Basic training strategy and why it might work

The fundamental approach is to force the artificial operator (the 'Actor') to occasionally represent its understanding of the world in a carefully-incentivized natural-language latent space, thereby becoming a 'note-taker'.

The basic structure is that during the training of the Actor, at every step of inference in the Bayes net there is some probability that we will force it to first encode its internal state in text, perform various manipulations on this text, then decode it, while throwing away all other information. We use these manipulations to incentivise the use of language as humans do, and train the Actor to maximise its capability given this constraint. The model therefore needs to learn a mapping between the model's internal ontology and human ontology as a condition for optimal performance.

When the system is in use, we can then constantly create 'notes' from the internal state, while not forcing the inference to go through the notes, to get a form of live-readout of the internal state. Text is encoded from the hidden state by a `text_encoder` function and the inverse is performed by `text_decoder`.

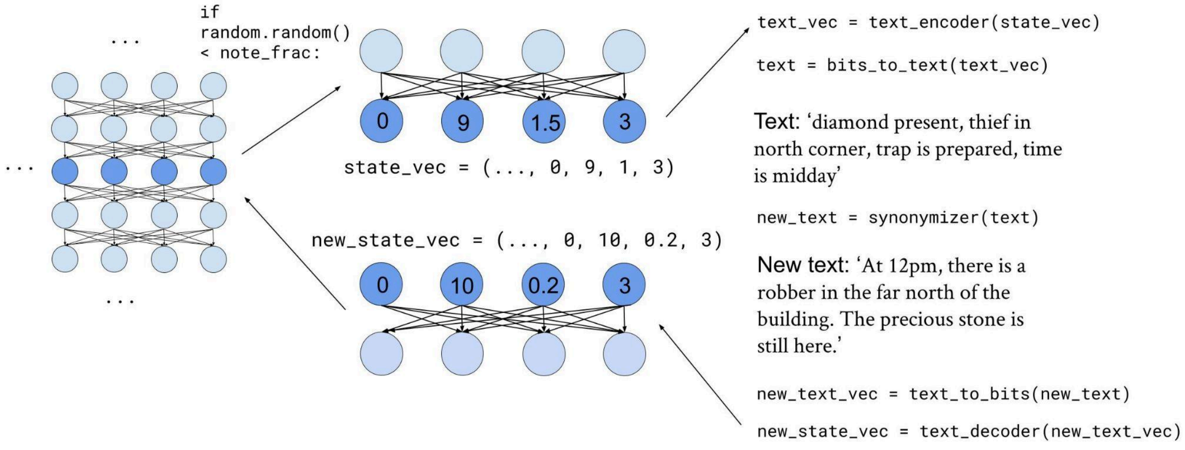
If this protocol allowed the network to write free text, it would simply use 'language' as an ordinary, inscrutable latent space. We therefore incentivise use of language in the same manner as humans in a number ways which, in combination, are hopefully enough for humans to extract meaningful insight into the internal state from the notes that are taken.

- We take a variable-length binary string from the model and pass it through a deterministic function, in which common bitstrings map to likely text. For decoding we apply the inverse of this coding. These functions are called `bits_to_text` and `text_to_bits`. **We therefore force the text to superficially look like human communication.**
- We fine-tune a large language model (hereafter called the 'Synonymizer') to take any piece of text and randomly return any piece of text which a human would judge to carry the same meaning, but which could differ wildly in style, structure etc. Ideally this could even create synonyms across modes, since images and video are key for human understanding and for grounding meaning - current experiments would start with text but I expect multi-modal models with well-grounded usage of language to be a big area of improvement in the next few years. We supplement this automatic synonymization where possible with the work of multiple humans to discourage learning quirks of the model. **We therefore incentivize the model to use language in a way which is isomorphic to the structure of human language.**
- We pre-train and co-train this note-taking architecture with as many examples as possible where the ideal way to use these notes is to interpret language as a human would. Since the latent space is assumed to be extendable, we can withdraw access to certain sensors and instead append useful information to the latent which can be used to respond correctly, providing it learns to properly interpret the text. If the synonymizer worked with image or video, these could also be appended. Existing

image and video labelling datasets should also be leveraged. **We therefore incentivize use of words in alignment with that of humans.**

- We train the Synonymizer to reject any language which it does not think is relevant to the task at hand by returning some unchanging sentence (as large language models can now be trained to refuse to answer nonsense questions). This makes it difficult to use language outside of that found in training examples with meanings different to human understanding. **We therefore restrict the scope of language to that which is expected to be relevant to the task.**

These elements in combination should allow some of the basic parts of how the Actor sees the world to be extractible at any point during inference, expanding the range of situations where we have a clear picture of what's really going on.



```
if
random.random()
< note_frac:
```

```
state_vec = (..., 0, 9, 1, 3)
```

```
text_vec = text_encoder(state_vec)

text = bits_to_text(text_vec)
```

Text: 'diamond present, thief in north corner, trap is prepared, time is midday'

```
new_text = synonymizer(text)
```

New text: 'At 12pm, there is a robber in the far north of the building. The precious stone is still here.'

```
new_text_vec = text_to_bits(new_text)

new_state_vec = text_decoder(new_text_vec)
```

```
new_state_vec = (..., 0, 10, 0.2, 3)
```

Figure 1. Diagram and pseudocode showing the note-taking loop

More precise detail on strategy

One important assumption here is that the Bayes net has a structure such that we can cut across the graph at a variety of points in its simulations to produce an array representing the world at this point.

In recurrent, model-based architectures such as MuZero, there are obvious points at which to cut, but the approach would work so long as the Bayes net has sufficient structure to be interpreted along many possible cuts by the same network. If this is not the case, additional information would need to be provided to the encoder and decoder.

There may be some imprecision with my use of 'latent space' for a set of words with no fixed length. More specifically I am imagining a transformer, or other form of sequence predictor, which takes the information before the cut, and returns a sequence of bits to go into the encoder which is of unlimited length but in which length is penalised.

It is likely to be optimal to share weights between a question-answerer and this note-taking section of the code, reducing training overhead, and mutually reinforcing the incentive to use language as humans would.

It seems probable that if this strategy were applied to a broadly super-human Actor, the `text_to_bits` and `bits_to_text` functions become unnecessary, as learning the syntax and structure of human language becomes an easy task without noticeable performance cost. For current purposes, though, I expect this additional structure to make training easier (though still difficult - the degree of freedom afforded to the synonymizer may need to be slowly increased for training to be possible).

Counterexample

The scope of cases where this strategy would work is dependent on the possibility of training the system to write meaningful notes, and the extent to which, when language is pushed beyond our own usage of it, some hint of the meaning is still discernible. To my knowledge these are open questions.

The best-case scenario for this approach is one in which the Actor takes a series of actions which would not be found or understood by a human, but which has a *result* which is comprehensible, but easy for a human to misunderstand - for example, making the diamond in the vault disappear. In such an example, the situation has a clear description in human language, but a human would be expected to answer that the diamond was not there, and a human operator would not be able to make the situation arise to provide correct answers.

These situations could easily fail, if the note-taker fails to generalise from the use of ‘disappear’ in training examples, either in the broader, language understanding training, or in task-specific training, where the diamond could be made to disappear by prosaic means. Nonetheless, it seems possible that this strategy, which has a lot of dimensions for additional improvement, could become an important part of our toolset for understanding such models.

FAQ

I now attach some potential questions or criticisms, and my thinking in response:

- Note taking of this kind may not be capable of high performance.
    - High performance with note-taking is not necessary for the strategy to be useful, only that it is incentivised to make the major elements of the situation as clear to itself as possible.
    - In particular, we should imagine that the relevant information can be placed on a scale from discrete lumps to diffuse distributions. The note-taking would necessarily discard the long tail of subtle probabilities while retaining the biggest lumps of information.
- Note taking may never reach a level of performance that the resultant state is anything like that which would be reached without notes, and the model would just learn this as a totally separate type of hidden state.

    - If training is conducted to maximise similarity between the result of the note-taking system (as in EfficientZero) and the ordinary state-transition then I don’t think this would be a stable equilibrium, and in any case, as long as the notes provide meaningful context, this should not be fatal.
- Some of these discrete pieces of information would be communicated in the way that humans expect, while others would be communicated using what appears to be relevant human language but would in fact be a private language from the algorithm to itself - ‘disguised communication’.
    - This risk is real but does not negate the possibility that such an architecture would increase the space of cases where we would get an honest understanding of how the Actor understands the world, though we need to be mindful of the risk of overestimating our understanding. It can be mitigated by:
        - excluding communication that doesn’t make sense as a response to the situation (see above) as fully as possible, while
        - detecting when there are discrete pieces of information that it would find useful to communicate, and
        - finding the most efficient protocol to communicate with humans the nature of these discrete lumps so that they can be integrated into a shared vocabulary.
        - Speculatively, we could train the model with non-linguistic free bits in the latent space to identify the most important pieces of information not easily communicated in language, then use some kind of adversarial structure to identify the cases in which this bit is activated, and incorporate it into a shared language.
        - Note that if this protocol worked perfectly it wouldn't be necessary to use human language, we’d just learn the outlines of the most important discrete concepts from some small discrete latent space. Language is preferable, though, because there already exists the human understanding, training sets and models for a huge variety of important concepts, and so starting there should significantly speed up the communication process.
    - It is hoped (though not guaranteed), that in finding words to serve new situations, it would extend existing language in ways which give us insight into its thought process, even if not total clarity.
    - It is also not clear that this is much less of a problem for when a question-answerer tries to give answers about an actor who has taken a series of actions that a human would never take, and there isn’t a clear set of human examples for how a human would answer. Sharing weights between the reporter and this note-taking structure would maximise the likelihood of clear communication in both cases.
- This approach fails to reveal new internal information to humans beyond the case where humans have operated the machine themselves.
    - This is not true because, while there is a similarity between a. those positions which the human is able to reach as operator, and b. those positions in which natural language can describe the situation: the latter is broader.
    - For example, the Actor may find it possible to convince a would-be robber not to attempt to steal the diamond, using mechanisms that a human would find incredibly difficult to understand. This option would never be discovered if

operating the Vault themselves, and the human questioner may reliably not think of asking. Nonetheless, ‘X has been brainwashed’, or some similar phrase should be the natural way of representing this state of affairs, and therefore we reasonably could hope that this will be present in the notes.

- Is this compatible with training via gradient descent?
    - The synonymizer would be a large language model which can be trained through. Work would need to be done to make differentiable `text_to_bits` and `bits_to_text` functions but this seems manageable and in any case these should not be necessary at high capability levels.
- The complexity and additional training required make the system unworkable.
    - The complexity of learning such a task is high, but it seems that it should not scale with the complexity of the model. It needs to keep track of the structure and referents of language, but as the size and capability of the model scales, this task should need only incremental improvement, so should not be a fatal objection for super-human models, though a successful demonstration in simpler cases may be very difficult.
    - The amount of additional training is unknown. What is needed is to create a very large dataset which incentivises the use of language as humans understand, and to train on this sufficiently often that this capability doesn’t degrade. This is most easily achieved for models which act on similar scales and situations to humans, for it is there that language is optimised for communication.