w_2^{\mathsf{T}}) \odot (1 - H^2).$$

$$\nabla_{W_1} L = X_{\text{flat}}^{\mathsf{T}} D_1.$$

Vertical stripe: directional filter responses



Horizontal stripe: directional filter responses



A finite-difference gradient check

For a parameter coordinate θ_j,

$$\partial_j L \approx \frac{L(\theta + \epsilon e_j) - L(\theta - \epsilon e_j)}{2\epsilon}.$$

Use a moderate ϵ: too large gives truncation error, too small gives cancellation error.

Local cross-correlation

$$R_{nijf} = \sum_{u=0}^{2} \sum_{v=0}^{2} X_{n,i+u,j+v,0} K_{uv0f}.$$

The output shape is $(n, 6, 6, F)$ for F filters.

Output size

For input width W, filter width k, padding p, and stride s,

$$W_{\text{out}} = \left\lfloor \frac{W + 2p - k}{s} \right\rfloor + 1.$$

Here $W = 8$, $k = 3$, $p = 0$, $s = 1$, giving 6.

Filter response and orientation

$$K = \frac{1}{3}\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \qquad \sum_{u,v} K_{uv} = 0.$$

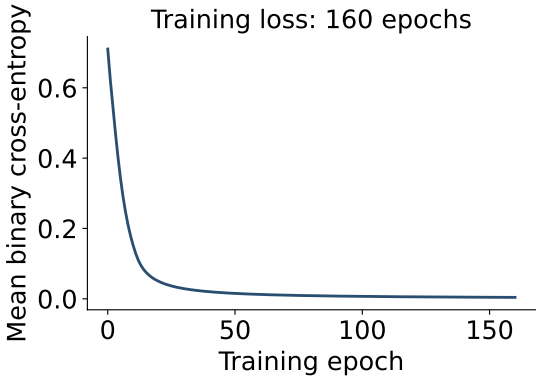
A constant patch has response $c \sum K_{uv} = 0$.

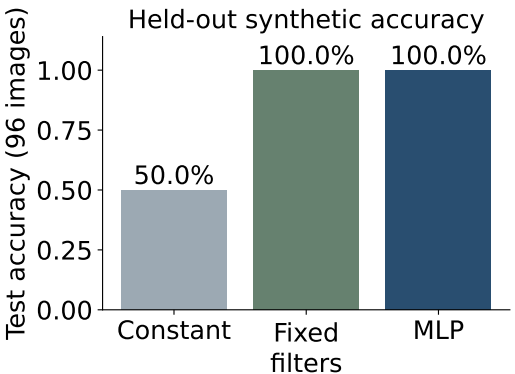
Rectification of a signed response

$$\mathrm{ReLU}(a) = \max(0, a), \qquad \mathrm{ReLU}(-2) = 0, \quad |-2| = 2.$$

ReLU retains positive responses and removes negative responses. Absolute value retains the magnitude of either sign. For a signed edge filter, these operations therefore treat opposite edge orientations differently. The companion's fitted network instead uses $\tanh$.

Learning and baselines





Parameter sharing

The same 3×3 coefficients act at every valid image position. This reuses nine parameters per input/output channel pair. A dense map would assign separate weights to each input coordinate for each output.

Baselines and generalization

The comparison holds the training/test split fixed and evaluates

$$\text{constant rule}, \quad \text{fixed spatial filter}, \quad \text{trained network}.$$

A higher-capacity model is useful only if it improves the relevant task.

Distribution shift

A test set drawn from the same generator checks interpolation within that experiment. A changed generator tests a different claim. Specify which image transformations and noise changes are included before evaluating robustness.