A Model of a Mind

Tyler Neylon

264.2024

[Formats: html | pdf]

This post explains a simple model of how minds may work. I'm motivated by the success of AI-based language models to look at the future of digital minds. The model presented here is not detailed enough for a full implementation, but it's more detailed than the popular models of minds I've been able to read about in my background research.

I can imagine two goals of a mind model: to understand human brains, or to create digital minds. These goals overlap because the most impressive mind we know of is the human brain[1]. Because of this overlap, I'll aim for a mind model that can account for human behavior — though my primary motivation is the creation of digital minds.

There's still plenty of debate about whether or not a digital mind can ever be truly conscious, or have emotions or subjective experiences as humans do. I'm convinced they can. Rather than focus on that debate, I'd like to work in the hypothetical world where digital minds are indeed capable of all the internal experiences of human minds. If I'm wrong, then this becomes a fun collection of speculative blueprints; if I'm right, then this post is something more — hopefully, actual progress toward both the creation of digital minds as well as some insight into how our own brains may work.

1 Goals of the Model

I'm trying to make a system that can behave like a human. Consciousness is a personal motivation, but I'm not going to focus on it as a goal because it's difficult to define well and people often disagree about it. This posts instead looks at some aspects of minds that — while still challenging — are a little easier to discuss.

Specifically, I'm trying to build a system that has these features:

- Agency

[1] Please don't mistake ignorance for hubris! I'm sure other minds can exist that are better.

- Learning
- Thinking
- Introspection

I'll show you the simple model, argue why it can enable behavior like each of the above points, and I'll finish with some notes about the elusive word "consciousness."

2 The Model

I'm thinking about minds in terms of data flow between simultaneously-acting modules. If you have a computer with a GPU, a multi-core CPU, and a camera attached, then each module (GPU, CPU, camera) can do its own work in parallel. The modules in a system like this talk to each other, but they can always process information as it's received.

Human brains are incredibly parallel machines. Neurons don't wait for each other, but apparently react to signals as soon as they receive them. So it makes sense to think of a brain as a vast neural network — one we can understand better by seeing its architecture as a data flow diagram between modules that continuously act in parallel.

2.1 An action model

A central concept in this model is what I call an *action model*. The name is a natural evolution of *language models*, being systems that understand and can produce language. Thus an action model understands and can produce *actions*.

You can think of an LLM, in simple terms, like this:

```
context -> LLM -> next_token
```

By analogy, an action model works like this:

```
context -> Action Model -> next_action
```

Conceptually, I'm thinking of an "action" as a superset of words. If I wanted to say "hello," then *say hello* is an action. If I want to walk to the kitchen, that's an action. And if I want to ponder the meaning of life, that pondering is also an action.

2.2 The model at a high level

Here's the model:

Each arrow represents a flow of information. Solid arrows are what I consider to be the most important flows.

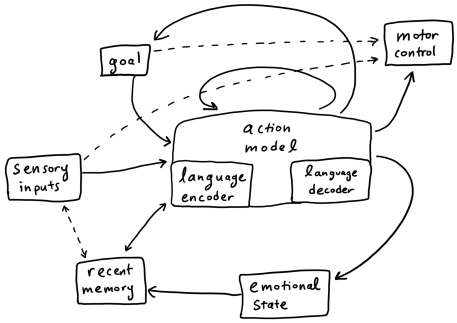

Figure 1: Data flow diagram for a model of a mind.

A couple modules do a lot of work for us, but are easier to understand: The *sensory inputs* provide everything we sense, including vision, taste, temperature, pressure, and so on. I'm letting this module perform some work as well, since (for example) our vision system quickly provides us some analysis of what we see, so that we tend to perceive visual objects rather than a raw image. The other somewhat-simple module is the *motor control* which we can think of as receiving a conceptual vector (for example, "scratch left ear"), and it can do some processing to translate that high-level command into a series of individual muscle commands. When you memorize a piano song well enough, it feels as if your fingers know the song better than you do, and I believe that indicates some kind of learning has happened within the motor control module.

The *action model* has already been introduced. I've included within it a *language encoder*, which translates incoming signals — seeing written words, hearing spoken words, seeing sign language — converting those into a vector space understood by the system. Since I'm imagining an action model can be a slight generalization of a language model, I'm expecting that such an action model could naturally incorporate within itself a way to standardize lexical concepts into consistent vectors. Similarly, the *language decoder* is good at converting those conceptual vectors back out to actions, such as speaking a sentence out loud, or writing something down.

The *emotional state* module is doing a lot of work: It's meant to represent all of our bodily needs, such as feeling hungry or tired, as well as our state of mind, such as feeling elated, frustrated, nostalgic, or intrigued. In this model, our emotional state can change based on what's coming out of the action model, and it also filters that output into the *recent memory* module.

I've chosen this flow of data carefully. In effect, there are two filters on what we store in recent memory: First, when the action model processes a lot of incoming information, it will effectively pay more attention to some information than the rest. As in a language model, the unused information essentially disappears from the network as it passes through later layers; the attended ideas persist until

the end. The second filter is based on our emotional state. When we're bored, what's happening is not considered important, and not flagged for longer-term memory. When we're experiencing an emotional spike, a lot more data is kept around in more detail. Our usual life tends to be somewhere between these extremes.

Finally, I've called out one particular piece of data called a *goal*. This is not a computational module, but rather a part of the data feedback loop coming out of the action model and fed back into itself. I'm imagining the action model as receiving a lot of data that we could view as one giant vector, and likewise producing another large vector. These large vectors might begin life in new brains as "unformatted," meaning that a person can learn to use that space as they grow, rather than thinking of the vector data as pre-assigned to given purposes. Within the vector representations, there's room to learn / define specific variables, and one of the most important variables we learn is our current goal.

Just as a word can be captured by a vector, so can an action or a (closely related) goal that we have in mind. In this mind model, our current goal fundamentally shapes how we filter the incoming information, and can be edited by the action model itself. We may even have an effective *stack* of goals, a small data structure that we can push new goals onto, and pop them off as we complete them. Or, if you're like me, a limited-size stack where tasks are often forgotten because I keep thinking of new things to do.

That's the gist of the mind model. In the next few sections, I'll explain how this model can provide agency, learning, thinking, and introspection.

3 Agency

I'll explain how agency can be achieved first because it's the simplest of our goals to accomplish, and it's somewhat independent from the present mind model.

A large language model doesn't have agency because it can only react to input; it can't independently take action.

However, we can imagine a change that adds agency to any LLM-like system. Think of a model that receives two interwoven input streams. One input stream is the person talking to the model, and the other is the model being able to see its own output. Current LLMs see both of these streams, but they're set up so that only one person at a time can talk — the LLM or the user. The difference here is that the model is designed from the start to see its own feedback, constantly, as well as real-time input from "the outside," such as the user.

Now the LLM can choose to switch, at its own discretion, back and forth between a talking and listening mode. When the LLM wants to listen, it can

produce a special `<listening>` token many times in a row, until it wants to say something. When it wants to speak, it outputs what it wants to say instead of the `<listening>` token.

In this way, the model can run continously while enabling a meaningful two-way conversation that includes pauses for the other speaker. It can independently say whatever it likes whenever it likes. This is the lexical version of agency, and it applies perfectly well to the mind model sketched above, which does indeed receive both sensory inputs as well as feedback from its own output.

4 Memory and Learning

It might sound surprising to say that a "machine learned" LLM doesn't learn. What I mean is that, in their standard mode of operation, modern LLMs don't modify any internal state in reaction to the conversations they have. The first wave of LLMs would completely forget what everyone said as soon as its context window was full. As I'm writing this, some systems like ChatGPT, have been augmented so that they "remember" certain facts. While I can't confirm details internal to OpenAI, my educated guess is that these facts are available to the model because they can be selectively added to the prompt. That is, I believe the only common way for LLMs to "learn" today is to implement an additional system to store data to be learned, and to selectively insert that data into prompts when we think it might be useful.

This is different from the way we experience life because we gain new abilities, and often the things we remember don't seem to be part of some internal prompt. For example, when you speak out loud, you don't feel as if your brain chose a subset of 100 candidate words to present to you, and you chose from amongst those.

Rather, it does seem that the behavior of our organic neurons is updated in response to what happens to us. The equivalent of this in the mind model is to update weights based on experiences.

4.1 Story memory and action memory

To explain the idea in this mind model, I'll split memory into two broad categories:

- *Story memory* is the memory of everything that's happened to you; and
- *action memory* is the modification of how you act based on positive or negative feedback.

I'll motivate these categories with a simple example. If a stranger says to you, "hey, you can definitely trust me!" then you can immediately store this narrative element of your life: this person said these words. Now, is what they said *true*? That's a different matter, and one you should probably decide based on more evidence. The *fact* that they said these words can safely go into story memory

without fact-checing. The *idea* that they're trust-worthy is an uncertain claim we can keep around, flagged as "dubious" until further notice.

When it comes to decisions we make, it's not always obvious if it was a good decision until some later point in time. Consider making a move in chess. If your opponent surprises you with an unseen checkmate two moves later, you might retrospectively realize a particular move had been a mistake. This is an example of delayed feedback on the quality of your decision. When you have delayed feedback, it's useful if you can later reinforce good decisions, or discourage repetition of mistakes.

The motivation for the "recent memory" module in the mind model is a place that can essentially memorize exactly what has happened recently before it's baked into the action model. I suspect this is useful because, as you fine-tune LLMs, you can easily cause catastrophic forgetting, which is the effective erasure of old memories. In other words, in practice it seems that new memories are added carefully, perhaps in order to keep old memories intact. Another use of the recent memory module is to provide a delay to considering my own actions as good until after I've received feedback about that action.

A third motivation to have a separate recent memory module is that a detailed memory of the past few hours is much more valuable that an equally-detailed memory of some random window of a few hours from when you were four years old. The usefulness of story memory decreases rapidly with time, and there's a need to filter what's stored due to the sheer volume of sensory input in comparison with the finite capacity in your action memory. Because recent memories tend to be more useful, it's convenient to have a rolling window of accurate memories that are forgotten as enough time passes.

This breakdown of memory types might account for these features of human memory:

- We seem to have a small memory capacity that we receive with almost no effort or special attention spent on the thing being remembered. George Miller did work to establish that most people can quickly remember about seven items from an arbitrary list. That memory might fit into the feedback vectors of the action model itself. This memory disappears as soon as we think enough about something else.
- Different people have different recent memory capacities, but it's common to remember what you ate for breakfast this morning, but not what you ate for breakfast several days ago, ignoring predictability (such as if you cheat by eating the same thing for breakfast every day). This type of memory matches what can fit into the recent memory module.
- Longer-term memories don't seem to have a pre-determined time limit, but they do tend to fade over time. This pattern is consistent with knowledge baked into LLMs, and so can match the way an action model would effectively remember things — without a time limit, but with the ability to fade over time, especially if not referenced for a long time.

Human brains seem to have separate locations for long-term memories and whatever our equivalent of an action model is. Cases of amnesia suggest this: People can forget much of their past while otherwise acting normally. If our memories and behavior depended on the same set of neurons, then this wouldn't be possible. However, in the mind model above, I've let the long-term memory be implicitly part of the action model because this is effectively how language models currently store their version of memories.

People tend to remember events in more detail when their emotions were strong at the time. Conversely, people tend to forget moments when they were bored or not paying attention. Think of the last time you took a well-worn route, such as your daily commute to work. You probably don't remember how you spent much of that commute, or at least, you probably don't remember the details that don't matter, such as the color of the car in front of you at a certain intersection.

The mind model accounts for this by filtering memories through emotional states. In order for the model to remember something, it must be both (a) something the action model has paid attention to, and (b) something the mind cares to remember based on the emotional state. In addition, the emotional state is part of the context for the action model, so that goals are influenced by how the mind feels, and what the mind pays attention to is likewise influenced by feelings. For example, if the mind is in a happy mood, it's more likely to appreciate the positive aspects of a conversation; if it's feeling defensive, it's more likely to notice a perspective from which a conversation can be seen as judgmental.

I'm using the word "emotion" in a broad sense meant to include pleasure, pain, boredom, happiness, frustration, and any combination of states of mind that have a not-purely-rational feeling associated with them. The most basic aspect of this — akin to simple pleasure or pain — can be seen as a relatively quick feedback loop to inform if the recent action memories are good are bad for the sake of learning. If a recent action was akin to hitting your thumb with a hammer, then you'll have pain as a clue to no longer take that same action. The model captures pain as negative feedback from the emotional state.

4.2 Meta-learning

Another kind of learning happens at a higher level, which requires longer-term thinking. For example, suppose you write a first draft of a book, and then give that book to some beta readers for feedback. You can view this as a process with many months between the action first taken — writing your first word of a new book — and receiving useful feedback on that action. The recent memory is no longer a useful vehicle for this kind of learning.

In this case, I suspect humans learn a process in a more explicit manner. I'm convinced that humans learn rational behaviors as action sequences which are initiated by triggers. For example, when I want to write an idea that's already well formed in my mind, I'll either record a voice memo of the outline, or I'll type an outline draft in google docs. That's part of my personal process. The

trigger is the combination of (a) wanting to write an article, and (b) not needing to do more research, that is, feeling confident I'm ready to write. The action sequence, at a high level, is to make the outline.

Now suppose I get feedback on my action sequence. For example, maybe the voice recorder app on my phone deletes a file due to a bug. Then I'll make a mental note to use a different voice recorder app. This kind of learning is not happening at the level of weight updates in a neural network. Rather, it's a more conceptual idea that is best seen as over-writing the key-value pair:

```
[I want to record an outline] -> [open voice app A]
```

by re-using the same key, and replacing the value, like so:

```
[I want to record an outline] -> [open voice app B]
```

I've phrased things this way specifically because human brains don't seem to be good at erasing past memories, but rather they seem to be able to *replace* values associated with pre-existing keys.

4.2.1 Key-value memory in humans and AI models

Consider a person with a bad habit, such as biting their nails. It's notoriously difficult to enact a strategy of simply stopping such a habit. If you do this and your thought is "I'll just stop," you're likely to fail. However, if you *replace* the bad habit with something else, you're more likely to succeed. For example, you can notice the situations where you're most likely to bite your nails — such as sitting in a classroom and somewhat bored — and then teach yourself to take a *different action* in those same contexts. For example, you might use a fidget spinner instead of nail-biting. This is a human-oriented example of key deletion being hard ("key deletion" here would be like ignoring the trigger — *bored in a classroom* – that tends to elicit your bad habit), but value-updating being possible ("value-updating" meaning that the trigger, *bored in a classroom*, still means something to you, but now your reaction is updated).

The internal mechanisms of modern language models are similar. They fundamentally rely on the transformer module, which is built on key-value lookups. Transformer-based models learn to ask internal queries (key lookups) encoded as vectors (a list of specific, but somewhat noise-tolerant, numbers). Once a model has learned to look for a certain key, it's hard to unlearn. To change the model's behavior, it seems easier to change what the key points to rather than to get the model to change so that it ignores the trigger altogether.

The similarity between these two "add-only" mechanisms may not be a coincidence; perhaps brains internally use something akin to the key-value pairs, just as the transformer does.

4.2.2 How the mind model can meta-learn

Meta-learning can happen in the mind model in a few ways:

- **Planning**: When you understand you want to take on a new behavior in the future, you can perform explicit planning for your eventual actions. For example, you might put something on your calendar, or write down a list of things you want to do today. In this case, the model can simple capture the actions of using a calendar, or of writing a list, and the higher-level goals of these actions are only indirectly captured by the neural weights.
- **Association**: Often you don't know when you'll need to use a new piece of knowledge, such as learning to ask directions in a new language. In this case, it's useful if you can recall a relatively unpracticed action based on the correct context. The model could account for this in the following way: When you learn ahead of time, you have an understanding of the future context where the action will be useful, so that future context can be linked with the knowledge. The action itself can be stored as well as possible either through practice (such as language learning) or through understanding (such as reading a how-to guide).
- **Problem-solving**: There are other kinds of meta-learning, separate from either planning or receiving knowledge. If you're faced with a problem you've never solved before, and you don't know where to look up an answer (or don't want to), then you can try to simulate the probem in your head, and mentally consider potential solutions. If you arrive at an idea you like, this is it's own kind of learning.

I'd say this last kind of learning is based on *thinking*, so now is a good time to switch gears — let's take a look at how the mind model can capture sophisticated thoughts.

5 Thinking

Suppose you've just learned how to play tic-tac-toe, and it's your turn. This is an example of thinking that's easy to think about. You're "naughts" (circles), and it's your turn on this board:

You're considering the center square for your next move. I'm suggesting this example because, if you're brand new to tic-tac-toe, it's not immediately obvious that crosses (x's) will win. After a little thinking or experience, you can see this.

The mind model captures thinking as an internal feedback loop. Some of the output of the action model is received again as input for the next cycle.

In the tic-tac-toe example, the thought process might work like this:

- It's circle's turn. X's will win if they go in the middle next, so I better go in the middle.

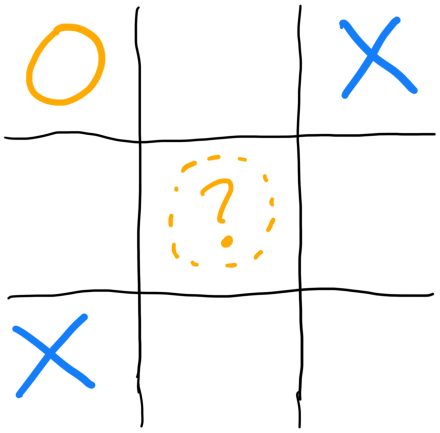

Figure 2: It's your turn. Should you go in the middle of the board?

- Then it's x's turn. Similarly, the x player better go in the lower-right corner.
- Now, imagining that board, I can see that the x player has two lines that can win on the next move. Circle can't block them both, so x must win.

In the mind model, each of these bullet points may be one iteration of thought through the action model. It would be more difficult to imagine a single iteration of an action model noticing that conclusion if it was new to tic-tac-toe. So each iteration is useful as a smaller step in a kind of search process toward better understanding of what's happening, or in a protocol of more carefully deciding what to do. That example is more of a caricature compared to the exact calculations that actually happen, but it illustrates the way in which a feedback loop can support internal thoughts building on each other.

5.1 What is thought?

I won't try to completely answer that question — but I'm still working in the hypothetical scenario where the behavior of the model is similar to a human brain's. While I'm not going to claim to understand all of human thought, we can notice a few interesting things about both this model, and how people seem to think. I'll talk about thoughts in an abstract, human-oriented manner, and then circle back to the model and explain how these can be captured by the mind model.

One mode of thought may be **predictions about the future**, including the future actions of other agents. This is clearly useful in a game-playing context, but it's also useful in many other scenarios. For example, if you're negotiating with someone (such as navigating the tricky terrain of a bed-time routine with a young child), it's useful to predict how the other agent will react to different

ways to communicating about the situation.

Another mode of thought can be **creativity**, wherein you're coming up with new ideas. An example of this would be in writing fiction, poetry, painting, or creating new music. In this mode of thought, it feels to me as if there's a general direction to the creativity, and we alternate between trial-and-error discovery of pieces of the work being created, or a mode in which we know what we want to achieve and simply put in effort to translate that goal into an actualization, such as painting an image we have clearly in mind.

A kind of thinking related to both of the above is **problem-solving**, in which case we want to achieve something but are uncertain about the best way to move forward. A toy example would be someone asking me a riddle. What's better than pizza but worse than taxes?[2] There's an interesting asymmetry to many problems we can try to solve: Often it's easier to *recognize* a good solution than it is to *find* that good solution.

So when it comes to problem-solving, our mode of thought may be a feedback loop in which a creative component suggests candidate solutions, and an analytic part of the action model decides whether or not this is a good candidate.

5.2 Advanced thinking

More sophisticated versions of each of these processes can exist.

For one thing, human brains clearly learn from experience. When you're better at tic-tac-toe, you can first see patterns that allow you to skip ahead in predicting the outcome of different boards — and eventually you can simply memorize the best possible moves. Similar pattern-recognition exists for more interesting contexts, from games like chess to real-world challenges, such as writing fiction (understanding tropes, audience reactions, dealing with narrative road-blocks) or running a business.

Related to pattern recognition is the concept of an internal mental vocabulary. A simple perspective is that mental "words" match words in the language we know best. By the time you learn the word "dog," you have an idea for what a dog is. But there are differences between our verbal and mental vocabularies. You can recognize an animal you've seen before without having to know what it's called. More abstractly, you can know how to deal with a situation you've been in before without needing a name for that situation.

Many people experience an inner voice, which seems to be just one particular way of thinking. I often thinking without an inner voice. But I do hear one, often, when I'm faced with a decision or problem that takes me a little more time to solve. Often my inner voice acts, to me, as a simple tool to help organize my own thoughts. For example, if I'm analyzing a list of options, I find it useful to "say" the options out loud in my mind to crystallize my comprehension of the

[2] Nothing.

full list. If I'm trying to solve a tricky math or coding question, I'll ask "aloud" (in my mind), title-like questions, such as: What's the simplest toy version of this problem? What other problems does this remind me of?

Whether or not you use an inner voice, there are still meta-protocols available to modes of thought. For example, in whatever job you have, you probably have faced many different variations of similar challenges. When those challenges can be helped with a lot of thought, you probably develop *templates* for solving similar problems. Because I like math, I'll use that as an example. In 1945, the mathematician George Pólya published a small book called *How to Solve It*, in which he outlined conceptual guidelines for tackling difficult math problems. These are examples of meta-protocols available to modes of thought. They are processes that are not learned the way you memorize how to play a piano song, but rather that seem to exist at a higher level in a hierarchy of thought because there are so many abstract and unknown variables involved in each specific implementation of this process.

5.3 How the mind model captures modes of thought

The mind model can capture prediction about the future by implicitly asking: What will happen next in this context? Or, more specifically, *what will this one agent do next in this situation*? This is captured by the action model just as a language model can simulate many different tones of voice. The default mode of the action model is to decide what the "self" actor will do, but, by adjusting the model's analog of a system prompt, we can ask the same module what another agent would do.

Creativity might be captured in a manner similar to stable diffusion. Specifically, we may have a context for what we want the creativity to achieve — this is like the text given to a text-to-image model. Then we have vague, noisy thoughts to begin forming our solution, and over time we work to solidify those vague thoughts into more concrete realizations that align with the context. If you're a novice musician, you can probably hum a short tune, or drum a simple beat with your fingers. With more focus and experience, you can begin to turn those simple ideas into more complete songs. While I have not explicitly called out a stable diffusion component within the action model, the idea is that part of the feedback loop can include a partially-solidified (and thus partially-noisy) vector representing the eventual output of the stable diffusion component, and one pass through the action model has the ability to serve as a stable-diffusion-style denoiser.

The problem-solving mode of thought is simply a combination of the above two pieces. Your creativity can suggest uncertain or incomplete pieces of solutions, and your prediction mode of thought can work to ansewr the question: If I tried to use this solution, would it solve my problem? This question probably takes on more specific formats that depend on the challenge at hand, such as: If I communicated this solution, would it convince someone else? Or: If I took the

actions of this solution, do I predict the outcome I'm aiming for?

The more advanced forms of thought also fit within the model.

For one thing, once we learn a word, that word must have a vectorized representation as an output of the language encoder. This output vector is an internal mental concept used by the action model — this kind of vector is exactly analogous to the internal token vectors used by large language models. This mechanism shows how learning to understand words adds to our internal mental vocabulary.

It's one thing to understand what a word means, but another to produce the word while writing or speaking. Generally, people have a larger reading vocabulary than a spoken vocabulary. The mind model can explain this because it's easy for the model to receive a word that it is unlikely to produce as output, since the language encoder and decoder are different systems. This can explain how pushing yourself to use a word in a sentence several times helps to add that word to our output (spoken or written) vocabulary.

All of the above, taken together, helps to show that the action model does indeed have an internal mental vocabulary which aligns closely with, but is in no way limited to, the concepts captured by a verbal vocabulary.

Another example of a thinking style is an inner voice, which is a special case of the feedback loop where the output of your action model makes use of the language decoder, translating non-verbal concepts into a verbal sequence. That internal verbal sequence is then received by the language encoder, and your internal percpetion is very similar to hearing a voice spoken aloud.

When you develop habits of thought, such as trying to solve a math problem by beginning with a simplified version of the problem, then we're touching on processes that aren't directly part of the action model, but rather emerge at a higher level. This is analogous to the way we can drive a new car in a new country on the other side of the road (perhaps with some stress), even though there is certainly no single neuron, or even really a subset of neurons, dedicated to this kind of activity. Put another way, when you're studying a CPU at a given level of abstraction, it's possible for the system to handle more complex operations than what can be done by a lower-level perspective. The mind model is captures a low level picture, so that sophisticated actions and ideas are out of scope. We just need to know that these more complex actions and ideas are enabled, just as a simple Turing machine can support any potential program.

Old stuff below here.

5.4 Enabling agency

I'll explain this idea first because it's easy to understand even before I present the mind model.

A large language model doesn't have agency because it can only respond to input.

However, we can imagine a change that adds agency to an LLM-like system. Think of a model that receives two input streams that are interwoven together. One input stream is from the person talking to the model, and the other is the model being able to see its own output (which is the way current LLMs operate; they need to see their own output to keep talking).

When the LLM wants to listen, it can produce a special `<listening>` token for a long sequence. In this sense, the LLM can always be running, while still enabling a meaningful two-way conversation that includes pauses for the other speaker. That is, such an LLM can independently say whatever it likes whenever it likes, which is the lexical version of agency.

5.5 An action model

You can think of an LLM, in simple terms, like this:

context -> LLM -> next_token

A core piece of the mind model here is an analogy to an LLM that I'll call an *action model* because its output is a sequence of actions to take, rather than tokens to speak:

context -> LLM -> next_action

Conceptually, I'm thinking of an "action" as a superset of words. If I wanted to say "hello," then "say hello" is an action. If I want to walk to the kitchen, that's an action. And if I want to ponder the meaning of life, that's an action. I'll elaborate on this below.

5.6 The model at a high level

To-do: Explain the diagram briefly first.

6 Learning and memory

How does this model account for the ability of a mind to learn?

I think this is an important question because it's a key difference between what a human mind can do and what an LLM can do. A typical LLM today forgets everything as soon as you start a new conversation.

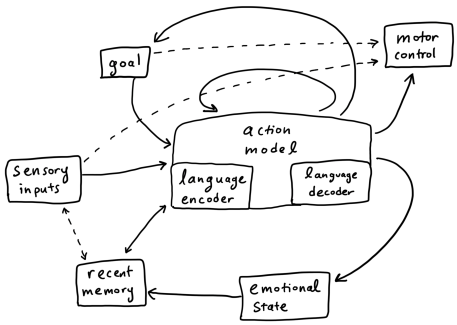

Figure 3: Data flow diagram for a model of a mind.

You might reply by saying that augmented LLMs can do better by either having a long context window (which is a kind of way of storing and using the full chat); or by using retrieval-augmented generation (another way of storing and using the full chat). However, in both of these approaches, the weights of the LLM remain the same, so there's a disconnect between these ideas and how brains seem to work.

Another way to phrase my goal is: How can a mind model learn without explicitly storing the chat history?

To talk about how the above mind model can learn, I want to categorize learning into two types:

- Remembering what has happened — I'll call this *story memory*; and
- changing how I act based on feedback, such as learning from a teacher how to play chess — I'll call the memory which helps with this *action memory*.

There's one big difference between what goes into story memory and what goes into action memory: When you make decisions, such as in playing chess, you might be doing well or poorly, but you don't always know immediately if the action was good or bad. The goal is to evaluate if your action was good or bad. If it was good, you want to reinforce that action, and if it was bad, you want to avoid repeating it. On the other hand, what you experience as the story of reality does not require you to judge if the story is right or wrong (ignoring, for now, the possibility of us making perception errors, such as having a hallucination).

In terms of how the model works, story memory is saying "this happened," and can be baked directly into the action model, just as many facts about life are baked into the weights of an LLM. We don't want to auto-learn that our actions were correct, however, when we make decisions. So the model has a way to store actions it has taken, and to later choose to reinforce the quality of those actions.

The motivation for the "recent memory" module is a place that can essentially memorize exactly what has happened recently before it's baked into the action

model. I suspect this is useful because, as you fine-tune LLMs, you can easily cause catastrophic forgetting, which is the effective erasure of old memories. In other words, in practice it seems that new memories are based added carefully, by sprinkling them in between recalls of old memories to help keep those old memories intact. Another use of the recent memory module is to provide a delay on considering my own actions as good until after I've received feedback about that action.

A third motivation to have a separate recent memory module is that a detailed memory of the past few hours is much more valuable that an equally-detailed memory of some random window of a few hours from when you were four years old. The usefulness of story memory decreases rapidly with time, so there's a trade-off between the capacity in your action memory versus the volume of sensory input you receive. It's convenient to have a rolling window of accurate memories that are forgotten as enough time passes.

This breakdown of memory types might account for these features of human memory:

- We seem to have a small memory capacity that we receive with almost no thought about the thing being remembered. George Miller did work to establish that most people can quickly remember about seven items from an arbitrary list. That memory might fit into the feedback vectors of the action model itself. This memory disappears as soon as we think enough about something else.
- Different people have different recent memory capacities, but it's common to remember what you ate for breakfast this morning, but not what you ate for breakfast several days ago, ignoring predictability (such as if you cheat by eating the same thing for breakfast every day). This type of memory matches what can fit into the recent memory module.
- Longer-term memories don't seem to have a pre-determined time limit, but they do tend to fade over time. This pattern is consistent with knowledge baked into LLMs, and so can match the way an action model would effectively remember things — without a time limit, but with the ability to fade over time, especially if not referenced for a long time.

People tend to remember events in more detail when their emotions were strong at the time. Conversely, people tend to forget moments when they were bored or not paying attention. Think of the last time you took a well-worn route, such as your daily commute to work. You probably don't remember how you spent much of that commute, or at least, you probably don't remember the details that don't matter, such as the color of the car in front you at a certain intersection.

The mind model accounts for this by filtering memories through emotional states. In order for the model to remember something, it must be both (a) something the action model has paid attention to, and (b) something the mind cares to remember based on the emotional state. In addition, the emotional state is part of the context for the action model, so that goals are influenced by how the

mind feels, and what the mind pays attention to is likewise influenced by feelings. For example, if the mind is in a happy mood, it's more likely to appreciate the positive aspects of a conversation; if it's feeling defensive, it's more likely to notice a perspective from which a conversation can be seen as judgmental.

I'm using "emotional" in a broad sense meant to include pleasure, pain, boredom, happiness, frustration, and any combination of states of mind that have a not-purely-rational feeling associated with them. The most basic aspect of this — akin to simple pleasure or pain — can be seen as a relatively quick feedback loop to inform if the recent action memories are good are bad for the sake of learning. If a recent action was akin to hitting your thumb with a hammer, then you'll have pain as a clue to no longer take that same action.

6.1 Meta-learning

Another kind of learning happens at a higher level, which requires longer-term thinking. For example, suppose you write a first draft of a book, and then give that book to some beta readers for feedback. You can view this as a process with many months between the action first taken — writing your first word of a new book — and receiving useful feedback on that action. The recent memory is no longer a useful vehicle for this kind of learning.

In this case, I suspect humans learn a process in a more explicit manner. I'm convinced that humans learn rational behaviors as action sequences which are initiated by memory triggers. For example, when I want to write an idea that's already well formed in my mind, I'll either record a voice memo of the outline, or I'll type an outline draft in google docs. That's part of my personal process. The trigger is the combination of (a) wanting to write an article, and (b) not needing to do more research, that is, feeling confident I'm ready to write. The action sequence, at a high level, is to make the outline.

Now suppose I get feedback on my action sequence. For example, maybe the voice recorder app on my phone deletes a file through a bug. Then I'll make a mental note to use a different voice recorder app. This kind of learning is not happening at the level of weight updates in a neural network. Rather, it's a more conceptual idea that is best seen as over-writing the key-value pair:

```
[I want to record an outline] -> [open voice app A]
```

by re-using the same key, and replacing the value, like so:

```
[I want to record an outline] -> [open voice app B]
```

I've phrased things this way specifically because human brains don't seem to be good at erasing past memories, but rather they seem to be able to *replace* keys associated with pre-existing keys.

6.1.1 Key-value memory in humans and AI models

Consider a person with a bad habit, such as biting their nails. It's notoriously difficult to enact a strategy of simply stopping such a habit. If you do this and your thought is "I'll just stop," you're likely to fail. However, if you *replace* the bad habit with something else, you're more likely to succeed. For example, you can notice the situations where you're most likely to bite your nails — such as sitting in a classroom and somewhat bored — and then teach yourself to take a *different action* in those same contexts. For example, you might use a fidget spinner instead of nail-biting. This is a human-oriented example of key deletion being hard ("key deleltion" here would be like ignoring the trigger — bored in a classroom – that tends to elicit your bad habit), but value-updating being possible ("value-updating" meaning that the trigger, bored in a classroom, still means something to you, but now your reaction is updated).

The internal mechanisms of modern language models are similar. They fundamentally rely on the transformer module, which is built on key-value lookups. Transformer-based models learn to ask internal queries (key lookups) encoded as vectors (a list of specific, but somewhat noise-tolerant, numbers). Once a model has learned to look for a certain key, it's hard to unlearn. To change the model's behavior, it seems easier to change what the key points to rather than to get the model to change so that it ignores the trigger altogether.

The similarity between these two "add-only" statuses may not be a coincidence; perhaps brains internally use something akin to the transformer architecture. But, even if this similarity is a coincidence, the stance of this article is to look for functional equivalence in a mind model, so the similarity in behavior suffices to make this implementation model useful.

6.1.2 How the mind model can meta-learn

stuff (I think thinking is needed here, so I suggest segueing into the thinking section here.

7 Thinking

I think when we're thinking, we have a feedback loop where the actions are purely internal and we seem to have an implicit candidate-and-test cycle where we have an idea and that idea can be verbally-encoded or pre-verbally-encoded. The way that works is that it depends on what meta-level we want to operate.

One meta-level is a group of people talking and deciding something. The next level down is writing down on a piece of paper. Another level down is talking to yourself. And another level down is thinking in your head. (I feel like I think a little more clearly when I think out loud versus in my head, for whatever that's worth.)

So there are different meta-levels.

There are different meta-levels. The verbalized encoding level would correspond to us doing everything we do when we speak except that it short-circuits us actually saying the thoughts out loud. That way we can have candidate ideas that we can consider as if someone else had said them.

This is better than not having a feedback loop because we have a way to iterate. What can we do with internal thoughts? We can judge them, we can edit them, we can discard them.

As soon as we can take an action on thoughts from others in the external world, we can take those actions on our own thoughts. As soon as we can model someone else's attention, we can model our own attention. And modeling other peoples' attention is awareness (Ganziano), and modeling my own attention is self-awareness.

Thinking can be pre-verbal encodings. That's a feedback loop that doesn't have to use all of the language encoder/decoder pieces within the action model.

Side note that each module in the mind model is meant to be its own, always-running, computationally-capable mini-CPU. For example, motor control can keep us walking forward without needing constant supervision.

Also, the action model itself might be some complicated that we may end up thinking of it as a mixture of experts, with many independently-running pieces. In that case, maybe some additional process happens to decide which of many independent outputs is used to decide the next action.

Two questions: blindsight question, and then how can I express the idea of consciousness in this model?

7.1 Consciousness

I think consciousness is the ability to answer questions like why did I do that, or did you notice what just happened — were you aware of something?

7.2 Blindsight

If we think of there as existing many pieces of an action model in the brain, then what I can imagine happening in the blindsight case is that there is an action model that receives from vision, and is able to suggest things to motor control. But the action model that's in the loop with vocalization is not connected to vision.

So, specifically, the way I'm modeling something going wrong with blindsight is that vision is going to part of the brain, but not all of the brain. There's a big part of the brain where what we've perceived as conscious vision, where we receive a signal in a specific action model, that's not getting anything from the eyes.

And so what we think of consciousness in this model basically indicates availability — not only availability for attention — but effectively achieving attention. It's not critical that we have to talk about something for us to be conscious of it, but it's how we know someone was conscious of something. So, for internal consciousness, the feedback loop. Is there one feedback loop that's entirely responsible for what we do? I don't think that's true.

I think there's more than one feedback loop. I think it's more about what we remember than about vocalization.

The relationship is a little complicated here.

I think what we tend to feel we are conscious of, I think corresponds most with what we remember. Because awareness is a precursor for memory. And even with attention, there's sort of a do-we-care / don't-care about it.

There's this question of the sliding dots example. How ... I don't even see why that's such a big paradox, to be honest.