COMPUTATIONAL ASTROPHYSICS

Observatorio
Astronómico
Nacional

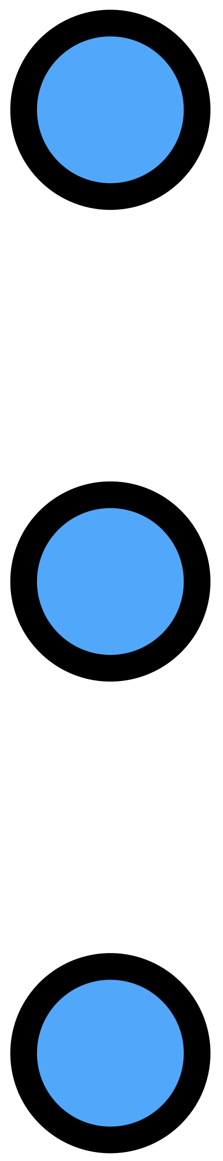
Computational Astrophysics

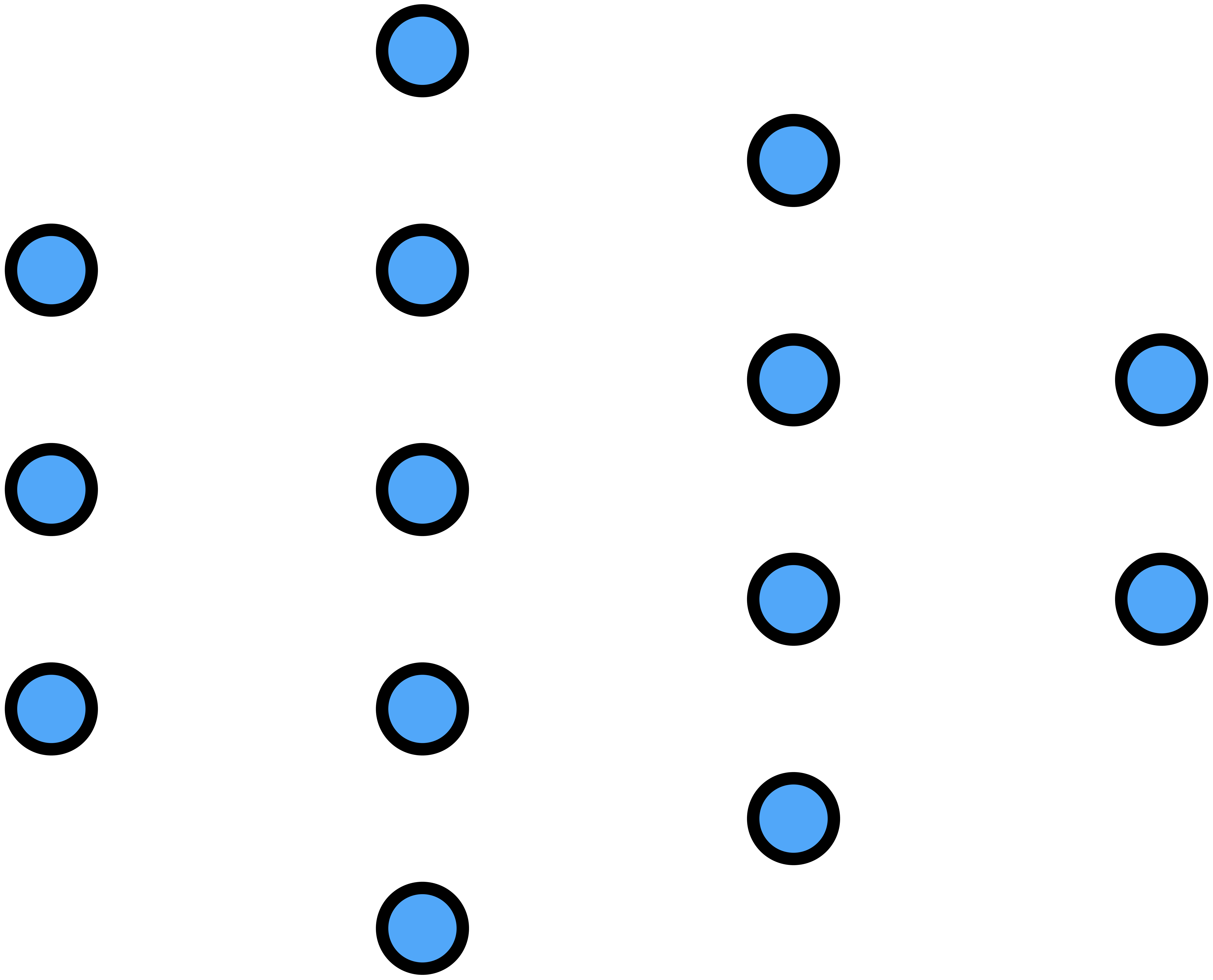
Neural Networks. Classification

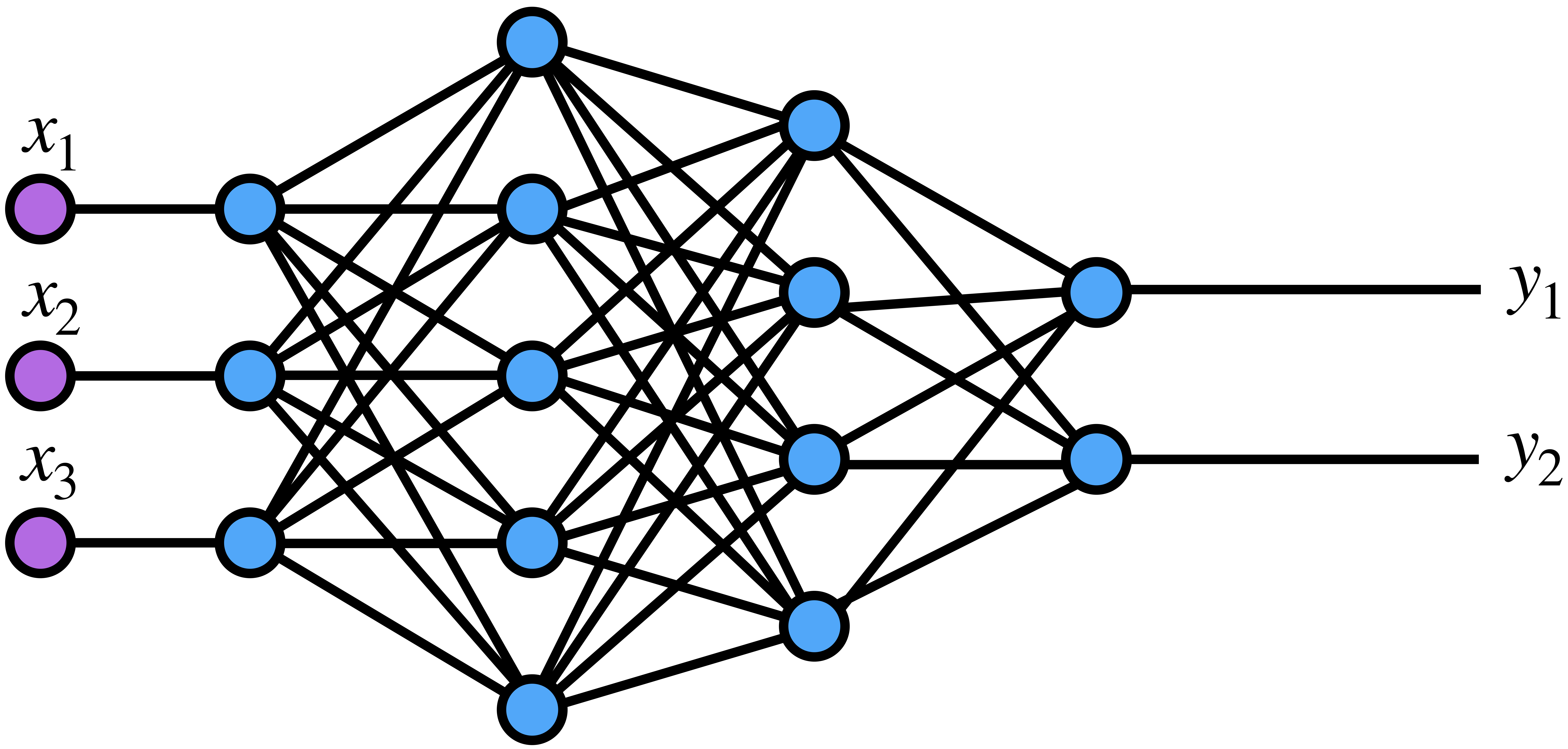
Eduard Larrañaga
Observatorio Astronómico Nacional
Universidad Nacional de Colombia

Classification Neural Networks

Layers

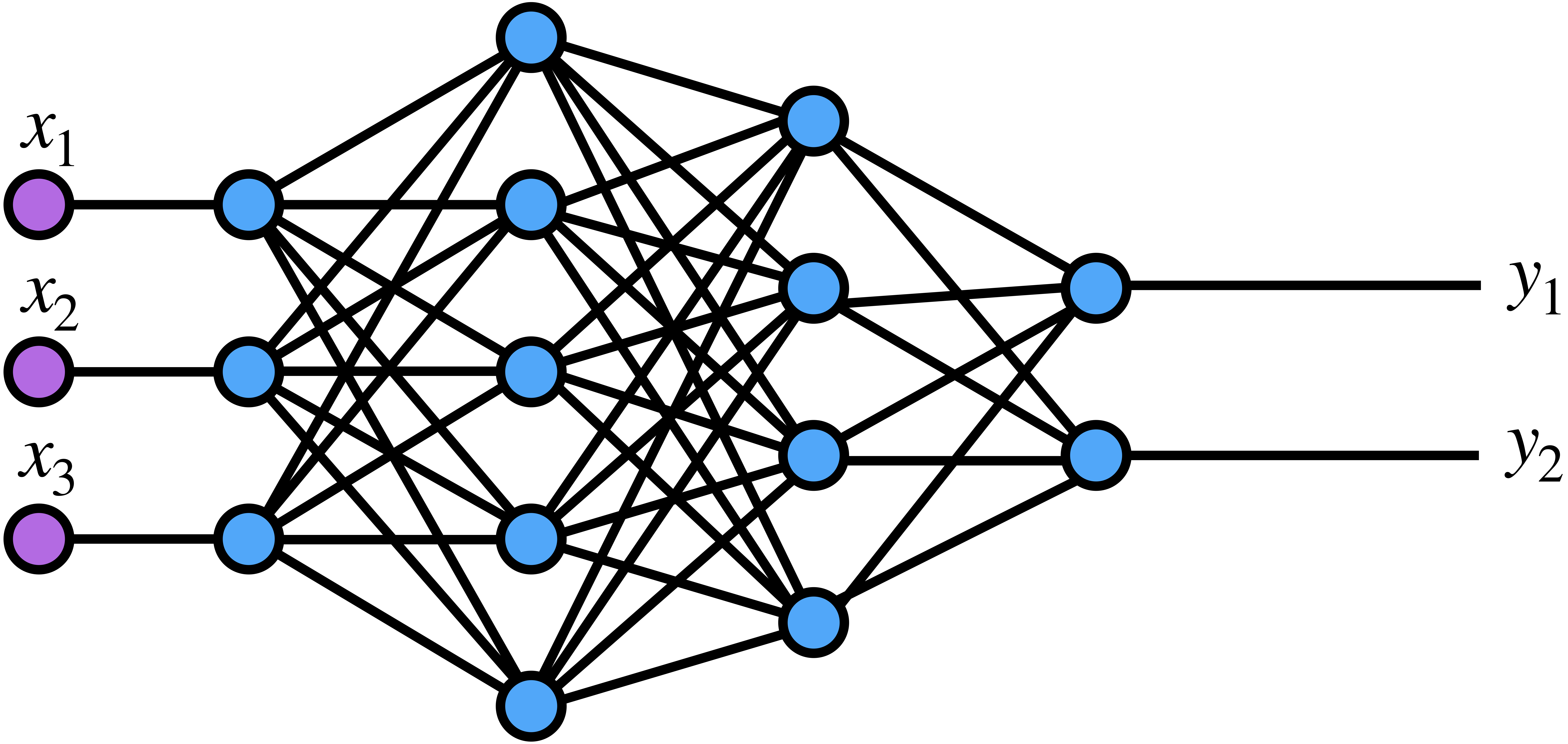






x_1
x_2
x_3
y_1
y_2

Neural Network



Cross Entropy

Information and Surprise

Information quantifies the number of bits required to encode and/or transmit an event.

Lower probability events $\longrightarrow$ More Information (surprising!)

Higher probability events $\longrightarrow$ Less Information (unsurprising)

Quantifying Information

Information $h(x)$ for an event x is calculated in terms of the probability of occurrence $P(x)$.

$$h(x) = -\log\left[P(x)\right]$$

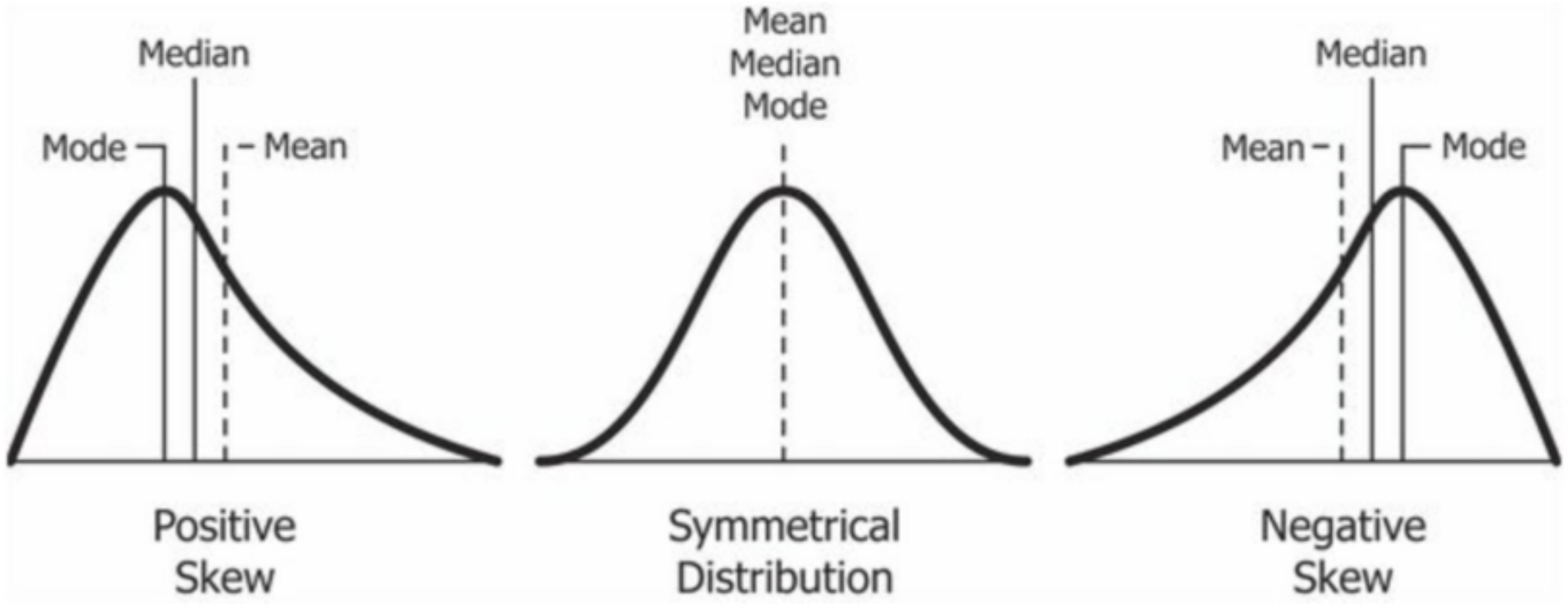
If $P(x) = 1 \longrightarrow h(x) = 0$ (No surprise)

If $P(x) \to 0 \longrightarrow h(x) \to \infty$ (Surprise!)

Entropy

Entropy is defined as the number of bits required to transmit a randomly selected event from a probability distribution.

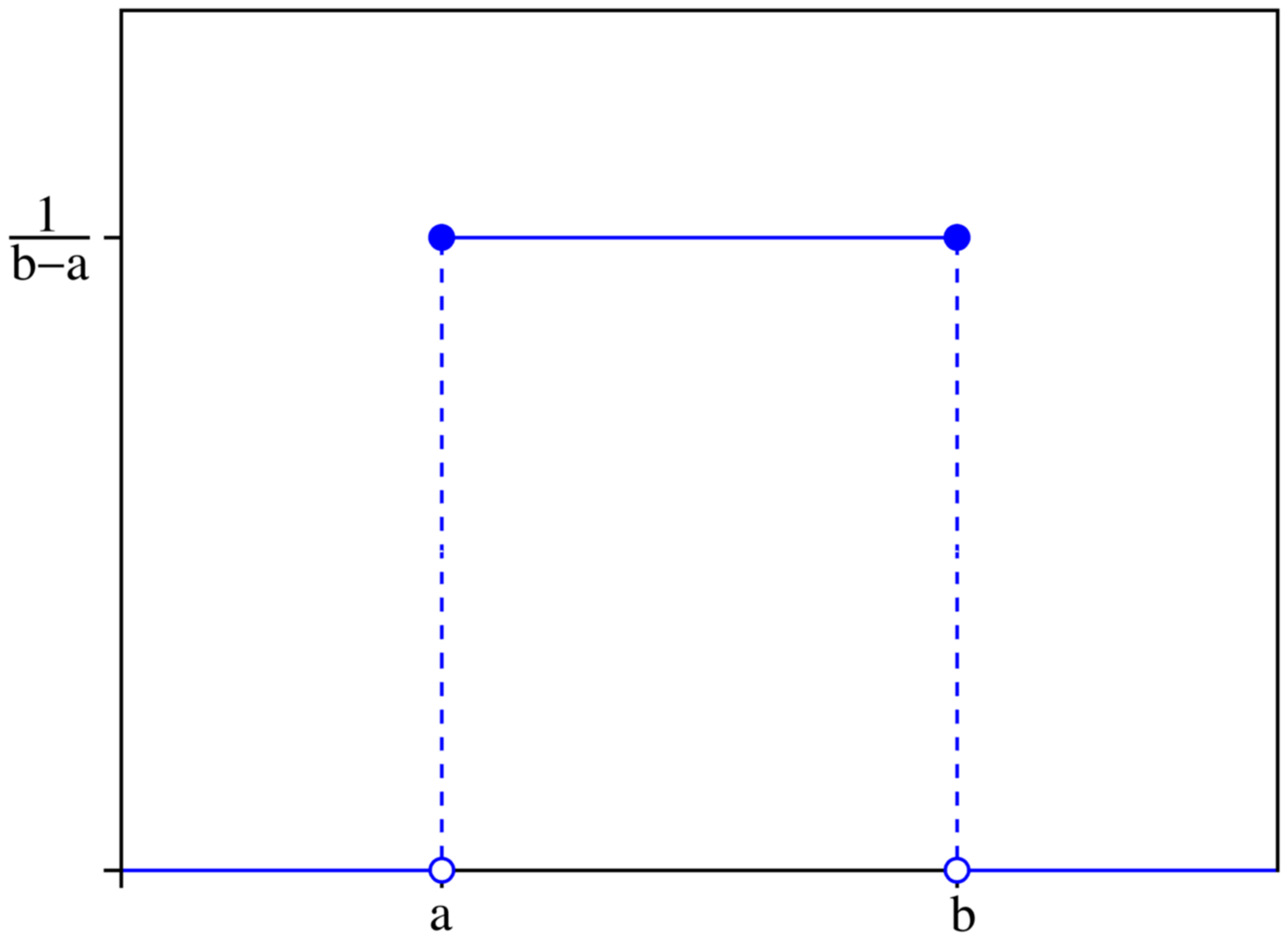
A skewed or a normal distribution has low entropy (low information associated)



Entropy

Entropy is defined as the number of bits required to transmit a randomly selected event from a probability distribution.

A uniform distribution has high entropy (high information associated)



Entropy

Mathematically, the entropy associated to a set X of discrete states x, with probabilities $P(x)$ of occurrence, is defined as

$$S[P(X)] = -\sum_{x \in X} P(x) \log[P(x)]$$

Using the base-2 logarithm, the results is given in **bits**.

If the base-e (natural) logarithm is used, the results will have units called **nats**.

Example 1

$$X = [x_1, x_2, x_3]$$
$$P(X) = [P(x_1), P(x_2), P(x_3)] = [0,1,0]$$

$$S[P(X)] = -\sum_{x \in X} P(x)\log[P(x)]$$
$$S[P(X)] = -P(x_1)\log[P(x_1)] - P(x_2)\log[P(x_2)] - P(x_3)\log[P(x_3)]$$
$$S[P(X)] = -\log[P(x_2)]$$
$$S[P(X)] = -\log[1] = 0,$$

Example 2

$$X = [x_1, x_2, x_3]$$
$$P(X) = [P(x_1), P(x_2), P(x_3)] = [0.5, 0.3, 0.2]$$

$$S[P(X)] = -\sum_{x \in X} P(x) \log[P(x)]$$
$$S[P(X)] = -P(x_1)\log[P(x_1)] - P(x_2)\log[P(x_2)] - P(x_3)\log[P(x_3)]$$
$$S[P(X)] = 1.49$$

Cross Entropy

Cross-entropy calculates the number of bits required to represent or transmit an average event from one distribution compared to another distribution.

Target distribution: P

Approximation of the target distribution: Q

$$H[P,Q] = -\sum_{x \in X} P(x)\log[Q(x)]$$

Example 3

$$X = [x_1, x_2, x_3]$$

$$P(X) = [P(x_1), P(x_2), P(x_3)] = [0,1,0]$$

$$Q(X) = [Q(x_1), Q(x_2), Q(x_3)] = [0.6,0.2,0.2]$$

$$H[P, Q] = -\sum_{x \in X} P(x)\log[Q(x)]$$

$$S[P(X)] = -P(x_1)\log[Q(x_1)] - P(x_2)\log[Q(x_2)] - P(x_3)\log[Q(x_3)]$$

$$S[P(X)] = -P(x_2)\log[Q(x_2)]$$

$$S[P(X)] = 2.32$$

Example 4

$$X = [x_1, x_2, x_3]$$

$$P(X) = [P(x_1), P(x_2), P(x_3)] = [0,1,0]$$

$$Q(X) = [Q(x_1), Q(x_2), Q(x_3)] = [0.3,0.5,0.2]$$

$$H[P, Q] = -\sum_{x \in X} P(x)\log[Q(x)]$$

$$S[P(X)] = -P(x_1)\log[Q(x_1)] - P(x_2)\log[Q(x_2)] - P(x_3)\log[Q(x_3)]$$

$$S[P(X)] = -P(x_2)\log[Q(x_2)]$$

$$S[P(X)] = 1.0$$

Example 5

$$X = [x_1, x_2, x_3]$$

$$P(X) = [P(x_1), P(x_2), P(x_3)] = [0,1,0]$$

$$Q(X) = [Q(x_1), Q(x_2), Q(x_3)] = [0.1,0.8,0.1]$$

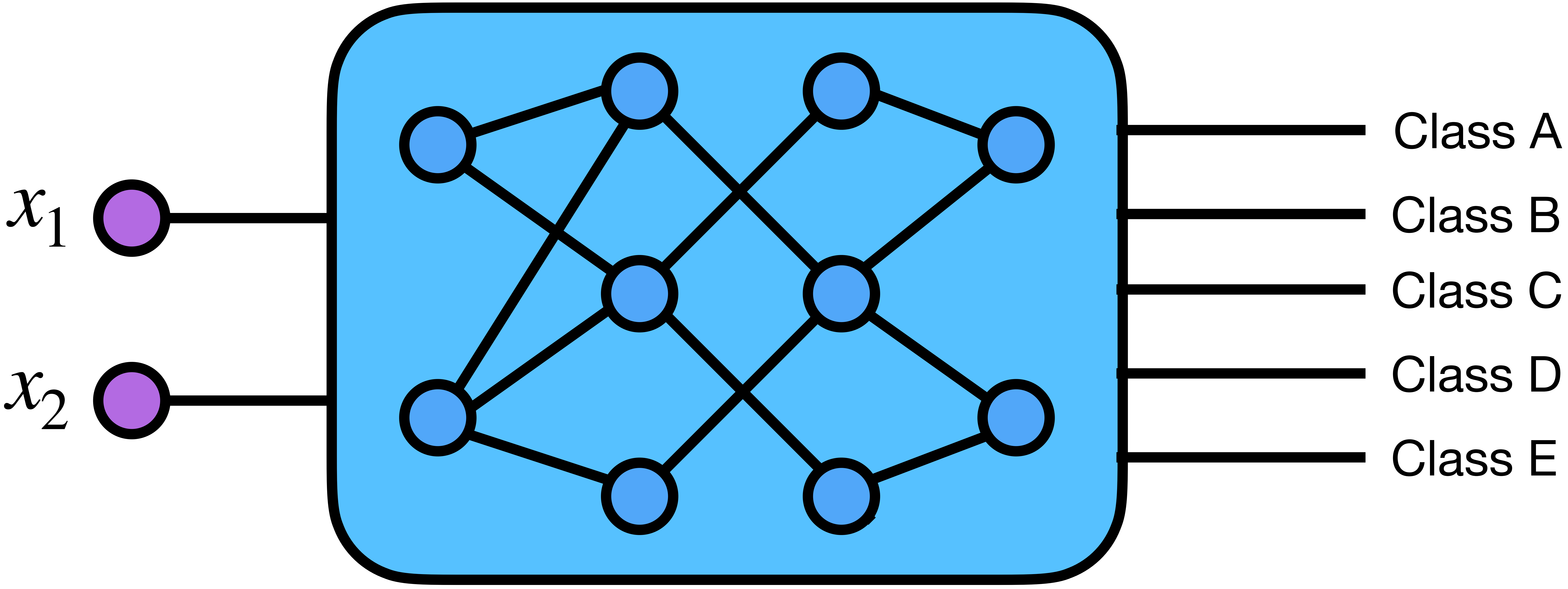
$$H[P, Q] = -\sum_{x \in X} P(x) \log[Q(x)]$$

$$S[P(X)] = -P(x_1)\log[Q(x_1)] - P(x_2)\log[Q(x_2)] - P(x_3)\log[Q(x_3)]$$

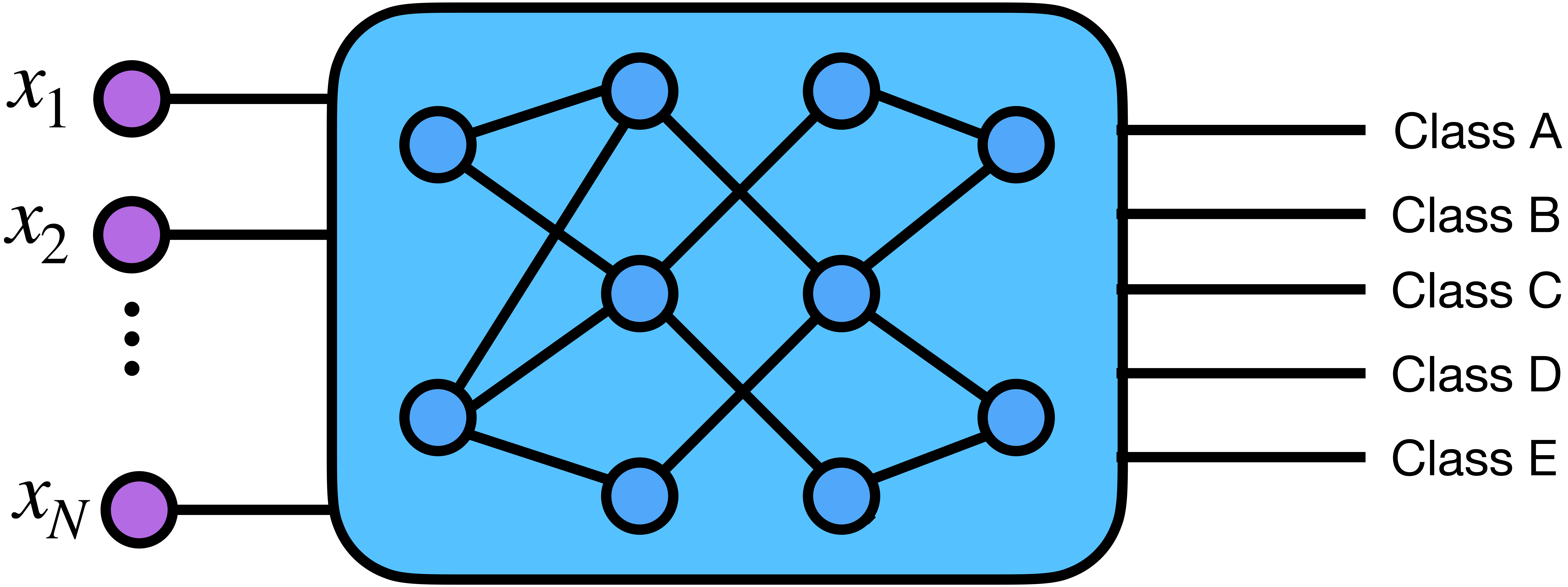
$$S[P(X)] = -P(x_2)\log[Q(x_2)]$$

$$S[P(X)] = 0.32$$

Neural Network



Classification Neural Network



Codification

One Hot Encoding

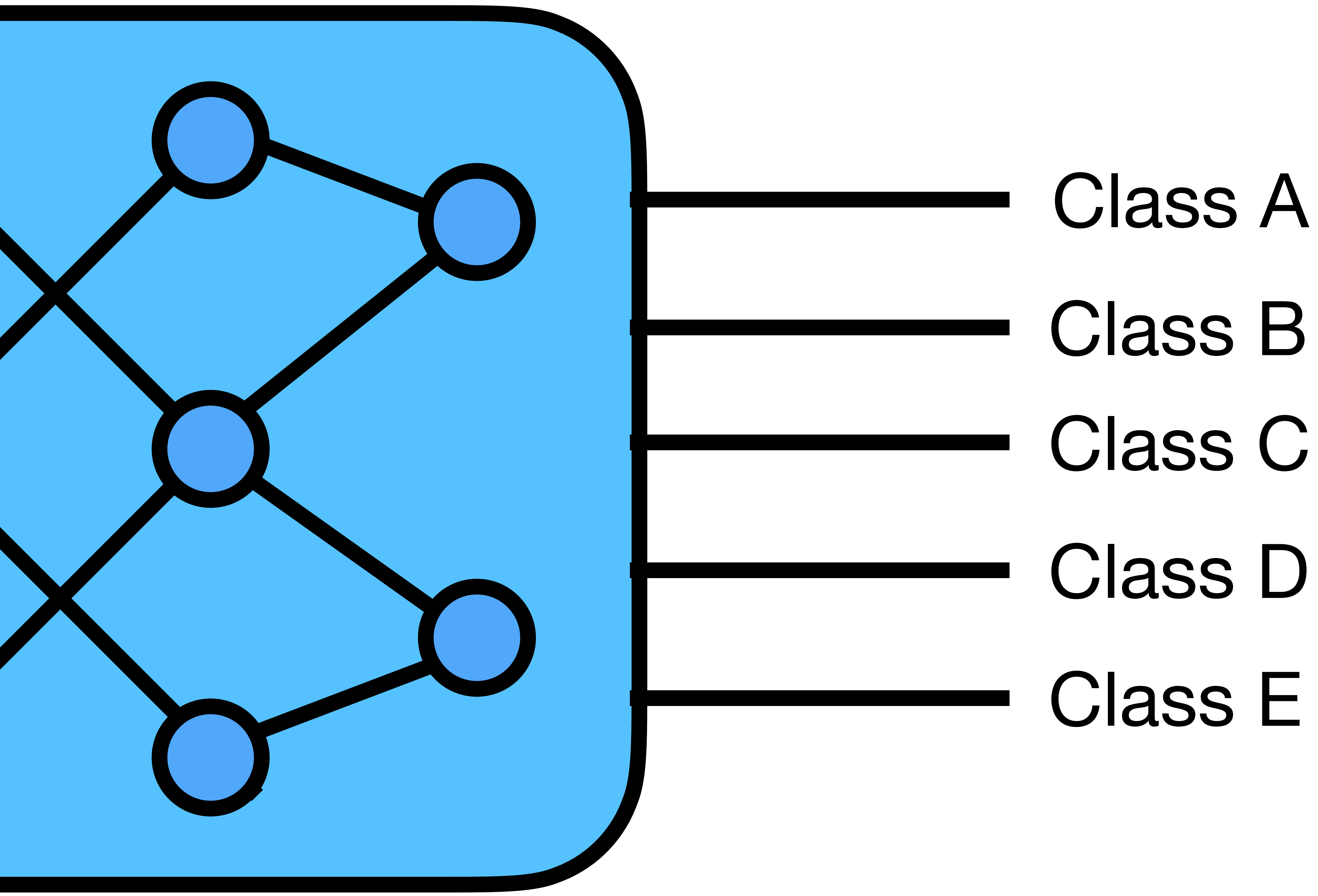
Class A →	0 →	[10000]
Class B →	1 →	[01000]
Class C →	2 →	[00100]
Class D →	3 →	[00010]
Class E →	4 →	[00001]

SparseCategoricalCrossEntropy

CategoricalCrossEntropy

$$H = -\sum_{i=1}^{N} y_i \log[y_i^p]$$

Classification Neural Network



The output of the neural network is a vector with a number for each of the final neurons which are proportional to the probability of each class. However, these numbers can be either positive or negative and they can be greater than one.

In order to obtain a probability interpretation, it is possible to use an adequate activation function in the output layer.

SoftMax Activation Function

The SoftMax activation function takes an input vector with N real numbers and normalizes it into a probability distribution consisting of N probabilities proportional to the exponentials of the inputs.

$$[\sigma(\mathrm{z})]_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$$

The sum of all the components of the final output is 1.

@astronomiaOAN

AstronomiaOAN

@astronomiaOAN