Week 4, Part 2:
Prediction, Learning, and Cross-Entropy Loss

Prof. Michael Gastpar

Slides by Prof. M. Gastpar

Spring Semester 2025

Outline

In today's lecture, we explore the role of entropy in prediction and learning problems.

Example : Classify Images

Example : Classify Images

Image

Neural Network

Label

Example : Classify Images

Image

Neural Network

Label

Label
Ibex
Kangaroo
Lynx
Wombat
Dog
Cat
Turtle
Dolphin
Elephant
Kookaburra
Other

IS OUR "NEURAL NETWORK" PERFORMING WELL ?

IMAGE: x

OUR MACHINE (NEURAL NETWORK) OUTPUTS $Q(x)$

TRUE LABEL: $\text{Label}(x)$

"ZERO-ONE Loss":

$$\mathbb{1}\{Q(x) \neq \text{Label}(x)\}$$

$$= \begin{cases} 0, & \text{if } Q(x) = \text{Label}(x) \\ 1, & \text{if } Q(x) \neq \text{Label}(x) \end{cases}$$






)

CLASSIFICATION ERROR:

$$\frac{\sum_x \mathbb{1}\{Q(x) \neq \text{label}(x)\}}{\text{number of images}}$$

is the fraction of mis-labeled images.

WHAT'S TO LIKE?

- VERY INTUITIVE
- INTERPRETABLE

WHAT'S NOT TO LIKE?

- NOT DIFFERENTIABLE

Example : Classify Images

Image

Neural Network

Label

Label	*Probability*
Ibex	0.98
Kangaroo	0.005
Lynx	0.002
Wombat	0.002
Dog	0.001
Cat	0.001
Turtle	0.001
Dolphin	0.001
Elephant	0.001
Kookaburra	0.001
Other	0.005

Example : Classify Images

- Our Neural Network produces also called Q in these slides.

$$P_{machine}(label|image).$$

- The true label distribution is

$$P_{true}(label|image) = \begin{cases} 1, & \text{correct label,} \\ 0, & \text{wrong label.} \end{cases}$$

(assuming for simplicity that for each image, there is a single correct label).

- Ideally, we would like

$$P_{machine}(label|image) = P_{true}(label|image)$$

for *every* pair $(image, label)$.

- Clearly, this is not going to happen in the real world!

Example : Classify Images

- Instead, people like to consider **cross entropy loss**.
- That is, we wish for our $P_{machine}(label|image)$ to **minimize**

$$\begin{aligned} & L(P_{true}(label|image), P_{machine}(label|image)) \\ & = -\sum_{label} P_{true}(label|image) \log_D P_{machine}(label|image) \end{aligned}$$

- Given training data $(image_i, label_i)$, for $i = 1, 2, \ldots, n$, we select $P_{machine}(label|image)$ to minimize the cross entropy loss.

Cross Entropy Loss

- Cross Entropy Loss:

$$L(P, Q) = -\sum_{y} P(y) \log_D Q(y).$$

where

- P is the true distribution
- Q is our approximation (via the neural network).

Why is it popular?

- Good properties for training with "gradient descent" in certain standard architectures.
- Theoretical properties.

We will now discuss these in turn.

IMAGE CLASSIFICATION

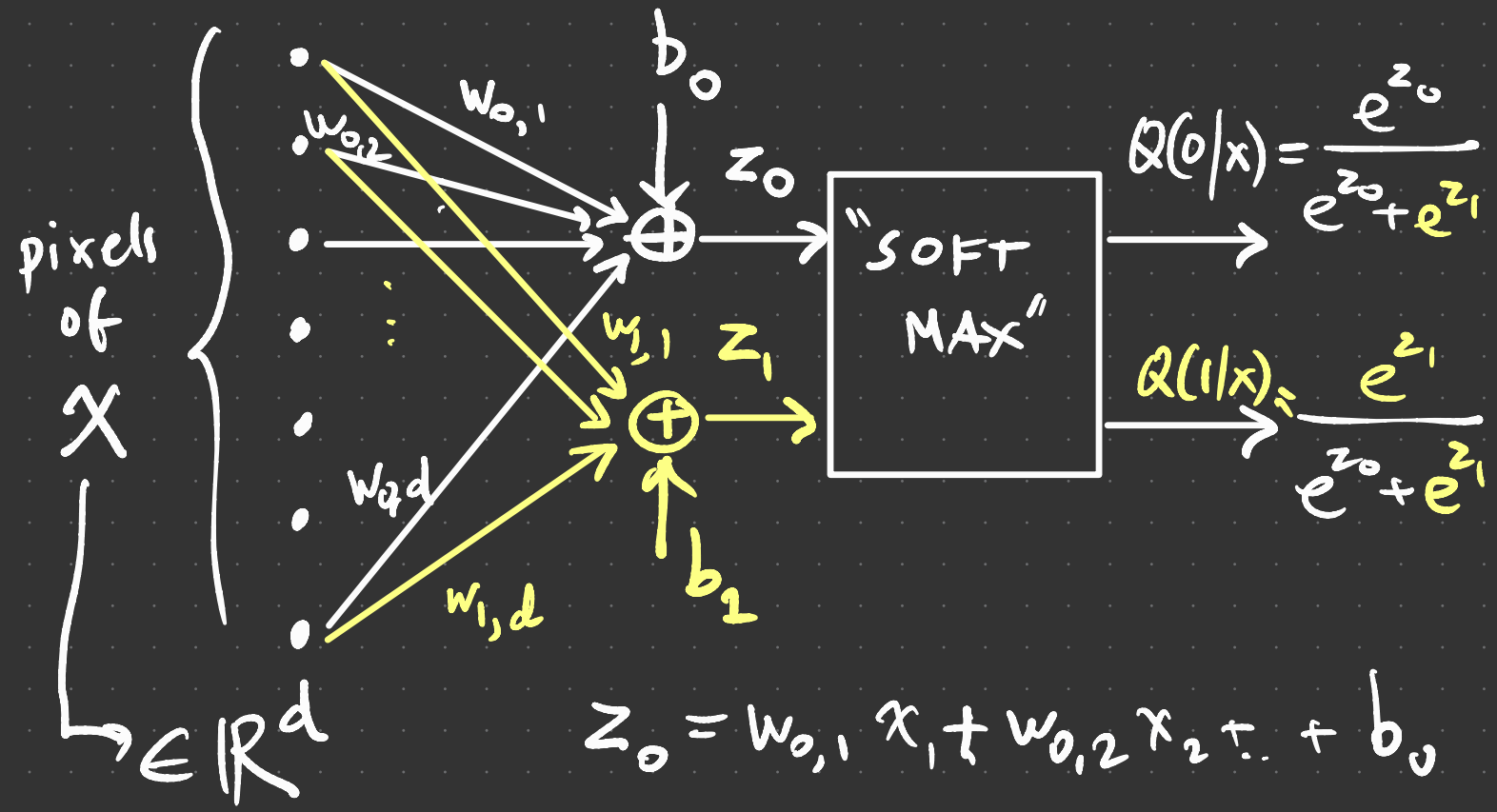
LABEL "0"



LABEL "1"



A (VERY) SIMPLE NEURAL NET



$$z_0 = w_{0,1}x_1 + w_{0,2}x_2 + \dots + b_0$$

GOAL :






GOAL:

GIVEN TRAINING DATA, $\longrightarrow$

SELECT:

w_0, b_0, w_1, b_1

CLEVERLY!

FOR US:

SELECT

W_0, b_0, W_1, b_1

SUCH AS TO MINIMIZE

CROSS-ENTROPY LOSS

FOR A SINGLE IMAGE x:

$$L\left(P(y|x), Q(y|x)\right)$$

$$= -\sum_{y} P(y|x) \log Q(y|x)$$

$$= -P(0|x) \log Q(0|x)$$

$$- P(1|x) \log Q(1|x)$$

FOR A SINGLE IMAGE x:

$$L\left(P(y|x), Q(y|x)\right)$$

$$= \sum_y P(y|x) \log Q(y|x)$$

$$= -P(0|x) \log Q(0|x) - P(1|x) \log Q(1|x)$$

$x \in \mathbb{R}^d$ ← total # of pixels

$w_0 \in \mathbb{R}^d$ $\quad$ $w_1 \in \mathbb{R}^d$

$$Q(0|x) = \frac{e^{z_0}}{e^{z_0} + e^{z_1}} = \frac{e^{x \cdot w_0 + b_0}}{e^{x \cdot w_0 + b_0} + e^{x \cdot w_1 + b_1}}$$

$$L\left(P(y|x), Q(y|x)\right)$$

$$= -P(0|x) \log Q(0|x) - P(1|x) \log Q(1|x)$$

$$= -P(0|x) \log \frac{e^{x \cdot w_0 + b_0}}{e^{x \cdot w_0 + b_0} + e^{x \cdot w_1 + b_1}}$$

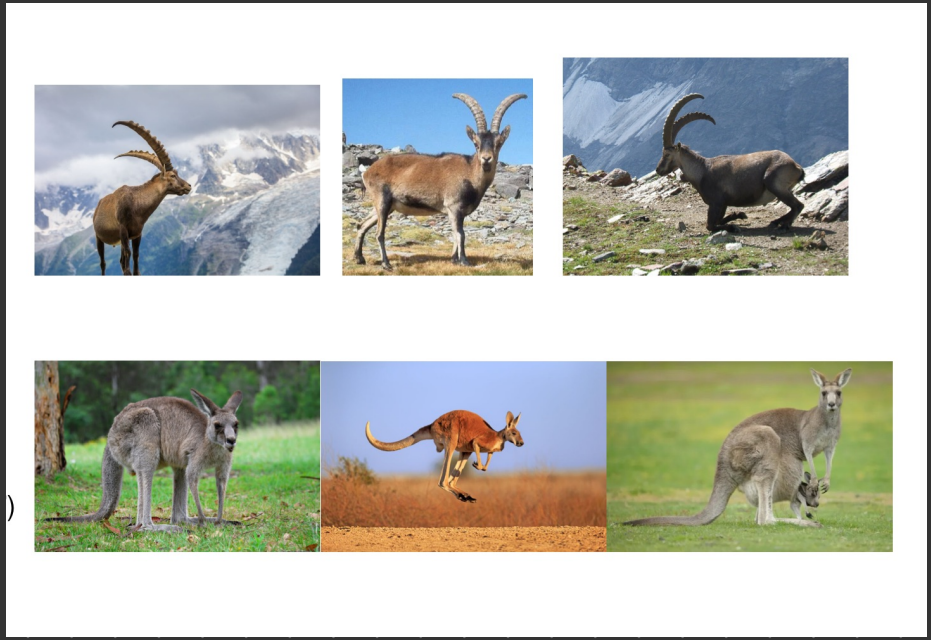
$$- P(1|x) \log \frac{e^{x \cdot w_1 + b_1}}{e^{x \cdot w_0 + b_0} + e^{x \cdot w_1 + b_1}}$$

$$L\left(P(y|x), Q(y|x)\right)$$

$$= \begin{cases} \log \dfrac{e^{x \cdot w_0 + b_0}}{e^{x \cdot w_0 + b_0} + e^{x \cdot w_1 + b_1}}, & \text{if } x \text{ is Ibex} \\ \log \dfrac{e^{x \cdot w_1 + b_1}}{e^{x \cdot w_0 + b_0} + e^{x \cdot w_1 + b_1}}, & \text{if } x \text{ is Kangaroo} \end{cases}$$

BACK TO THE FULL TRAINING SET

NOW, ADD UP THE CROSS-ENTROPY LOSS OVER THE TRAINING SET.



TOTAL LOSS:

$$L_{total}(w_0, b_0, w_1, b_1) =$$

$$= -\sum_{\text{ibexes}} \log \frac{e^{x_i \cdot w_0 + b_0}}{e^{x_i \cdot w_0 + b_0} + e^{x_i \cdot w_1 + b_1}}$$

$$-\sum_{\text{kangaroos}} \log \frac{e^{x_j \cdot w_1 + b_1}}{e^{x_j \cdot w_0 + b_0} + e^{x_j \cdot w_1 + b_1}}$$

IDEA # 1:

SELECT

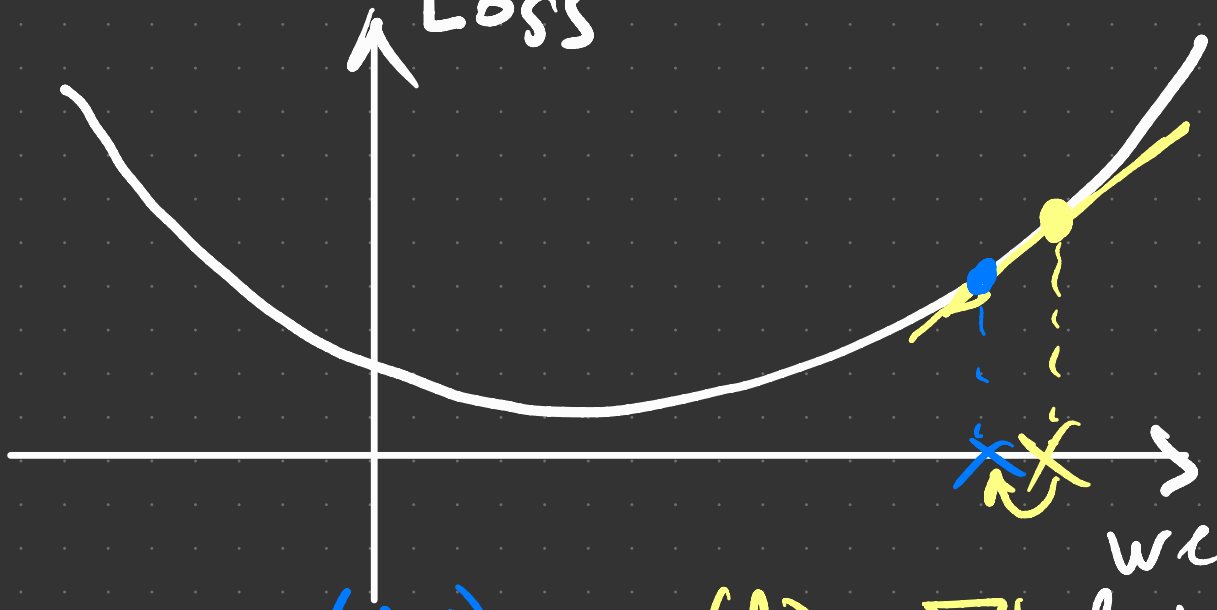
$$w_0, b_0, w_1, b_1$$

TO MINIMIZE

$$L_{total}(w_0, b_0, w_1, b_1)$$

IDEA #2

"GRADIENT DESCENT"



$$w_0(k+1) = w_0(k) - \mu \nabla L_{\tau(w_0(k))}$$ & biases

IN EITHER CASE, WE WANT

DERIVATIVES

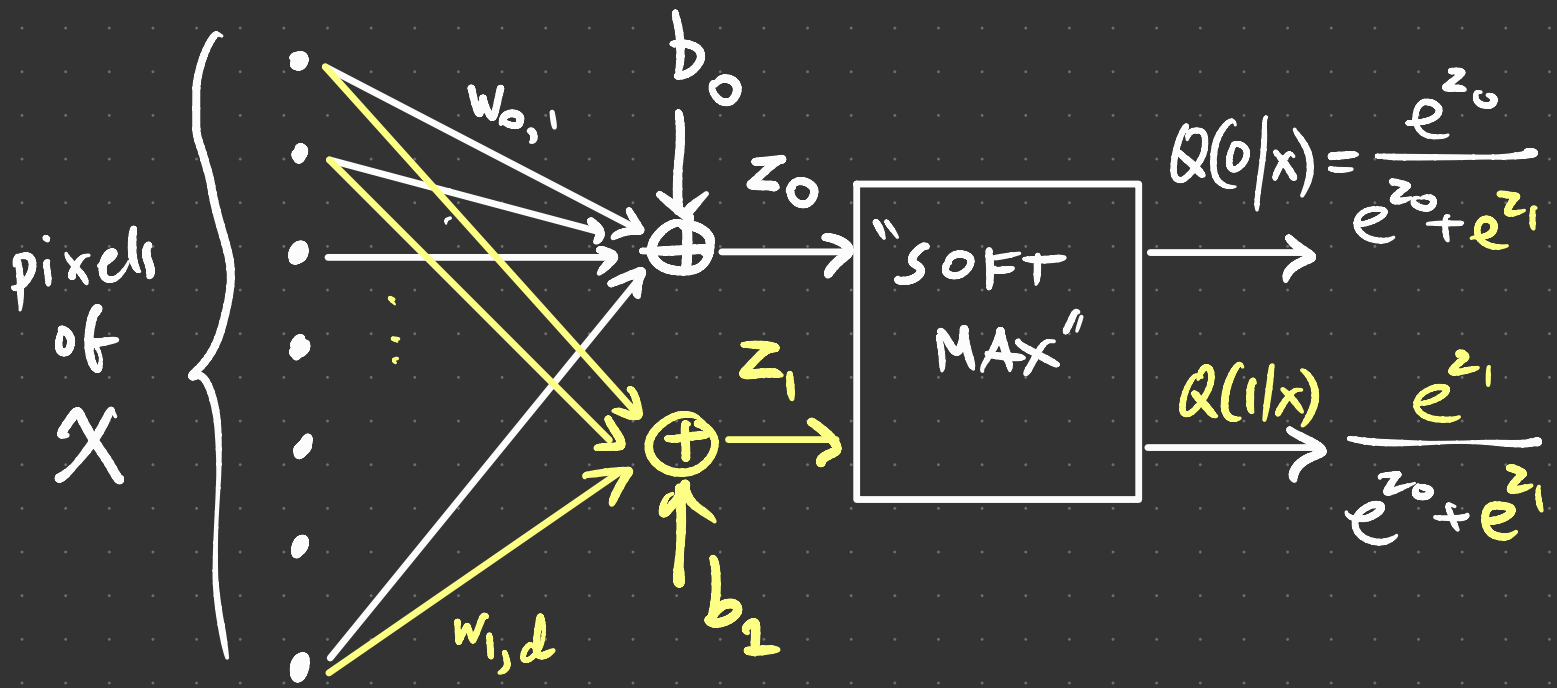
OF $L(w_0, b_0, w_1, b_1)$.

$\longrightarrow$ SIMPLE FOR LOG-LOSS!

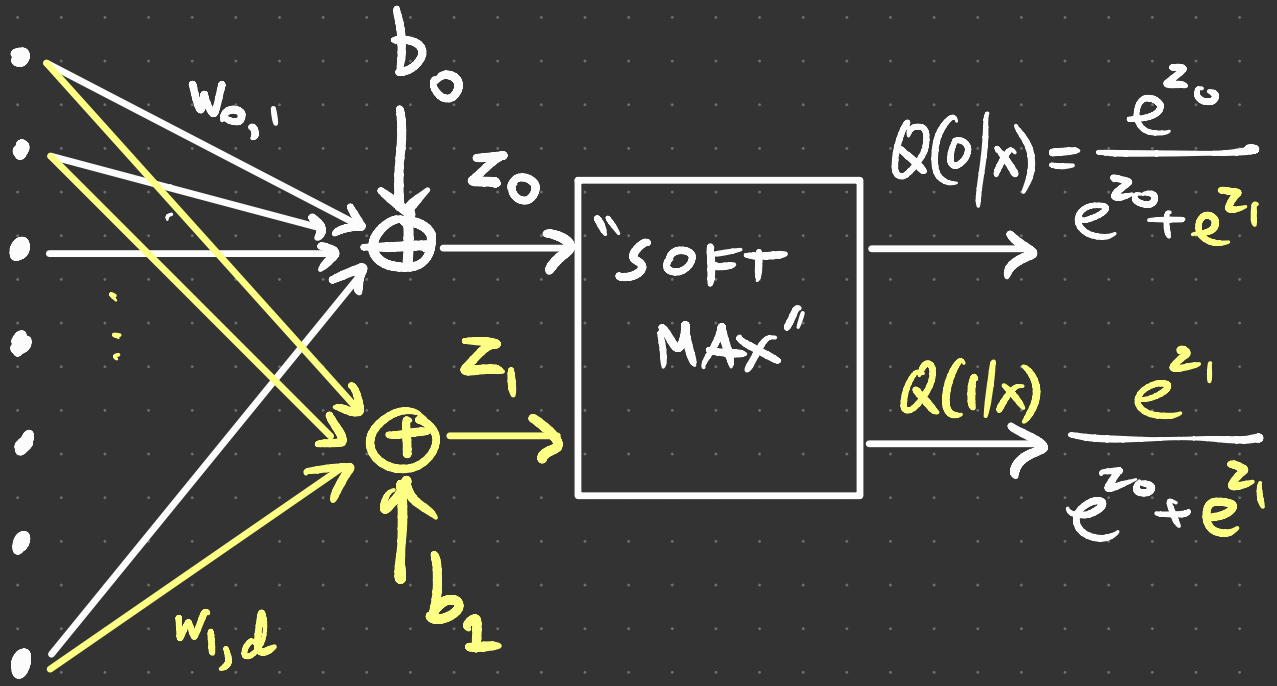
SEE HOMEWORK 4!

SOME FOLLOW-UPS ...

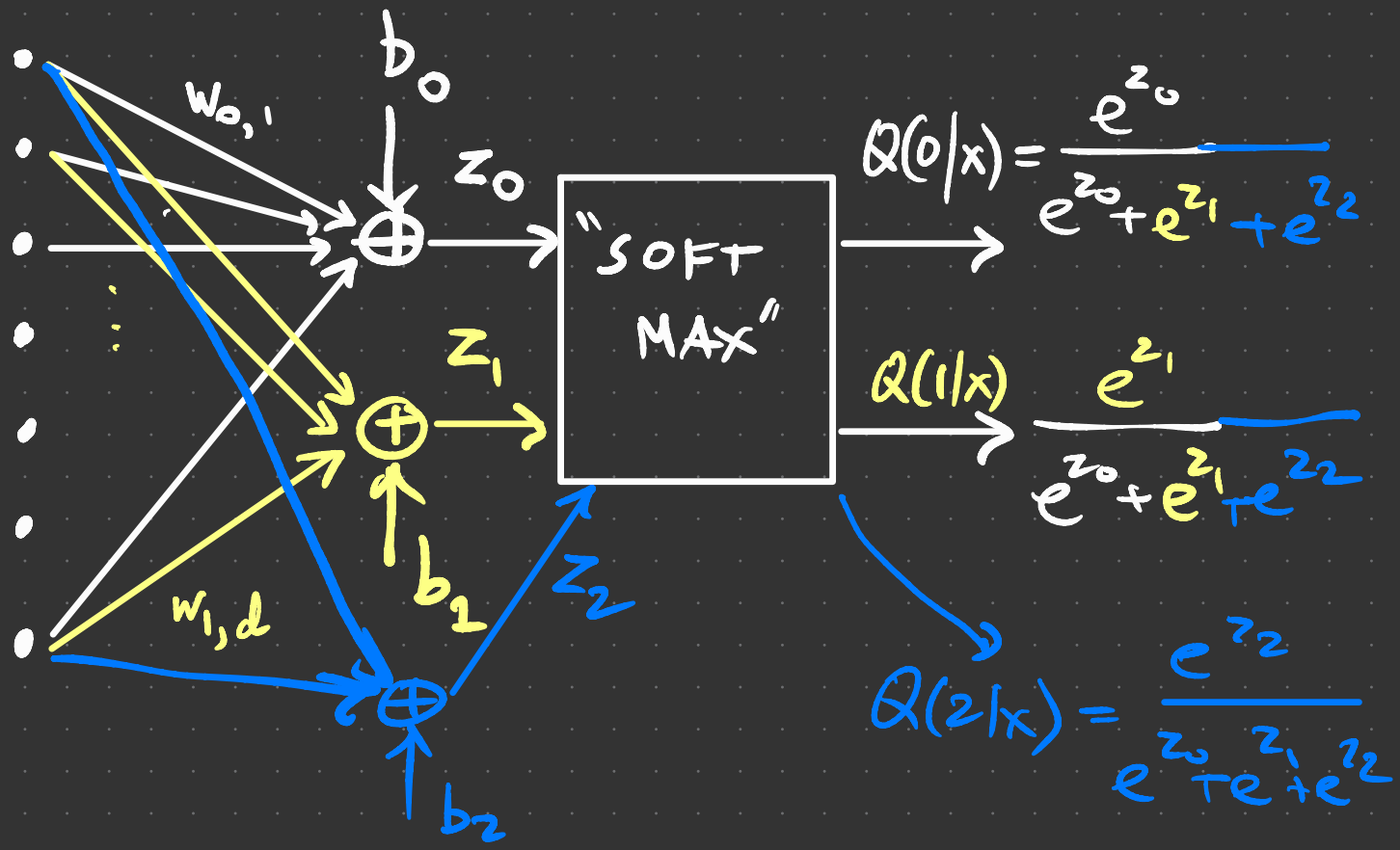
A (VERY) SIMPLE NEURAL NET



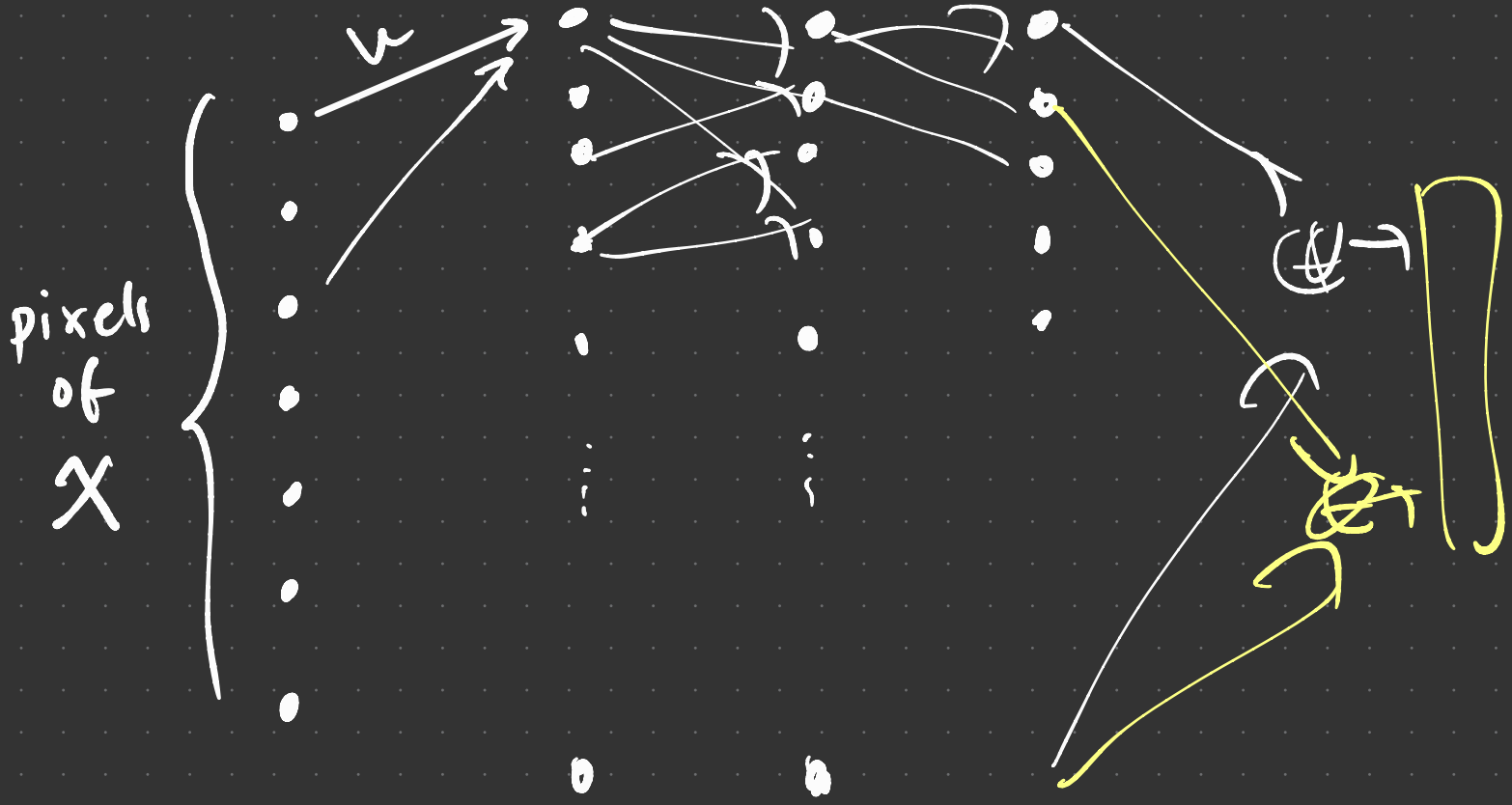
• MORE THAN 2 CLASSES ?



• MORE THAN 2 CLASSES ?



"DEEP LEARNING" ?



Example : Classify Images, Training with cross entropy loss

- The Neural Network takes in an image. Let us call this x.
- It outputs a label distribution $Q(y|x)$ over the set of labels.
- Let us restrict to just two labels. Only "Ibex" ($y = 0$) and "Kangaroo" ($y = 1$).
- In simplified terms, the Neural Network outputs:

$$Q(y = 0|x) = \frac{e^{z_0}}{e^{z_0} + e^{z_1}},$$
$$Q(y = 1|x) = \frac{e^{z_1}}{e^{z_0} + e^{z_1}} = 1 - Q(y = 0|x).$$

where

$$z_0 = w_0 x + b_0$$
$$z_1 = w_1 x + b_1$$

where w_0 and w_1 are called *weights* and b_0 and b_1 are called *biases*.

- The key is to select the weights and biases cleverly.
- This is done by *training* with data.

EXAMPLE : CLASSIFY IMAGES, TRAINING WITH CROSS ENTROPY LOSS

Label = 0
("Ibex")



Label = 1
("Kangaroo")



EXAMPLE : CLASSIFY IMAGES, TRAINING WITH CROSS ENTROPY LOSS

- For fixed weights and biases, calculate the loss over all n training samples:

MINUS SIGN MISSING

$$L_{training} = \sum_{i=1}^{n} \left(P(y=0|x_i) \log_D \frac{e^{z_{0,i}}}{e^{z_{0,i}} + e^{z_{1,i}}} + P(y=1|x_i) \log_D \frac{e^{z_{1,i}}}{e^{z_{0,i}} + e^{z_{1,i}}} \right)$$

where $z_{0,i} = w_0 x_i + b_0$ and $z_{1,i} = w_1 x + b_1$.

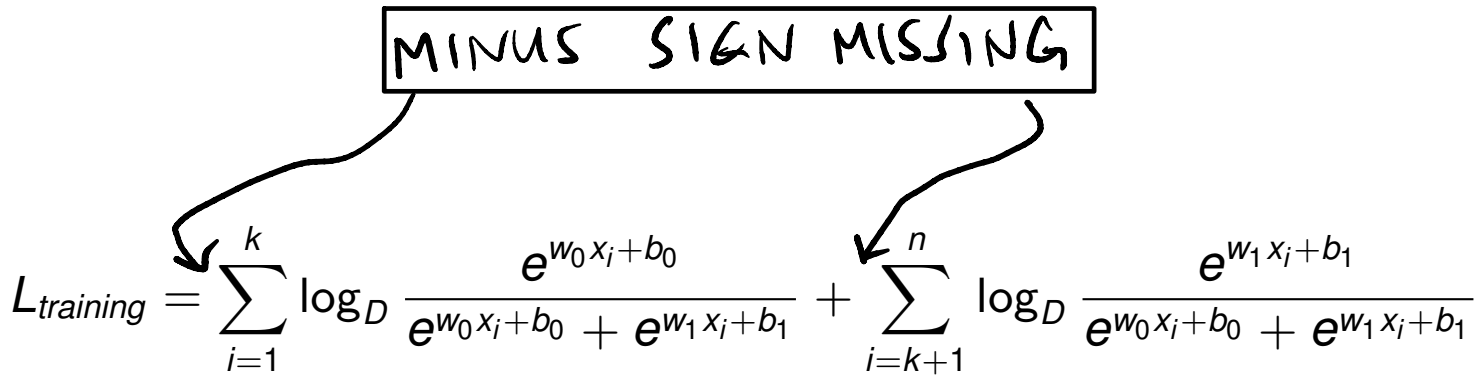
- Suppose images $i = 1, 2, \ldots, k$ are ibexes (label 0), and images $i = k+1, k+2, \ldots, n$ are kangaroos (label 1). Then, we can write

$$L_{training} = \sum_{i=1}^{k} \log_D \frac{e^{w_0 x_i + b_0}}{e^{w_0 x_i + b_0} + e^{w_1 x_i + b_1}} + \sum_{i=k+1}^{n} \log_D \frac{e^{w_1 x_i + b_1}}{e^{w_0 x_i + b_0} + e^{w_1 x_i + b_1}}$$

- Now minimize this over all weights and biases!

MINUS SIGN MISSING

Example : Classify Images, Training with cross entropy loss



- Find gradient (derivative) with respect to weights (and biases).
- Most commonly, *gradient descent* is used.
  - Start with a **random choice** of the weights.
  - Then, proceed in "small" steps against the gradient.

Cross Entropy Loss : Theoretical Properties

- Cross Entropy Loss:

$$L(P, Q) = -\sum_{y} P(y) \log_D Q(y).$$

Theorem

For a fixed probability distribution P, the minimum

$$\min_{Q} L(P, Q)$$

is attained if and only if we select $Q^* = P$, and in this case,

$$L(P, Q^*) = L(P, P) = H(P),$$

where $H(P)$ is the entropy of the probability distribution P.

The proof, which will be done in class, uses once again the "IT inequality."

PROOF:

$$H(P) \leq L(P, Q)$$

w.eq. iff $P = Q$.

Claim:

$$H(P) - L(P, Q) \leq 0$$

$$-\sum_y p(y) \log p(y) + \sum_y p(y) \log Q(y)$$

$$= \sum_y p(y) \log \frac{Q(y)}{P(y)}$$

$$\underset{\text{w.eq. iff}}{\leq} \sum_y p(y) \left[\frac{Q(y)}{P(y)} - 1 \right] \log(e)$$

$$= \sum_y \left(Q(y) - P(y) \right) \log(e)$$

$$= \left\{ \underbrace{\sum_y Q(y)}_{=1} - \underbrace{\sum_y P(y)}_{=1} \right\} \log(e)$$

$$= 0$$

□

NOTE:

KL-DIVERGENCE
(aka KL DISTANCE):

$$D_{KL}(P \| Q) := \sum_y P(y) \log \frac{P(y)}{Q(y)}$$

Fact 1: $D_{KL}(P \| Q) \geq 0$
with equality iff $P = Q$.

Outline

Entropy:

$$H_D(X) = -\sum_x p(x) \log_D p(x).$$

For $D = 2$, we simply write $H(X)$, and we call the unit *bits.*

Entropy has many useful properties, including:

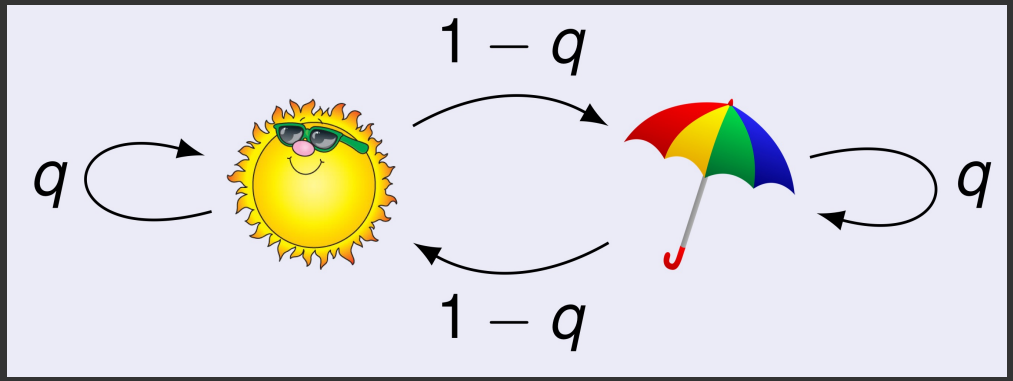
- $0 \leq H_D(X) \leq \log_D |\mathcal{X}|$
- $H_D(X|Y) \leq H_D(X)$ with equality if and only if X and Y are independent.
- $H_D(X, Y) = H_D(X) + H_D(Y|X)$

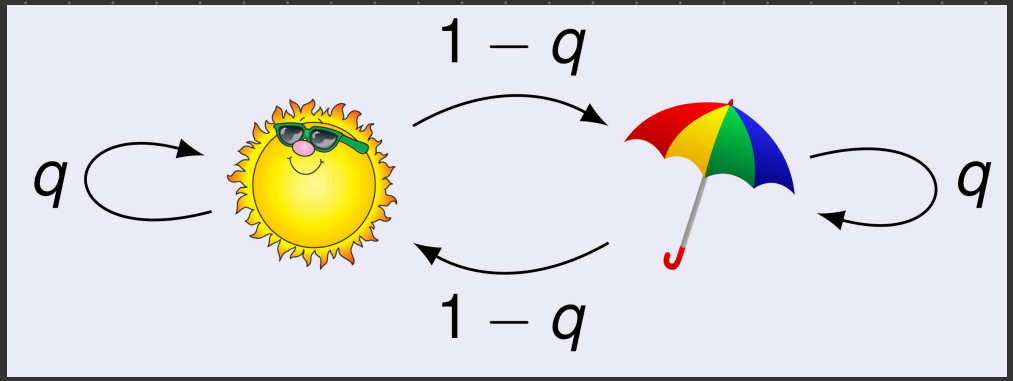
Data Compression:

- Every uniquely decodable binary code must use at least $H(X)$ bits per symbol on average.
- There exists a binary code that uses between $H(X)$ and $H(X) + 1$ bits per symbol on average.
- Hence, for a source string of length n :
  - every uniquely decodable binary code must use at least $H(S_1, S_2, \cdots, S_n)/n$ bits per source symbol, and
  - there exists a binary code that uses between $H(S_1, S_2, \cdots, S_n)/n$ and $H(S_1, S_2, \cdots, S_n)/n + 1/n$ bits per source symbol.

- COIN FLIP

- SUNNY - RAINY





$s_1, \quad s_2, \quad s_3, \ldots$

QUIZ: ARE s_1 AND s_3 INDEPENDENT?

$$p(s_1, s_3) = \sum_{s_2} p(s_1, s_2, s_3)$$

$$= \sum_{s_2} p(s_1)\, p(s_2 | s_1)\, p(s_3 | s_2)$$

Entropy and Algorithms

- We explored examples where entropy can give a lower bound on algorithmic performance.
  - *Example:* in search-type problems, give a lower bound on the minimum number of necessary queries.

Cross-Entropy Loss

- Machine (e.g., Neural Network) outputs a distribution $Q(y)$ over all possible labels.
- Cross-Entropy Loss: Select $Q(y)$ to minimize $L(P, Q) = -\sum_y P(y) \log_D Q(y)$.