Core Concepts in Mechanistic Interpretability - Examples

Neural Networks as Grown Systems

Imagine you're trying to create a garden that produces the perfect tomatoes. You have two approaches:

1. Engineering approach: Trying to build a tomato from scratch by assembling cells, designing DNA, and controlling every aspect
2. Growing approach: Preparing the soil, planting seeds, providing water and sunlight, and letting nature do the complex work

Neural networks are like the second approach. We don't write code that explicitly tells the system "here's how to recognize a cat" - instead, we:

- Design the architecture (like choosing the right soil and garden layout)
- Set the learning objective (like providing sunlight and water)
- Expose it to examples (like letting the plants experience different weather conditions)
- Let it develop its own internal organization (like how plants develop their own root systems)

This is why we often can't directly explain how neural networks make decisions - we cultivated the capability rather than engineered it, just as a gardener might struggle to explain exactly how their tomato plant decides where to grow each root.

Features and Circuits

Think of how humans recognize a school bus:

1. We notice it's yellow (a simple feature)
2. We see it has wheels (another feature)
3. We spot black horizontal stripes (another feature)
4. When we see these specific features arranged in a particular way, our brain says "school bus!"

Neural networks develop similar hierarchical recognition patterns:

- Level 1: Simple features like edges and colors
- Level 2: More complex features like wheels and windows
- Level 3: Complete object recognition by combining lower-level features in specific arrangements

The breakthrough insight is that these patterns form naturally during training - the network discovers that "wheels below + windows in middle + yellow color = school bus" is a useful pattern, just as humans naturally learn to recognize objects without being explicitly taught the step-by-step process.

Linear Representations

Imagine an artist's paint palette with three primary colors: red, blue, and yellow. Any color can be created by mixing these base colors in different proportions. Neural networks represent concepts in a surprisingly similar way:

Word embeddings example:

- "King" = Royal direction + Male direction
- "Queen" = Royal direction + Female direction

- Therefore: King - Male + Female = Queen

This works because concepts are stored as "directions" in the network's "concept space", just like colors are "directions" in color space. More surprisingly, this even works for complex concepts:

- "Puppy" = Dog direction + Young direction
- "Kitten" = Cat direction + Young direction
- Therefore: Puppy - Dog + Cat = Kitten

This isn't just wordplay - the network actually represents concepts this way, allowing it to understand and generate new combinations it hasn't explicitly seen before.

Superposition

Imagine you have a tiny apartment but lots of stuff to store. You might use clever solutions:

- Murphy bed that folds into the wall when not in use
- Dining table that converts to a desk
- Stackable storage boxes that serve different purposes at different times

Neural networks use a similar trick called superposition. Instead of having one neuron for each concept (which would require too many neurons), they:

1. Share the same space between multiple concepts
2. Use clever "time-sharing" - like how your Murphy bed is a bed at night and a wall during the day
3. Rely on the fact that you rarely need all concepts at once (you don't need your bed while using your desk)

Real example: A single neuron might activate:

- 80% for cats when processing pet-related content
- 80% for cars when processing transportation content
- 80% for carrots when processing food content
  ...but never needs to represent cats, cars, and carrots simultaneously, just like you never need your Murphy bed to be both a bed and a wall at the same time.

Dictionary Learning

Imagine you're at a party where everyone is talking at once. It sounds like noise, but with practice, you can focus on one conversation - this is called the "cocktail party effect." Now imagine you have a recording of this party, and you want to separate out each individual conversation.

This is exactly what dictionary learning does for neural networks:

1. Start with mixed signals:
   - Neuron A activates for both "dogs barking" and "cars honking"
   - Neuron B activates for both "birds singing" and "cars honking"
2. Dictionary learning separates these into clean features:
   - "Dog Feature" only for dogs
   - "Car Feature" only for cars
   - "Bird Feature" only for birds

Real-world example from Claude:

- Before dictionary learning: A neuron activates for "backdoors in code" and "hidden cameras" and "secret passages"
- After dictionary learning: We discover it was actually representing the abstract concept of "concealment" in different contexts
- This helps us understand that the model has learned the general concept of hiding things, not just memorized specific examples

Universal Building Blocks

Imagine giving the same LEGO building challenge to 100 different people who have never met:

- Task: Build a stable bridge
- Result: Most people discover similar solutions (support pillars, cross-bracing, etc.)
- Why? These are optimal solutions given the constraints of physics and LEGO bricks

Neural networks show the same phenomenon:

- Different networks trained independently discover the same basic features
- Early layers always develop edge detectors and curve detectors
- Higher layers develop more complex but still universal features

Even more surprisingly, biological brains show the same patterns:

- AI vision systems develop "Gabor filters" for edge detection
- Later, scientists found the same patterns in monkey brains
- This suggests these aren't just arbitrary solutions, but optimal ways to process visual information

Monosemanticity vs Polysemanticity

Imagine trying to organize a library with very limited shelf space:

1. Ideal situation (Monosemantic):
    - "History" books on the history shelf
    - "Science" books on the science shelf
    - Clear and organized, but requires lots of shelves
2. Reality due to space constraints (Polysemantic):
    - Shelf A: "History AND Science books from authors A-M"
    - Shelf B: "Science AND Fiction books from authors N-Z"
    - More efficient use of space, but harder to find specific books

Neural networks face the same challenge:

- They'd like to have one neuron per concept (monosemantic)
- But they need to store too many concepts with limited neurons
- So they end up with neurons that respond to multiple concepts (polysemantic)

The breakthrough is that we can now "reorganize the library" using sparse autoencoders, converting messy polysemantic neurons into clear, single-purpose features.

Multimodal Features

Imagine how humans understand the concept of "sharp":

- Visual: We can see a sharp knife
- Tactile: We can feel a sharp point
- Auditory: We can hear a sharp musical note
- Abstract: We can understand a sharp criticism

Modern neural networks develop similar concept detectors that work across different types of input. Real example from Claude:

- A "deception" feature activates for:
    - Text describing lies
    - Images of hidden cameras
    - Code containing backdoors
    - Articles about fraud

This shows the model has developed a genuine understanding of abstract concepts, not just pattern matching. It's similar to how humans can recognize "deception" whether we're reading a story, watching body language, or analyzing financial data.

Mechanistic Interpretability vs Traditional Studies

Imagine two scenarios for understanding a complex system:

1. Studying the human brain:
    - Can only observe a small number of neurons at once
    - Can't modify connections without damage
    - Can't reset to a previous state
    - Can't run the same exact experiment multiple times
2. Studying neural networks:
    - Can observe ALL neurons simultaneously
    - Can modify any connection
    - Can reset to previous states
    - Can run identical experiments repeatedly

Despite these advantages, understanding neural networks is still challenging. It's like having:

- A program's complete source code
- All variable values during execution
- Ability to modify any part
  ...but the code is written in an alien programming language we're just beginning to decipher.

This suggests that even with perfect information, understanding complex systems requires new tools and ways of thinking - a challenge that applies to both artificial and biological intelligence.