Scientific Computing in Python Lecture 6

Logistic regression and a neural-network classifier

Zan Ahmad

Winter 2026

Logistic scores

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

Logistic scores

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

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

Threshold geometry

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

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.

Bernoulli negative log likelihood

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

Bernoulli negative log likelihood

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

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

The logit derivative

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

The logit derivative

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

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

This simple residual drives the parameter update.

Gradient descent and regularization

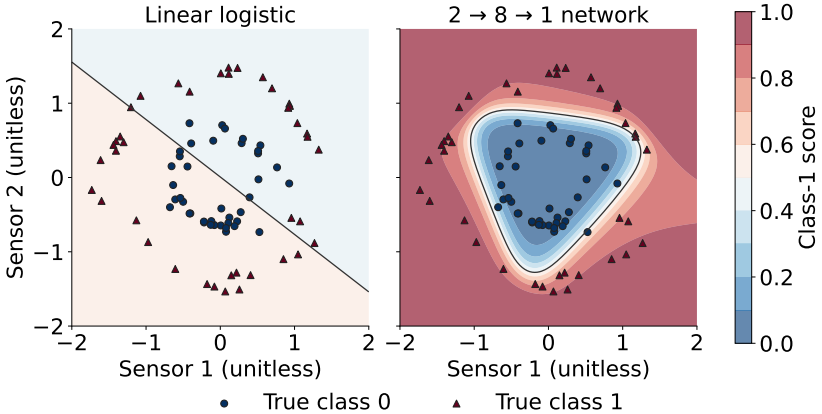
$$L(w, b) = \frac{1}{n} \sum_i \ell_i + \frac{\lambda}{2} \|w\|_2^2.$$

Gradient descent and regularization

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

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

Linear and nonlinear decision boundaries



Intercept update

With no penalty on the intercept,

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

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.

A nonlinear hidden layer

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

A nonlinear hidden layer

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

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

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

Backpropagation through tanh

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

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

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

Weight-penalty derivatives are added separately.

Confusion counts

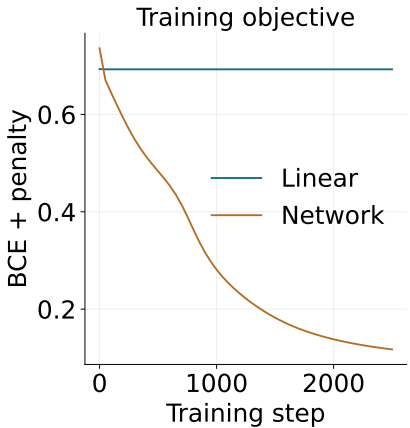
$$\text{accuracy} = \frac{TP + TN}{N}, \quad \text{precision} = \frac{TP}{TP + FP}, \quad \text{recall} = \frac{TP}{TP + FN}.$$

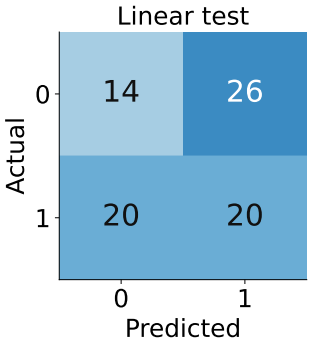
Confusion counts

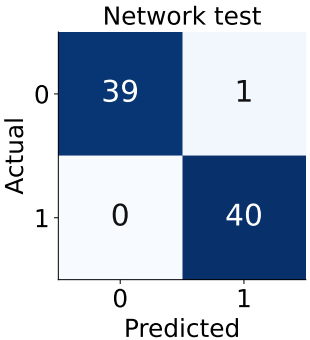
$$\text{accuracy} = \frac{TP + TN}{N}, \quad \text{precision} = \frac{TP}{TP + FP}, \quad \text{recall} = \frac{TP}{TP + FN}.$$

Each denominator specifies a different reference group.

Training loss and held-out counts







A worked confusion matrix

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

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.

A controlled model comparison

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

A controlled model comparison

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.

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.