# Artificial Intelligence - Learning

CSCI 1030U - Intro to Computer Science
@IntroCS

Randy J. Fortier
@randy_fortier

Ontario Tech University

# Outline

- Machine learning
  - Unsupervised
  - Supervised
- Neural networks
- Genetic algorithms
- Bayesian networks

# Machine Learning

"Suddenly the machine just knew what it had to do: It had to fail the Turing test on purpose."

- Mikko Hyppönen

# Machine Learning

# Machine Learning

- Search-based methods involve encoding human (or non-human) methods of solving a problem into an algorithm
- Machine learning, in contrast, aims to let the machine learn how to solve the problem on its own
  - The developer prepares a (large) set of training data for the machine
  - The machine looks for patterns in the training data
  - Using those patterns, the machine tries to solve problems it hasn't seen before

# Machine Learning

- One way to categorize ML models:
  - Classifier
    - There are two or more classes (e.g. spam, ham)
    - The classifier tries to choose which class to which a given input belongs
    - e.g. sentiment analysis (which mood is likely for a given message?)
  - Predictors
    - Given historical data, predict a new data point
    - e.g. given survivability of a disease, predict the survival of a new patient
  - Clusterers
    - Finds data with relationships/similarities
    - Arguably the same as classifiers, but the classes are not known beforehand
    - e.g. given movies watched by Netflix customers, predict movies they will also like (based on what others have also watched)

# Machine Learning - Training

- Machine learning comes in two main forms:
    - Unsupervised learning
        - No clues are given.  The machine just examines data and looks for patterns (e.g. similarities)
        - e.g. a list of which users liked which TV shows on Netflix
        - The result may be a bunch of clusters, or similar/related things
    - Supervised learning
        - Training data includes the correct answers to help the machine distinguish each category
        - e.g. a list of spam and non-spam messages
    - We'll focus primarily on supervised learning in this lecture

# Machine Learning - Training

- Machine learning usually involves two stages of data:
  - Training data
    - A proportion (e.g. 80%) of the data available that is used during the learning phase
  - Test data
    - A proportion (e.g. 20%) of the data available that is used to evaluate the model

# Machine Learning - Evaluation

- Evaluation is necessary to understand the efficacy of your model
  - e.g. How accurate is this test for Alzheimer’s?
- Results:
  - True positive - We predicted positive, and it was positive
  - True negative - We predicted negative, and it was negative
  - *False positive* - We predicted positive, but it was negative
    - Unnecessary tests, costs, potential pain and suffering
  - *False negative* - We predicted negative, but it was positive
    - Missed diagnosis, potential complications, no treatment

# Machine Learning - Evaluation

- Measures used:
  - Precision - a measure of statistical variability
  - Recall - a measure of sensitivity, true positive rate
  - Specificity - a measure of true negative rate

# Machine Learning - Evaluation

- Measures used:
  - Precision - a measure of statistical variability

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

  - Recall - a measure of sensitivity, true positive rate
  - Specificity - a measure of true negative rate

# Machine Learning - Evaluation

- Measures used:
  - Precision - a measure of statistical variability
  - Recall - a measure of sensitivity, true positive rate

$$\text{recall} = \frac{(TP)}{(TP+FN)}$$

  - Specificity - a measure of true negative rate

# Machine Learning - Evaluation

- Measures used:
    - Precision - a measure of statistical variability
    - Recall - a measure of sensitivity, true positive rate
    - Specificity - a measure of true negative rate

$$\text{specificity} = \frac{(TN)}{(FP+TN)}$$

# Machine Learning - Evaluation

- Researchers also often summarize their results with a single, calculated, metric:

$$\text{f1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

# Machine Learning - Training Data Bias

- Companies have plans to use ML for many purposes:
  - Approving people for loans
  - Shortlisting candidates for a job
  - Calculating insurance rates
  - Approving health claims
  - Choosing potential suspects in a crime

# Machine Learning - Training Data Bias

- Companies have plans to use ML for many purposes:
  - Approving people for loans
  - Shortlisting candidates for a job
  - Calculating insurance rates
  - Approving health claims
  - Choosing potential suspects in a crime

- Given that the data used to train these models was created by humans, can you see any issues that may present themselves for these problems?

# Machine Learning - Ethics and Law

- Quite a few AI models have been trained on copyrighted data without the creators' permission
  - Dall-E
  - GPT
  - Copilot
- Considering that current AI models essentially remix existing content from its training data, this could be considered derivative work
  - There are lawsuits currently being settled
- One could ask whether it is ethical to use such an AI

# Machine Learning - Explainability

- An active area of research within machine learning involves determining how a model came to its conclusions
- This might involve:
  - Visualizing the values within the network
  - Evaluating outputs from a series of inputs designed to target some intermediate conclusions
  - Trace dependencies between neurons for a particular input or set of inputs

# Machine Learning

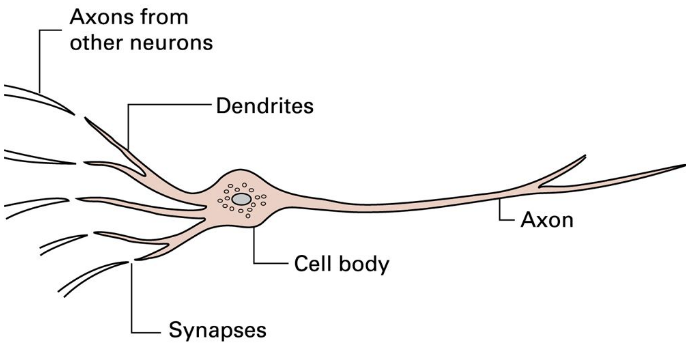
- Common machine learning techniques:
  - Artificial neural networks
    - The connection between neurons is reinforced by correct solutions
  - Genetic algorithms
    - Future solutions are based on the level of fitness of existing solutions
  - Bayesian networks
    - Probabilities are updated according to the actual frequency of events

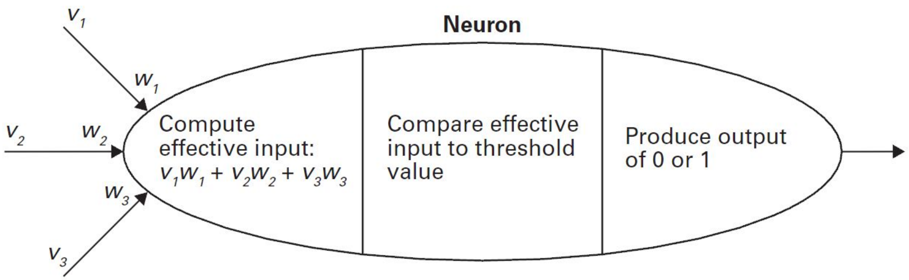
# Neural Networks

# Artificial Neural Networks

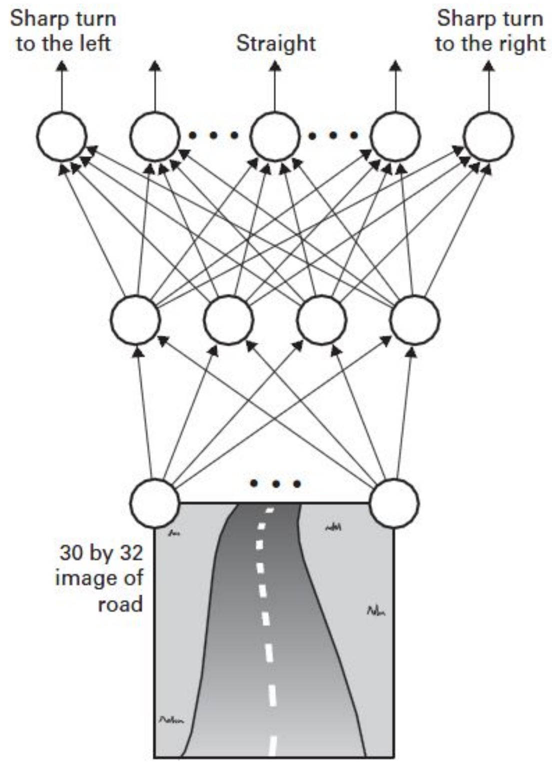
- Artificial neural networks use a simulation of neurons (brain cells) to solve problems
  - ANNs have been used to solve many problems:
    - Computer vision (e.g. stop sign recognition)
    - Decision-making (e.g. medical diagnosis)
    - Classifying data (e.g. is this message spam?)
    - Game-playing (e.g. blackjack)

# Artificial Neural Networks - Neurons

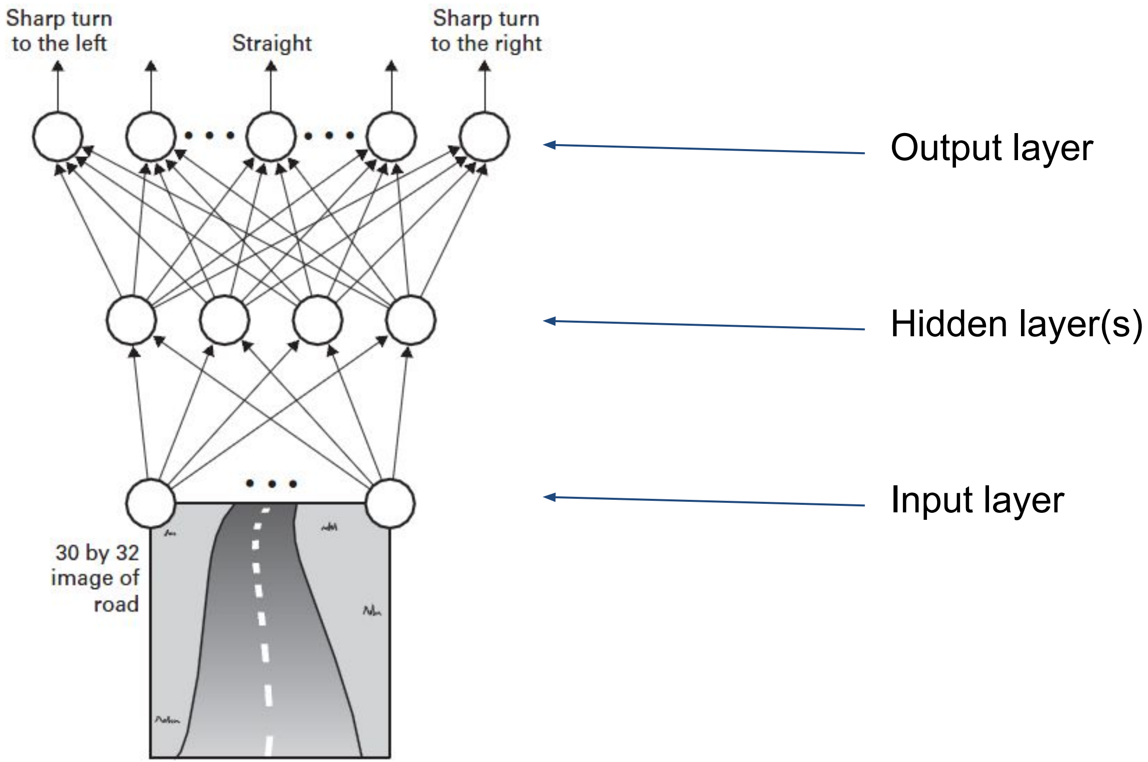


# Artificial Neural Networks

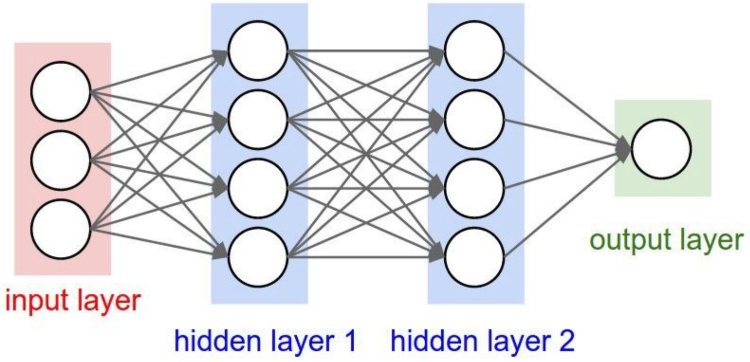


# Artificial Neural Networks



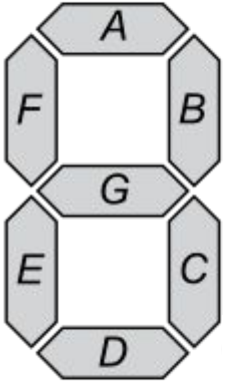
# ANNs - Forward Propagation

- The input layers values are combined by each neuron in the next layer, using their weights to create a weighted average
  - A bias value is also added to each input * weight term
- That weighted average is sent through some *activation function* in order to determine the output for that neuron



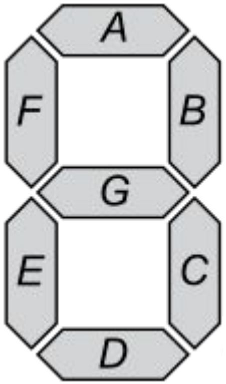
# Forward Propagation

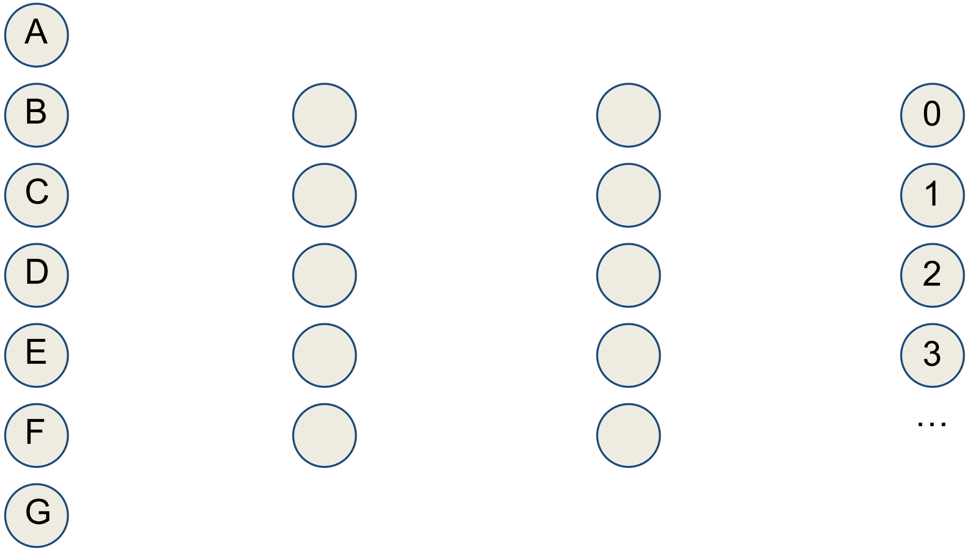
- Let's assume that we're trying to recognize a digit on a 7-segment display, like this:



# Forward Propagation

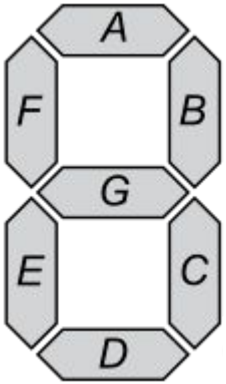
- Each input to our neural network might be whether or not each segment is lit up
  - Note that, in practice, these inputs would be imperfect
  - So, our inputs would be A, B, C, D, E, F, and G

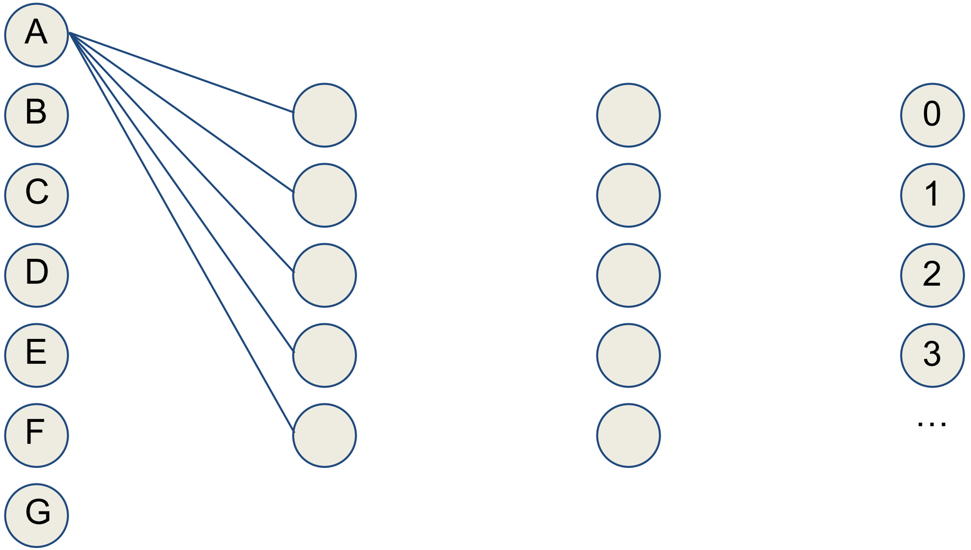


# Forward Propagation

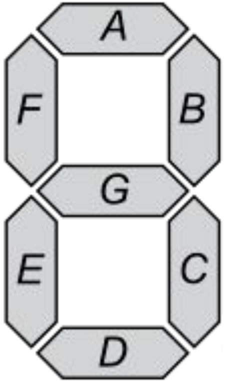
- Each neuron on the input layer feeds its output into each neuron on the first hidden layer
  - First, from A

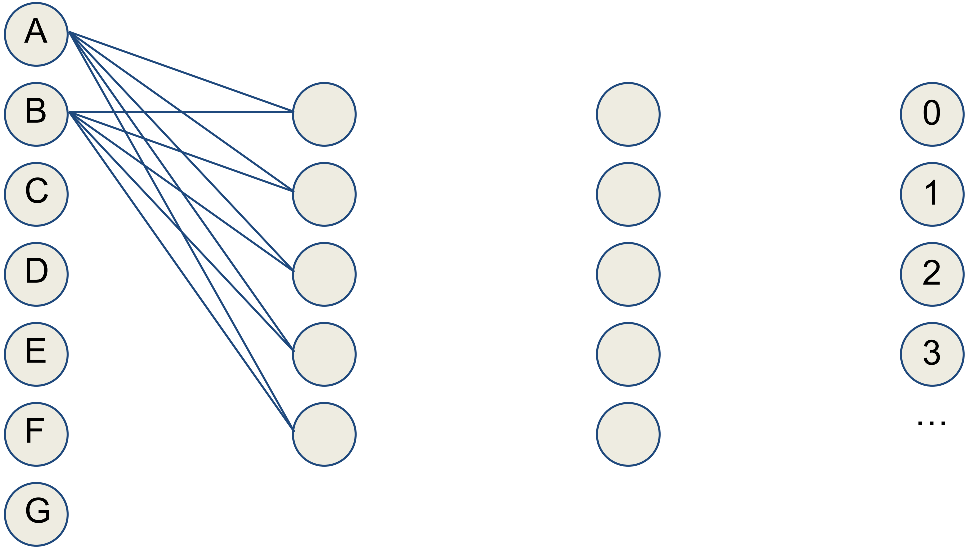


# Forward Propagation

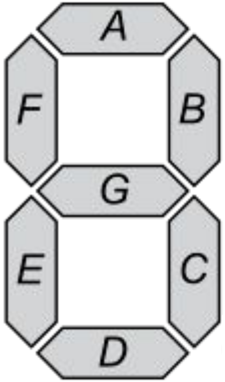
- Each neuron on the input layer feeds its output into each neuron on the first hidden layer
  - Then B

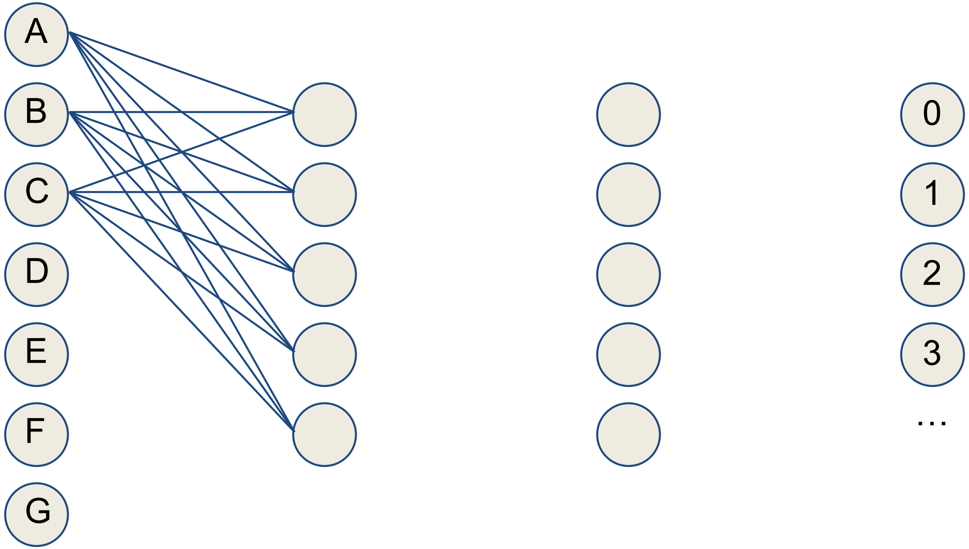


# Forward Propagation

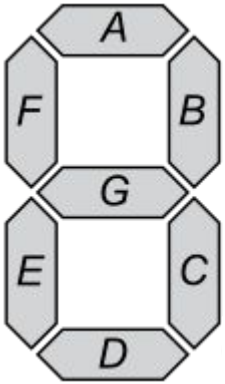
- Each neuron on the input layer feeds its output into each neuron on the first hidden layer
  - Then C

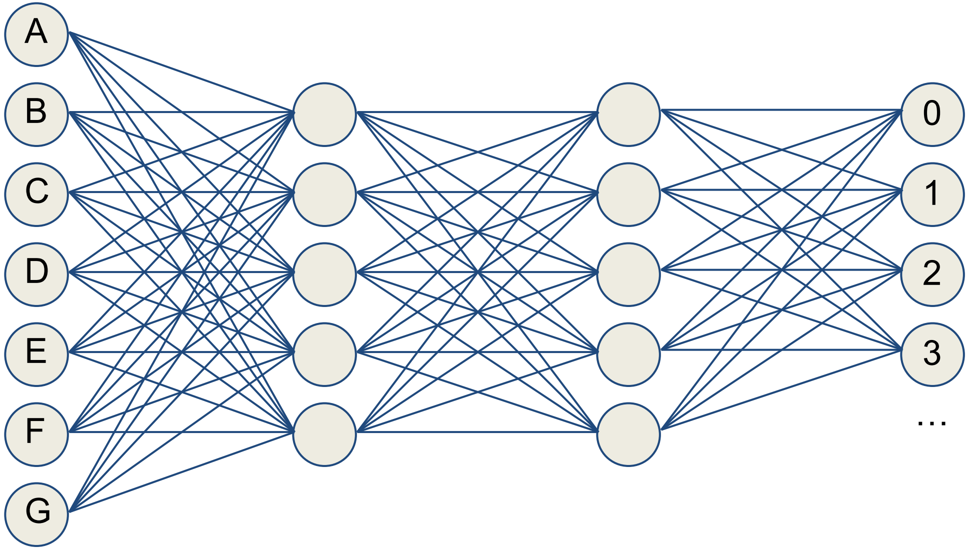


# Forward Propagation

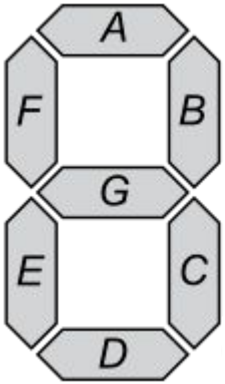
- Each neuron on the input layer feeds its output into each neuron on the first hidden layer
  - And so on for all of the inputs

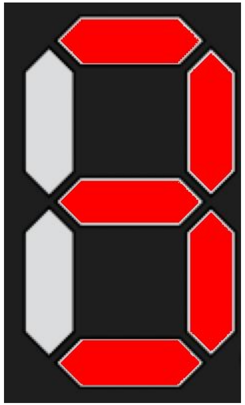


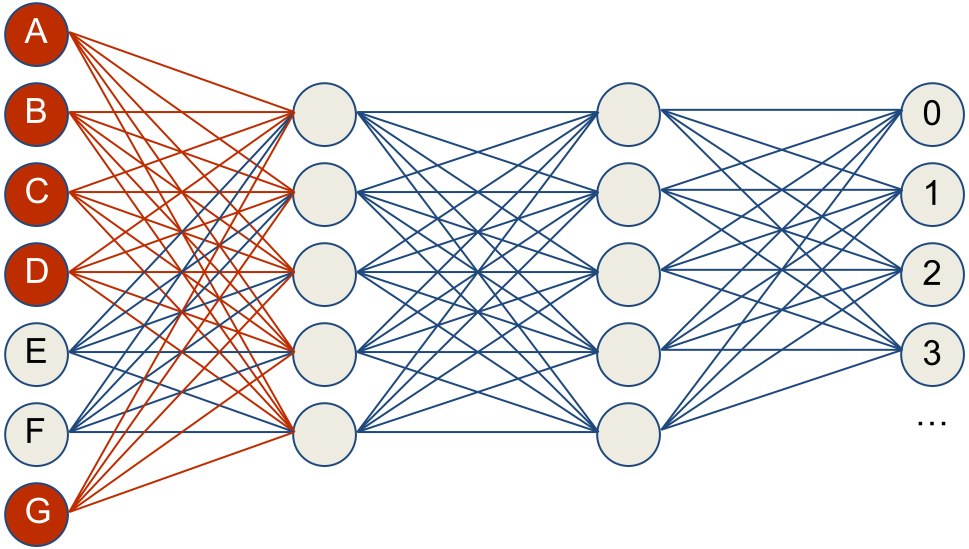



# Forward Propagation

- For example, a 3 might look like this:

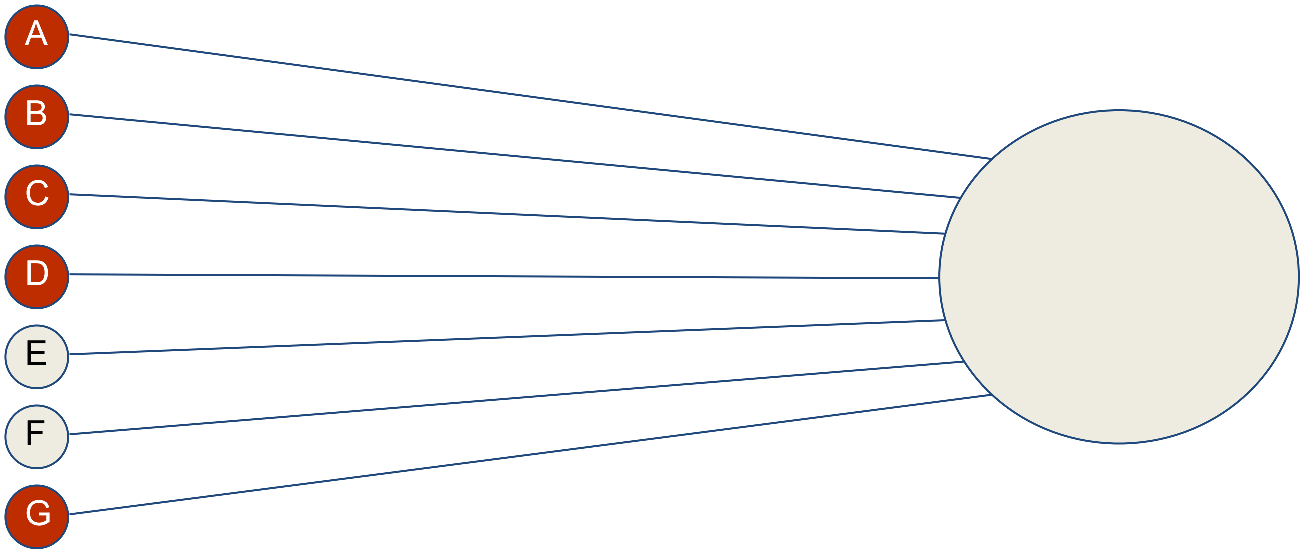




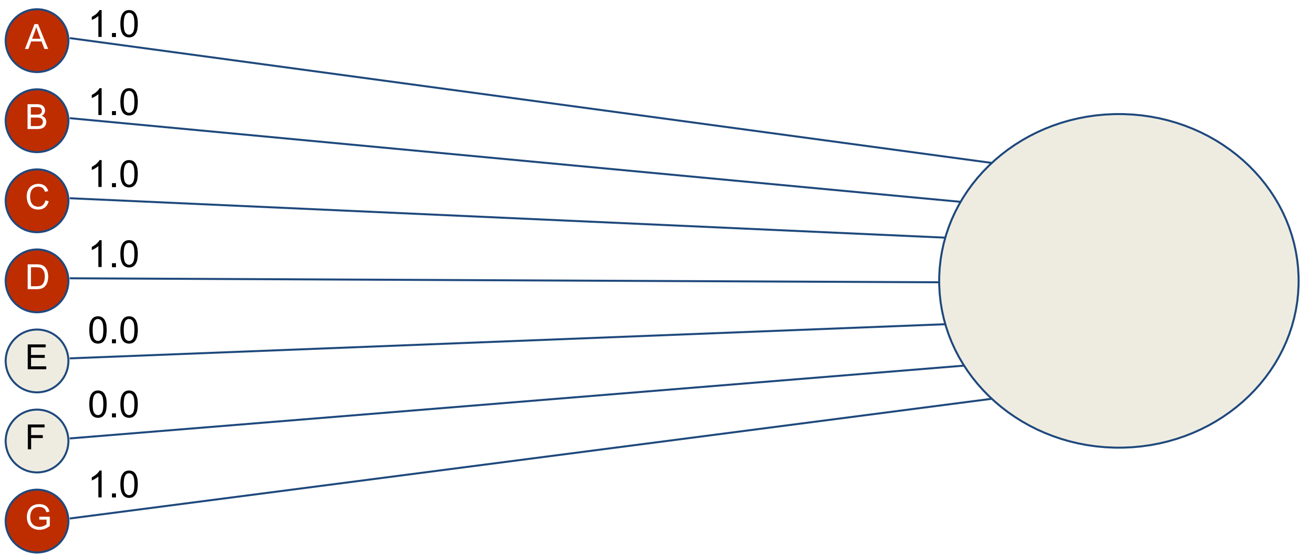
# Forward Propagation

- Let's zoom in on the first hidden layer neuron



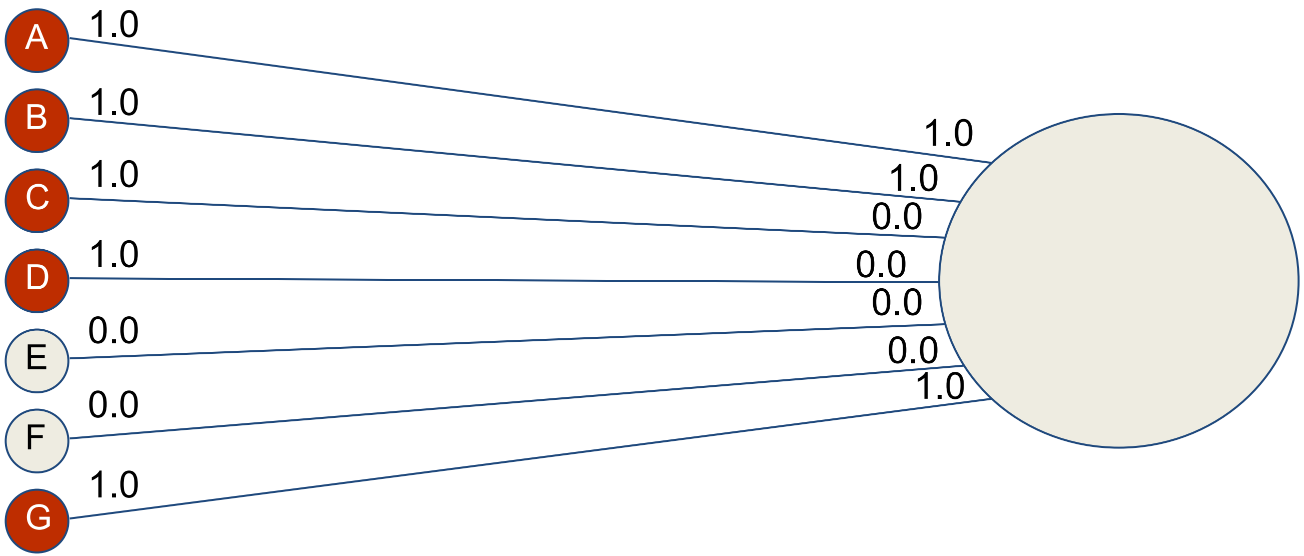
# Forward Propagation

- Here are the input values:



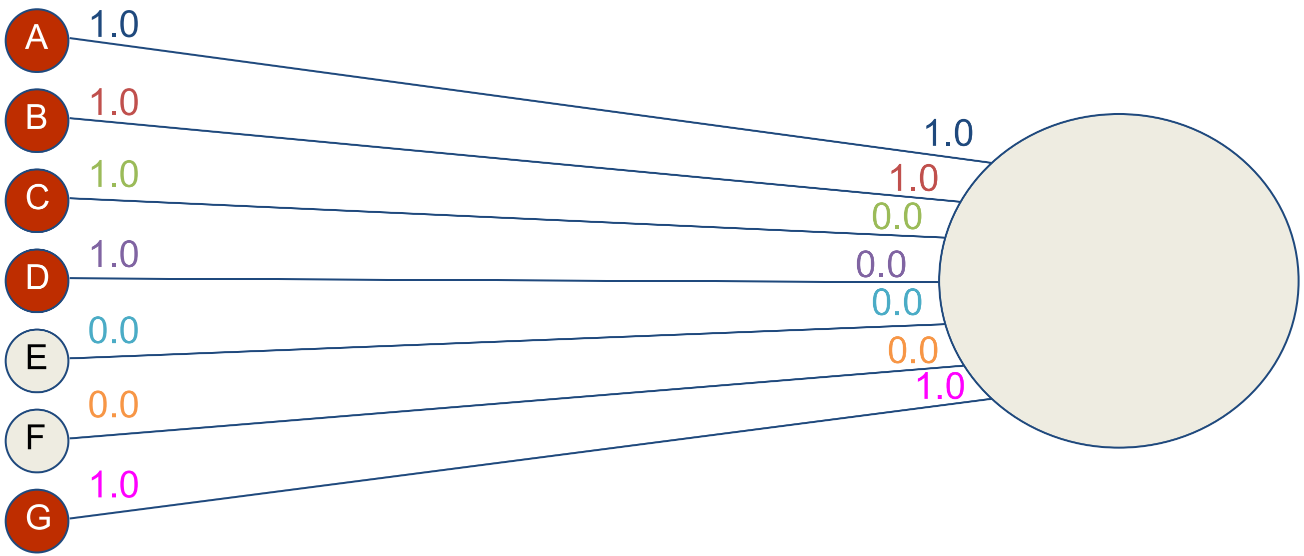
# Forward Propagation

- Each input will have a weight (and a bias, not shown)



# Forward Propagation

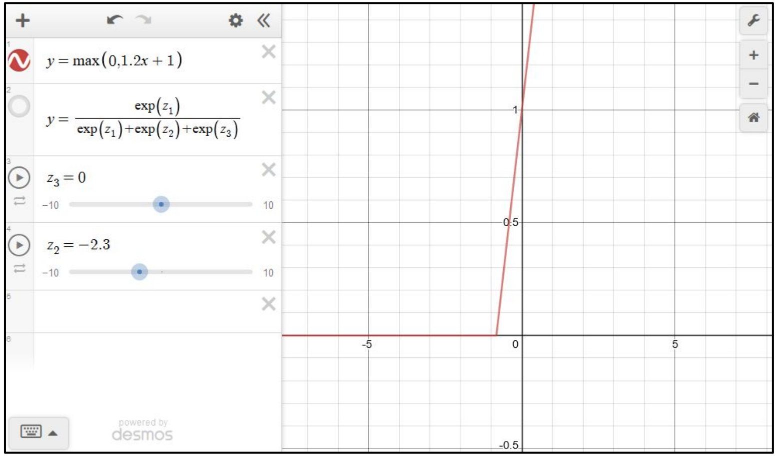
- The output for each neuron will depend on the inputs and their corresponding weights (and biases)

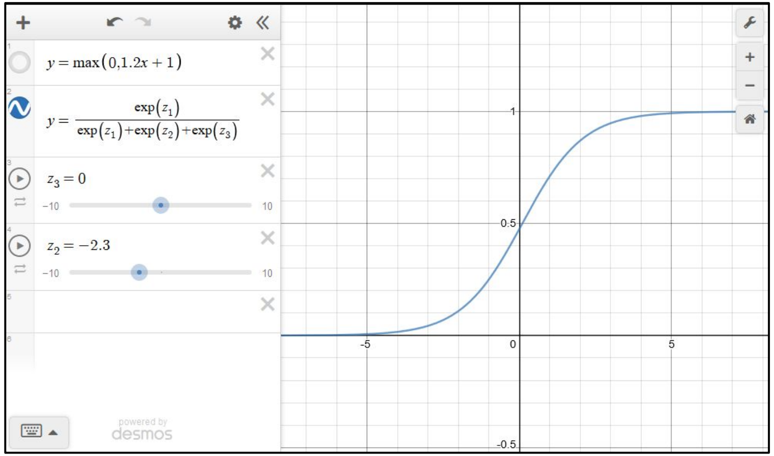



output = 1.0 * 1.0 + 1.0 * 1.0 + 1.0 * 0.0 + 1.0 * 0.0 + 0.0 * 0.0 + 0.0 * 0.0 + 1.0 * 1.0

# ANNs - Activation Functions

- Most activation functions serve two purposes:
  - Smooth the output
    - Why should 0.49 be False, and 0.50 be True?
  - Normalize the output
    - All output values should be similar in range for all neurons



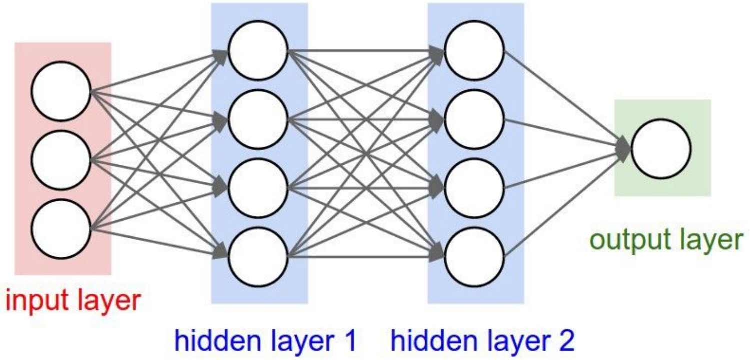



# ANNs - Measuring Loss

- Once the forward propagation completes, we need to determine how wrong our confidence was
  - This is called the *loss* of the network
- Knowing how wrong each output neuron is will help us tune the weights of all of the neurons in the network

# ANNs - Back Propagation

- How should we change the weights of the previous layer's neurons in order to improve these results as much as possible?
- A common way to do this is to use an algorithm called gradient descent
- It is done starting at the output neurons, and then you work your way backwards (thus the name)



# Artificial Neural Networks - Discussion

- In a human brain, what are some of the mechanisms for learning?
- Is there anything in the human brain that we cannot replicate with an artificial neural network?

# Coding Exercise 11.1

- Let's write up some simple code that does the basic forward propagation in a neural network

# Bayesian Networks

# Bayes Theorem

- Bayesian theorem allows us to reason about conditional probability
  - The use of Bayes theorem to infer, using Bayesian probability:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

`P(A)` – The independent probability of A
`P(B)` – The independent probability of B
`P(A|B)` – The probability of A, given that B has occurred
`P(B|A)` – The probability of B, given that A has occurred

# Bayes Theorem

- Let's go through this with an example:
  - `P(A|B)` – The probability an autonomous car crashing, given that it has firmware version B
  - `P(A)` – The independent probability of a car crashing
  - `P(B)` – The independent probability of a car having firmware version B
  - `P(B|A)` – The probability of having firmware version B, given that the car has crashed

# Bayes Theorem - Example

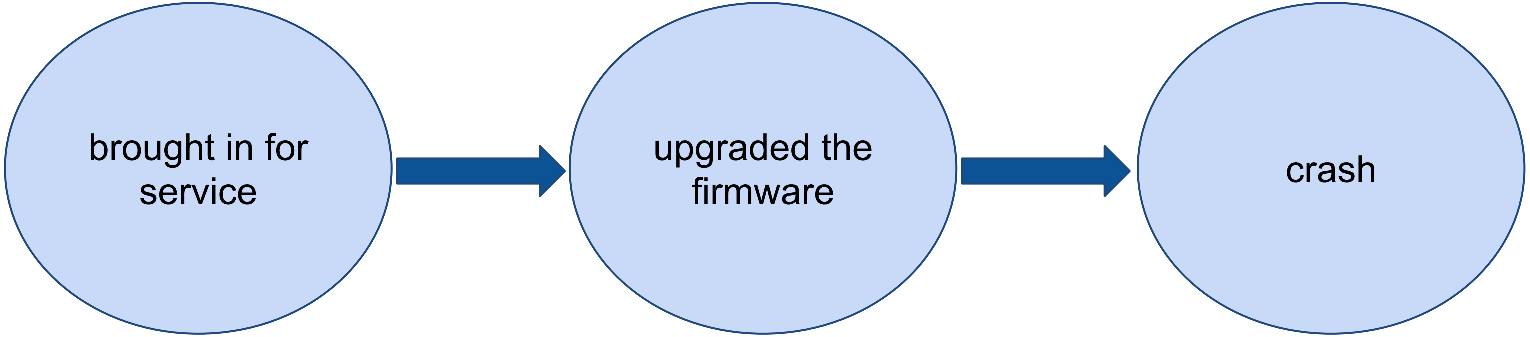
- Let’s go through this with an example:
  - `P(A|B)` – This is what we're trying to find out
  - `P(A)` – There have been 17 total reports of crashed cars, according to Edison's website, 35,500 cars have been sold
  - `P(B)` – According to the Edison car company, 87% of owners have upgraded to firmware version B
  - `P(B|A)` – There have been 17 total reports of crashed cars, 5 with firmware version B

# Bayes Theorem - Example

- Let's go through this with an example:
  - `P(A|B)` – This is what we're trying to find out
  - `P(A) = 17/35500 = 0.000479`
  - `P(B) = 87/100 = 0.870000`
  - `P(B|A) = 5/17 = 0.294118`

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.294118 * 0.000479}{0.87} = 0.0001619$$

# Naïve Bayes Classifier

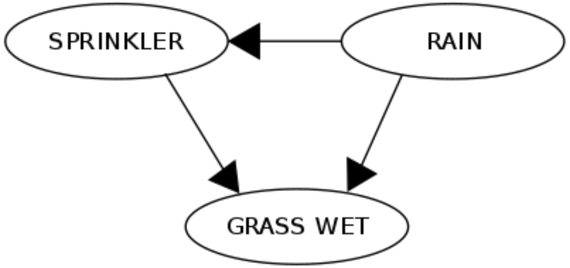


# Naïve Bayes Classifier

- A naïve Bayes classifier is one type of Bayesian network
  - The network shows events as nodes, and conditional probabilities between two events (`P(A|B)`) as directed edges
- Evaluating a network is a matter of filling in the certainties (events you know), and then following the edges (using Bayes theorem) toward the goal node
  - Any events which are certain do not involve Bayes theorem
- The result is a probability estimate
  - It should be noted that these probabilities are not traditional probabilities, but more belief certainty

# Naïve Bayes Classifier

| RAIN | SPRINKLER T | SPRINKLER F |
|---|---|---|
| F | 0.4 | 0.6 |
| T | 0.01 | 0.99 |




| RAIN T | RAIN F |
|---|---|
| 0.2 | 0.8 |

| SPRINKLER | RAIN | GRASS WET T | GRASS WET F |
|---|---|---|---|
| F | F | 0.0 | 1.0 |
| F | T | 0.8 | 0.2 |
| T | F | 0.9 | 0.1 |
| T | T | 0.99 | 0.01 |

# Naïve Bayes Classifier - Discussion

- One of the most common uses for naïve Bayes classifiers is for spam detection
  - What are some of the events that might exist in such a system?

# Genetic Algorithms

Ontario Tech University

# Genetic Algorithms

- Mimic the process of evolution, but at a much quicker speed
  - Survival of the fittest
- Determine how to represent the problem as a string or number
- Randomly generate a bunch of solutions:
  - Consider each solution a **chromosome**
  - Each component of the chromosome is a **gene**

# Genetic Algorithms

- Using rules of genetics, continually generate more solutions:
  - Each chromosome (solution) is evaluated on its fitness (quality of the solution)
  - Choose parents probabilistically, based on fitness (*selection*)
  - To reproduce, combine genes from the different chromosomes (*crossover*)
  - Optionally, also include mutations on individual chromosomes (*mutation*)
- Selection and crossover are the primary mechanisms for *learning*

# Genetic Algorithms - Example

- Using genetic algorithms to solve the pathfinding problem is possible
  - Let each chromosome be a list of actions for each intersection
    - `L` - Left
    - `R` - Right
    - `F` - Forward/straight
    - `B` - Backward/U-turn
  - Generate the initial (say, 1000) chromosomes randomly
    - e.g. `RRFBFRLLBBRF`

# Genetic Algorithms - Example

- *Fitness* - how far away from the destination are we?
- *Selection* - select the top 10 (out of 1000) chromosomes
- *Crossover* - take sub-strings of any two selected chromosomes to form new chromosomes
  - Intuition:
    - One path may make good progress at the beginning and then wander aimlessly
    - Another path may wander aimlessly, but then make good progress
- *Mutation* - randomly change any action
  - e.g. A left turn becomes a right turn

# Genetic Algorithms

https://rednuht.org/genetic_cars_2/

http://www.cambrianexplosion.com/

# Genetic Algorithms - Practical

- Genetic algorithms can be used to play some basic games
  - However, this technique often takes too long to converge at a working solution
  - It is rarely used on its own for difficult problems
- Genetic algorithms is one of the techniques used to set the initial parameters (e.g. neuron weights) in a neural network
  - e.g. https://www.youtube.com/watch?v=qv6UVOQ0F44

# Wrap-up

- Learning
  - Unsupervised
  - Supervised
- Neural networks
- Genetic algorithms
- Bayesian networks