CONNECTIONISM:

- A theory where cognition and intellectual ability can be modeled and explained in terms of neural networks.

- The connectionist model is where minds are explained by neural networks.

- Generally, cognition, mental states, etc can be defined in terms of these things called neural networks.

NEURAL NETWORKS:

- Speaking at the middle Marr level where we define representation and algorithm.

- They consist of input units, output units, and hidden units.

- Information is being transmitted between these units within hidden layers that consist of output layers.

- They are transferred through things called nodes.

- **Example of the nervous system as a neural network:**
    - We look at the Nervous system in our brain.

    - The input layer would be the sensory neurons receiving stimulus from your limbs and needing information from the outside world.

    - Between the input and output layer, there is something called the hidden layers, which are all the other neurons.

    - All the neurons get processed in the hidden layers.

    - The output layer is the motor neurons. It can allow you to recoil from harmful stimuli as an output.

- The activation of a node depends on the activation values of surrounding nodes.

- Different nodes could have positive or negative values and their connections between multiple nodes at higher levels.

- If all nodes in the input layers are positive then all the nodes in the hidden layers would be positive.

- The abstract idea here is that the activation of any node is dependent on the nodes that surround it.

- **Activation:**
    - Positive weight.

    - Nodes are activated when they have positive weight.

- **Inhibition:**
    - Negative weight.

    - Nodes are inhibited when they have negative weight.

- **Activation Function:**
    - Sums up contributions from all the units in the system.

    - It allows it to determine the various weights of these neurons and takes the positive-negative activation of the neurons.

- We are only talking about the middle Marr level for neural networks that have representations.

- Modern computers are not exactly like turing computers and are deterministic.

- They utilize a lot of serial processing in a more classical computational way.

- However, those networks are able to support those sorts of computers, which can support very complex neural networks.

- People think classical computation and connectionism are both true at the same level of how our minds work and how computers work.

- Connections and computation are at different levels.

- The system settles on a solution.

- There are no explicit rules of algorithms.

- This is a point where neural networks structure is going to be different from classical computation structure.

- Neural networks do not have anything explicit like the turing machine instruction book.

- It's just different nodes that are connected to each other with particular weights that are then being summed by a mathematical activation function.

- It will activate a set of weights and go through a set of levels and result in an output pattern, which represents some information the system will be able to use.

  Different Neural Network Structures:
  - **Feed Forward:**
    - Activation flows forward from inputs to hidden nodes to outputs.
    - It maps activation flowing forward from the input nodes to the output nodes.
    - You're not getting activation of the nodes going from output to input. It can't go backward. It all goes forward.
  - **Recurrent**
    - Signals are sent from higher to lower levels (short-term memory, habituation).
    - You can have a node from the output node activate and go towards the input node.
    - Sometimes a system takes a while to fall into equilibrium because they continue to activate each other in these recurrent ways.

- **Example of a neural network looking at handwritten digits:**
  - You can see the number 3 at different resolutions and your brain will still recognize you're looking at the number 3.
  - **Conventional neural networks** → Good for image recognition
  - **Long short-term memory networks** → Good for speech recognition
  - Neural networks are inspired from the brain.
  - A neuron is a node that holds a number between 0 and 1.
  - The network starts with a bunch of nodes corresponding to a 28 x 28 image, which is 784 in total.
  - Each one of these nodes represents the gray scale value of the corresponding pixel of the image.
  - 0 represents black pixels.
  - 1 represents white pixels.

- The number inside the neuron is called the **Activation.**
- Each node is lit up when it has a high activation number.
- The way the network operates determines the activation of the next layer.
- The hidden layers are composed of lines and loops that make up a number.
- There is some specific neuron whose activation is going to be close to a certain line or loop that represents the number you are trying to scan.
- Going from the 3rd layer to the last one requires learning which combination of sub-components corresponds to which digits.
- Recognizing a loop would break down into subproblems.
- You would have to teach the neural network to recognize the various edges that make up a loop.
- Each node in the second layer of the network will correspond with the various amount of edges.
- Many recognition tasks break down into layers of abstraction.
- The goal is to have some mechanism combine pixels into edges or edges into patterns
- Then make the patterns into digits.
- You want activation to be some value between 0 and 1.
- You would pump the weighted sum into some function that squishes the number line into 0 and 1 using the **Sigmoid Function**.

- **The Sigmoid Function**
  - It is known as a logistic curve.
  - Very negative inputs end up close to 0.
  - Very positive inputs end up close to 1.
  - It just steadily increases around the input 0.

- The activation of the nodes here is a measure of how positive the relevant weighted sum is.

- **Example of a speech recognition neural network process:**
  - It involves taking raw audio.
  - It picks up distinct sounds which are combined to make a syllable.
  - The sounds make up syllables.
  - Then the syllables make up words.
  - It then combines to make phrases.

UNSUPERVISED LEARNING - HEBBIAN:

- If you're inputting information into a system of input layers, then the hidden layers will settle upon some output.
- Since you don't have explicit representation in the neural networks is how you get the right weights between the input, hidden, and output layers to get a useful answer.
- This is a variant of training the neural network.
- You are training these neural networks to be maximally attuned to the things you want.
- Neurons that fire together wire together.
- You give a system a bunch of different stimuli and you let it in an unsupervised way form a connection between the nodes.

- **Example of teaching a network to identify cat pictures:**
  - If 2 nodes are frequently connected by the certain stimuli you give it, then this connection would be strengthened.
  - The next nodes would be more likely to go from an input node to a hidden node.
  - If you try to train a network to recognize a picture of cats, then you give it a bunch of cat pictures.
  - If you give it a lot of cat pictures, the pattern that emerges between all the node activation will be sensitive enough to get novel cat pictures.
  - Nodes that tend to be activated (the neurons firing together) will cause the activation connection to be strengthened.

- It really happens from a very complicated mathematical algorithm to train neural networks to recognize something.

- The theory behind this is you will see the right set of emergent patterns that will map on the kind of thing you're giving to a neural network that you want it to identify.

- Once you have so many attempts of unsupervised learning, the networks should be able to correctly identify what you're trying to teach it to scan.

SUPERVISED LEARNING: BACKPROPAGATION:

- To manipulate a neural network via supervised learning.

- Requires a training set of inputs with desirable outputs for a given task.

- It requires you have a set that it's labeled with your desired output to train the system.

- **Example of training a system to identify someone's gender:**
  - Train a neural network to tell you somebody's gender based on an image.

  - The input nodes are going to be the pixels of an image and look at it pixel-by-pixel.

  - For your output node, you would want a probability to tell if a person is a man or a woman.

  - Then whatever of those 2 probabilities are, it will determine the gender.

  - You give this system a bunch of images of people and ask if the system is it a male or female.

  - You want to train this system to make that determination.

  - You would give the system a bunch of pictures that have a label saying whether a person is a male or female to train it properly.

  - You are making it more likely to make those determinations in the future.

  - **The Process:**
    - You set all the nodes to random and give the system an image of a girl to break down.

    - It has a bunch of random weights on the hidden layers.

        - The output nodes are just 2 probabilities of the image being a man or a woman.

        - If it does get it wrong, all the weights are adjusted slightly to get the desired input.

- This process can take hundreds of thousands of rounds of weight adjustment and takes weeks of training a computer.

- You need labeled images every time for every trial during the training process.

- The back propagation is recurrent so the system is taking in the stimuli spotting out some particular solution.

- If the output is wrong then the nodes are tweaked and the system takes in another training image.

ARTIFICIAL INTELLIGENCE:

- Teaching computers to mimic human behavior

- **Machine Learning:**
    - A subset of AI that uses statistical methods to make computers improve with experience.

    - Enables to learn from experience over time.

    - **Example of Machine Learning:**
        - Spotify recommending you songs based on your interest

        - The YouTube Algorithm.

    - **Deep Learning:**
        - A subset of Machine Learning.

        - Makes use of computation at multiple layers of neural networks.

        - It's able to learn from past experience but doing it via neural networks.

        - It adapts from its use of neural networks.

        - **Examples of Deep Learning Methods:**
            - **Unsupervised Learning:** Hebbian

- **Supervised Learning:** Backpropagation

PROS FOR THE CONNECTIONIST MODEL OF THE MIND:

- Can the mind be usefully represented as a neural network?

PROS FOR NEURAL NETWORKS:

- Models cohere with computational neuroscience.
- It has units in the form of nodes, which are similar to neurons.
- The different nodes have connections between the nodes from one level to the next.
- They can be activated or inhibited.
- The structure of a neural network looks similar to the structure of neurons.
- They are able to translate information from one part of the brain to another.
- In that sense neural networks are similar to neurons.
- Connectionism and computational neuroscience are at different levels of explanations.
- Computational neuroscience is all about implementation
- Connections care more about the algorithm level and do not make a lot of commitments to the implementation level.
- Computational neuroscience models are supposed to be biologically realistic.
- They focus on things such as firing rate, action potentials, and tuning curves.
- It's all about the specifics and they want to know how information is transferred from one neuron to another.
- Neuron networks aren't supposed to be literal neurons.
- They just show the similarity that you have units and connections.
- However, neurons are so much more than units and connections.
- Nodes in computational neuroscience look similar to connectionist networks, which represent literal neurons.

- Nodes and neural networks are supposed to represent more of an abstraction.

- Computational neuroscience is not interested with multiple realizability

- Connectionism is supposed to be multiple realizable.

- Connectionism comes from the middle Marr level.

- A connectionist network should be able to be implemented in a silicon life form, humans, or an octopus.

MORE PROS FOR NEURAL NETWORKS:

- Noisy inputs or unit destruction causes **graceful degradation.**

- **Unit Destruction Example:**
    - If you got hit on the head and part of your brain got damaged.

    - Unless the damage killed you or was located in a vital part, you would probably survive.

    - If you got hit on the fusiform face area, you would have trouble recognizing faces.

    - If you had subcortical damage to your hippocampus, you would have difficulty consolidating short-term memory and long-term memory.

    - However, other aspects of your mental life would be preserved.

    - You can have damage to one part of your brain while other parts still seem to work.

    If a mind like classical computing:
    - A turing machine is deterministic, straight forward, and explicitly programmed.

    If a mind is like the big connectionist network:
    - You have a network of information coming from input and moving through the hidden layers.

    - The different nodes will activate or inhibit each other.

    - The system is then able to produce an output.

- We think about which of these 2 systems will allow for unit destruction like hurting or damaging part of your brain.

- **Connectionism** makes the claim to be able to accommodate for unit destruction.

- Neural networks do a great job in dealing with conflicting parallel constraints.

- **Example of conflicting constraints:**
    - When performing a task, two big factors such as accuracy and speed will impact your performance.

    - You can either be very accurate or fast.

    - It's ideal to be accurate and fast.

    - There is always a trade off between accuracy and speed.

    - For some tasks it may be important to be very accurate or fast.

- Your mind is having to construct representations of the world that deal with the speed and accuracy trade-off all the time.

    Snake Detection Theory:
    - The idea is that our visual system was evolved to be able to identify snakes.

    - In our evolutionary history, it was important to identify snakes.

    - There have been studies done where people will see a movement in a periphery and they'll jump and be afraid.

    - They think they saw a snake due to similar movement, but there was no snake there.

    - Their system is very sensitive to snake detection.

    - In this sense, the snake detection system is not accurate since it will just be a false alarm.

    - However, it's very fast and allows you to have quick reaction time.

    - You're able to get fast split second decisions your system makes to alert you.

- You would want your neural network system to be fast, but also pretty accurate, so you set your weights at random and train it using a lot of images.

- You teach your system the right balance of speed and accuracy for the most effective output.

- Whatever the pattern of activation is, it's going to be doing that in a way that's accurate and fast.

- You don't have to know what the right balance is, you just have to train the system over lots of different trials.

- It will help the emergent pattern appropriately balance speed and accuracy.

- This is a benefit over classical computation systems where you have to explicitly represent it and you have to add it to your big book of code.

ANOTHER PRO FOR NEURAL NETWORKS:

- It provides an attractive account of concepts in our minds.

  Example:

  - **Condition for a chair:** A thing with four legs you can sit on.

  - How do we characterize what makes up a chair if they have different designs?

  - There is a chair that has no legs but can sit down on it

  - There is another chair with 4 legs, but you aren't able to sit on it.

  - The idea is that connectionism is able to provide us with a way to understand kind membership as being something graded.

  - A neural network may activate a chair category in your brain but does not do a good job since the chairs have questionable designs.

  - Connectionism allows for a more graded idea that can accommodate our concepts for any symbolic representations.

WHY THE MIND IS MORE LIKE CONNECTIONISM AND A NEURAL NETWORK:

- If a classical computation system (turing machine) were to be damaged, the whole thing would collapse and break because it wouldn't be able to go to the next step.

- If a part of your brain is to be damaged, the others parts would still function.

- So long as all the nodes aren't damaged, you'll still be able to input things into the system to arrive at some output.

- It may be less accurate, but the process would still happen.

- If you were missing a page from a turing instruction, the whole thing would break apart.

- The brain does not work like that and will still function even if something is damaged or missing.

- Neural networks can deal with noisy inputs

- Noise can refer to random firing of neurons in neuroscience.

- There is a lot of noise in our brain, but we usually tune it out in the way we build our interpretation of the world.

- Neural networks can do that also.

- We hear noise everywhere and all time, but we're able to tune it out because the stimuli we care about are those consistent ones that make patterns through your neural network.

- You could represent noise in a classical network, but it's difficult since turing machines are very serial and do not accommodate noisy inputs.

OBJECTIONS: ONE OFF LEARNING:

- Humans are able to do one off learning.

- In order to get a neural network that has the right measure of weight for the task, you must do lots of training.

- However, humans don't need that amount of training.

- **Example of One Off Learning:**
    - If you get food poisoning you become extremely averse (strong dislike) of whatever food gave you food poisoning.

    - A person ate chickpeas and felt sick after leaving it in the fridge for too long.

    - There's nothing inherently wrong with chickpeas.

    - the person who ate got sick from it.

    - If you get food poisoning from something, it will make you very averse to eating it again.

    - You're biologically repulsed in a strategy where you assume the last thing you ate gave you food poisoning.

    - We are able to perform one off learning after any experiences.

- If we think of connectionism as our mind, it's difficult to imagine how a connectionist model can accommodate this one off learning.

OBJECTIONS: JUST DOING ASSOCIATION:

- Can connectionist models truly master abstract rules or are they just learning associations

  Example of a neural network being trained on recognizing trees:
    - You give it many images of trees hoping you get some emergent pattern of activation for the nodes.

    - Activation of the nodes will help the network allow it to identify a try like a person with a mind can.

    - We have a concept of trees and it's easy for us to think about a tree and recognize them.

    - If you gave a neural network multiple images of different kinds of trees, will it be able to identify a new tree?

    - You gave a training set to a neural network to extrapolate to new input that are part of the training.

    - Neural networks are really good at recognizing images that are part of a training set, but not very good at a new set at all.

    - The idea is that neural networks are doing association.

    - Humans are doing something else more advanced than just association unlike neural networks.

- Connectionist models sometimes have difficulty generalizing performance from the training set.

OBJECTIONS: SYSTEMATICITY:

- Thought is systematic.

- If you can think ‘John loves Mary’ (aLb) .

- Then necessarily you can think ‘Mary loves John’ (bLa).

- Even though both have different meanings.

- These are predictable patterns in the thoughts we can understand.

- The idea is that language is systematic such that some sentences are intrinsically linked to our ability to entertain other sentences.

- **jLm** and **mLj** are broken down into these 3 subcomponent parts:
    - John
    - Mary
    - The love relation.

    - You don’t need any other information from **jLm** to **mLj** because they are just re-combinations of these 3 parts

    - It can be broken down into sub-components.

    - Then the sub-components can be rearranged to form new thoughts.

    - There are these intrinsic connections between some thoughts and other thoughts.

- They are these **predictable patterns** in the thoughts we can understand or not**.**

What does this have to do with connectionism and classical computation?:

Classical:

- In classical computation you have rules and explicit representations in your computational book that are manipulated.

- A classical computation will have an easy job in telling us what the sentence of **jLm** or **mLj** means by the explicit reference to those 3 components of the relation.

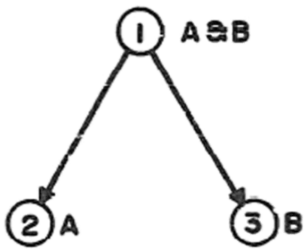
Connectionism:

- There are no nodes like Mary, Love, and John in a connectionist system.
- There are only distributions of nodes that will get you the proposition ‘Mary loves John’ once activated.

- There is no explicit manipulation of different symbolic parts.
- Therefore absent explicit representation and symbolic symbol manipulation to have a system to map the sentence **jLm.**
- It can understand what **jLm** means from the training sets, but it won't understand **mLj.**
- That might produce an entirely different set of node activation and the system may not recognize at all due to lack of training.
- It's not able to capture the fundamentals of the mind and thought because they don't adhere to the system.
- You could train it to identify what jLm and mLj means, but it is not necessarily systematic unlike a classical system.

OBJECTIONS: CONSTITUTION:

- Thought is constitutive.
- To understand the constitution, you need to entertain some thoughts that's a conjunction.
- A and B are constituents of A&B.
- Not clear how this is possible on a connectionist picture.
- **Example of constitution:**



- When you entertain your mind with peas and carrots, what does that mean?
- Peas and Carrots would represent A & B at the top.
- The claim could be breaking down making peas at A and carrots at B.

- **Classical Picture:**
    - This explanation looks easy to accommodate for a classical system because classical computation utilizes explicit representations and rules.

    - It's impossible to separate peas or carrots from these 2 constituent parts.

    - In a classical picture, thought being constituent in this way is necessary and is something you get for free.

- **Connectionist Picture:**
    - Imagine feeding your system peas and carrots.

    - We can imagine a network that can be trained to recognise the conjunction like peas and carrots.

    - However, it is not able to recognize the constituent peas or carrots when they are fed into the system independently.

    - You could train a connectionist network to be able to recognize peas and carrots, but it's not a facet of a system.

    - This seems like a shortcoming for a connectionist system.

    - Thought is necessarily these ways that any model of the mind is going accommodate that

OBJECTIONS: PRODUCTIVITY:

- You will only entertain a **finite number** of thoughts. But in principle, there are **infinitely** many thoughts you might entertain.

- **Examples of Productivity:**
    - 'Mary gave the test tube to John's daughter.'
    - 'Mary gave the test tube to John's daughter's daughter.'
    - 'Mary gave the test tube to John daughter's daughter's daughter.'

    - These sentences can be grammatically accurate no matter how many times you add daughter to it.

    - All of these different components to the sentence such as Mary, test tube, and the daughter are all going to be explicitly represented.

    - This looks difficult on a connectionist model.

- It's still possible to train it, but it's certainly given in virtue of the architecture like a classical computation system.
- Thought has a productive structure built into classical computation systems.

- That is something connectionist networks have to try hard to be able to accommodate what classica networks get for free.

LANGUAGE OF THOUGHT HYPOTHESIS:

- **LOTH:** Thought is made up of language-like components with compositional structure.

- **Mentalese** components combined in systematic ways have a syntactic structure.

- Supposed to exist at the algorithmic level.

CONNECTIONIST REPRESENTATION:

- Non-atomic representations (unlike **LOTH**).

- The representations can remain intact when parts of the model are destroyed.

- Distributed (rather than local!) representations.

- These types of representations remain intact when parts of models are destroyed.

- Representations can just be thought of as giant vectors(the number input, all the different manipulations across hidden nodes, and the number output).

CLASSICAL VS CONNECTIONIST:

- **Implenational connectionists:** connectionism might be the implementational way to realize classical computation algorithms (i.e. brain's neural nets implement a symbolic processor).

- **Radical connectionists:** reject this view; seek to eliminate the idea of symbolic processing.

  Classical:
  - Information represented by strings of symbols.

  - Mental processing = digital processing.

  - Strings produced in sequence according to program.

  - Better at explaining one-off learning.

- Coheres with “Language of Thought”:
    - Systematically
    - Constitution
    - Productivity

Connectionist:

- Connectionism is a lower-level explanation for higher-level classical computational algorithms.

- Information is stored non-symbolically in connection strengths between the units of a neural net.

- Mental processing = activity in a neural net.

- Each unit’s activation depends on the connection strengths and activity of neighbors.

- Coheres with neuroscience.

- Graceful degradation.

- Conflicting parallel constraints
    - Speed vs. accuracy

- Concepts