# Neural Networks and Interpretability

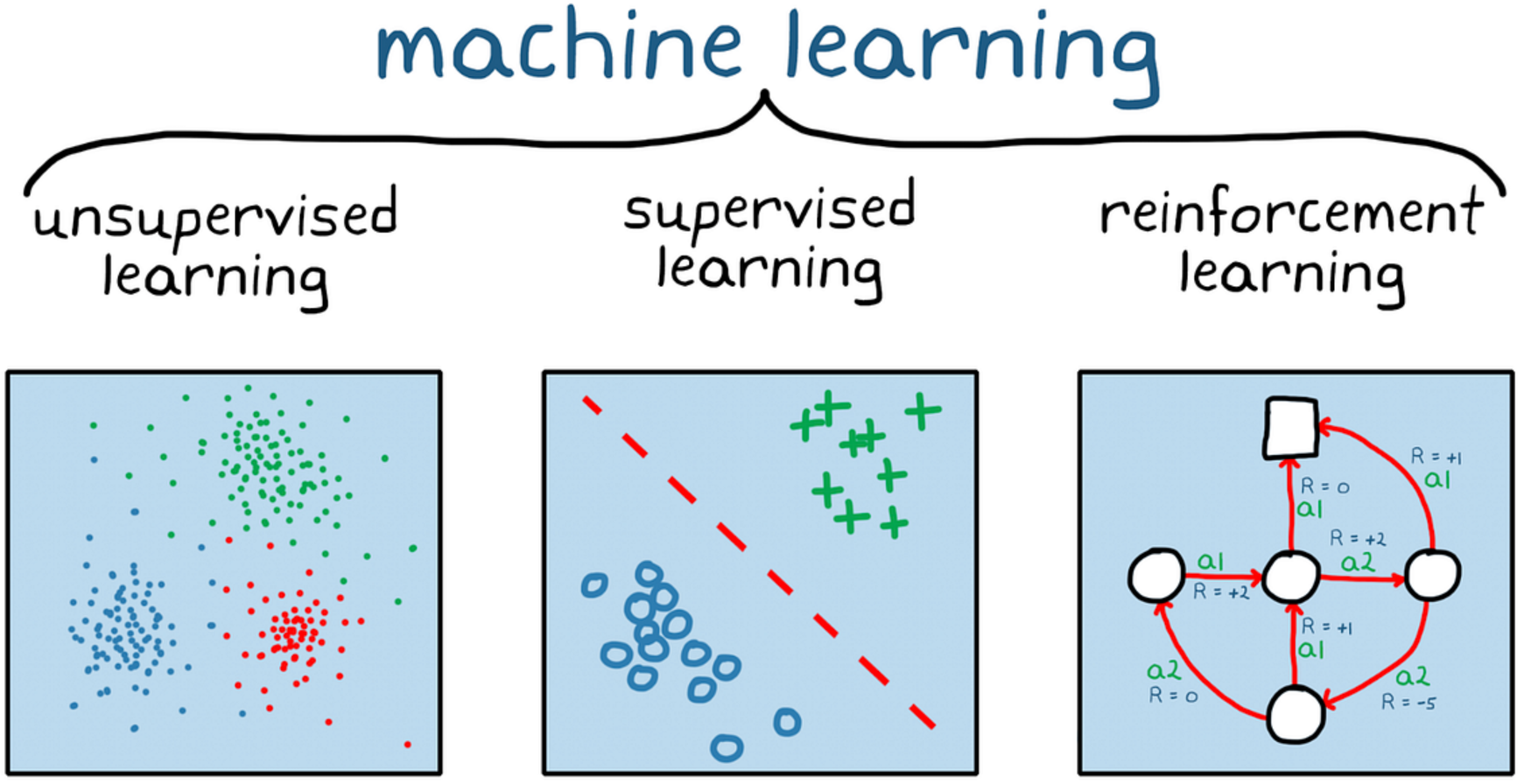
Indira Ocampo,
Instituto de Física Teórica, Madrid

Santiago, 15 - 16 de Enero de 2025

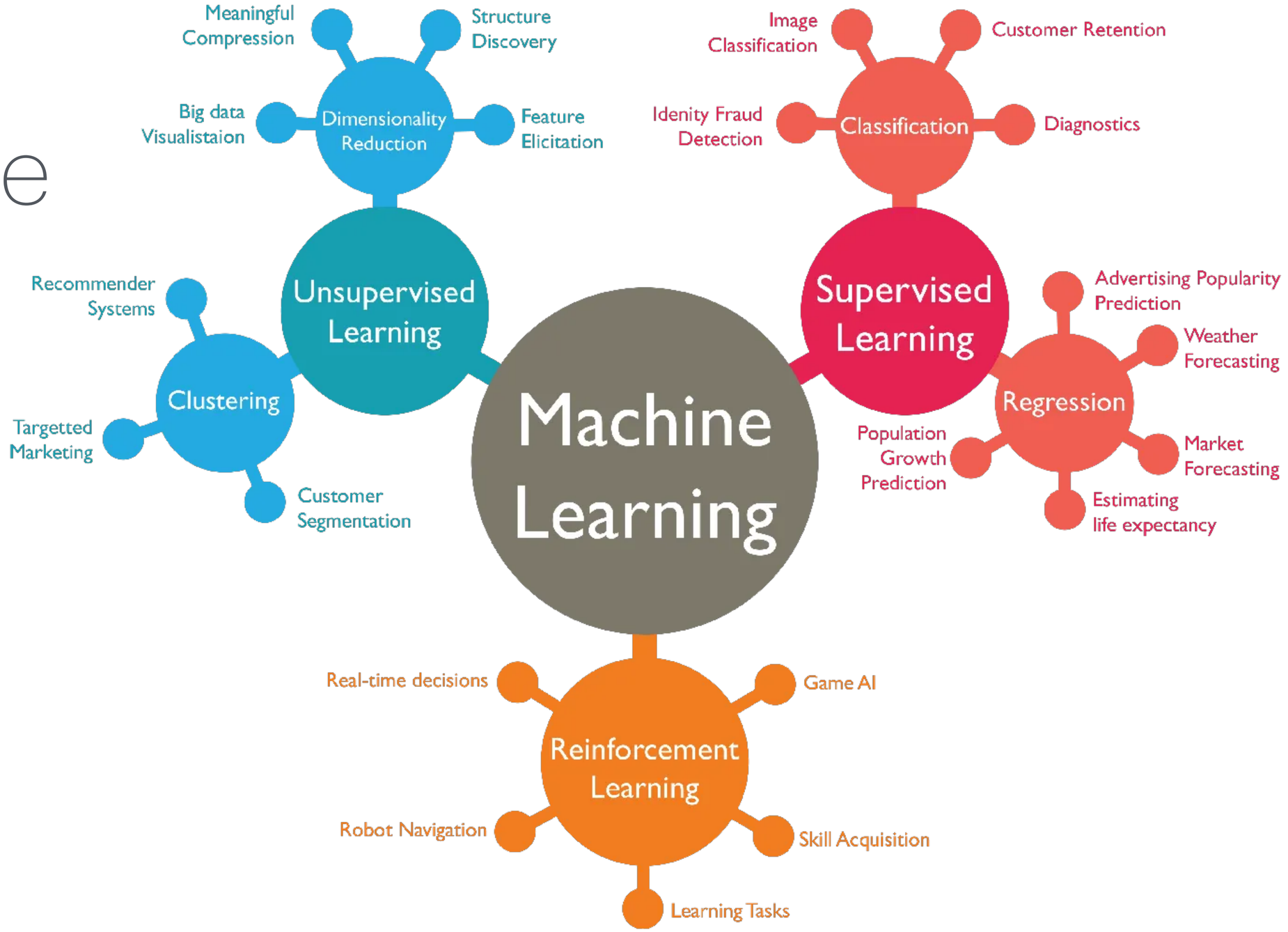
# Outline (Part 1)

- Introducción
- Machine Learning landscape
- Linear Regression
- Logistic Regression
- Neural Networks
- Some examples

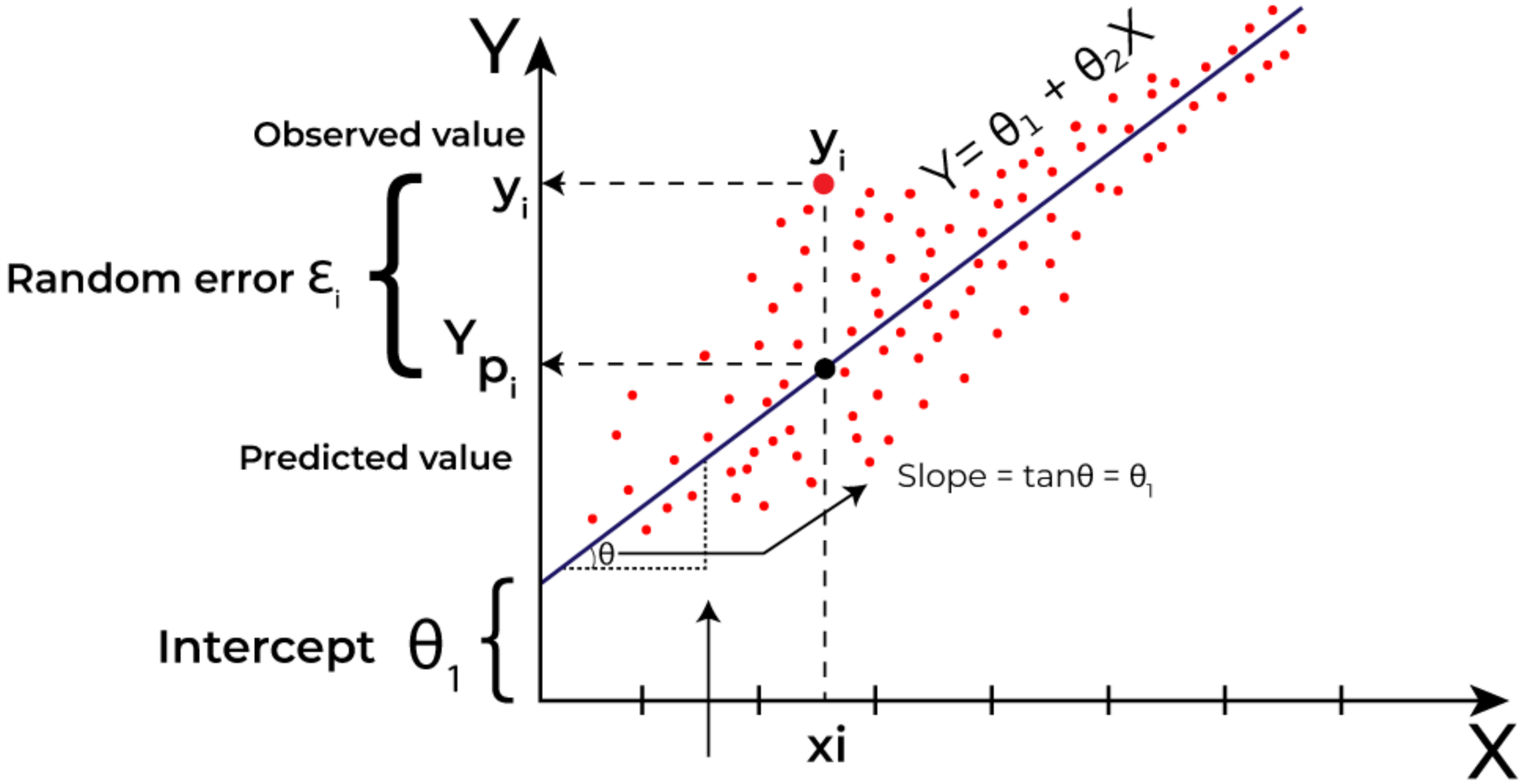
# Introducción



# ML landscape



# Linear regression




$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

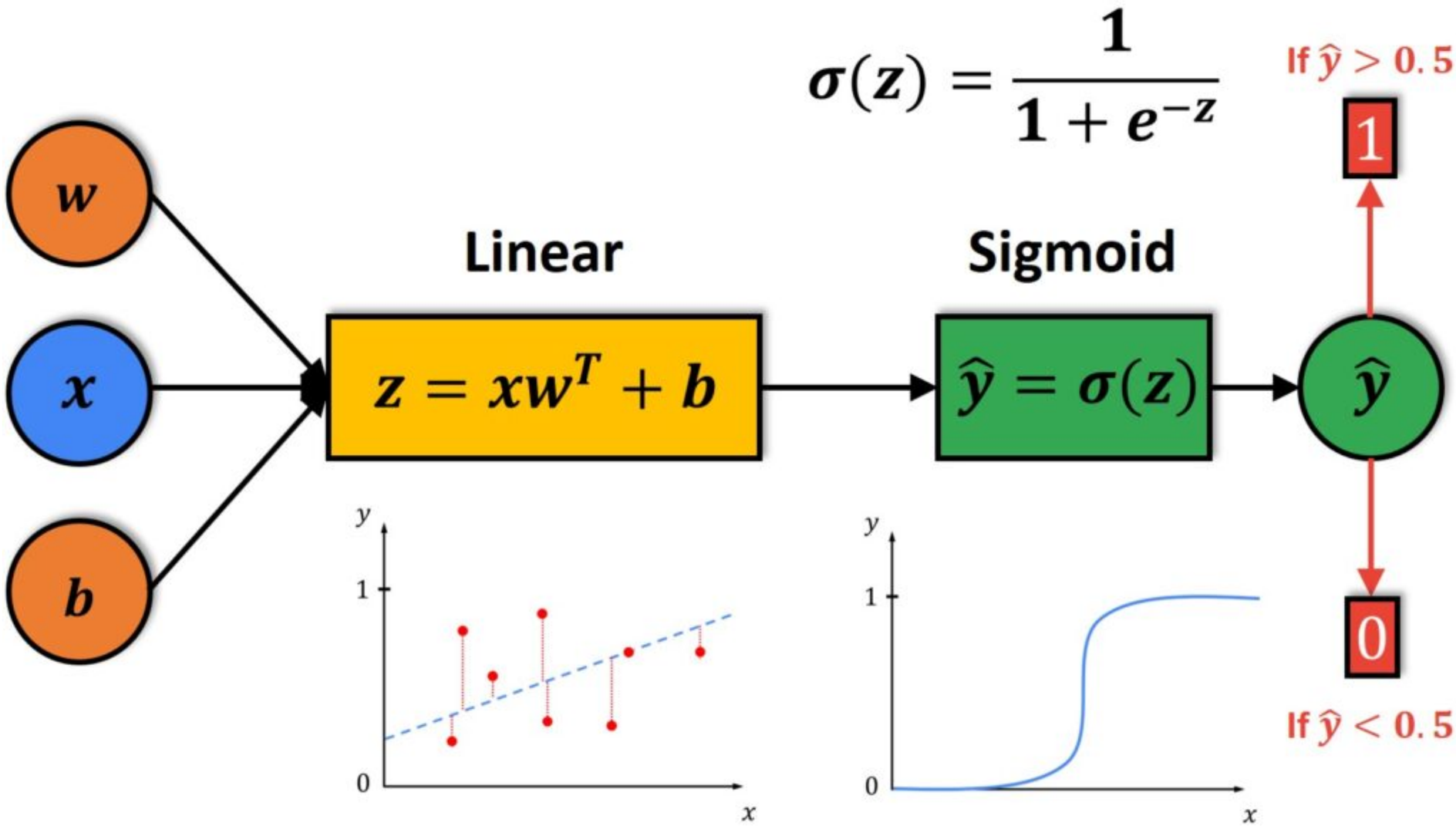
$\mathrm{MSE}$ = mean squared error

$n$ = number of data points

$Y_i$ = observed values

$\hat{Y}_i$ = predicted values

# Logistic regression



# Logistic regression

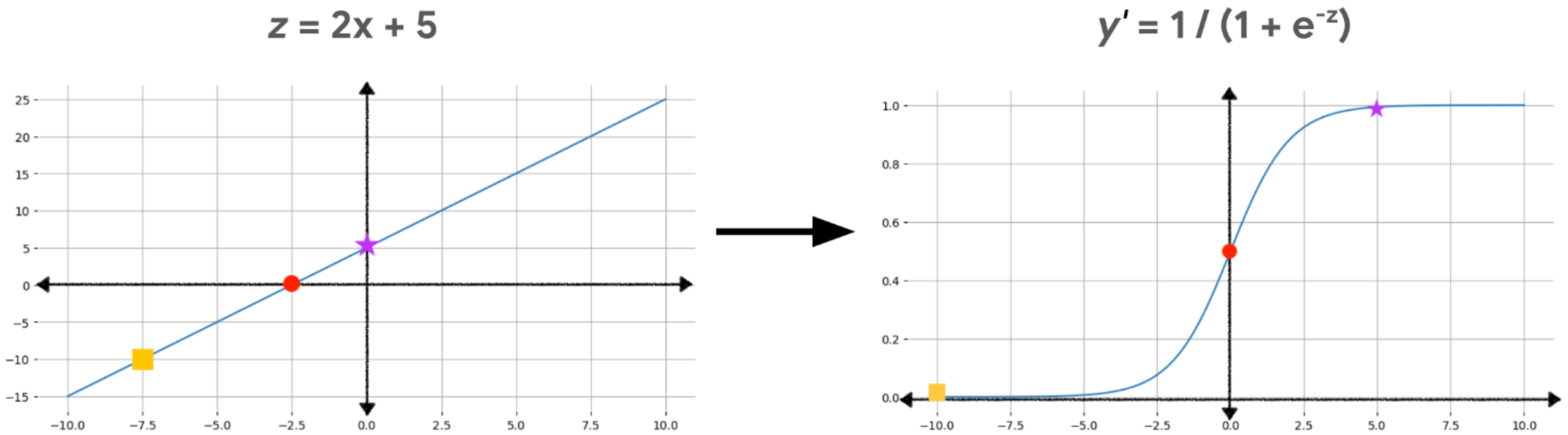
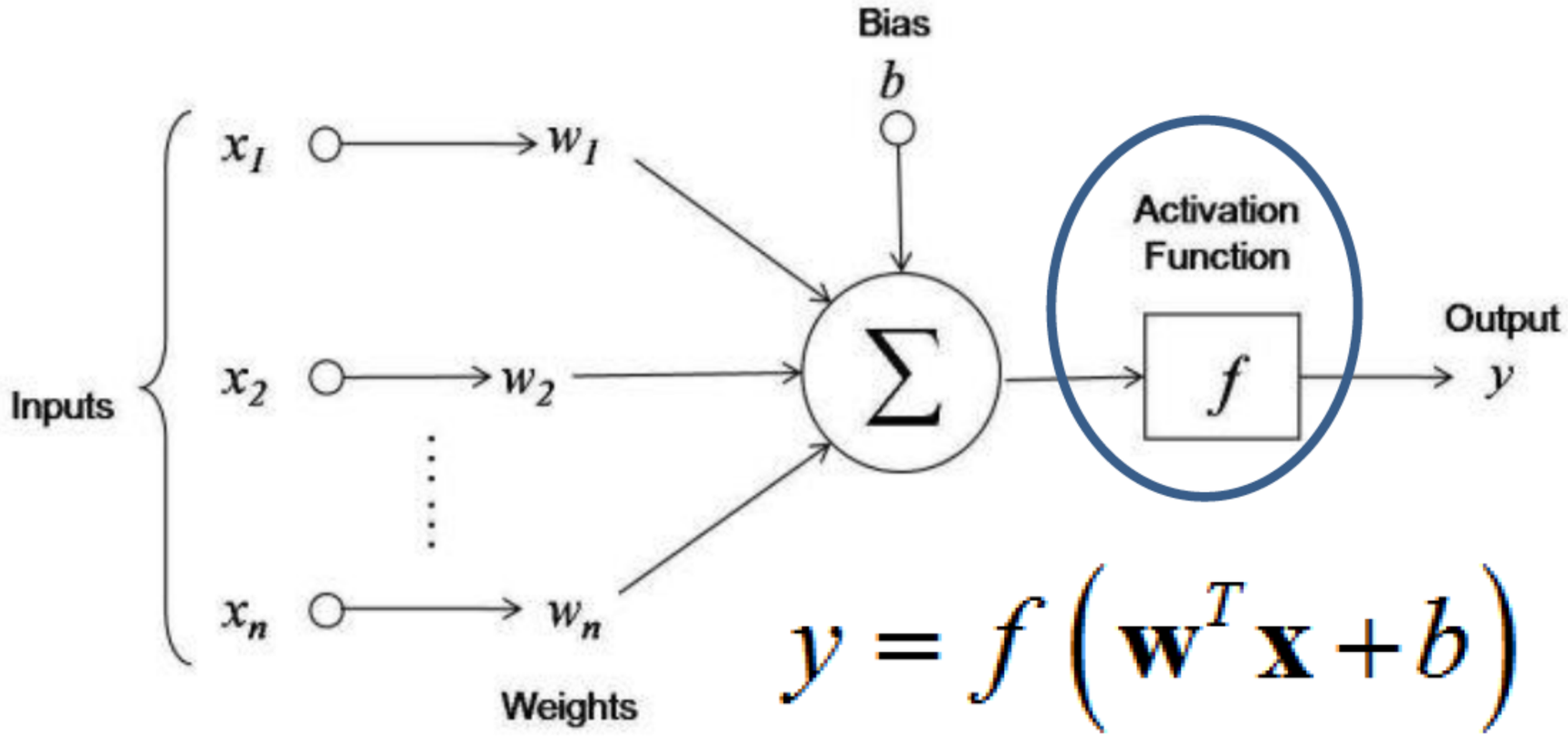



***Figure 2.*** *Left: graph of the linear function z = 2x + 5, with three points highlighted. Right: Sigmoid curve with the same three points highlighted after being transformed by the sigmoid function.*

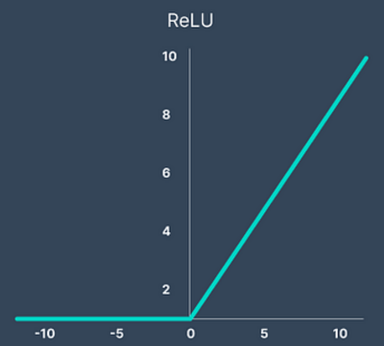
# Neural Networks

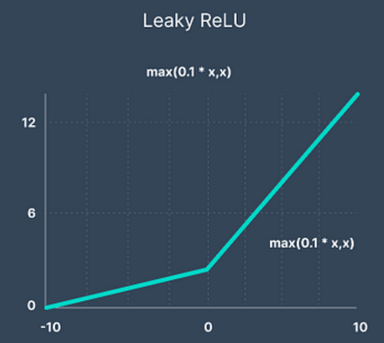


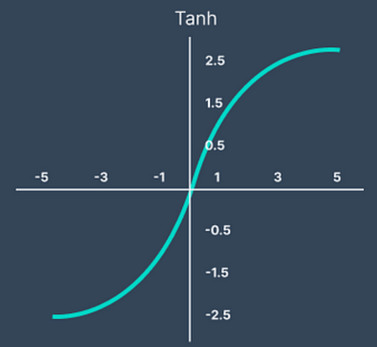

$$y = f\left(\mathbf{w}^T\mathbf{x} + b\right)$$

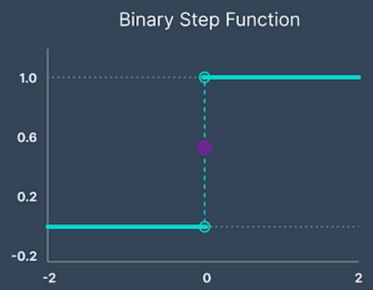
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

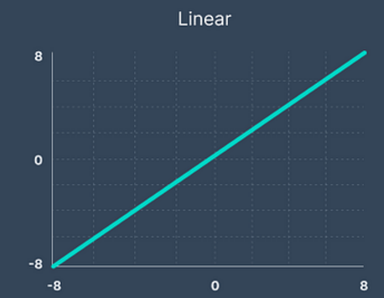
# Activation functions

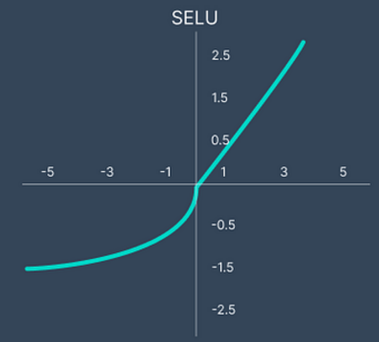


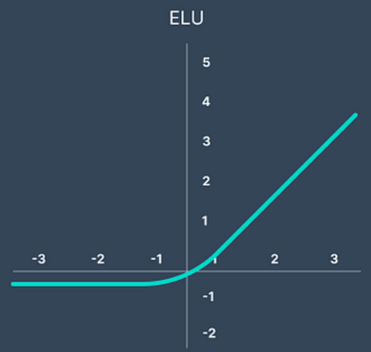



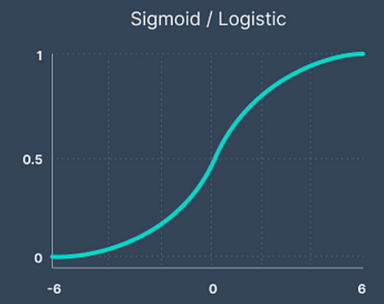



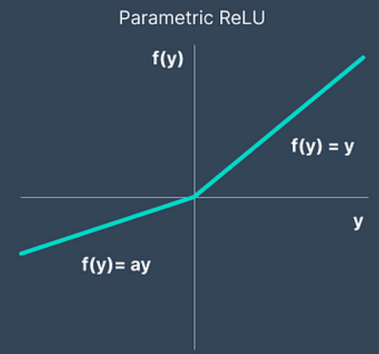



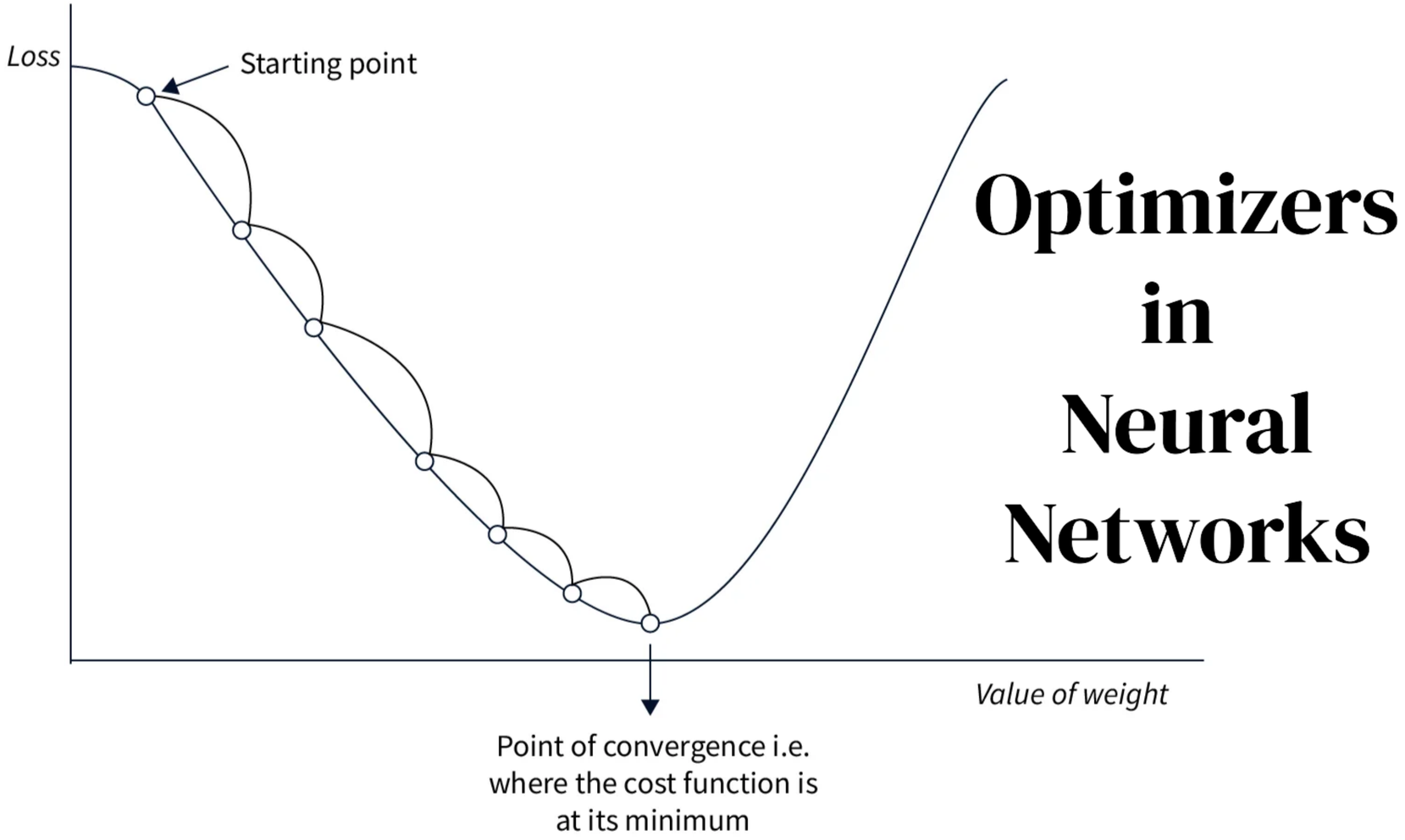



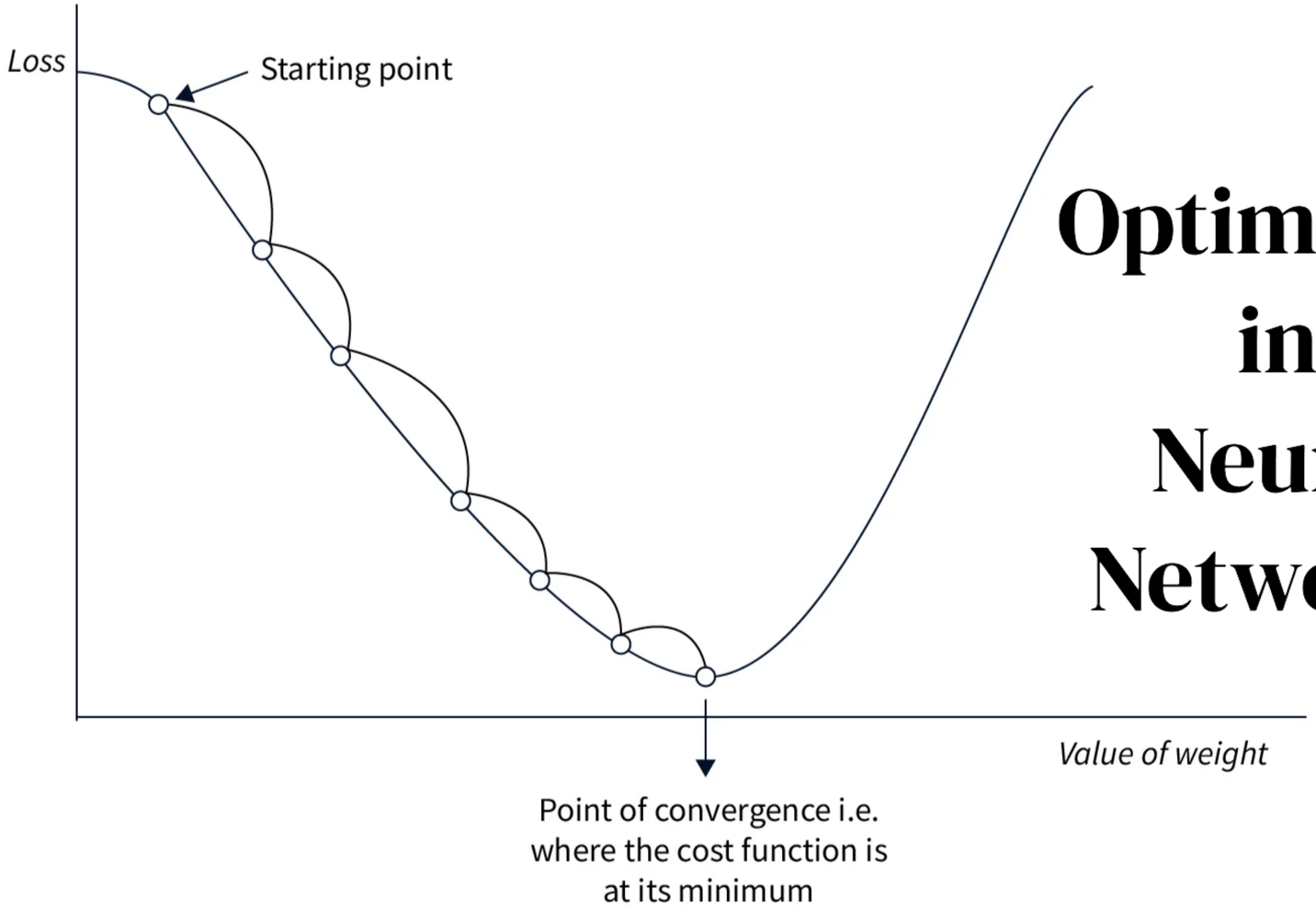



# Optimizers




**Optimizers in Neural Networks**

# Optimizers

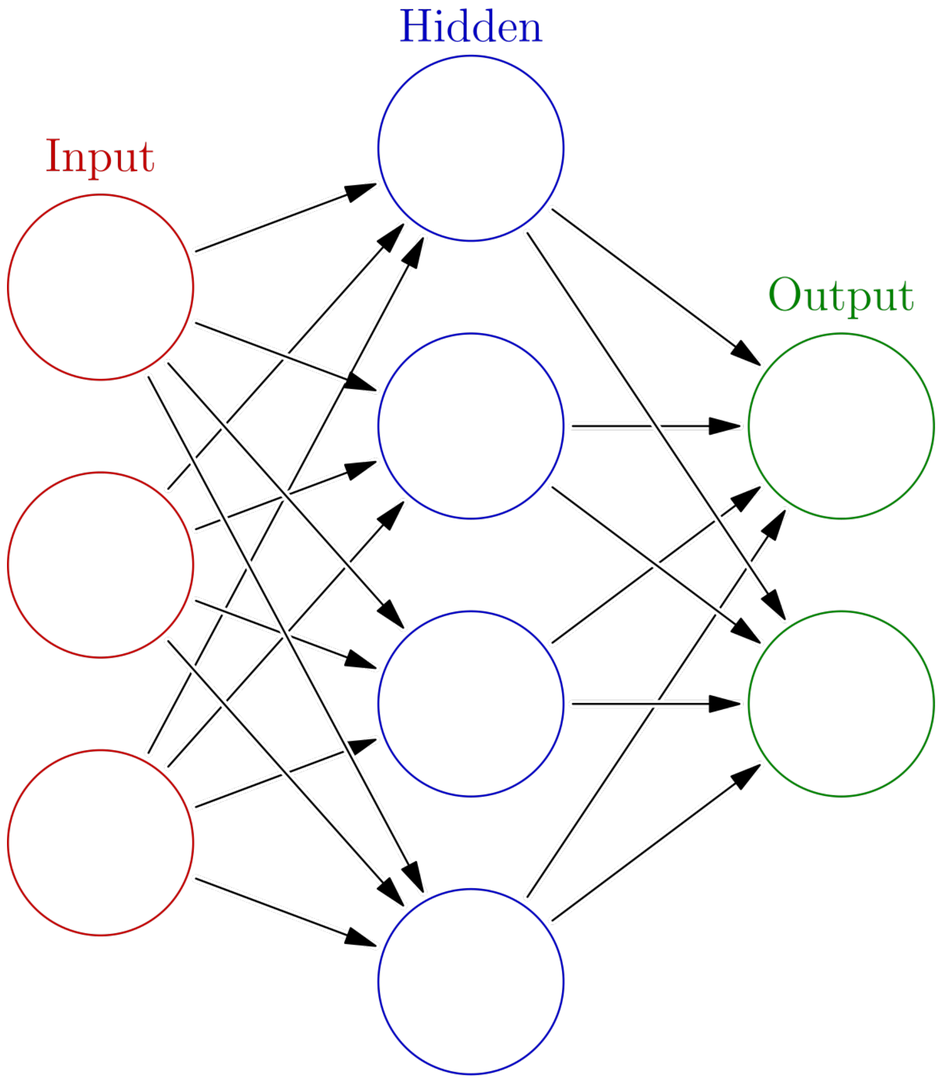



**Optimizers in Neural Networks**

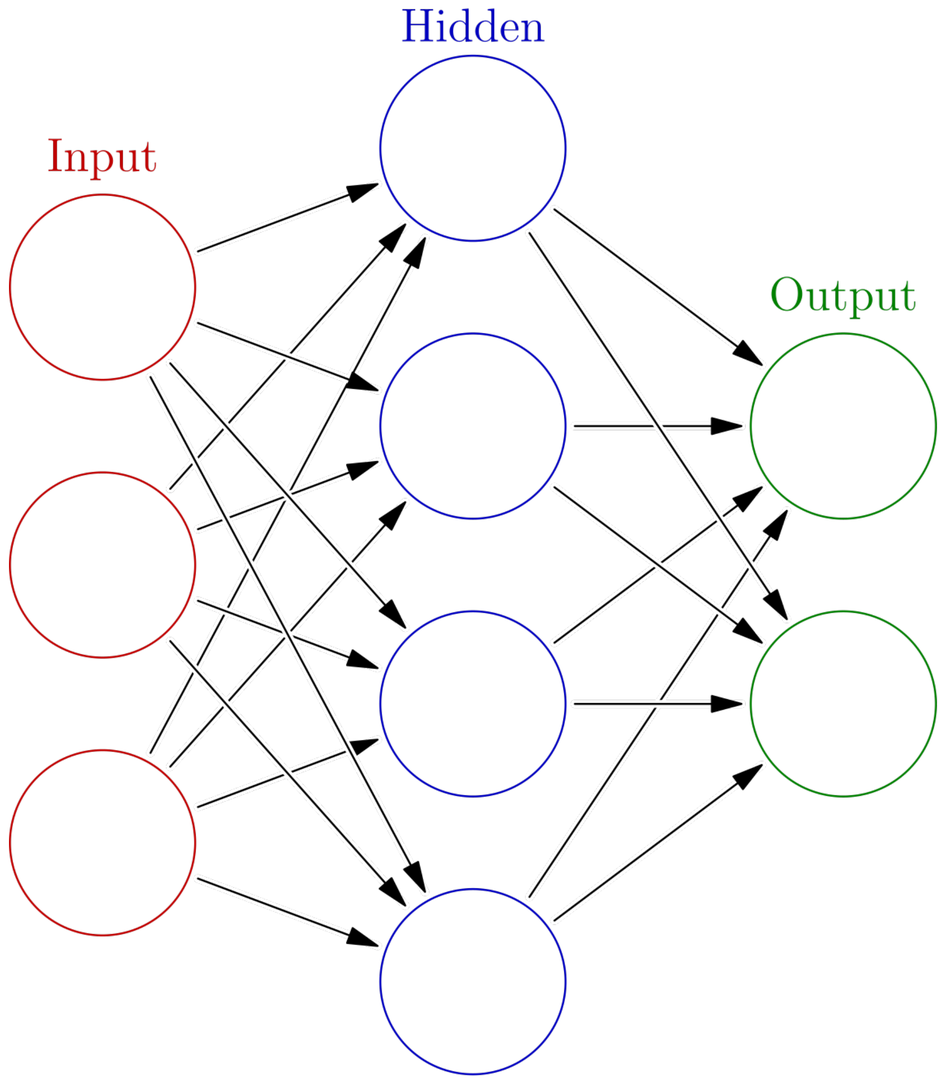
$$w_{\text{new}} = w_{\text{old}} - \alpha \frac{\partial L}{\partial w}$$

$$b_{\text{new}} = b_{\text{old}} - \alpha \frac{\partial L}{\partial b}$$

# Neural Networks



# Neural Networks




- Forward Pass

$$y = w \cdot x + b \qquad \sigma(z) = \frac{1}{1 + e^{-z}}$$

- Compute the Loss

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

- Backpropagation of errors (gradient descent)

$$w_{\text{new}} = w_{\text{old}} - \alpha\frac{\partial L}{\partial w} \qquad b_{\text{new}} = b_{\text{old}} - \alpha\frac{\partial L}{\partial b}$$

- Repeat (until minimizing the loss)

epoch

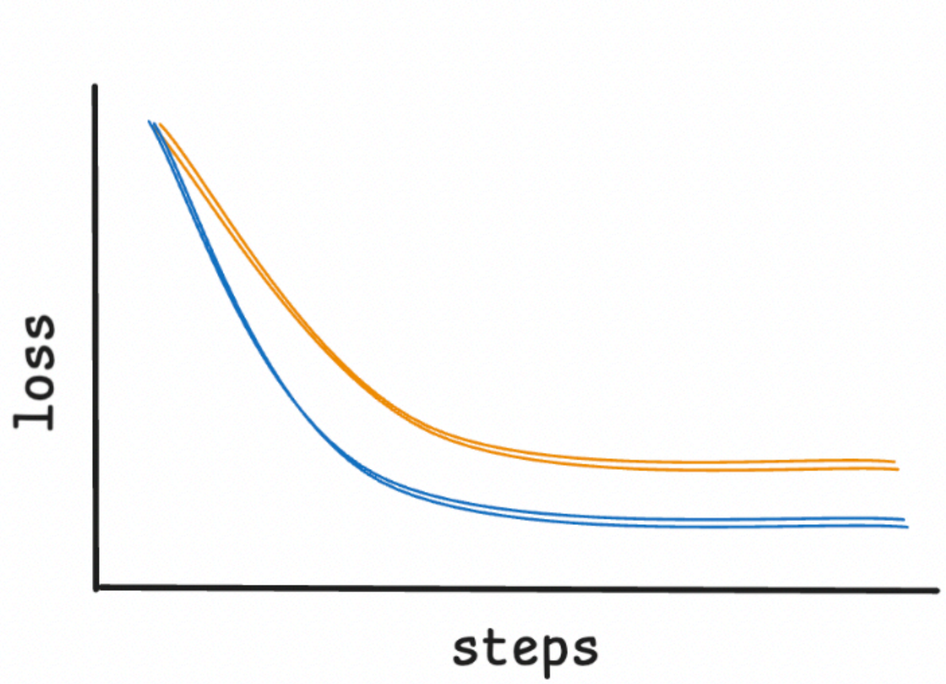
# Evaluation metrics

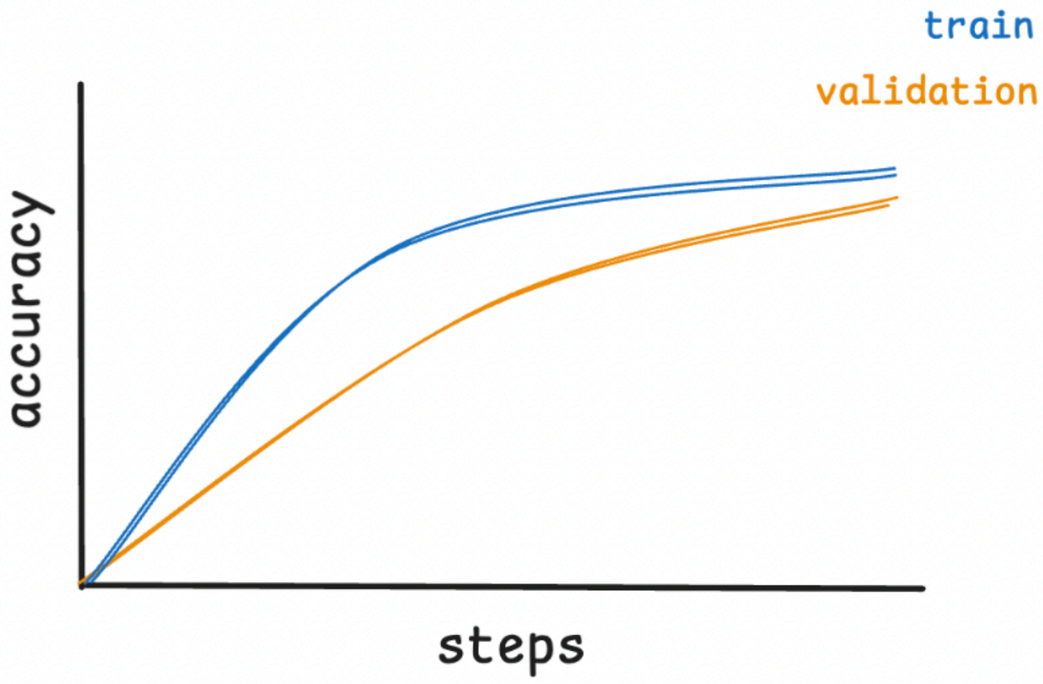
## Classification tasks

- Learning curves
- Confusion Matrix
- ROC curve
- AUC

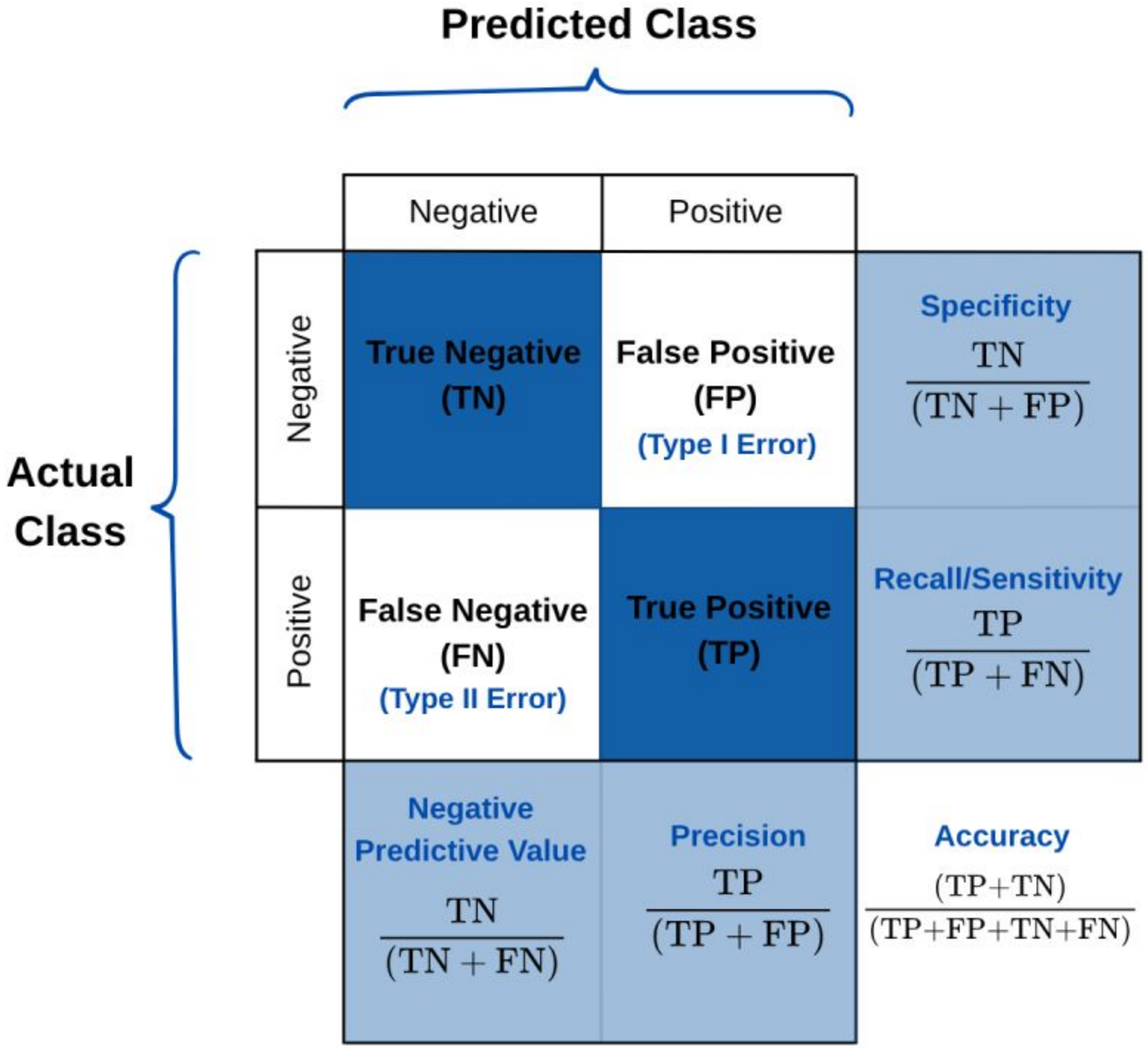
# Learning Curves

- Accuracy: percentage of correct predictions over time
- Loss: how the model's error decreases over time

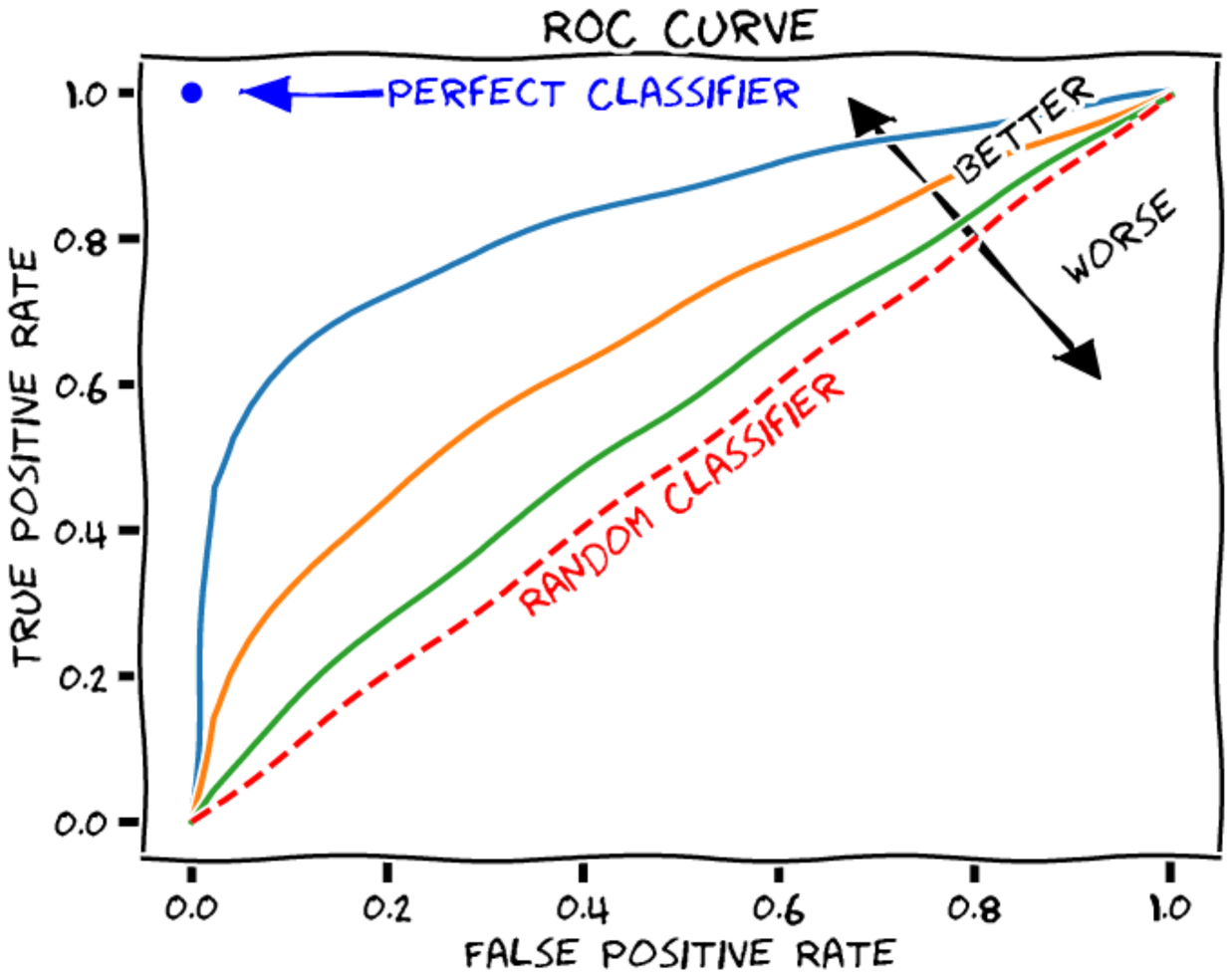


# Confusion Matrix



# Receiver-operating characteristic curve (ROC)




| Acquired Class \ True Class | T | F |
|---|---|---|
| Y | True Positives (TP) | False Positives (FP) |
| N | False Negatives (FN) | True Negatives (TN) |

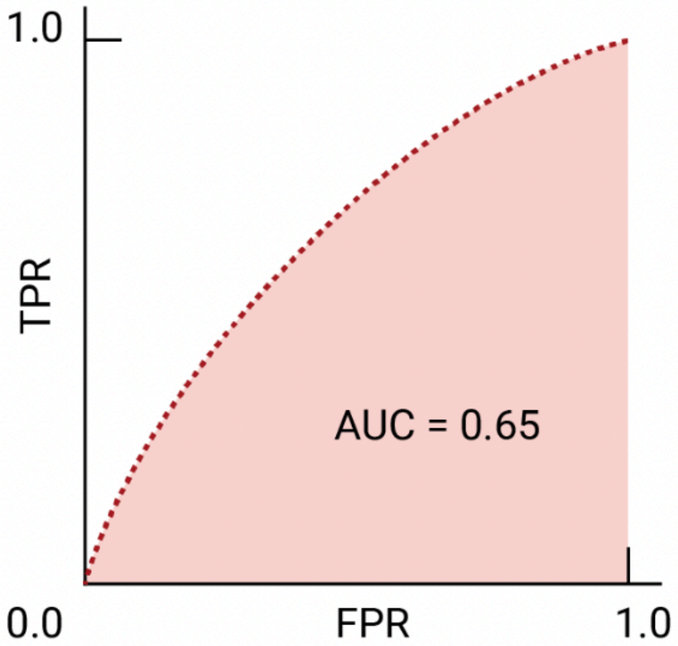
$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$

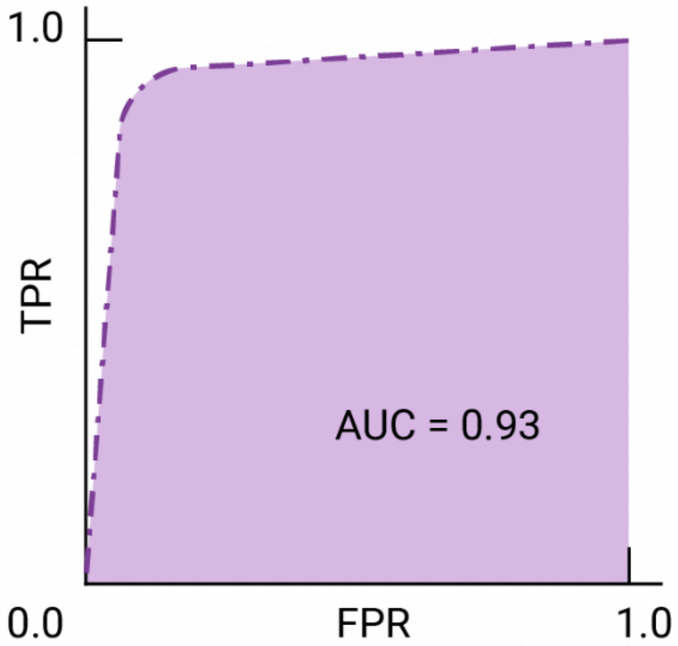
$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

$$\text{Accuracy (ACC)} = \frac{TP + TN}{TP + FP + TN + FN}$$

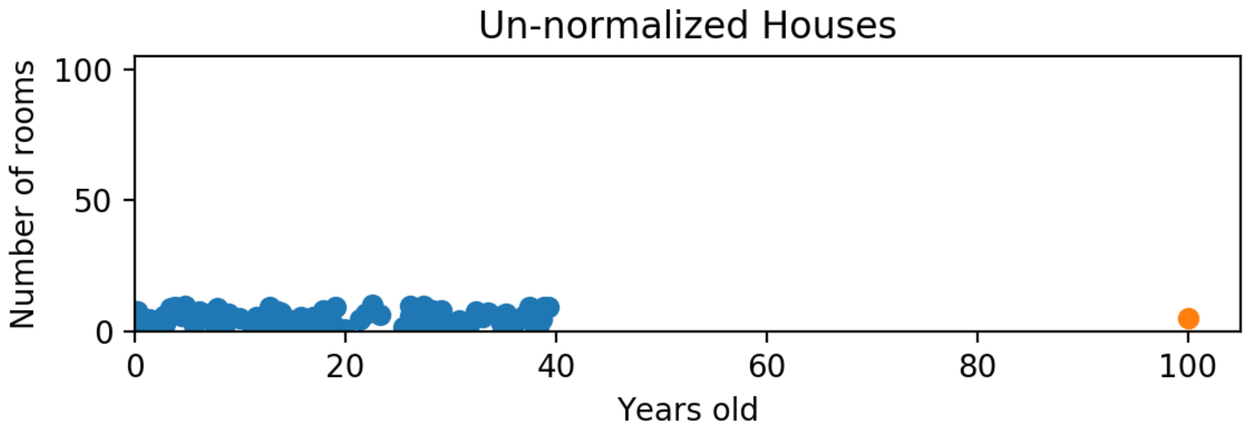
# Area Under the ROC Curve

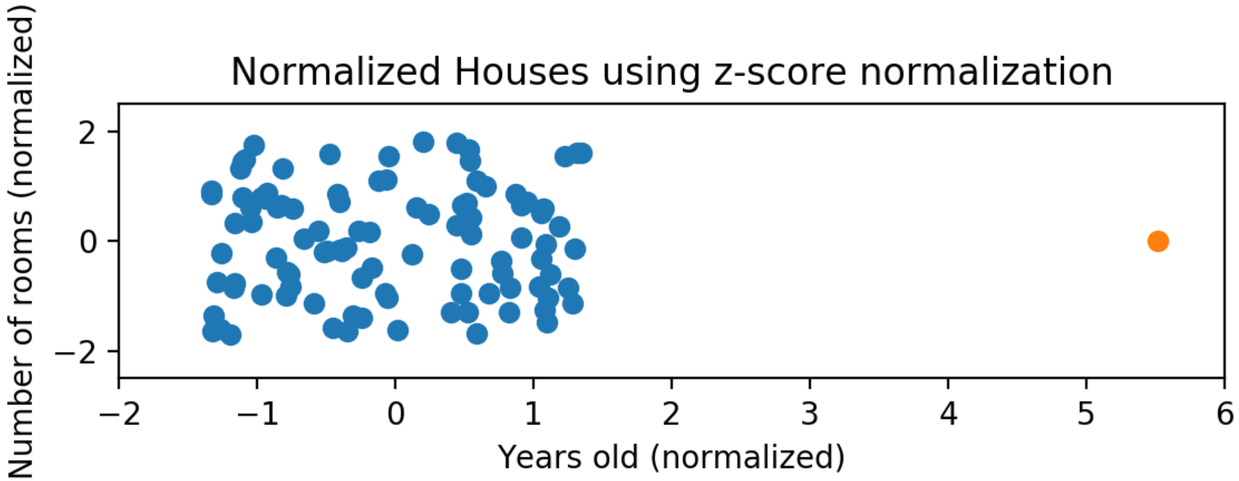
- >> How well the model distinguishes between positive and negative classes across all possible classification thresholds



# Normalization



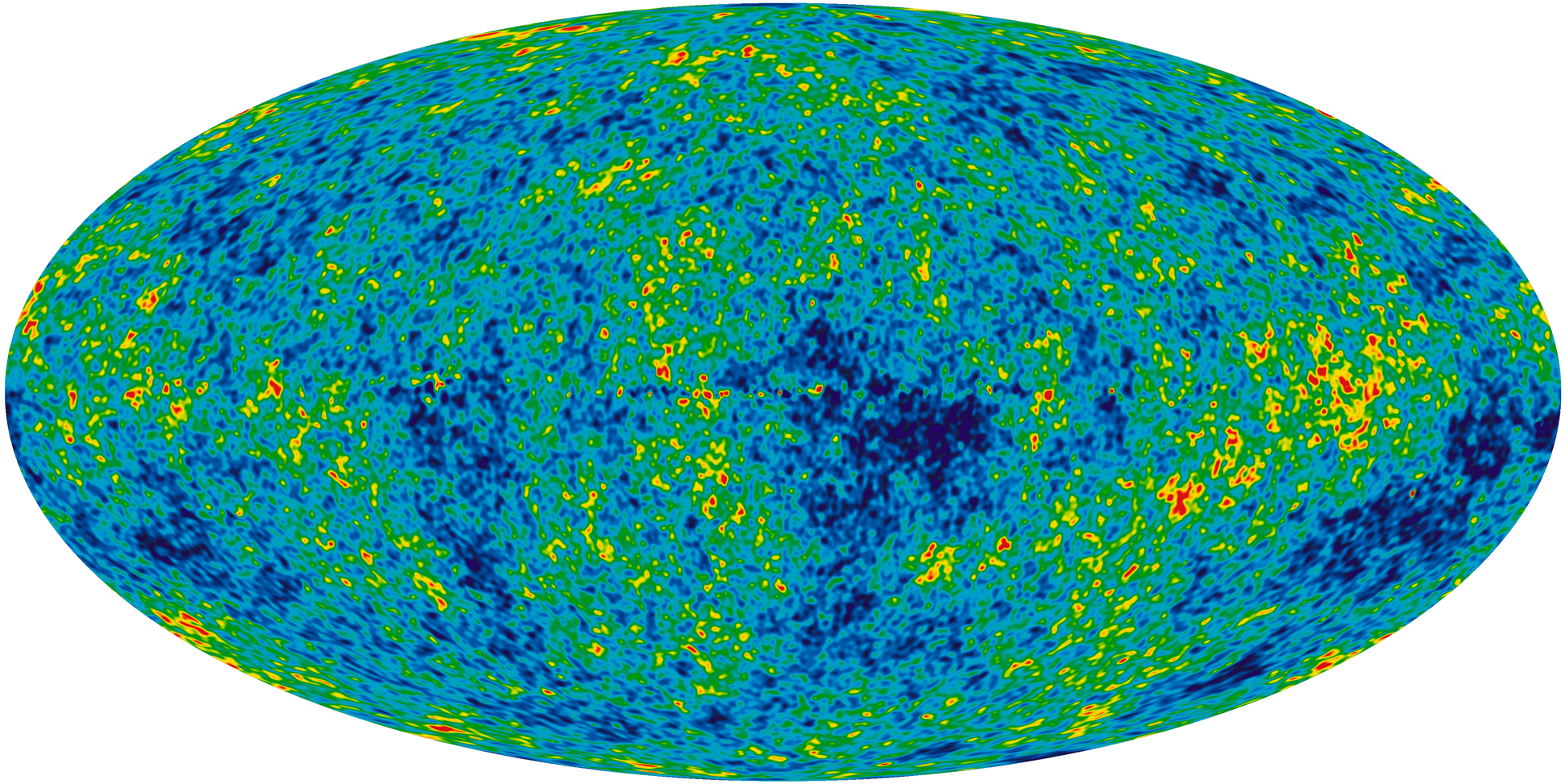




**Normalization Formula**

$$X_{normalized} = \frac{(X - X_{minimum})}{(X_{maximum} - X_{minimum})}$$

# An example

## Cosmic Microwave Background



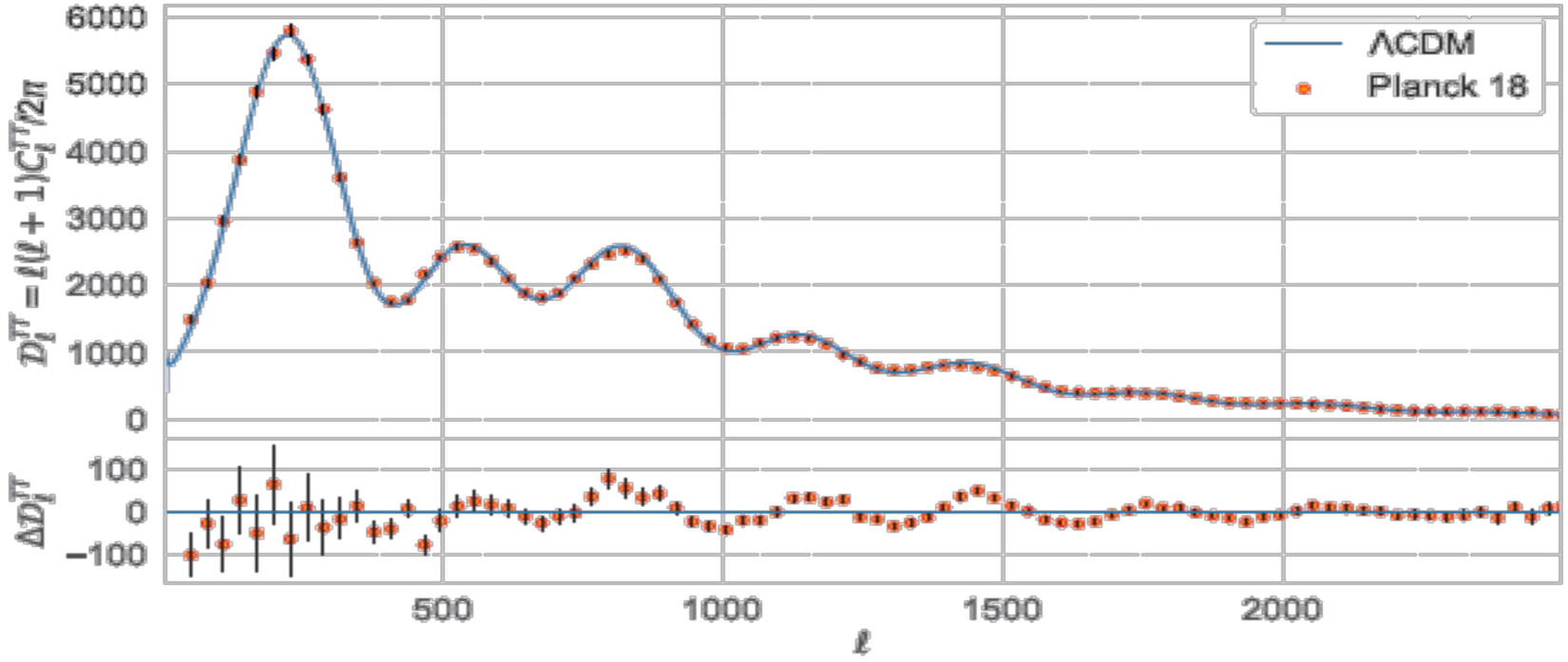
# The CMB

## Angular Power Spectrum

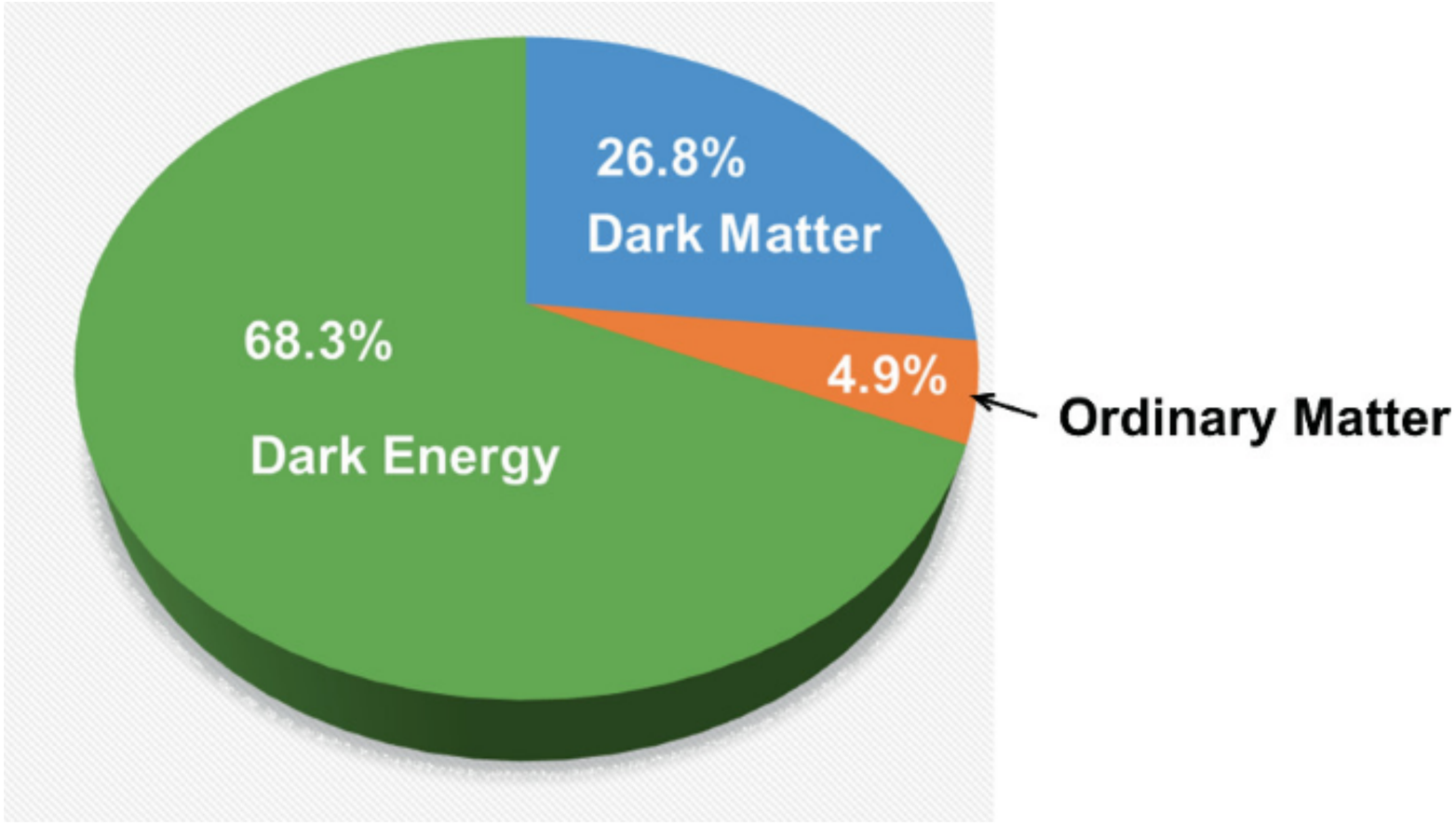
$$\frac{\Delta T}{T} \sim 10^{-5}$$

$$\frac{\Delta T}{T}(\hat{n}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\hat{n})$$

$$C_\ell = \langle |a_{\ell m}|^2 \rangle$$

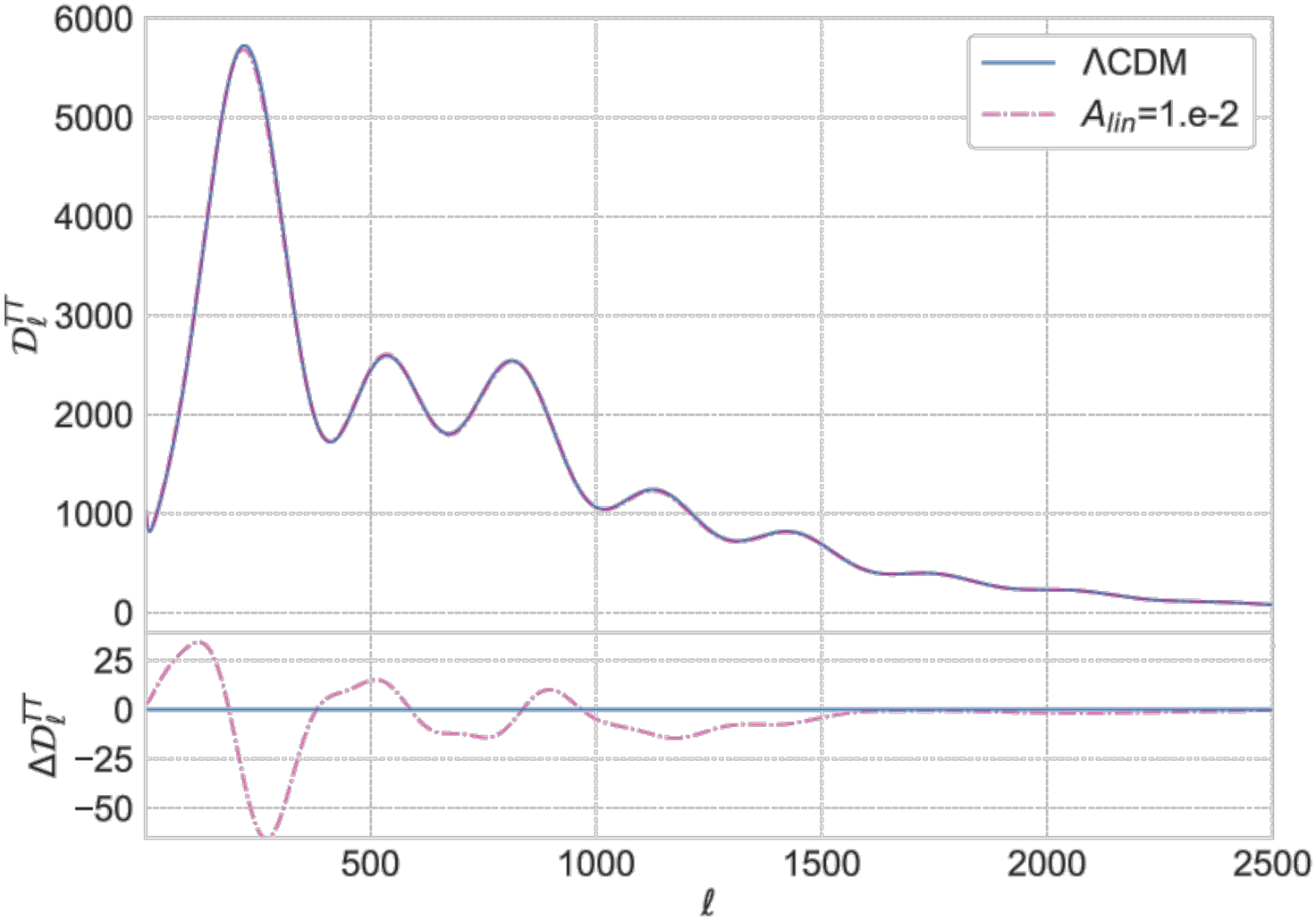


# The ΛCDM model

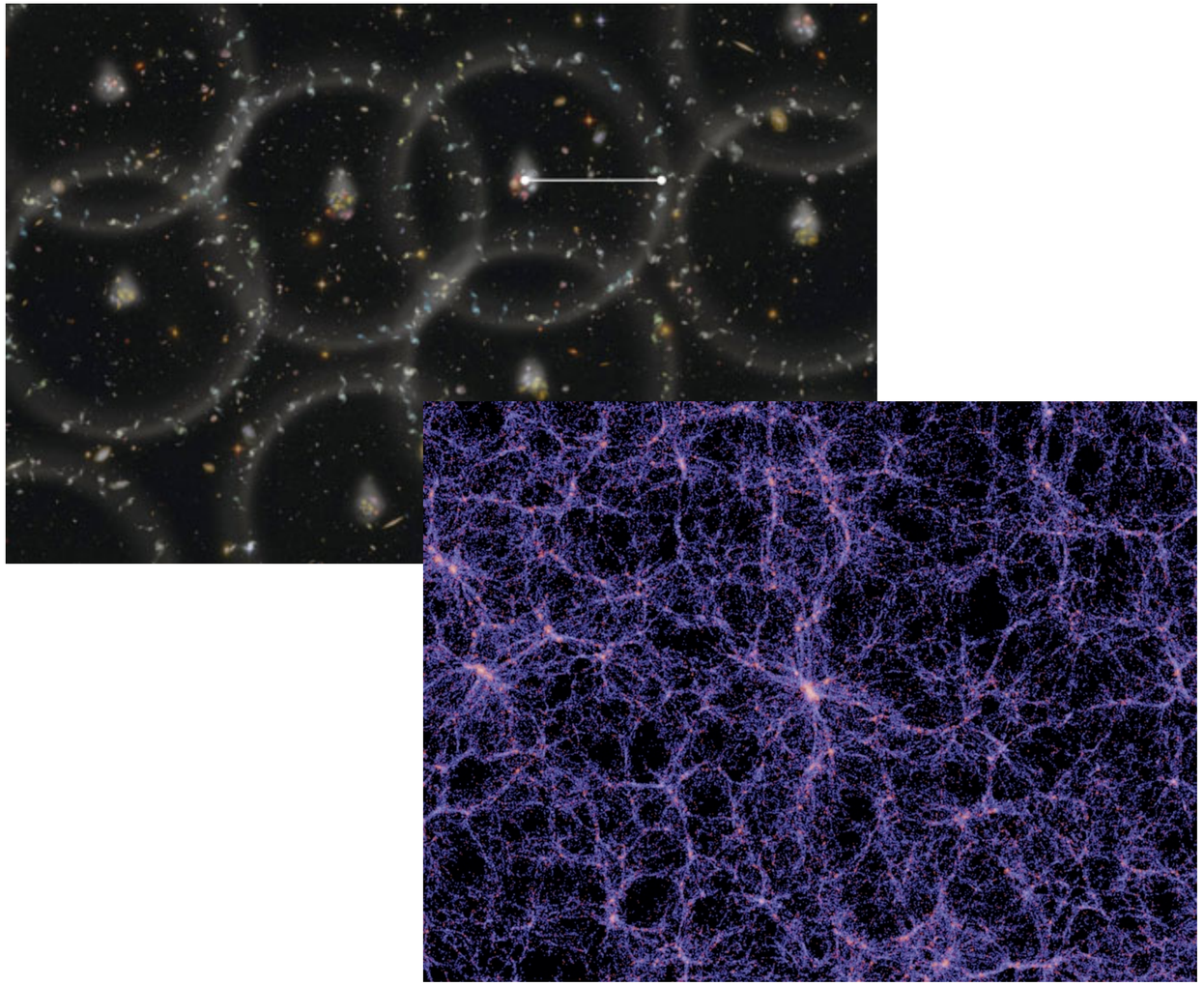


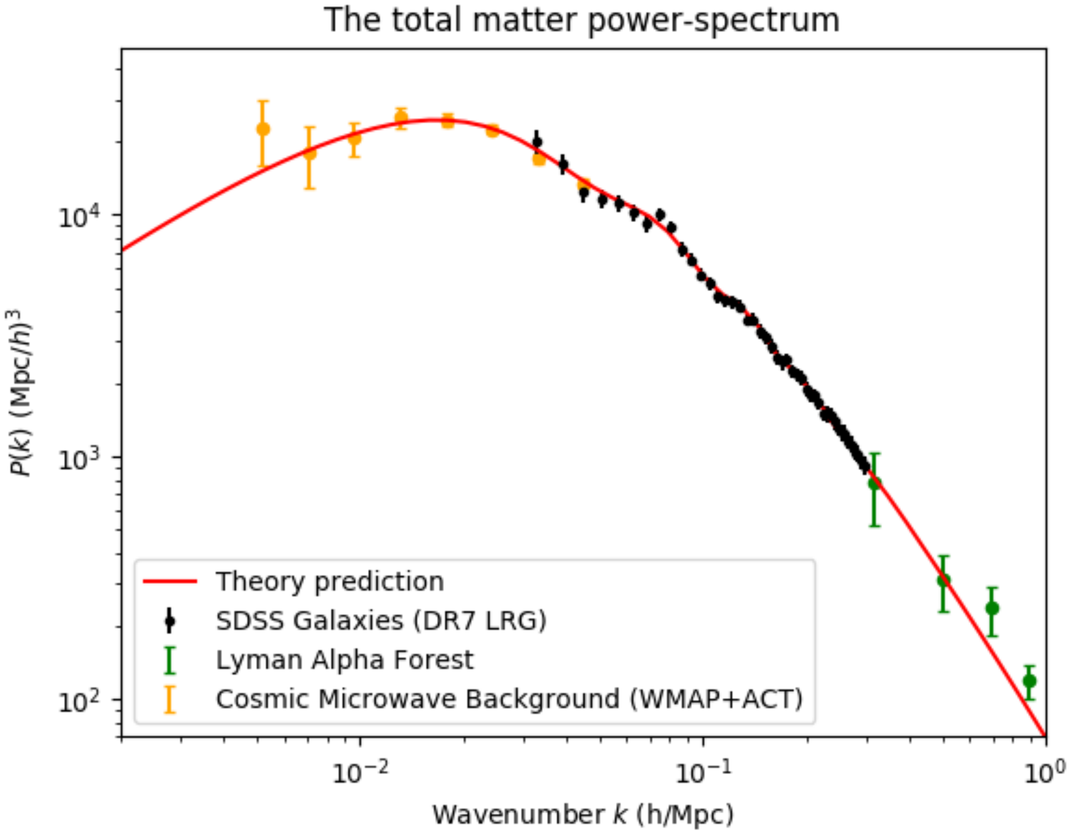
# Alternatives?

- Modified gravity
- Non-standard primordial physics
- Dynamical Dark Energy
- …



# Your turn!




The total matter power-spectrum
$P(k)$ (Mpc/$h$)$^3$
$10^4$
$10^3$
$10^2$
$10^{-2}$
$10^{-1}$
$10^0$
Wavenumber $k$ (h/Mpc)
Theory prediction
SDSS Galaxies (DR7 LRG)
Lyman Alpha Forest
Cosmic Microwave Background (WMAP+ACT)

# Your turn!

## 2. Exercise Tasks

### ✅ Task 1:

Split the dataset into train and validation sets.

### ✅ Task 2:

Build and train a neural network classifier.

Suggested architecture:

- Dense(64, relu)
- Dense(64, relu)
- Dense(1, sigmoid)

Train for 40 epochs.

### ✅ Task 3:

Evaluate the learning curves, plot ROC, AUC and the confusion matrix.

### ✅ Task 4 (important):

Normalize the inputs using `StandardScaler` and retrain the model, what happens?

# Gracias!