# Introduction to Optimization

Jikai Wang

# What is Optimization?

- Definition
  - Optimization is the process of **adjusting the parameters** of a model to **minimize** the error (or loss) and improve its performance.
- Key Points
  - It involves finding the **best set of parameters** (weights) that result in the lowest possible loss.
    - Pose estimation: find the best hand/object pose to minimize the loss.
  - In the context of neural networks, this means tuning the weights of connections between neurons.

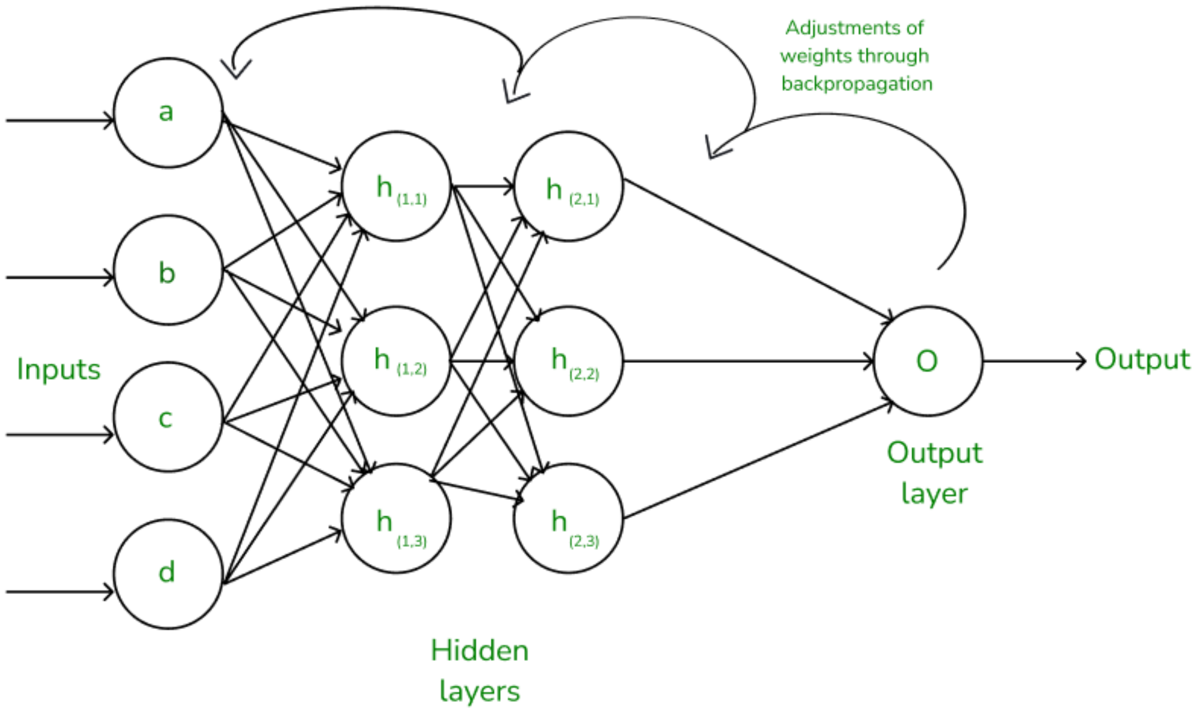
# What is Loss Function?

- A loss function measures how well the model's predictions match the actual target values.
- A common loss function is the Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values.
- The goal is to minimize the loss function during training.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(pred_i - true_i)^2$$

# Backpropagation in Neural Network

- **Backpropagation** is an iterative algorithm, that helps to **<u>minimize</u>** the cost function by determining which weights and biases should be adjusted. During every epoch, the model learns by adapting the weights and biases to minimize the loss by **moving down toward the gradient** of the error.

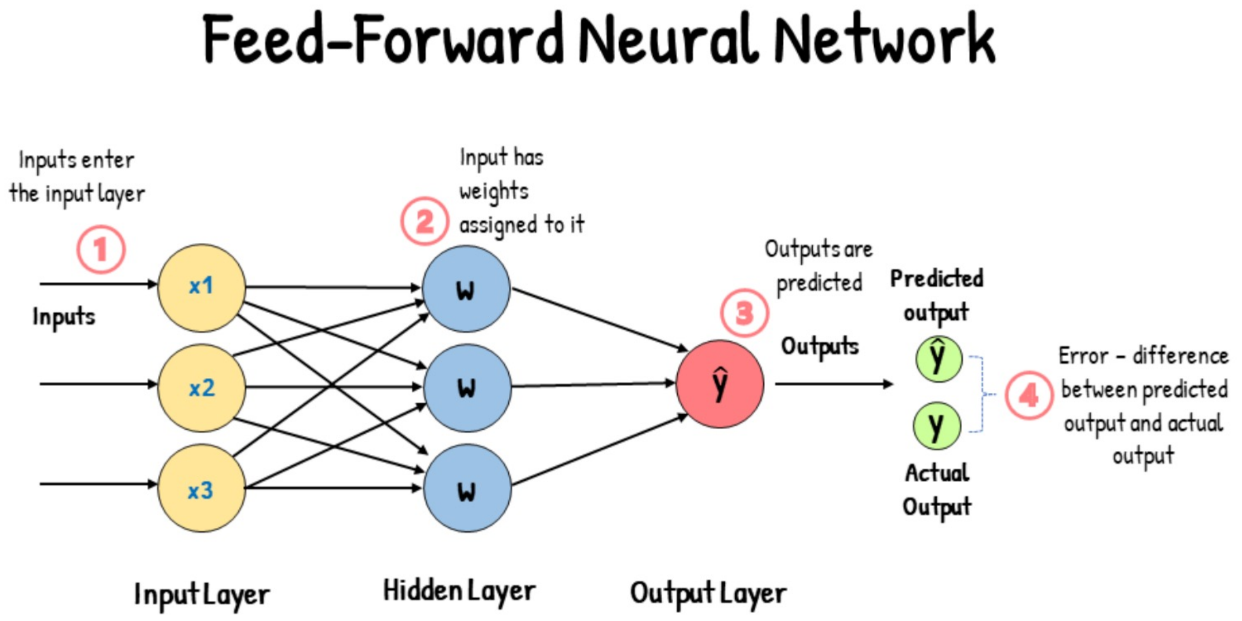


# How Backpropagation Algorithm Works

- The Backpropagation algorithm works by two different passes
  - Forward pass
  - Backward pass

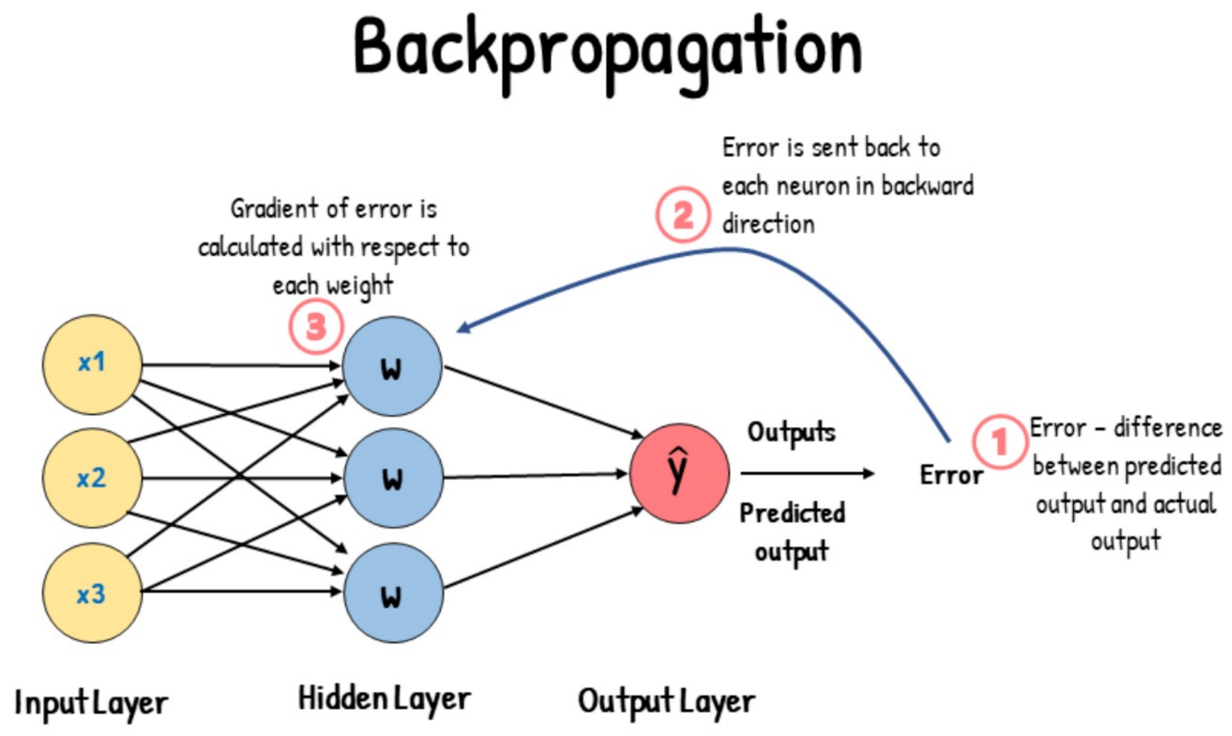
# Backpropagation - Forward pass

- Initially, the input is fed into the input layer.
- The inputs and their corresponding weights are passed to the hidden layer.
- Finally, the weighted outputs from the last hidden layer are fed into the output to compute the final prediction.



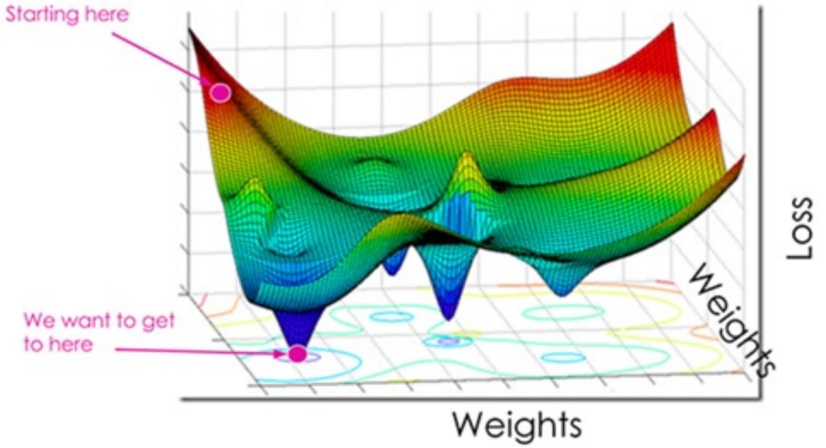
# Backpropagation - Backward pass

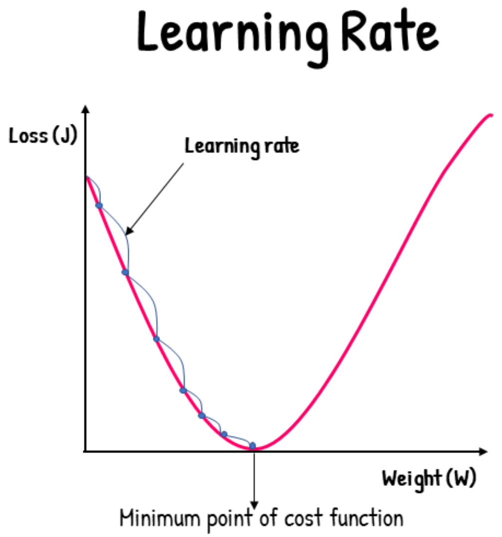
- In the backward pass process, the error is transmitted back to the network which helps the network, to improve its performance by learning and adjusting the internal weights.
- The weights are adjusted using a process called **gradient descent**.
- To find this weight, we must navigate down the cost function until we find its **minimum point**.



# Backpropagation - Gradient Descent & Learning Rate

- Gradient Descent
  - **Gradient Descent** is an optimization algorithm that is used to find the weights that minimize the cost function. Minimizing the cost function means getting to the minimum point of the cost function. So, gradient descent aims to find a weight corresponding to the cost function's minimum point.
  - **Learning Rate** is a tuning parameter that determines the step size at each iteration of gradient descent. It determines the speed at which we move down the slope.



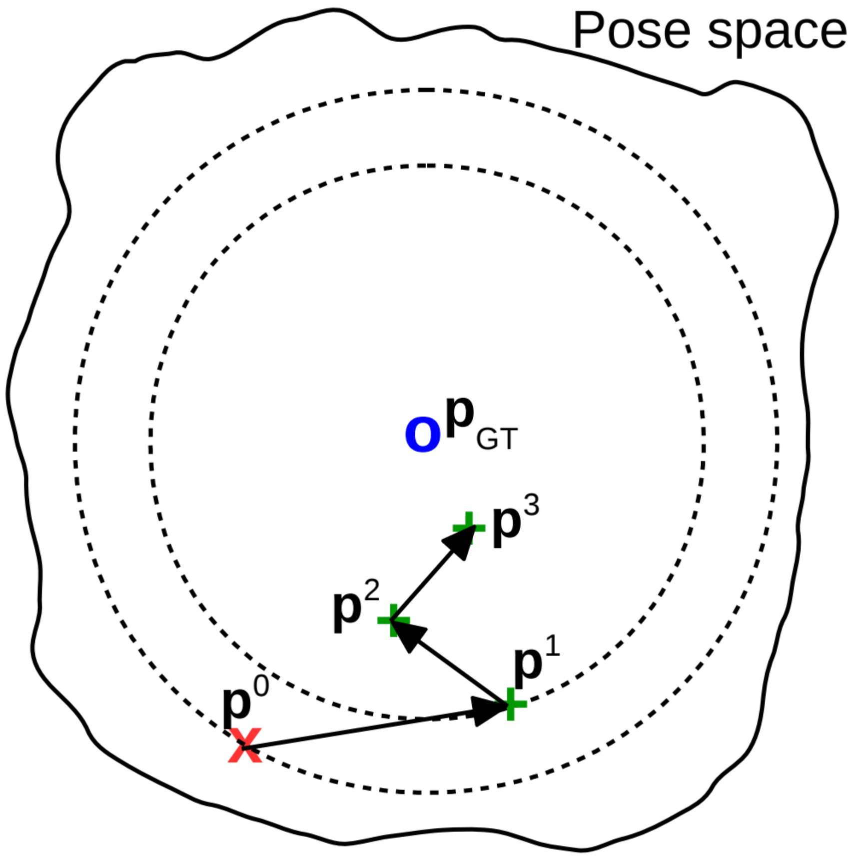



# Gradient Descent Optimization Algorithms

- A gradient descent optimization algorithms updates the model's parameters to minimize the loss function.
- Different Variants of Gradient Descent
    - Batch Gradient Descent
    - Stochastic Gradient Descent (SGD)
    - Mini-batch Gradient Descent
    - Momentum-based Gradient Descent
    - Nesterov Accelerated Gradient (NAG)
    - Adagrad
    - RMSprop
    - Adaptive Moment Estimation (Adam)
        - it combines the benefits of Momentum-based Gradient Descent, Adagrad, and RMSprop
- Visualize Optimization

# Pose Optimization

- Start at an initial pose (x). And want to converge to the ground truth pose (o), that minimizes the loss.
- Loss
  - $L_{2D}$: the difference between the target 2D keypoints and projected keypoints.
  - $L_{3D}$: the difference between the target 3D keypoints and predicted 3D keypoints.
  - $L_{SDF}$: measure the distances of surrounding points to the posed mesh.
  - $L_{smooth}$: punish the jitter poses across the whole sequence.
- Optimization Algorithm
  - Adam
  - Learning rate 0.001



THE UNIVERSITY OF TEXAS AT DALLAS

# Thank You