Workshop

AI in a nuthsell

Kamel Brouthen

- CS Student at ESI Algiers
- GDG Algiers Dev Core Team Member (AI)
- SOAI Algiers Technical Manager

What is AI?

Artificial Intelligence is the **science** and **art** of making computers **smart**!

How to do AI?

1. Problem Statement
2. Data Collection
3. Data Pre-processing
4. Data Modelling
5. Model Validation
6. Deployment

AI in action

Problem Statement

Diabetes Health Indicator 1/0

BMI

Smoking

Blood Pressure

Age

Features

Diabetes Health Indicator

Gender

Cholesterol

Physical Activity

Income

Building Intuition

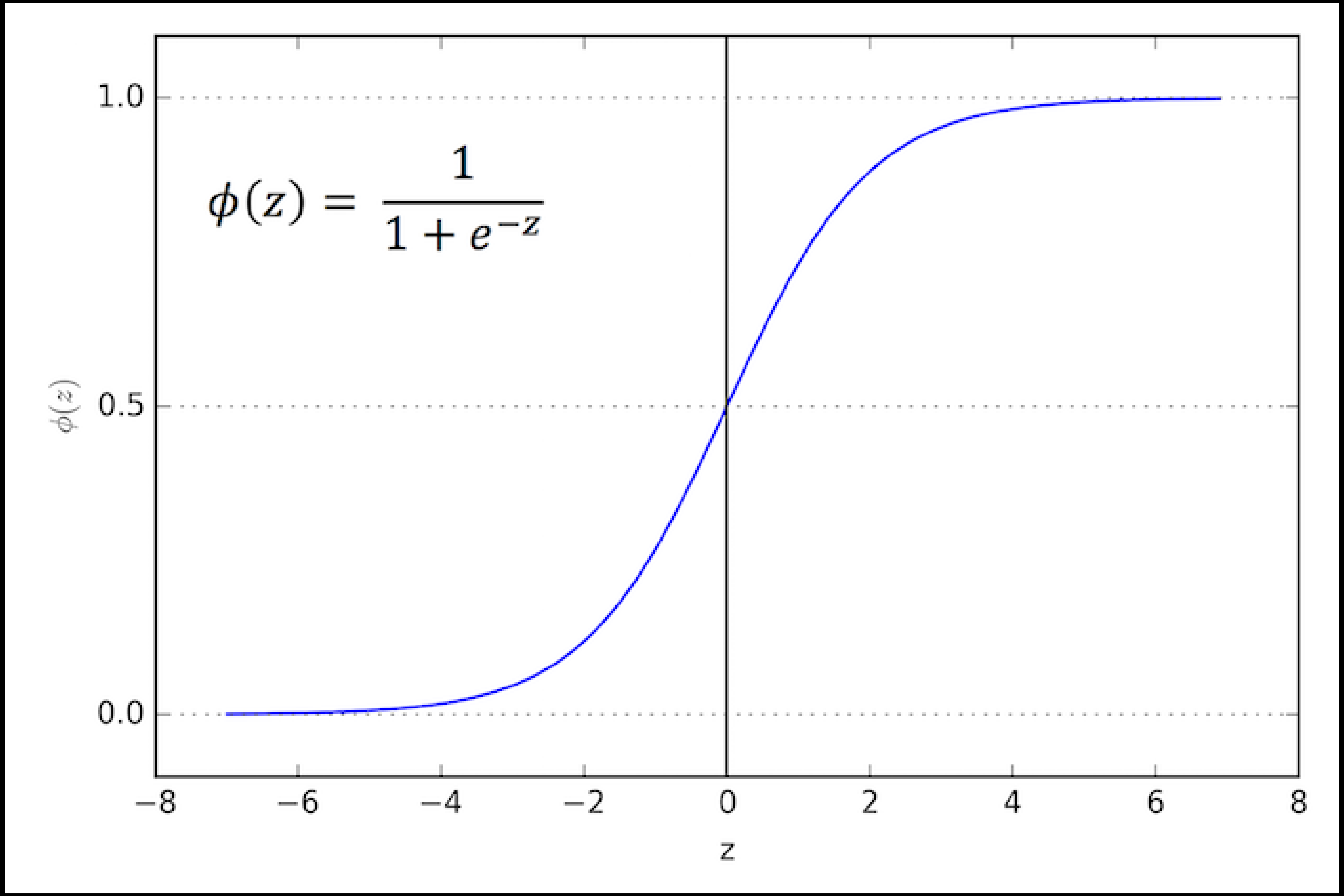
Target

Does the person have diabetes ? 1/0

Features

BMI	Smoking	Blood Pressure	Age	Gender
w1	**w2**	**w3**	**w4**	**w5**
Cholesterol	Physical Activity	Income		
w6	**w7**	**w8**		

Sigmoid Activation Function



Prediction

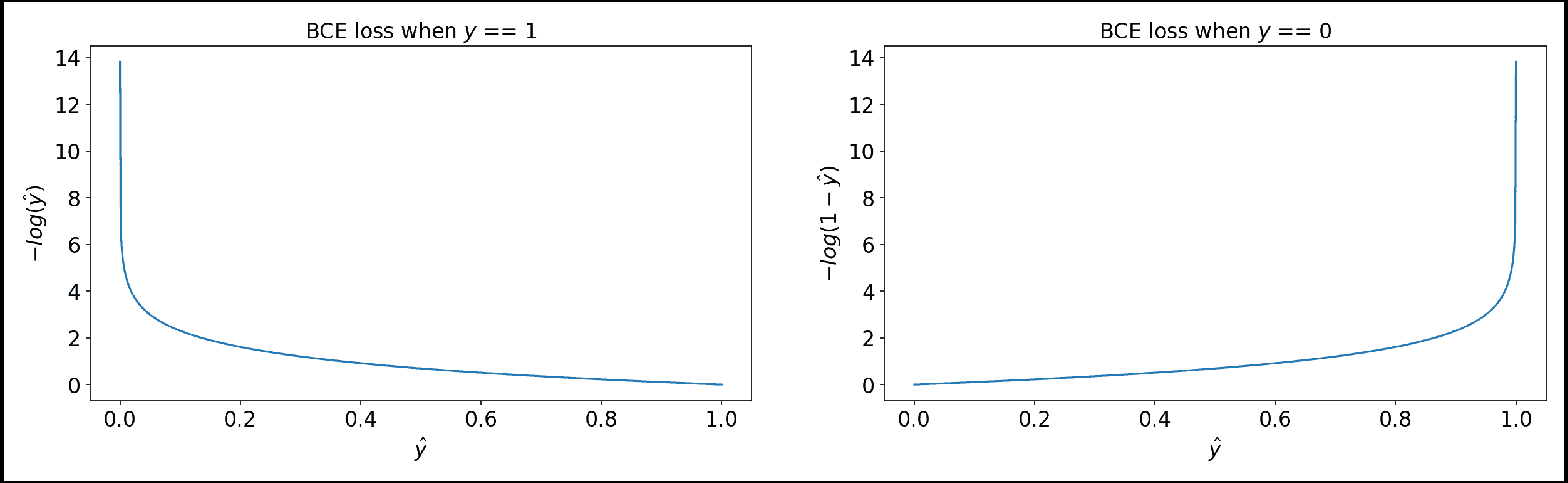
The **prediction** will be evaluated by **weighting** every **feature** in our input additional to a **baseline** value

$$\text{Prediction } (y) = \text{sigmoid}(\, w1 * x1 + w2 * x2 + \ldots + wn * xn + b\,)$$

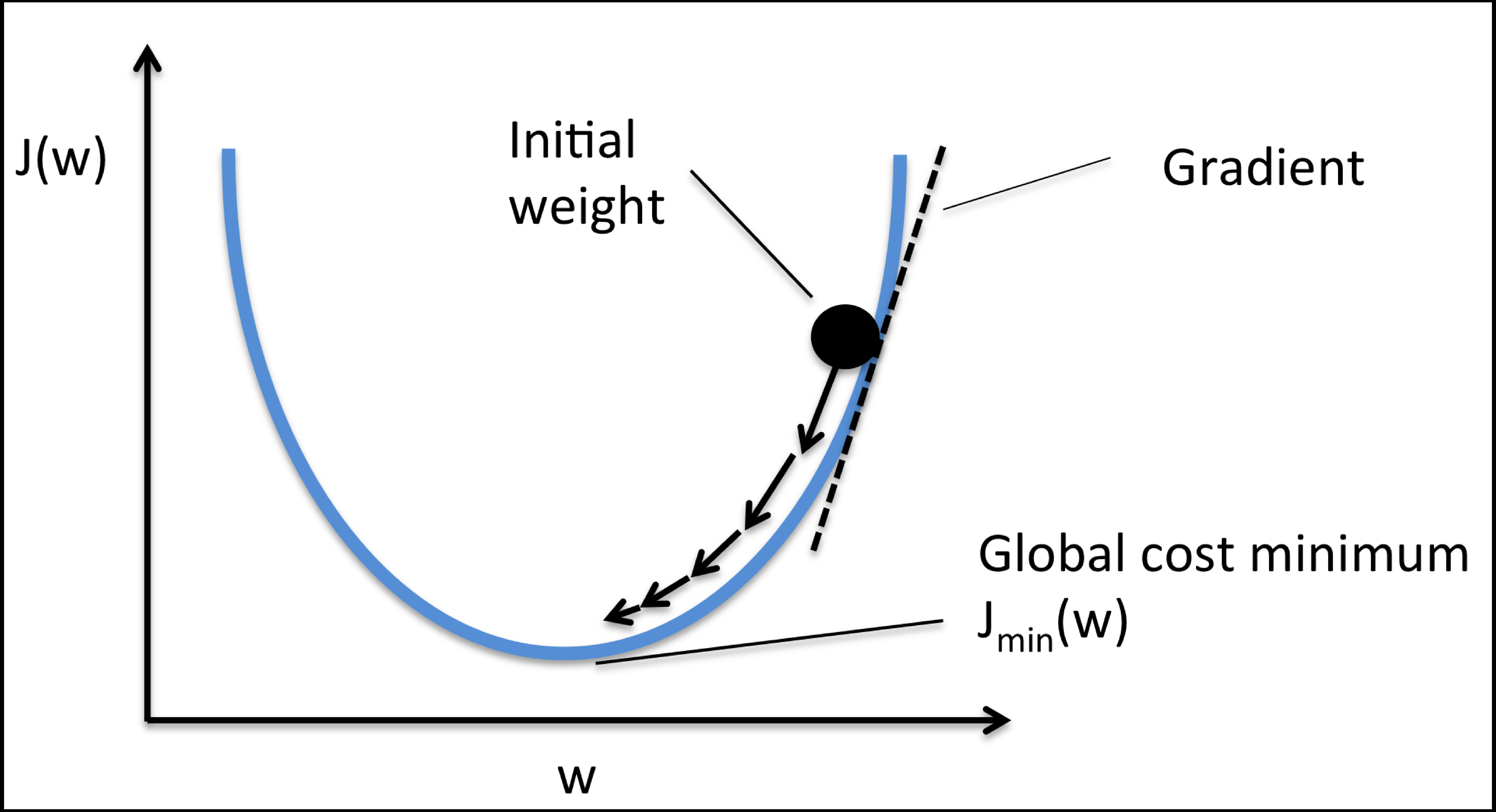
Loss Function

Binary Cross Entropy

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$



Gradient Descent

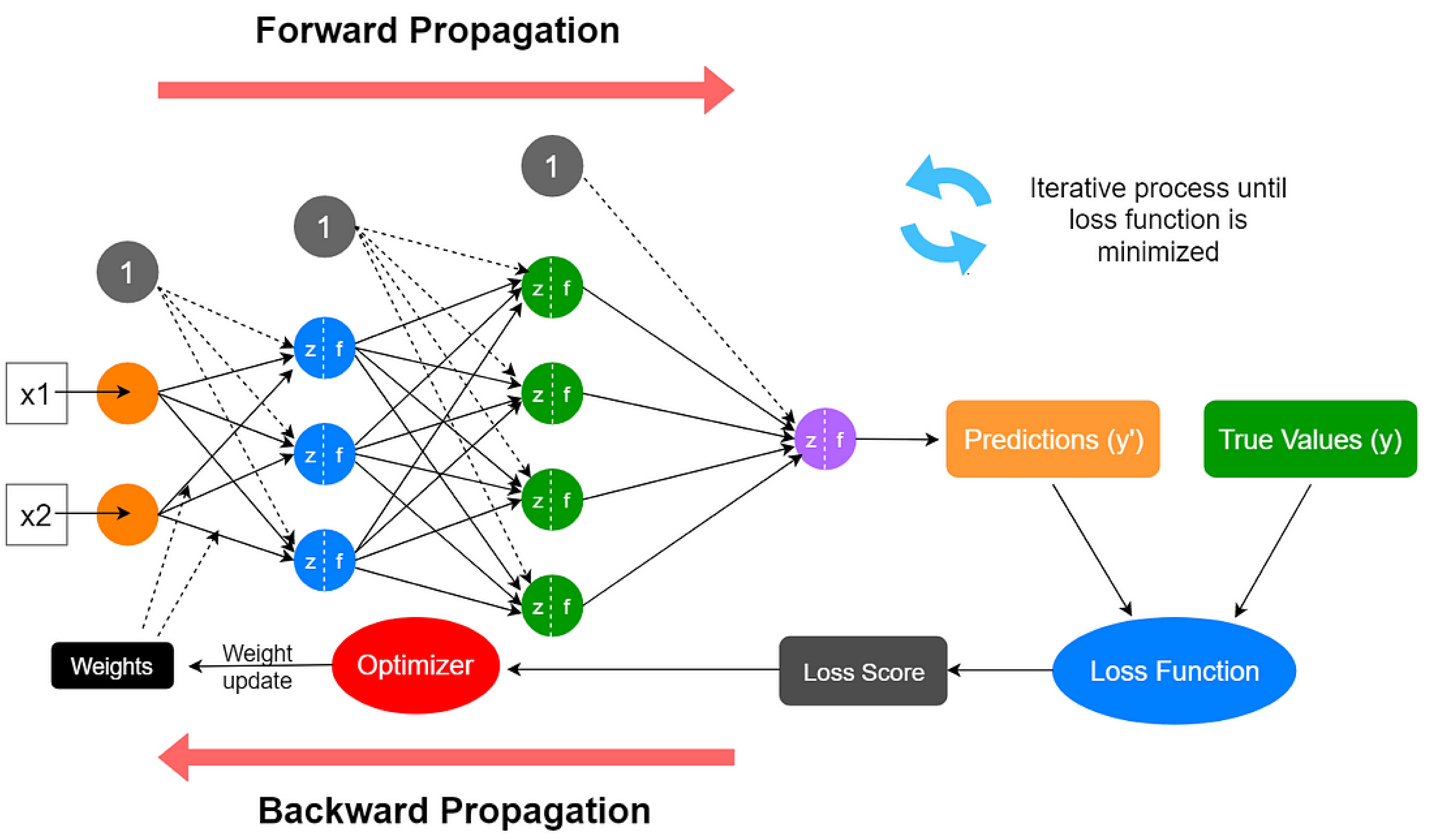


Model Training

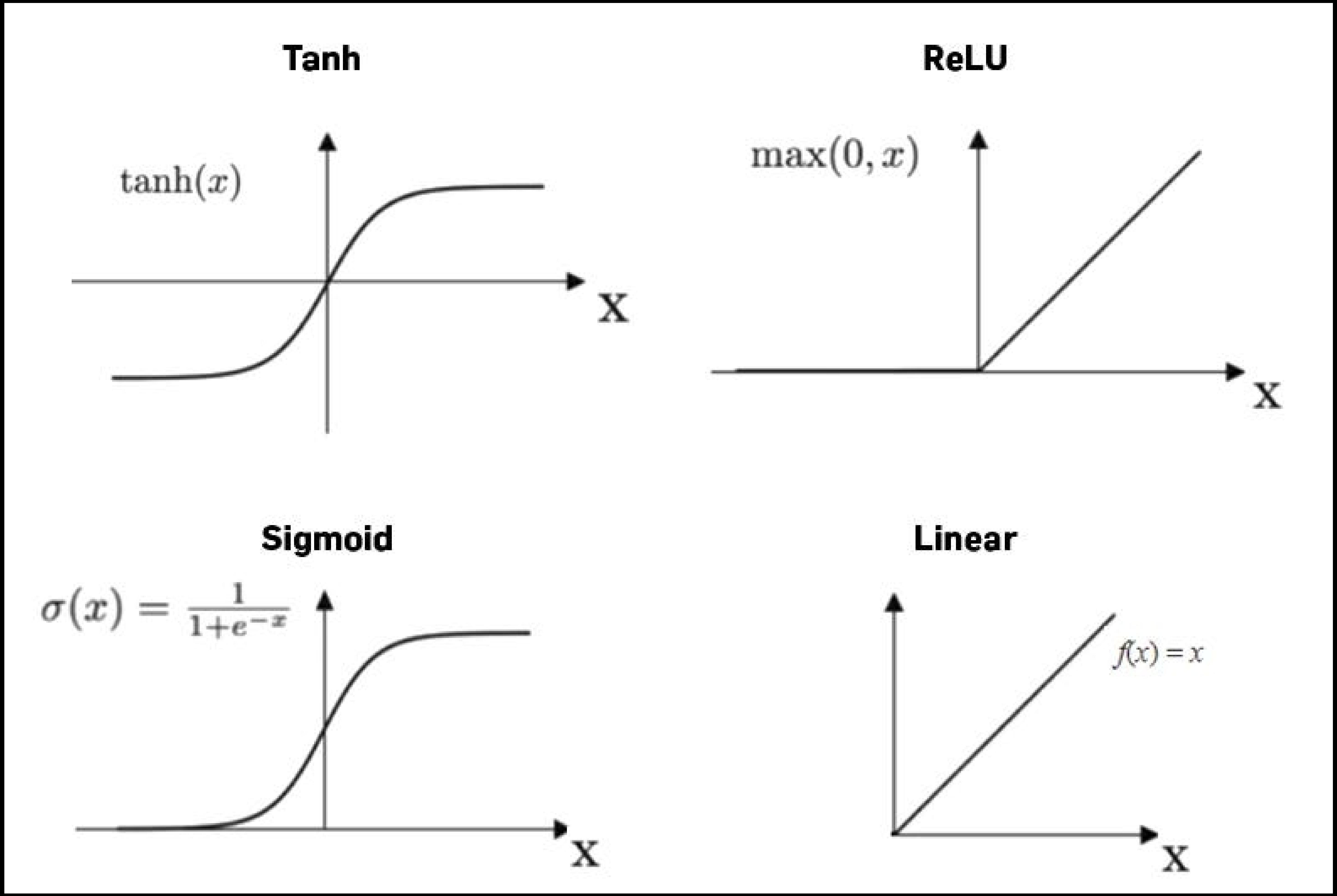
- W, b initialized randomly, learning_rate
- For every epoch:
  - For every pair (X,y):
    - Prediction = sigmoid(WX + b)
    - Loss = BCE(Y, Prediction)
    - For every Parameter:
      - Parameter -= learning_rate x gradient

Deep Learning

Problems can become harder implying complex data patterns and representations

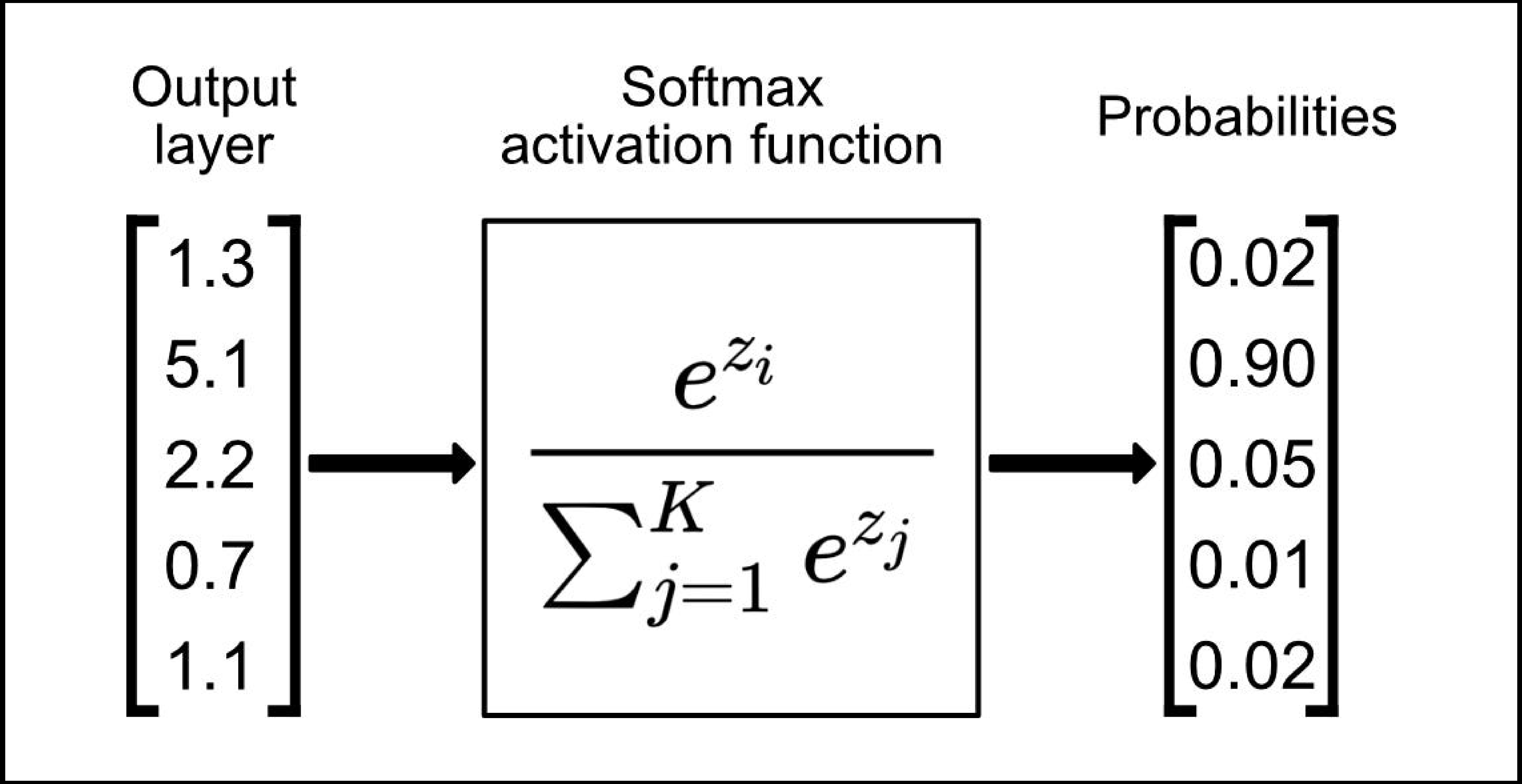

Forward Propagation
Iterative process until loss function is minimized
1
1
1
x1
x2
z f
Predictions (y')
True Values (y)
Weights
Weight update
Optimizer
Loss Score
Loss Function
Backward Propagation

Activation Functions



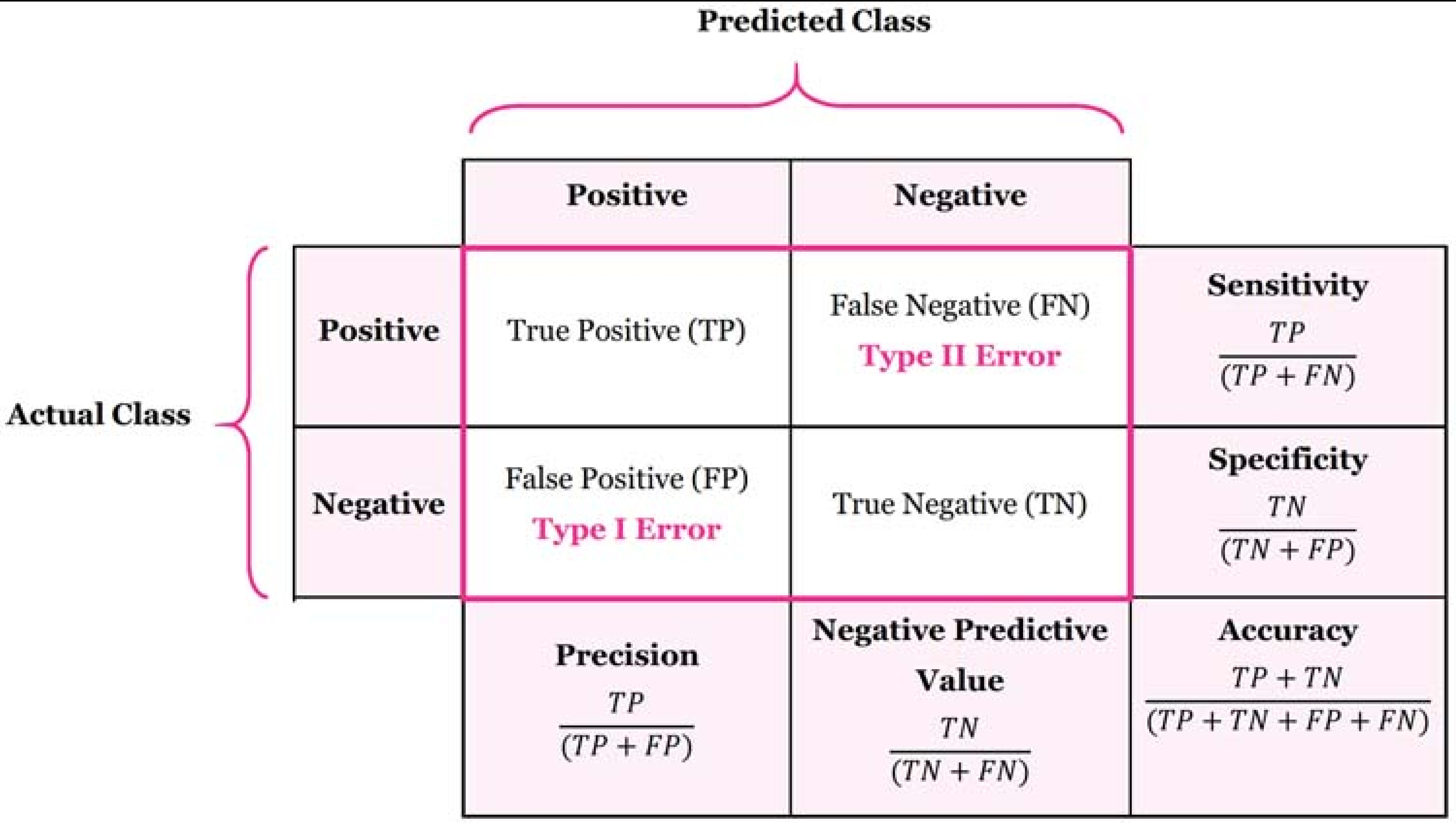
Softmax Activation Function

Generalizing to Categorical Classification



$$\begin{bmatrix}1.3\\5.1\\2.2\\0.7\\1.1\end{bmatrix} \rightarrow \boxed{\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}} \rightarrow \begin{bmatrix}0.02\\0.90\\0.05\\0.01\\0.02\end{bmatrix}$$

Model Evaluation



		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	**Negative Predictive Value** $\frac{TN}{(TN + FN)}$	**Accuracy** $\frac{TP + TN}{(TP + TN + FP + FN)}$

Code

The complete workflow

Google Collaboratory

Tensorflow

Beyond

AI in a nuthsell

Convolutional
Neural Networks

Computer Vision

Object Detection

Image
Classification

Recurrent
Neural Networks

GRUs, LSTMS

Natural Language Processing

Sentiment
Analysis

Machine
Translation

Large Language Models
(Transformers)

Generative AI

Image Generation
(DALLE, Stable Diffusion)

Environment

Agent

Reinforcement Learning

Observation

Reward

Constantine

Thank you!