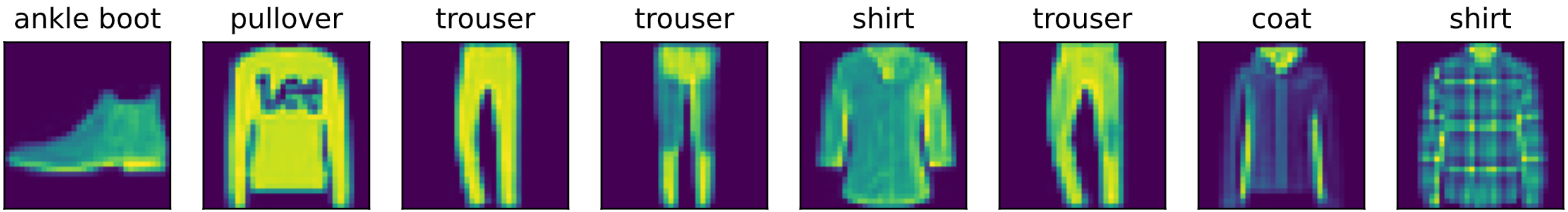
DSCI 565: LINEAR NEURAL NETWORKS FOR CLASSIFICATION

Ke-Thia Yao

Lecture 4: 2025 September 8

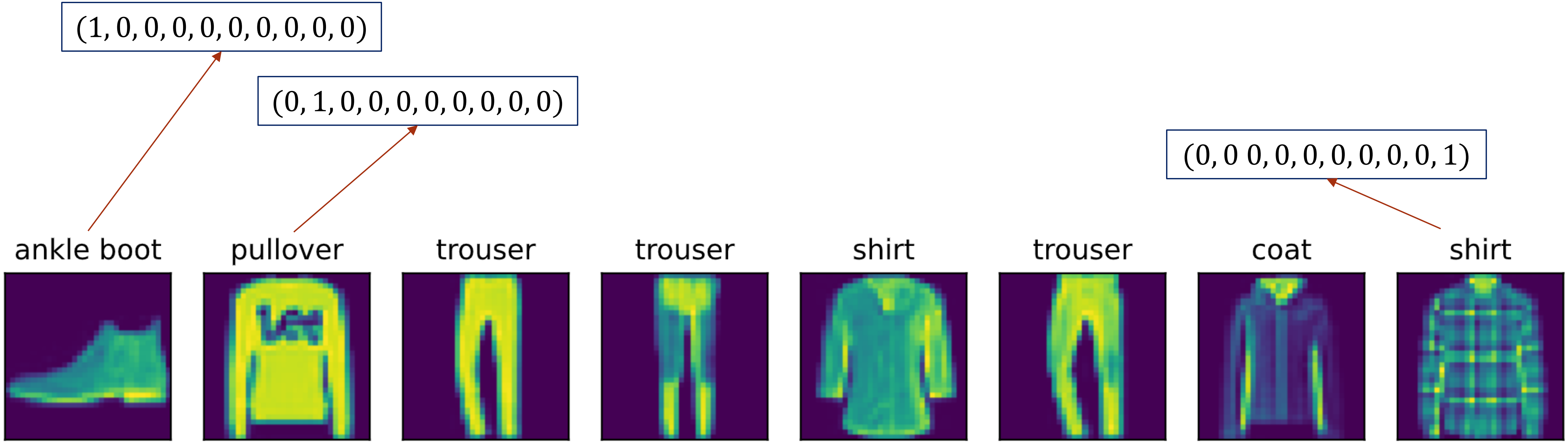
Classification

- Given a training set with labels
- Predict the label for a new instance that is not in the training set



One-hot Encoding

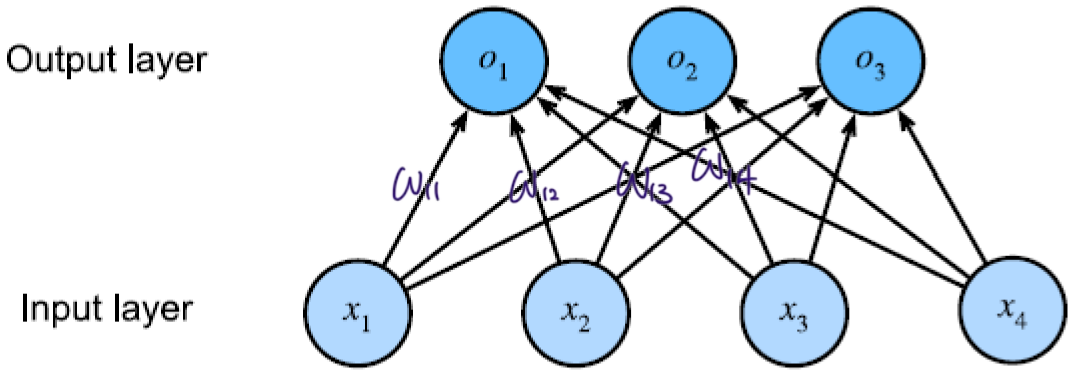
- Use unit vectors to represent labels
- Dimension of the unit vector is equal to the number of unique labels



Linear Model

- Fully connected layer
- One output node for each unique label
- One input node for each pixel

$$\boldsymbol{o} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$



$$\begin{aligned}
o_1 &= x_1 w_{11} + x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + b_1,\\
o_2 &= x_1 w_{21} + x_2 w_{22} + x_3 w_{23} + x_4 w_{24} + b_2,\\
o_3 &= x_1 w_{31} + x_2 w_{32} + x_3 w_{33} + x_4 w_{34} + b_3.
\end{aligned}$$

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ \vdots & & & \vdots \\ w_{31} & \cdots & & w_{34} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_4 \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_4 \end{pmatrix}$$

$$W\vec{x} + \vec{b}$$

Softmax

- Problem with simply using linear functions
  - There is no guarantee that the outputs o_i sum up to 1 in the way we expect probabilities to behave
  - There is no guarantee that the outputs o_i are even nonnegative, even if their outputs sum up to 1, or that they do not exceed 1
- Softmax function

Exponentiation to ensure nonnegative value

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o}) \quad \text{where} \quad \hat{y}_i = \frac{\exp(o_i)}{\sum_j \exp(o_j)}$$

Normalize to sum to 1

non-negative $\forall i, \hat{y}_i \geq 0$

sum to 1 $\sum_i \hat{y}_i = 1$

o_i : logit

Vectorization

- Given minibatch $\mathbf{X} \in \mathbb{R}^{n \times d}$ of n examples d dimensions
- Weights $\mathbf{W} \in \mathbb{R}^{d \times q}$ and bias $\mathbf{b} \in \mathbb{R}^{1 \times q}$, where q is # distinct labels

$$\mathbf{O} = \mathbf{X}\mathbf{W} + \mathbf{b},$$
$$\hat{\mathbf{Y}} = \mathrm{softmax}(\mathbf{O})$$

Cross-Entropy Loss Function

- The output of softmax can be interpreted as a probability
- Given dataset $\mathbf{X}$ and labels $\mathbf{Y}$

$$P(\mathbf{Y} \mid \mathbf{X}) = \prod_{i=1}^{n} P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$$

independent identically distributed (IID) assumption

- Negative log-likelihood

log to avoid multiplying small numbers; negative to minimize loss

$$-\log P(\mathbf{Y} \mid \mathbf{X}) = \sum_{i=1}^{n} -\log P(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)}) = \sum_{i=1}^{n} l(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}),$$

- Cross entropy loss function over q classes is defined as

$$l(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{j=1}^{q} y_j \log \hat{y}_j.$$

With one-hot encoding only one term is non-zero

Cross-Entropy Loss Function

$\hat{y}_i = \frac{\exp(o_i)}{\sum_{j=1}^{k} \exp(o_j)}$

- Cross-entropy loss drives the softmax output toward the ground truth

$$l(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{j=1}^{q} y_j \log \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)}$$

$$= \sum_{j=1}^{q} y_j \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j$$

$$= \log \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j.$$

Derivative is non-zero, if softmax differs from the ground truth

$$\partial_{o_j} l(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} - y_j = \mathrm{softmax}(\mathbf{o})_j - y_j.$$

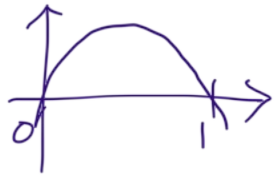
Information Entropy

- Introduce by Claude Shannon in "A Mathematical Theory of Communications" in 1948
- Entropy of a random variable X is

$$H(X) \equiv \sum_x p(x) \log\left(\frac{1}{p(x)}\right) = -\sum_x p(x) \log p(x)$$

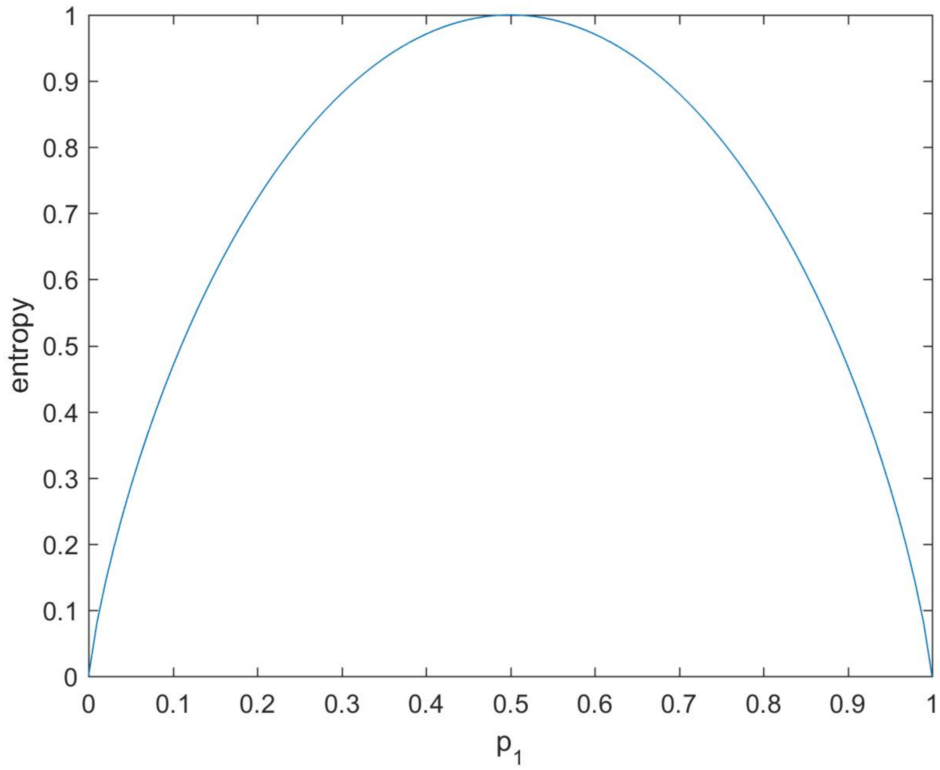
- Entropy is measured in bits
  - The entropy of fair coin is 1 bit
  - The entropy of a coin that always return heads is 0 bit
- A fair coin has more "surprise" or "uncertainty"

$H(X) = -\sum_X p(X) \log p(X)$



Entropy of a Bernoulli Random Variable

└ state $\{H, T\}$ $P(H)=p$, $0<p<1$, $P(T)=1-p$



- $= -p_1 \log_2 p_1 - p_2 \log_2 p_2$
  - define $0\log 0 = 0$
- Entropy maximized if
  $p_1 = p_2 = 0.5$
- Entropy minimized if
  $p_1 = 0$ and $p_2 = 1$,
  or $p_1 = 1$ and $p_2 = 0$
- Entropy measures randomness, impurity, surprisal

Cross-Entropy

- The cross-entropy from P to Q is the expected surprisal of an observer with subjective probabilities Q upon seeing data that was actually generated according to probabilities P

$$H(P,Q) = \sum_x -P(x) \log Q(x)$$

- The lowest possible value for cross-entropy is when $P = Q$, i.e.,

$$H(P,P) = H(P)$$

- Cross-entropy objective (make Q more like P) can be thought of as
  - Maximizing the likelihood of observable data
  - Minimizing the surprisal to communicate the label

Notebooks

- Image Dataset: chapter_linear-classification/image-classification-dataset.ipynb
- Base classifier class: chapter_linear-classification/classification.ipynb
- Softmax regression from scratch: chapter_linear-classification/softmax-regression-scratch.ipynb
- Concise Softmax regression: chapter_linear-classification/softmax-regression-concise.ipynb