Scientific Computing in Python Lecture 6

Logistic regression and a neural-network classifier

Lecture notes

A binary classifier maps a vector of measured features to a score and then applies a decision threshold. We begin with logistic regression and derive its loss gradient. A small neural network replaces the linear feature map by a nonlinear one. The models are compared on the same held-out observations, after their training choices have been fixed.

1 Logistic scores

Slides for this section

Let $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. Logistic regression forms an affine score $a_i = x_i^\mathsf{T} w + b$ and maps it to $(0, 1)$ using the sigmoid. The inverse relation is the log odds. At threshold one half, the decision boundary is the hyperplane $x^\mathsf{T} w + b = 0$.

$$p_i = \sigma(a_i) = \frac{1}{1 + e^{-a_i}}, \qquad a_i = x_i^\mathsf{T} w + b.$$

$$\log \frac{p_i}{1 - p_i} = a_i.$$

Threshold geometry

$$p_i \geq \frac{1}{2} \iff a_i \geq 0.$$

Changing the threshold changes the decision rule. It does not change the fitted score surface.

2 Bernoulli negative log likelihood

Slides for this section

Assuming conditionally independent labels under the fitted model gives a product of Bernoulli probabilities. Taking minus the logarithm produces a sum. Expressing the loss in terms of logits yields a stable computational form using logaddexp.

$$\ell_i = -y_i \log p_i - (1 - y_i) \log(1 - p_i) = \log(1 + e^{a_i}) - y_i a_i.$$

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

The logit derivative

$$\frac{\partial \ell_i}{\partial a_i} = \sigma(a_i) - y_i = p_i - y_i.$$

$$\nabla_w L = \frac{1}{n} X^\top (p - y).$$

This simple residual drives the parameter update.

3 Gradient descent and regularization

Slides for this section

A gradient step moves opposite the derivative of the objective. Adding a squared weight penalty discourages large coefficients and changes the gradient. The step size controls the numerical optimization and must be distinguished from statistical regularization. A decreasing training loss alone does not demonstrate good prediction on new observations.

$$L(w, b) = \frac{1}{n} \sum_i \ell_i + \frac{\lambda}{2} \|w\|_2^2.$$

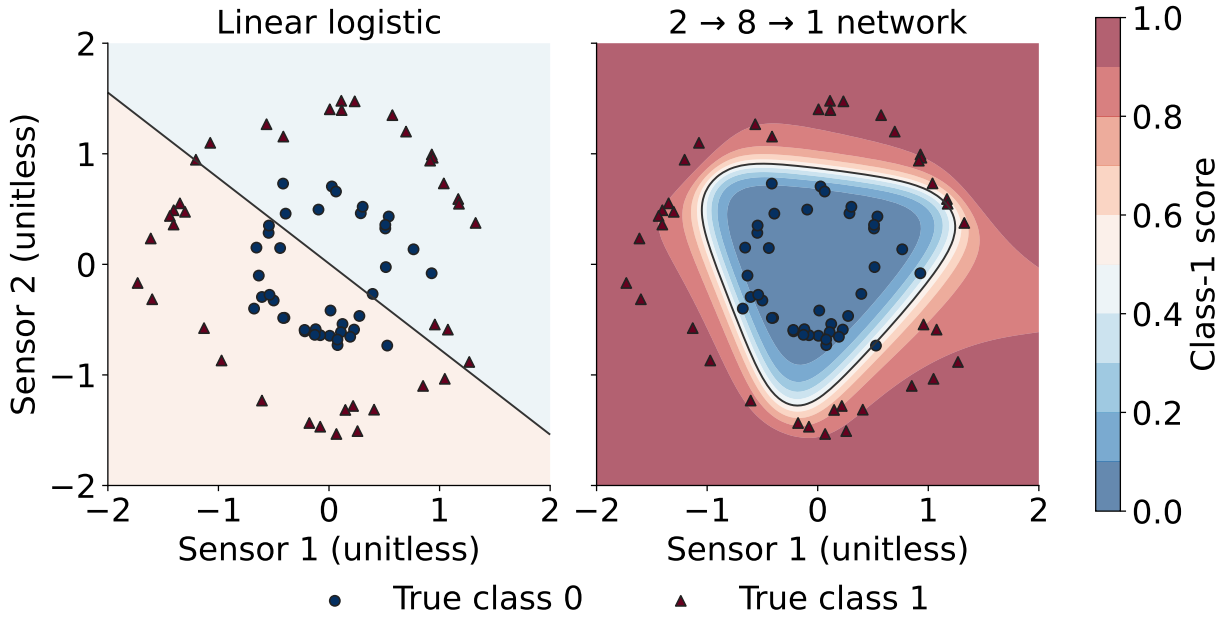
$$w \leftarrow w - \eta \left[\frac{1}{n} X^\top (p - y) + \lambda w \right].$$

Intercept update

With no penalty on the intercept,

$$b \leftarrow b - \eta \frac{1}{n} \sum_i (p_i - y_i).$$

The choice of which parameters receive a penalty belongs to the objective definition.



Both frozen models use the same training subset and scaling. Points are held-out examples, backgrounds are class-one scores and dark curves mark the one-half threshold. A separate marker key identifies true classes: circles for class 0 and triangles for class 1.

4 A nonlinear hidden layer

Slides for this section

For $X \in \mathbb{R}^{n\times d}$, choose $W_1 \in \mathbb{R}^{d\times h}$ and a row bias $b_1 \in \mathbb{R}^h$. Applying tanh entrywise gives $H \in \mathbb{R}^{n\times h}$. A second affine map produces one logit per row. The nonlinearity allows curved decision boundaries; without it, two affine maps collapse to one.

$$H = \tanh(XW_1 + b_1), \qquad a = Hw_2 + b_2, \qquad p = \sigma(a).$$

Shapes: $(n \times d)(d \times h) = n \times h$, then $(n \times h)(h \times 1) = n \times 1$.

Backpropagation through tanh

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

$$\nabla_{w_2} L = H^\mathsf{T} D, \quad D_1 = (Dw_2^\mathsf{T}) \odot (1 - H^2).$$

$$\nabla_{W_1} L = X^\mathsf{T} D_1.$$

Weight-penalty derivatives are added separately.

5 Confusion counts

Slides for this section

A confusion matrix records counts after thresholding scores. Rows denote actual classes and columns predicted classes in this course. Accuracy combines correct decisions; precision conditions on positive predictions; recall conditions on actual positives. Empty denominators require an explicit convention rather than an unreported numerical replacement.

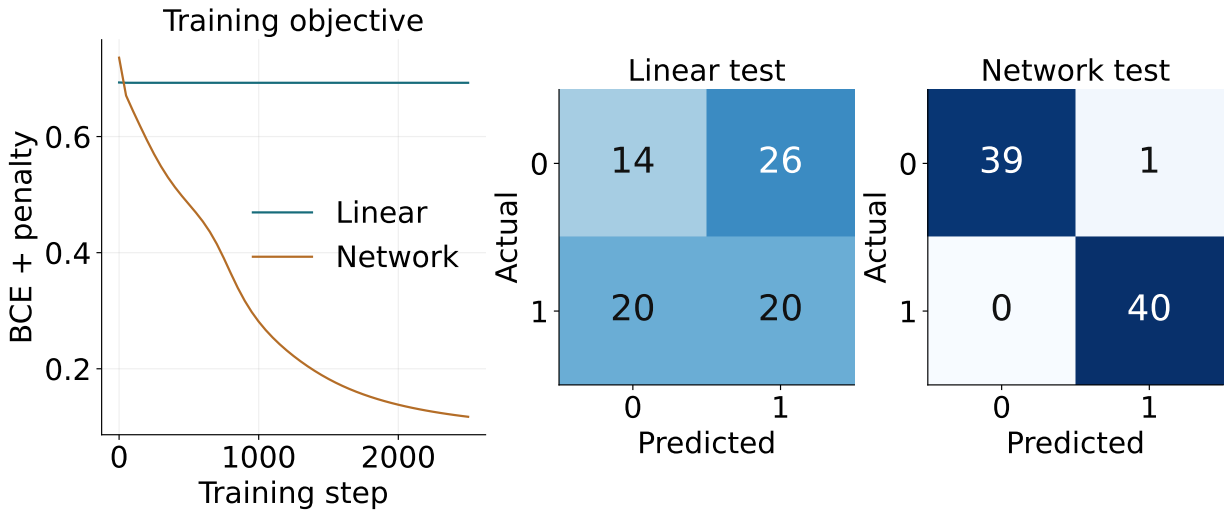
$$\text{accuracy} = \frac{TP + TN}{N}, \quad \text{precision} = \frac{TP}{TP + FP}, \quad \text{recall} = \frac{TP}{TP + FN}.$$

Each denominator specifies a different reference group.

A worked confusion matrix

	$\widehat{y} = 0$	$\widehat{y} = 1$
$y = 0$	30	10
$y = 1$	5	35

Accuracy = 65/80, precision = 35/45, recall = 35/40. This illustrative matrix is separate from the companion's computed result.



Training loss includes the stated weight penalty; confusion matrices use actual classes as rows and predicted classes as columns. Each test class contains 40 examples.

6 A controlled model comparison

Slides for this section

The same train/test split and training-derived standardization are used for both models. Decisions about architecture, penalties, training duration, and threshold precede reading the test outcome. Repeatedly choosing settings from test accuracy converts the test set into a tuning set. Synthetic success applies to the generator used and cannot establish clinical performance.

Both classifiers receive the same feature representation and the same training rows.

The held-out comparison measures the consequences of the fitted functions, not differences in which test observations were selected.

Reproducibility and variability

A fixed seed reproduces one random split and initialization.

It does not quantify variation over other samples. Repeated, properly separated experiments are needed to study that variation.

Exercises

H6-1. Read a confusion matrix

For a new classifier, TN = 36, FP = 4, FN = 9 and TP = 11. New practice; 10 feedback points.
a Write the 2-by-2 matrix with actual classes as rows, give the total sample count and calculate accuracy.

b Calculate precision and recall for class one and explain their denominators.

c Would always predicting class zero attain high accuracy here, and what would it miss?

H6-2. A threshold is a decision rule

Scores are [0.2, 0.4, 0.6, 0.8] and actual labels are [0, 1, 0, 1]. Predict one when score is at least the threshold. New practice; 10 feedback points.
a At threshold 0.5, give predicted labels and TN, FP, FN, TP.

b At threshold 0.4, give accuracy, precision and recall.

c If these were final test data, could selecting 0.4 because of this result support a fresh unbiased test claim?

H6-3. Trace a forward pass

Use the lesson network with two inputs, eight hidden tanh units and one output score. New practice; 10 feedback points.
a For a batch of five examples, state the shapes of X, W1, hidden values, W2 and output scores, and count the trainable numbers.

b In a separate two-hidden-unit illustration, hidden values are [0.5, -0.25], output weights [2, -4] and output bias -1. Find the logit and the label at threshold 0.5; sigmoid(1) is approximately 0.731.

c Explain what happens to the possible decision boundary if every hidden tanh is replaced by the identity function.

H6-4. Audit a comparison claim

A report standardizes all data, tries ten network widths, keeps the best test score and claims that neural networks are always necessary for the ring problem. New practice; 10 feedback points.
a Identify two distinct ways held-out information entered the reported workflow.

b Describe a corrected sequence for a final comparison.

c Explain how the radius-based comparator limits the claim that a neural network is necessary.

References

Original Python Lecture 5. PDF pages 5–6.
https://zanahmad.com/files/scientific-python/Lecture_5__Intro_to_Python.pdf

scikit-learn supervised neural-network guide. Sections 1.17.1 and 1.17.5–1.17.7: MLP, mathematical formulation and scaling.
https://scikit-learn.org/stable/modules/neural_networks_supervised.html

scikit-learn model-evaluation guide. Binary classification metrics: confusion counts, precision and recall.
https://scikit-learn.org/stable/modules/model_evaluation.html

scikit-learn common pitfalls. Sections 12.1–12.2.1: consistent preprocessing and training-only fitting.
https://scikit-learn.org/stable/common_pitfalls.html#data-leakage

Python source materials: Phillip Kerger (2023). Course adaptation and additional exposition: Zan Ahmad.