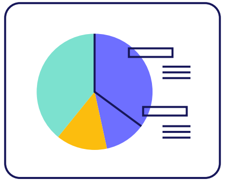
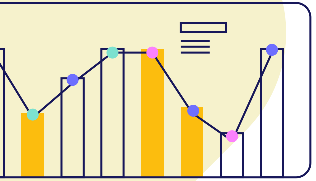
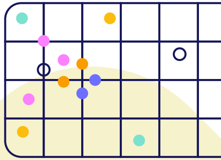


Machine Learning with Python

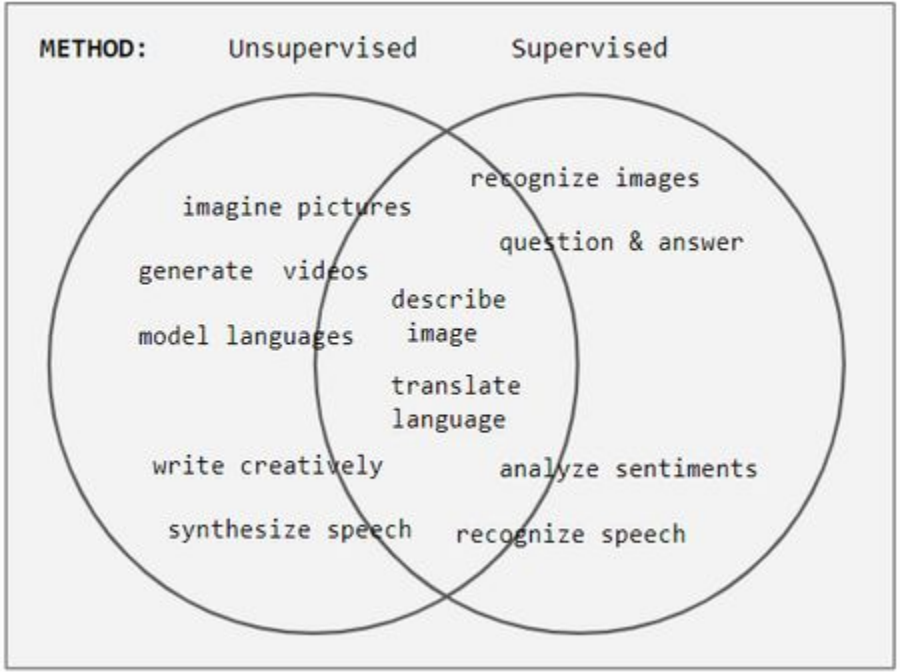
Dellie Wright and Dylan Kneidel





ML – Our Focus

Genetic Algorithms **Neural Networks**



METHOD: Unsupervised Supervised

recognize images

imagine pictures

question & answer

generate videos

describe image

model languages

translate language

write creatively

analyze sentiments

synthesize speech

recognize speech

Reinforcement Learning*

What Are Genetic Algorithms?

Algorithms Inspired by Natural Selection

- Solve Complex Problems by Mimicking the Process of Evolution
- Improves a population of potential solutions iteratively

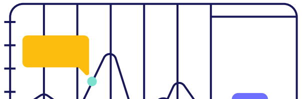
Genetic Algorithms explore the solution space, searching for the best one!

Used For:

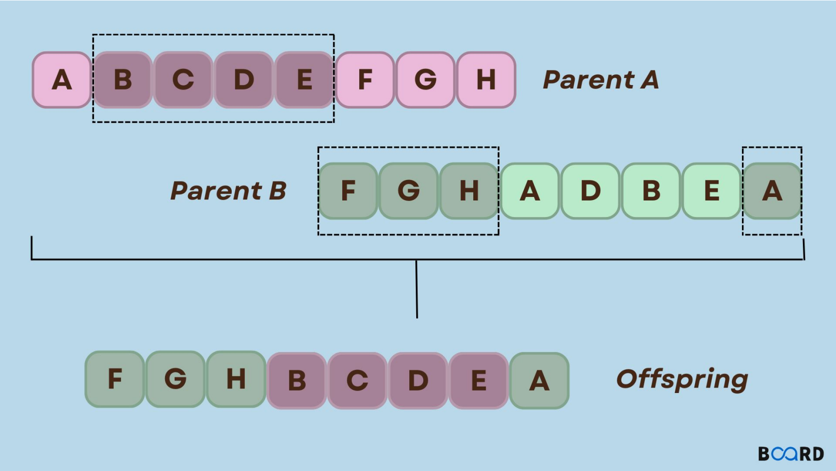
- Optimization Problems - FInding the best solution among a large set of possibilities
    - job scheduling, traveling salesman problem, traffic problems
- Machine Learning - Optimization of Machine Learning Models
- Financial Modeling - algorithmic trading, risk management, financial forecasting

Common Examples:

- Solving Sudoku Puzzles
- Tesla Self-driving
- Amazon’s Routing and Scheduling Problems



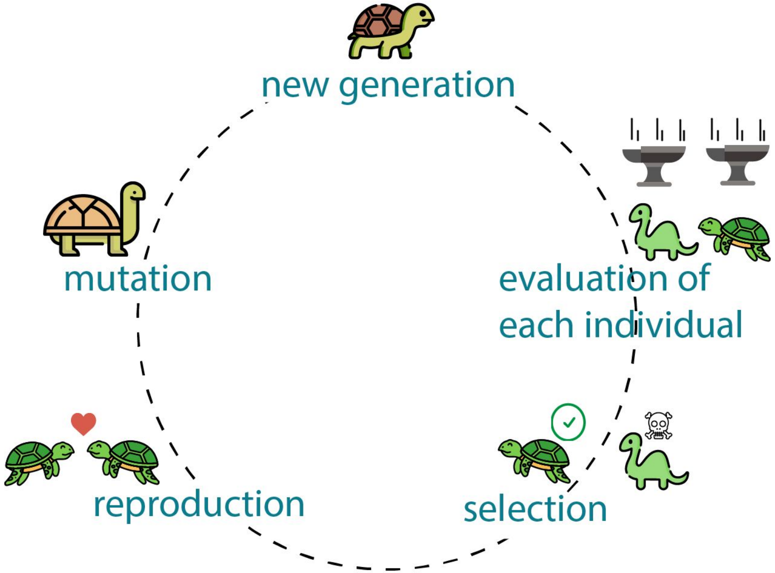
How Do Genetic Algorithms Work?



Genetic algorithms learn through darwinism.

- Individuals
- Genes
- Fitness
- Populations
- Evolutions
- Mutations

How Do Genetic Algorithms Work?



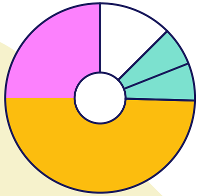
Our Example – Guessing Strings

For our example, we will use a genetic algorithm to guess a specific string!

Individual	Object with genes
Genes	Randomly generated strings
Fitness	“Closeness” of string to target
Population	Collection of Individuals with Genes
Mating	Combine two individuals to create a new one
Evolution	Building a new population by “mating”

Guessing Strings – Let’s Code Together!

1. Randomly Initialize an initial population
    a. Randomly initialize Individuals
    b. Randomly initialize a list of individuals
2. Determine Fitness of Individuals
3. LOOP until Target String:
    a. Select good parents from population
    b. Mate Parents to generate new population
    c. Add random mutations to new population



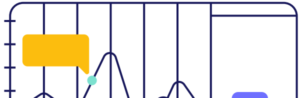
Our Example - Guessing Strings

Now that we’ve built this out, how do we improve generations?

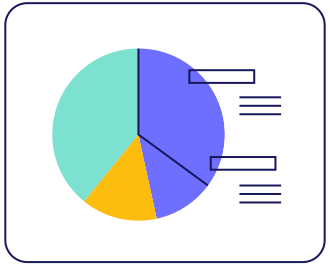
- **Choose best individuals**
- **Natural Selection**
- **Mate Best Individuals**

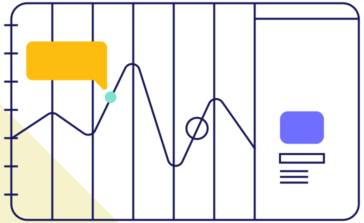
We’ve successfully created evolution!

Let’s implement it…



Important Math Libraries





Numpy

Overhauls the usage of arrays and adds more operations and finer-tuned control of said arrays

MatPlotLib

Helps with plotting graphs and images, essential to understand what data is being worked on

Pandas

Data analysis tool that packs our data into dataframes, easily exportable to excel files etc.

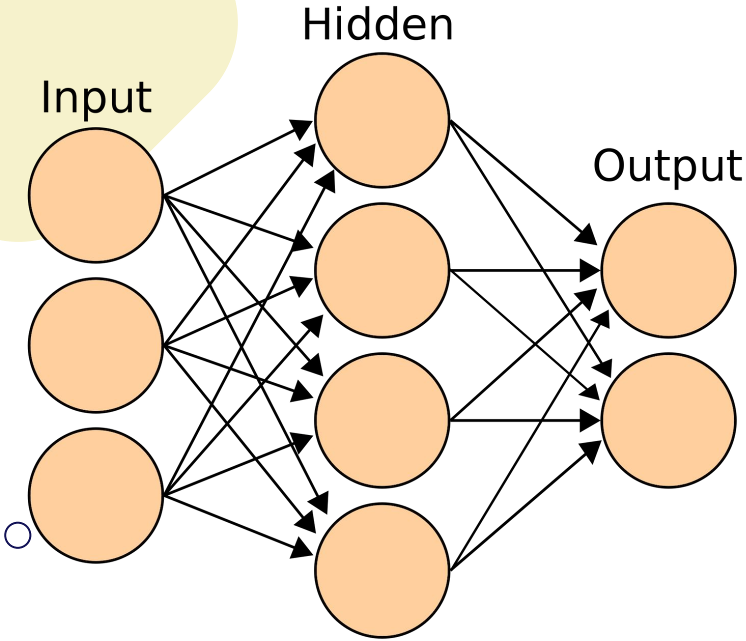
Important ML Libraries

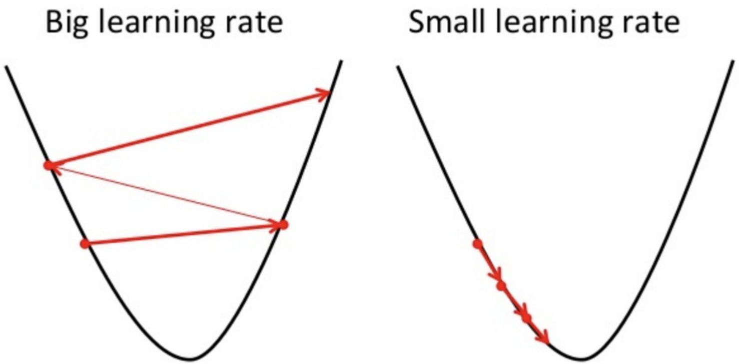
PyTorch

TensorFlow

Both of these libraries are widely supported and used by the community (and large companies!)

- Standard datasets for easy validation
- Preprocessing for data loading and formatting
- Support for web and mobile, as well as distributed computing
- Easy access on the cloud! (Microsoft Learn, Google Colab)





What is a Neural Network?

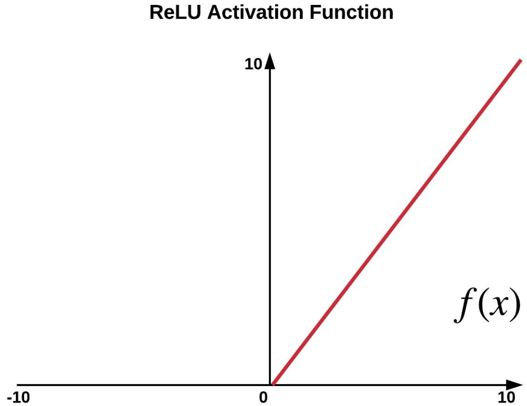
Neural networks are a form of *supervised learning*, which needs both the problem and the solution.

- Weights/Biases (Input * Weight + Bias)
- Activation Function
- Propagation
  - Loss Function
  - Gradient Descent
  - Learning Rate
- Multilayer Perceptron 😵‍💫

Our Example – Classifying Clothes

For our example, we will use PyTorch to classify types of clothes!

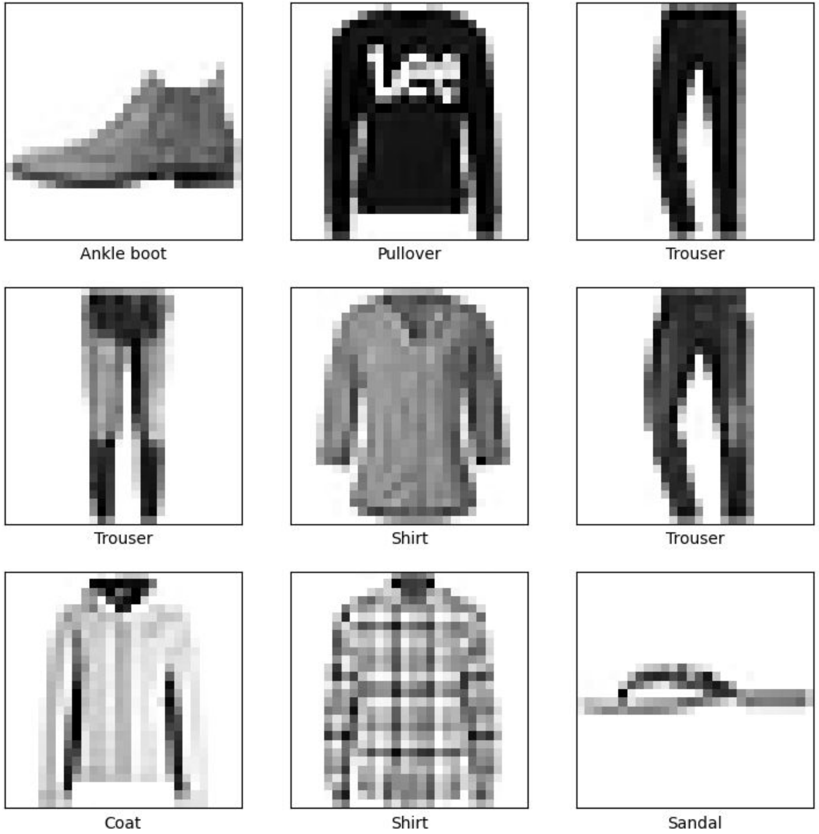
Dataset	FashionMNIST: 28×28 B/W images of clothes
Activation Function	ReLU (Rectified Linear Unit)
Loss Function	Cross Entropy (common for classification models)
Optimizer Function	Stochastic Gradient Descent (SGD)
Batch Size	64 Samples

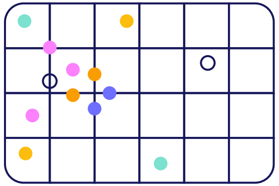


$$f(x) = max(0, x)$$

Guessing Clothes – Let’s Code Together!

1. Download and organize training data
2. Create iterables over data
3. Select processing device
4. Create our NN subclass!
5. Set the loss/optimizer functions
6. Create the train function
   a. Run forward prop with model.train()
   b. Find the error in prediction
   c. Run back prop with loss.backward()
   d. Rinse and repeat for all batches
7. Create the test function
   a. Sum up the correct guesses and loss
8. Run the model on our data!
   a. Train/test over x epochs (iterations)





Thanks!

Do you have any questions?

Further Learning!

01.

Machine Learning Specialization

https://www.deeplearning.ai/courses/machine-learning-specialization/

02.

Neural Networks - Zero to Hero

https://www.youtube.com/watch?v=gUmagAluXpk&t=343s

03.

Genetic Algorithms Introduction & Code

https://www.geeksforgeeks.org/genetic-algorithms/

04.

PyTorch Quickstart (what the NN part was based off of !)

https://pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html