doi:10.20944/preprints202106.0635.v1

Article

Fast Learning in Complex Domains

Robert Worden [1]

[1] Wellcome Centre for Human Neuroimaging, Institute of Neurology, University College London, London, United Kingdom; rpworden@me.com

* Correspondence: rpworden@me.com;

Abstract: Bayesian formulations of learning imply that whenever the evidence for a correlation between events in an animal's habitat is sufficient, the correlation is learned. This implies that regularities can be learnt rapidly, from small numbers of learning examples. This speed of learning gives maximum possible fitness, and no faster learning is possible. There is evidence in many domains that animals and people can learn at nearly Bayesian optimal speeds. These domains include associative conditioning, and the more complex domains of navigation and language. There are computational models of learning which learn at near-Bayesian speeds in complex domains, and which can scale well – to learn thousands of pieces of knowledge (i.e., relations and associations). These are not neural net models. They can be defined in computational terms, as algorithms and data structures at David Marr's [1] Level Two. Their key data structures are composite feature structures, which are graphs of multiple linked nodes. This leads to the hypothesis that animal learning results not from deep neural nets (which typically require thousands of training examples), but from neural implementations of the Level Two models of fast learning; and that neurons provide the facilities needed to implement those models at Marr's Level Three. The required facilities include feature structures, dynamic binding, one-shot memory for many feature structures, pattern-based associative retrieval, unification and generalization of feature structures. These may be supported by multiplexing of data and metadata in the same neural fibres.

Keywords: Fast learning; Bayes' theorem; navigation; language; spatial cognition; feature structures; unification; generalization; dynamic binding; metadata multiplexing.

.

1. Introduction

This opinion piece considers a new direction in machine learning and artificial intelligence that confronts a fundamental problem: namely, why have neural networks and deep learning [2] failed to provide an account of the fast learning and knowledge accumulation that characterises real-world sentient behaviour?

I suggest that current machine learning approaches fail to consider the underlying generative model, or structure, underlying the regularities that are to be learned (and their Bayesian priors). Failure to address model structures leads to inability to learn the parameters of those structures efficiently. For example, simply knowing that an association can exist – or not – structures learning by converting the learning problem into an inference problem; namely assessing the evidence for an association, relative to no association. In general, this kind of inference is substantially more efficient (and quicker) than weight learning in neural networks.

In what follows, I review key structural attributes that are needed to construct models of the sensed world, with a focus on scene construction (model building) in the spatial and temporal domains. I will take navigation and language as cardinal examples of the structure learning problem – and use them to suggest a re-examination of a long-known direction of travel in machine learning.

Specifically, I hope to take established constructs in computer science and engineering as the starting point for a mathematically-based approach to machine learning, which can rapidly learn composite feature structures. I show how these imperatives emerge under Bayesian first principles of learning, which define a maximum possible speed of learning in any domain (a speed which animals attain).

I relate this structural viewpoint to recent developments in the neural modelling of brains – notably the Free Energy Principle [3,4,5,6], the Thousand Brains model [7,8], and the Capsule Net/GLOM model [9,10]. The concepts of this paper may offer a common vocabulary in which to compare these different approaches, and re-frame some of their insights.

2. 2. Bayesian Inference in Complex Domains

In this section I propose a formalism to characterize Bayesian cognition in any domain - which captures some universal, cross-domain aspects of Bayesian inference, but which also captures key domain-specific aspects. This is a description in terms of multinode feature structures, and the operation of unification of feature structures. Unification is the key operation required for Bayesian scene construction – for building generative models of the animal's surroundings.

Animal brains perform impressive computations in many domains, most notably in the domain of three-dimensional spatial cognition – using sense data of all modalities to understand the 'what' and the 'where' of their surroundings. Animals infer not only what there is (an edible berry, an obstacle, a predator, ….), but also where it is, in a three-dimensional model of their surroundings which they use to control their movements. This computation is so complex that it is still a challenge for engineers in robotics.

I note three aspects of the 3-D spatial computation:

1. It involves composite things, which have a whole-part hierarchy. For instance, a face has a nose, mouth, etc.
2. It involves both whole-part relations and part-part relations. (for spatial cognition, these are 3-D geometric relations)
3. The goal of the computation is to construct the most likely possible model of the current situation, in the light of all sense data, as a basis for action. A model is a composite structure. Bayesian maximum likelihood inference governs the construction of models.

The same three features can be found, with variations, in other complex domains. In this paper I focus on two domains of complex cognition:

In **animal navigation**, (1) the composite thing is the geography of a landscape, with its whole-part relations of regions and places; (2) the relations are the relations of 2-D Euclidean geometry, between regions and places; (3) the model to be constructed is a cognitive map, containing at least the animal's present location and the place it needs to go to.

In **human language** (1) the composite thing is the syntax and semantics of a sentence; (2) the whole-part relations include syntax trees and tree-like meaning structures; (3) in language understanding, the goal of the computation is to find the syntax and the meaning of a stream of words; in language generation, the converse goal is to construct a sequence of words from a meaning.

For further illustration, I mention a simpler domain:

In **sequence processing** (1) the composite thing is a sequence in time (such as a bird song, or a sequence of movements); (2) the whole-part relations are relations between sequences, sub-sequences and individual events; the relations are temporal relations such as 'precedes' or 'is 5 seconds after' (3) the goal of the computation may be to recognize and classify a sequence, or to create a sequence to meet a goal.

All these domains require learning – to learn composite structures before they can be used. I will, for the moment, set learning on one side and ask: given some learning ability to acquire composite structures, is there any common model for the inferences required to use the composite structures, applicable across all these domains? There is such a model - the model of feature structures and unification.

A feature structure is a composite data structure - a graph consisting of several nodes and links (edges) between the nodes. Both the nodes and the edges carry information.

Feature structures can represent situations in the world, and they can do so in any of the domains of cognition above. For instance:

- In spatial cognition, there can be a feature structure for a face, with a top 'face' node, and whole-part edges to subordinate nodes for 'mouth', 'node' and so on. There can be edges at each level to represent geometric relations between the parts: 'nose' is 5cm above 'mouth' and so on.
- In navigation, there can be a feature structure for a piece of geographic information (or a fragment of a map) – like that shown below, involving a gate, a tree and a pile of logs. The map fragment has a node for each of these landmarks. Edges describe the 2-D geometric displacements between the landmarks.
- In language, a feature structure can represent the syntax of a sentence or of a part of it; or it can be a semantic representation of meanings, or syntax and semantics may be combined in the same feature structure. An example linguistic feature structure is shown below.

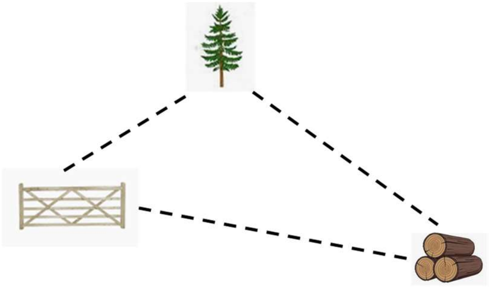

Figure 1: a feature structure in a computational model of animal navigation [11]. The nodes of the graph are landmarks, and hold their properties; the edges represent 2-D geometric displacements between landmarks

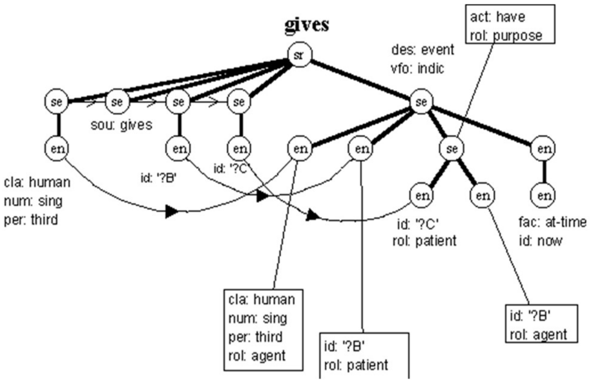

Figure 2: a feature structure representing the word 'gives' in a computational model of language processing and acquisition [12,13]. The left-hand branch represents the syntax of the word 'gives' (with donor, the sound 'gives', recipient, and gift). The right-hand branch represents the semantics of giving. Curved arrows relate the syntax to the semantics.

Feature structures, then, are a generic data structure which, with appropriate specializations (e.g., of the edges) can represent information in any complex domain of cognition.

I emphasise that feature structures are a logical data structure, defined at Marr's [1] Level Two; and their relation to physical structures at the neural Level Three may be a complex one.

To relate feature structures to Bayesian inference, I first consider their semantics – how they represent parts of the world. Any feature structure (denoted by F) has an information content of I(F) bits, which is given by summing the information content of its nodes and edges. Feature structures with larger information content represent less likely sets of possible situations, in the following sense:

If a feature structure F has I(F) bits of information, then it represents a set of situations in the world, whose probability is approximately $2^{-I(F)}$.

This is the fundamental link between feature structures and Bayesian inference. From it, we can see which operation on feature structures is needed to perform Bayesian maximum likelihood inference (or model construction) in any domain.

That operation is unification of feature structures. The mathematics and computation of the unification have been studied in computer science (for instance in computational linguistics) and are described in the Appendix. Unification is the basis of the computer language Prolog [14]. There follows an informal description of unification.

To unify two feature structures A and B (to construct their unification C = A U B), you try to match the nodes of A with the nodes of B, so as to maximise the shared information content on matched nodes, while respecting any constraints from the edges or nodes (incompatible values on nodes or edges cannot be matched). Then, if there is a possible match, the unification is the best matching feature structure containing all the matched nodes, and all the unmatched nodes and edges from both A and B.

Thus, the unification C (if it exists) contains all the information in either A or B; therefore, the set of situations described by A includes the set described by C (and similarly for B). But since C has maximized the amount of information matched between A and B, so it does not appear twice in C, the information content I(C) is less than I(A) + I(B), and is as small as it can be. Through the relation of information content to probability, C describes the most probable set of situations compatible with both A and B. If A and B are incompatible (represent disjoint sets of situations), they have no unification C.

This shows that unification is the fundamental operation of Bayesian maximum likelihood inference on feature structures. If a state of affairs is described by A, and is also described by B, then C = A U B describes the most likely state of affairs, satisfying the constraints of both A and B.

For instance, if feature structure S represents what an animal knows from current sense data, and a feature structure R represents some general rule or regularity, also known to the animal, then T = S U R describes what can be inferred from the sense data using the rule. Unification is the way to infer the consequences of any rule once it has been learned as a feature structure.

More generally, each feature structure is a partial generative model of reality. Feature structures have the capacity to represent complex, composite generative models. Several feature structures can be unified together, to form a more complete generative model – which, because of the properties of unification, is a Bayesian maximum likelihood model. Unification fills in the gaps in partial models of reality.

Feature structures and unification can be used to describe Bayesian maximum likelihood inference and model construction, in any domain.   This does not imply that unification is entirely generic and cross-domain; because there are different domain-specific constraints (e.g., for navigation, they are the constraints of 2-D Euclidean geometry). Depending on the nature of the constraints, an efficient unification procedure may take different forms. Generic hill-climbing may not be the most efficient procedure to unify feature structures.

The applicability of feature structures to describe inference has been demonstrated in (at least) two complex domains:

1. The computations of feature structures and unification were first applied in unification-based grammars, which give a good account of all human languages, including their great syntactic diversity, and their mixture of syntactic regularity and irregularity [15,16,17,18]. Together with Bayesian probability, unification-based grammars give a unified model of language semantics and syntax – for both language generation and understanding.
2. Animal navigation has been modelled [11] as a process of map construction by geometric fitting of map fragments, like the fragment shown above. This geometric fitting is a form of unification of map fragments (but the relation of feature structures to place cells [19] and grid cells [20] is a complex one, and is not yet fully understood [21,22])

I next turn to the problem of learning rules and regularities as feature structures, within a Bayesian formalism.

3. The Bayesian Theory of Learning

The central result of Bayesian learning theory is simple and memorable. The theory predicts that as soon as the evidence for some regularity in the environment is statistically significant (and no sooner) that regularity can be learned [23,24,12].

This implies that the number of training examples needed to learn some regularity can be very small; so that if animal brains are nearly Bayesian, they can learn regularities from small numbers of training examples. This result, which agrees well with extensive data on associative conditioning and other forms of learning, is at variance with the much slower rates of learning of neural net models [2], which typically require many thousands of training examples.

The speed of Bayesian learning can be illustrated by the example of a biased coin. Suppose a coin is biased, and gives heads on 80% of tosses. How many tosses will it take to learn that the coin is biased? When the coin has been tossed 20 times, and has given heads approximately 16 times, there is only a 1% chance that it is unbiased; so, there is statistically significant evidence that the coin is biased. The bias can be learned rapidly, from a few tosses.

To apply this to animal learning, we take the case of two events (called c and d) which may or may not be correlated with one another in the animal's environment. We take it as given (and known by the animal) that event a necessarily precedes event c, and that event b is a necessary (but not sufficient) antecedent for d. Then there is a correlation between c and d if

$$P(cd|ab) \neq P(c|a)*P(d|b)$$

Each time the preceding necessary events a and b occur together, this is a possible learning example, and the animal records whether or not c and d also occur. There are four possible outcomes – (c & d), (c & not d), (not c & d), and (not c & not d). The frequencies of these combinations will reveal any correlation, positive or negative, between c and d. If there is any correlation, then by elementary statistics (applying a simple special case of the Dirichlet distribution [25]), the evidence for that correlation will soon become statistically significant, as is shown in the figure:

doi:10.20944/preprints202106.0635.v1

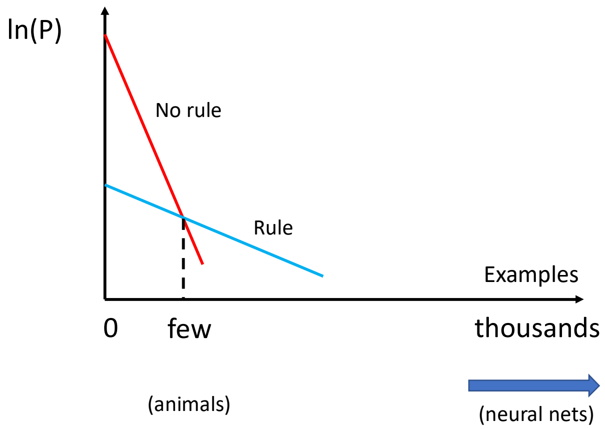

Figure 3: If there is a correlation between two events c and d, the evidence for that correlation becomes statistically significant after a small number of training examples. The prior probability of any rule is small, so the blue line starts lower than the red 'no rule' line'. But the blue correlated line gives a better account of the training events, so it has a smaller slope than the red line. When the blue line overtakes the red line, evidence for the rule is statistically significant. Slopes of the lines are given by elementary probability theory.

This shows the logarithm of the probability of a sequence of examples, as the sequence extends. When there have been no examples, the prior probability of any correlation between c and d is small; thus the 'rule' line starts below the 'no rule' line.

However, as examples accumulate, if there is a correlation between c and d, the 'rule' explanation gives a better account of the examples than the 'no rule' explanation; so, the red 'no rule' line descends linearly (on average), with a greater slope than the blue 'rule' line . There are random statistical fluctuations about the two lines. The slopes of the two lines can be calculated by elementary probability theory, as in [23, 24,26] and are special applications of the Dirichlet distribution [25].

After the two lines have intersected, the evidence for the rule is statistically significant- enough to believe the rule, in spite of its prior probability being lower than 'no rule'. At this point, an animal can learn the regularity, and act as if it is true.

It is in principle not possible to learn a correlation between events c and d any faster than this – because before the two lines intersect, any apparent correlation between c and d might just be a statistical fluke.

If there is only a weak correlation between c and d, the two lines will have similar slopes, and will it take many training examples before they intersect. More typically, the lines intersect quite rapidly, after a small number of examples.

In this mathematical analysis, a, b, c, and d can be any kind of event; any of them can be a composite event, described by a feature structure. So, the underlying relations a=>c and b=>d can be described by feature structures; and any correlation between the two, such as ab=>cd, is also described by a feature structure. Quite generally, a rule feature structure can describe a causal regularity in any domain, whether it is a simple or complex domain.

Such a rule feature structure will describe the bare minimum of information about a cause, which is necessary to lead to its effect. However, in the example situations which illustrate the rule, other things are happening at the same time – many of which are not relevant to the rule. In order to learn a rule feature structure from a set of example situations (in which the rule applies, but other things are going on at the same time) it is necessary to 'project out' the common factors in the cause and the effect, while throwing away the irrelevant details of each example - which differ randomly from one example to the next.

How does a learning brain project out the common factors from a number of learning examples, while throwing away irrelevant details which differ across examples?

If each learning example is described by a feature structure, then the operation to project out their common shared part is the operation of generalization. This operation is defined mathematically in Appendix A, and its definition mirrors the definition of unification. The two together form a mathematical structure - an algebra, which underpins the learning theory.

Informally, to make the generalization D of two feature structures A and B (written as D = A ∩ B) you again match the nodes of A and B, trying to maximise the amount of information on the shared nodes – but this time, you throw away any information which is only on A, or only on B, and is not on the shared nodes.

This is the Bayesian specification of the optimal form of learning in complex domains. A feature structure R for a regularity can be learned from a set of learning examples if:

- R is the generalization of feature structures F1, F2, which describe the individual learning examples.
- The evidence for R (from the intersecting probability lines, as described above) is sufficient to overcome its small prior probability, which is of order $2^{-I(R).}$

If animals are capable of this Bayesian optimal learning, they can learn complex regularities (which are feature structures, i.e., partial generative models of the world) from small numbers of learning examples, by generalization of the example feature structures. Once learnt, the regularities can be combined with current sense data, through unification, to form more complete generative models of the current situation. Combining learned rules with current sense data by unification is maximum likelihood scene construction. This is how optimal Bayesian learning is linked to optimal Bayesian inference, in any domain – particularly in complex structured domains, where structured models are needed.

The complementary nature of generalization and unification underpins the consistency of this scheme. As is described in Appendix A, these operations form an algebraic structure, which can be regarded as a formal model of inference and learning in these domains.

4. Selection Pressure for Fast Learning

The Bayesian analysis shows that it is possible in principle to learn and apply complex regularities in any domain (as feature structures), from small numbers of learning examples. This is the Bayesian speed limit for learning. There would be no point in learning any faster (from fewer examples) because faster learning would learn many spurious regularities, from statistical coincidences.

The question arises: is there sufficient selection pressure on animal brains, to bring their learning speeds very close to the Bayesian speed limit? Or could animals get away with learning that is slightly slower, say by a factor two?

If all species were content with slow learning, then perhaps any one species could get away with slow learning. But learning is fundamental to cognition, and is at heart a competitive business.

So, if there are variable learnable regularities in the habitat (for instance, about when certain food sources are available and edible), and if species A can learn those regularities twice as fast as species B – then the individuals from species A will get there first, and consume all the resources, leaving none for species B. Any species which can learn fast - close to the Bayesian speed limit – will out-compete a species of slower learners.

Therefore, we expect that if there is any possible neural implementation of fast Bayesian learning in complex domains, animal brains will have evolved to have that ability.

5. Learning in Animals and People

There is widespread evidence, from associative conditioning and other forms of learning, that animals learn at close to the Bayesian maximum speed – requiring only small numbers of learning examples to learn any regularity.

In associative conditioning, animals are exposed to different stimuli (such as buzzers, lights, or food) in different combinations, and rapidly learn about correlations between the stimuli. Anderson [23, chapter 4] has shown how this rapid learning is in detailed agreement with the Bayesian theory of learning, as outlined in the previous section – agreeing with a wide variety of data from associative conditioning experiments [27].

Animals show impressive abilities to navigate in the wild – for instance, concealing food caches in hundreds of places, and reliably returning to find the caches. This demonstrates that animals can build mental maps of their habitats, from only a few visits to each place. Introspectively, we know that we have the same capacity – to form mental maps linking thousands of places to each other, from only one or a few visits to each place. This speaks to both the speed and the high storage capacity of animal learning.

One of the most impressive demonstrations of human learning ability is every child's ability to learn the syntax and semantics of thousands of words of their native language, in a few years – in which time, they evidently learn each word from observing only a few examples of its use. This ability is so impressive that it has earned an incredulous label: 'the poverty of the stimulus' [28]. However, such rapid learning from only a few examples is entirely consistent with the Bayesian speed limit on learning; and it can be reproduced in computational models of language learning [12,13].

In summary, there is widespread evidence that animals and people are capable of rapid learning, at speeds close to the Bayesian speed limit, in domains where the thing to be learned is a composite data structure – which can be represented as a feature structure.

6. Computational Models of Fast Learning

It is possible to build a computational model of fast Bayesian learning, at Marr's [1] Level Two, in simple or complex domains, along the following lines:

- Every possible learning example is stored by a one-shot memory process. Each learning example is stored as a feature structure.
- There is an associative retrieval process, to retrieve all feature structures matching some pattern (which have that pattern as a sub-structure).
- There is an operation of generalization, which takes two or more feature structures and projects out their shared portion, as a candidate regularity.
- There is an offline learning process, which creates candidate regularities by generalization of learning examples.
- A candidate regularity with information content B bits has a prior probability of the order of 2-B, because there are approximately 2B such candidate regularities; the sum of their probabilities must be of order 1.
- A regularity is learned when the evidence for it passes a test of statistical significance - when the two lines in figure 3 have crossed.
- Once a regularity is learned as a feature structure, it can be applied by unification with current sense data – to find the Bayesian maximum likelihood model of the current situation.

This level 2 computational model can learn from small numbers of learning examples, close to the Bayesian limit. Working implementations of this model have been built in the following domains:

- Associative Conditioning [23]
- Navigation [11]
- Primate social intelligence [26]
- Human language [12]

The computer models of learning in language, social intelligence, and navigation are direct implementations of the general Bayesian learning model outlined above. The model of language learning gives learning at Bayesian speeds, from small numbers of examples

doi:10.20944/preprints202106.0635.v1

of word use, and has been compared with many other empirical facts of first language acquisition [12].

Associative conditioning could be implemented in a simple feature structure model of 'dinodes' (a dinode is a feature structure with only two nodes – a 'cause' node and an 'effect' node). It is a simple special case of sequence learning, with only two steps in the sequence (two nodes).

While these have been small-scale implementations (working with small numbers of landmarks, words, and so on), they have good computational scaling properties, and could easily handle larger-scale, biologically realistic problems. They require a memory capacity which scales only linearly with the number of pieces of knowledge that need to be retained or learned. This is a model of learning which learns fast, which handles complex domains, and scales well to realistic problems:

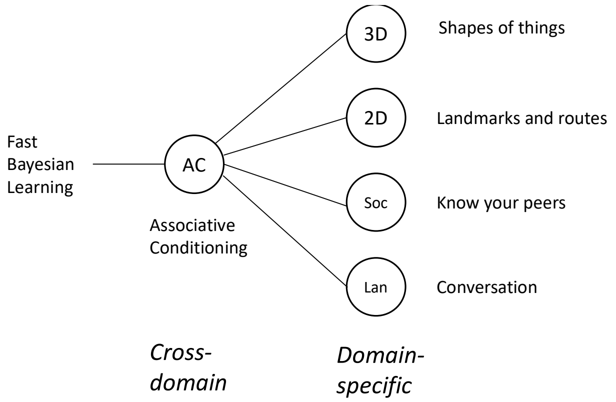

Figure 4: How the general Bayesian computational model of learning extends from associative conditioning to any complex domain, by elaboration of the feature structures.

The same learning examples can contribute to the learning of broad general regularities, and of narrower specific regularities. This is illustrated in the language learning case, where the same learning procedure can learn both broad syntactic regularities, and the irregular grammatical constructs which occur in every language.

Similarly, the example of navigational learning illustrates that there is a continuum all the way from one-shot episodic memory, through place-specific learning, to place-independent learning. An animal can acquire knowledge across a spectrum from the specific to the general:

a) The last time I was here, AB happened.
b) When I am here, A causes B
c) Whenever I am at a place like this, A causes B
d) At any place, A causes B

This continuum shows that one-shot memory is a necessary feature of learning. All these regularities can be represented as feature structures. (a) is episodic memory; (b), (c) and (d) require progressively greater levels of generalization, across larger sets of examples.

This continuum suggests that in any domain, learning is a fractal-like, incremental process – with rules at different levels of generality being learnt at different stages, and with knowledge of some rules being a pre-requisite for learning others. This is borne out in the case of first language acquisition (where some nouns must be known before verb syntax can be learnt [12]), and where the full learning algorithm requires unification as well as generalization of feature structures. Unification reveals hidden nodes of the generative model, which are sometimes needed to learn new regularities.

Good scaling to realistic learning problems can happen only when the pattern matching of composite data structures is invariant with respect to domain-specific constraints (such as two-dimensional geometry, in the case of navigation; or spatial invariance, for learning of object shapes). If the patterns to be learnt are not spatially invariant, there will be too many candidate patterns, and the prior probability of each one will be too small for any of them to be learnable (the intercept of the 'rule' line in figure 3 will be too low).

The need for invariant pattern-matching (with the type of invariance depending on the domain) is one example of the need for well-tuned Bayesian priors about the types of regularity which can be learned, or which are worth learning. It is essential for any species, in order to learn rapidly the regularities of its habitat, to have Bayesian priors for 'types of rule worth learning' which are well-tuned to the habitat. Natural selection tunes these priors. Good tuning of priors not only reduces the search space for complex rules; it also elevates the zero intercept of the 'rule' line in figure 3, reducing the number of examples needed for the blue 'rule' line to overtake the red 'no rule' line – giving faster learning.

7. Neural Implementation of Fast learning

The previous section described a working level-2 model of fast learning, in simple or complex domains. It was not a neural implementation of learning. I propose a hypothesis:

Learning hypothesis*: Fast animal learning, in simple and complex domains, is achieved by neural implementations of the Level 2 computational model of learning by generalisation.*

This hypothesis is in contradiction to the learning hypothesis implicit in neural nets. It implies that the level 3 implementation of fast learning comes about by neural implementations of the capabilities required by the level 2 computational model, which are:

- One-shot memory for learning examples as feature structures
- Storage of large numbers of learning examples as feature structures
- Fast pattern-matching retrieval of learning examples
- Domain-specific invariance in pattern matching (e.g., invariance under spatial translations)
- Generalisation of feature structures, to project out candidate regularities from examples

These capabilities evidently differ from the capabilities built into deep learning neural net architectures [2].

This is not a highly restrictive hypothesis, since each one of the capabilities above might have several different neural implementations. Rather, as described below in section 9, this hypothesis could give a common vocabulary, in which existing neural models can be compared and contrasted.

In general, one might expect the neural implementations of this model to be diverse and domain-specific. For instance, a neural model of navigational learning should be expected to be compatible with the overall architecture of place cells [19] and grid cells [20], which are involved in navigation.

It may be possible to show mathematically that some of the capabilities are necessary for fast Bayesian learning – that at the neural level 3, animal brains cannot do without them. This will be the subject of a future paper.

8. Dynamic Binding and Metadata

The previous section described a working level-2 model of fast learning, in simple or complex domains

The central requirement for a neural implementation of the fast-learning algorithm is the storage, processing, and retrieval of feature structures. Neural representation of feature structures is a primary requirement.

A simple feature structure can represent a composite situation of a red triangle and a blue square – by having a 'red triangle' node and a 'blue square' node, possibly with some spatial relation encoded in the edge between the nodes. Typical neural models

doi:10.20944/preprints202106.0635.v1

represent single properties (such as colour) by the firing rate of a single neural fibre or set of fibres.

If there are four sets of input fibres for the four pieces of information (red, blue, triangle, square), how does some part of the brain (say, a cortical column receiving them) know that 'red' goes with 'triangle' and 'blue' goes with 'square'? This requires binding information [29] – information about what data is bound together with what other data. Binding information is essential if operations such as unification are to be done correctly.

One can imagine solutions to the problem of feature structure binding, which are hard-wired in neural connections – say, that the 'red' fibres are physically close to the 'triangle' fibres, or share a set of synapses with them. However, it is soon evident that this kind of static neural binding has problems. Mainly, it lacks the flexibility to represent different feature structure graphs. If some part of the brain is hard-wired to represent a certain type of feature structure graph (say, with two nodes, each holding a colour and a shape) then that is all it can do. Such a lack of flexibility would be a massive impediment to feature structure learning (the ability to learn novel feature structure graphs) and inference.

Static neural binding will not suffice for complex feature structures. Dynamic binding is needed.

Binding information is data about data – which in computer science is called metadata. The use of metadata is universal in modern computing practice, for good reasons. Metadata underpins the huge flexibility of modern computing.

I propose the following hypothesis:

Metadata hypothesis: *feature structures are represented in the brain using dynamic binding, with data and metadata multiplexed in the same neural fibres.*

I illustrate this hypothesis with the simple example above, where a cortical column receives four sets of ascending fibres – representing two pieces of data on each of two nodes of a feature structure (called node 1 and node 2). The data are represented by tonic firing rates on these fibres. The column does not know which data belongs to which node. It therefore sends a message on descending fibres back to the cortical column which is the source of the ascending fibres. The message says: 'send me your metadata'.

The source column then switches from tonic mode to burst mode firing. It first sends a burst along the two sets of fibres representing node 1. Shortly afterwards, it sends another burst along the two sets of fibres representing node 2 of the feature structure. The receiving cortical column then knows which data are bound together on each node of the feature structure. Operations like unification or generalization can then be done correctly, without confusing what information is on what node.

This only the simplest possible illustration of metadata multiplexing. It is clear that in the brain (as in modern computing) metadata can convey much more:

- A feature structure could have any number of nodes, defined by metadata.
- Metadata can define the edges between the nodes – defining any graph structure.
- Metadata can define the type (i.e., the modality, or meaning) of data on each node or each edge of a feature structure – defining the semantics of the graph.
- Metadata can define the processing to be done on a feature structure – such as unification with some other feature structure, or generalization.

If metadata is to serve all these purposes, it is clear that some more or less elaborate metadata encoding is needed in the brain – possibly more complex than just the timing of bursts. This is not difficult to envisage. If the need for metadata in the brain is universal, there has been a long time to evolve metadata encodings, and they could be universal across different parts of the brain.

We need not be too concerned about the neural bandwidth consumed by metadata – since metadata needs to be sent less frequently than data, to occasionally 'switch' the processing of some cortical column; this need not consume much neural bandwidth.

Multiplexing could be by time division or by frequency division, or by some multi-fibre encoding. Use of the same fibres for data and metadata has the benefit of closely associating the data and metadata, defining which data the metadata are about.

How might a cortical column be sensitive to metadata? There could, for instance, be 'metadata-only' neurons, which are activated only by a metadata burst (i.e., a characteristic fast burst signature, followed by an encoding of the metadata), which then selectively act on nearby 'data-only' neurons – activating and deactivating their dendritic trees [38], or deactivating whole neurons. Metadata neurons could sustain the same firing patterns until the next metadata burst. Groups of linked metadata neurons could form metadata processors within cortical columns. A group of metadata neurons might activate only after receiving a specific metadata pattern.

This allows any cortical column to be reconfigured by metadata – for instance, selectively turning on 10% of its data neurons, or selecting some of their dendritic trees. Different non-overlapping subsets of 10% of the neurons could do 10 different jobs. Or subsets of its neurons activated in different combinations could do more than 10 different jobs.

Metadata can act like a program for a cortical column: 'keep running this program until you receive the next program'.

If there was any way to make neurons or cortical columns rapidly reconfigure to do different tasks (such as different forms of unification, or handling different graphs of feature structures), it would be surprising if brains had not evolved to use metadata configurability. Only if dynamic reconfiguration were in principle impossible would we expect neurons to act as static, non-reconfigurable processing elements. Nobody has shown that it is impossible – and as the sketch above illustrates, there are probably many ways to implement configuration of neurons by metadata. Selective switching of neurons has been discussed, while not labelling it as 'metadata' [30].

This could lead to a picture of each cortical column, or group of columns, as a configurable metadata-driven engine for processing feature structures – doing the universal operations of unification and generalization, as required for learning and inference in all domains, in a variety of domain-configurable ways. This could be the basis of the strong resemblances in the architecture of columns across the neo-cortex [31,8], and with the plasticity of cortical columns [32].

The principle of metadata-driven computing has achieved very high levels of economy and reuse of hardware and software in modern computing. In the same way, it could lead to economies in brain design – with neural connectivity not needing to be genetically specified in complete detail, because more generic (loosely specified) connectivity can be dynamically configured and selected by metadata. Economy of genetic encoding of brains could give simpler neural morphogenesis, and faster evolution to meet new cognitive challenges.

9. Relation to Existing Work

There are a few groups currently building working neural computational models of cognition, in both simple and complex domains. Usually, these groups express their results in their own specific language, and there have been few published comparisons between the different formulations on offer. Since the concepts described in this paper – such as feature structures, unification and metadata multiplexing – have cross-domain applicability, and relate to the key issue of fast learning, it may be possible to use these constructs to furnish a shared vocabulary in which different formulations can be compared. This section poses a few questions as a first step in that direction.

The strands of work which I shall discuss are:

- The Free Energy Principle and Active Inference [3, 4, 33, 34]
- The Thousand Brains theory [7,8]
- Capsule networks and the GLOM model [9,10]

doi:10.20944/preprints202106.0635.v1

Lack of space prevents the discussion of other models, such as [35]

I shall not attempt to do justice to those streams of work in their own terms; I aim only to relate them to the themes of this paper. If important points are lost in translation, I apologise to those authors.

The Free Energy Principle

The approach of this paper shares many common foundations with the Free Energy Principle, since both are based on a Bayesian analysis of cognition, and both rely on the construction of a generative model of the current situation with the greatest marginal likelihood or Bayesian model evidence. From this basis of strong commonality, I briefly touch on some areas of divergence or different emphasis:

1. In complex domains, FEP generative models are composite models – defined, like feature structures, by many discrete and continuous variables. Applications of the FEP have not used the vocabulary of feature structures, unification and generalization to describe these models – but they could do so. This is especially so with the advent of generative models based upon Markov decision processes with discrete states that are related through (vertical) likelihood mappings and (horizontal) prior transition matrices. Maximum (marginal) likelihood unification of composite generative models occurs in FEP models, as the basis of scene construction, but it is usually done by generic FEP hill-climbing procedures, and is not cast as unification.
2. In some implementations of FEP, the node graph of a feature structure (aka a composite generative model) may be realized by a given neural connectivity. If it is so tied, this raises questions about the ability of such generative models to represent open-ended feature structure – which would be possible using feature structures implemented using metadata. Do current FEP generative models assume a fixed neural connectivity structure, or can they take on new, unanticipated connectivity and semantics? See [36] for a related discussion.
3. FEP models of learning differ from the level 2 learning model described in this paper, and are not neural implementations of that model. This raises the questions: how close do FEP models of learning come to fast Bayesian learning? If they do not come very close, can we identify features of the level 2 learning models, which FEP neural models could implement for faster learning? For instance, do FEP models have one-shot retention of learning examples? Do they have an equivalent of feature structure generalization? Do they carry out statistical significance tests for a rule (possibly in a way that has not been made explicit)? See [37] for a related treatment.
4. The FEP procedure for inverting a complex generative model is a generic cross-domain hill-climbing method (minimising a K-L divergence, by neural message passing). In the feature structure models referred to in this paper, unification exploits domain-specific constraints and invariances – to reduce the dimensionality of the search space, making the search for the most likely model potentially more efficient. Within the FEP framework, are there more efficient, domain-specific procedures for finding maximum (marginal) likelihood models?

Questions like these could be fruitful future areas of investigation for the FEP.

The Thousand Brains Theory

Hawkins' [7,8] Thousand Brains model is not overtly grounded in Bayesian principles, but is consistent with them. It envisages the construction of composite internal models of the animal's environment, which are like Bayesian generative models. This is done by the collaboration of many cortical columns (the thousand brains), each of which holds a model of some part of the world, in its own 'reference frame'.

In spatial domains, a reference frame is like a fragment of a map – similar to the map fragments in the model of navigation discussed in this paper. Reference frames apply in all domains, including abstract domains and domains such as sequence processing [38]. How do they relate to the ideas of this paper?

If we identify a reference frame with a multi-node feature structure (or with a set of similar feature structures), there are links between the thousand brains model and the concepts of this paper. Some of the issues and questions raised by this link:

1. The identification of reference frames with feature structures, and the common operations of unification and generalization, could underpin the shared common architecture of cortical columns.
2. In metric domains, the representation of feature structures might involve grid-like cells with firing grids regularly spaced in one, two, or three dimensions; for abstract domains such as social intelligence or language, different representations might be used.
3. If a cortical column is driven by metadata, being configured by metadata to handle different types of feature structure in different domains, this allows a highly shared neutral architecture of configurable cortical columns working across different domains
4. Sensitivity to different metadata patterns could involve sparse pattern matching in dendritic trees, as in [38]. Metadata switching of neural circuits could underlie the low average firing rates in columns – with most circuits switched off at any one time.
5. The thousand brains model envisages cortical columns collaborating to build a larger generative model, by a process of voting. The perspective of feature structures and unification shows that this cannot be simple additive voting, but is a complex pattern match between partial models – a unification, like fitting fragments of a map together. There are important scaling questions about what level of connectivity between cortical columns is required to support this, and whether cortico-cortical links are sufficient. One way to address questions of connectivity scaling is for the thalamus to act as a central aggregator of hypotheses from cortical columns [39].
6. The thousand brains model emphases learning as a continual process, overlapping with episodic memory, as in this paper. The many cortical columns form a good basis for the scaling of this learning/memory to retain thousands of fragments of information.
7. In the thousand brains model, cortical columns learn, but not by neural implementation of the level-2 learning models of this paper. Can they learn regularities at speeds close to the Bayesian limit? If not, are there features of the level 2 learning model which can be implemented in cortical columns, to give faster learning?

The perspectives of this paper appear to raise interesting questions for the thousand brains model.

Capsule Nets and the GLOM model

Capsule nets [9] aim to overcome some of the limitations of neural nets, within an architectural framework which resembles neural nets.

Capsule nets are usually introduced in the context of learning, for familiar learnable shapes such as faces or cars. I suggest that capsules may be appropriate for the more primitive (and evolutionally prior) problem of inferring unique irregular 3-D shapes, such as rocks or vegetation, from motion – a problem which every animal must solve in order to move. The 3-D integration of 'pose' between wholes and parts, as performed by capsules, seems well suited to tackle this problem, which may be an essential prelude to learning 3-D shapes.

In his subsequent GLOM model [10], Hinton aims to enable neural networks to dynamically encode diverse whole-part hierarchies, as required for visual scene

doi:10.20944/preprints202106.0635.v1

construction. He says that for a neural network to represent a whole-part hierarchy *'…is difficult because a real neural network cannot dynamically allocate a group of neurons to represent a node in a parse tree … [because] what neurons do is determined by their incoming and outgoing weights and real neurons cannot completely change these weights rapidly*.'

Visual object hierarchies are an instance of the general problem of dynamically representing composite feature structures, discussed in this paper. The solution Hinton proposes is a large-scale neural-net-like architecture, with groups of neurons (capsules or columns) passing activity vectors between them. Many capsules or columns are used to represent a hierarchical feature structure – with the higher nodes represented by 'islands' of many columns sharing identical activity vectors.

This paper has suggested that while neurons cannot change their input and output weights rapidly, nevertheless groups of neurons (such as cortical columns) can change their functionality rapidly, being rapidly re-configured by metadata. This might be done as described in section 8, on dynamic binding and metadata – for instance, using 'metadata neurons' to switch the functionality of other 'data' neurons, in a way such as that described in [30]. Metadata would allow dynamic, novel feature structures to be represented within individual cortical columns – giving a more local micro-level solution to the problems raised by Hinton. Such a micro-level solution might be more economical.

Part of the motivation for the GLOM model is that, unlike modern computers, the brain cannot do dynamic allocation of storage – because synapse strengths change slowly. The problem of dynamic storage allocation can be divided into two parts: (a) dynamic repurposing of local memory units, such as cortical columns; and (b) dynamic changes in connectivity between memory units. The former can be done by metadata-driven switching. The latter problem raises harder issues of scaling. I suggest that the solution to (b) lies not in the cortex, but in the thalamus, which acts as a blackboard [39,40], dynamically switching data between cortical regions, depending on the physical location of the source.

The activity vectors of the GLOM model (vectors of firing rates of many neurons) clearly have the capability to represent composite structures, such as 3-D spatial hierarchies or parts of them. The question arises: how do activity vectors represent spatial hierarchies, and how do any two capsules agree with one another on how the vectors represent structure? There are broadly two classes of solution: hard-wired connections between the capsules, or connections whose meaning is configured by metadata. The latter would seem to have benefits of flexibility, to represent any structure. They also require less precise neural connectivity between modules, as 'metadata can sort out the problems'. This is like buying a computer, knowing that you will be able to program it.

I know of no proof that metadata-driven configurability in the brain is not possible – and as in section 8, there appear to be several ways to do it. So, it would seem remarkable if the brain did not use metadata-driven configuration. Adapting the GLOM model to use metadata is clearly feasible.

10. Conclusions

To recapitulate the main results of this paper:

1. Nearly all learning is learning about composite aspects of the world – aspects which have parts, and relationships between the parts.
2. These can be represented by feature structures – graphs of nodes, with information on the nodes and the edges.
3. The combination of feature structures and Bayesian inference is a powerful computing paradigm, and is applicable in any domain, simple or complex.
4. In this paradigm, feature structures represent composite generative models of the world.
5. The core operation for scene construction (for building larger, maximum likelihood generative models from partial models) is unification of feature structures.

6. Feature structures can express individual states of affairs (examples), or the regularities which hold across them, in any domain.
7. There is a Bayesian speed limit for learning, which defines the minimum possible number of examples needed to learn any regularity.
8. Regularities are inferred from examples by the generalization of feature structures.
9. Unification and generalization are complementary operations, and form an algebraic structure.
10. This gives a computational model of learning at Marr's Level Two, which can learn from small numbers of examples, at speeds close to the Bayesian limit – unlike neural nets, which require large numbers of learning examples.
11. *Hypothesis*: animals learn by neural implementations of the fast-learning model.
12. This requires neural implementations of feature structures, and of the operations on them.
13. *Hypothesis*: For a flexible dynamic neural implementation of feature structures, neuronal firing rates represent both data and metadata – particularly, metadata about the graph structure and meaning of a feature structure.
14. *Hypothesis*: Cortical columns are universal metadata-driven feature structure engines.

In summary, there are two evolutionary imperatives on brains – Bayesian inference, and fast learning. Taking those imperatives seriously in complex domains, we are led to the need for composite feature structures [15-18] (which have gone under many other names, such as scripts [41] generative models [6] or Non-Parametric Hierarchical Bayesian Models [42]), and a few universal operations on them, such as unification. Dynamic, configurable feature structures may require metadata-driven neural processing.

This approach to the neural modelling of learning - with more emphasis on mathematical and computational analysis, and less dependence on experimentation and tuning - challenges the approach of convolutional deep neural nets.

Funding: This research received no external funding.

Acknowledgments: I thank Karl Friston, Adam Safron and Noor Sajid for helpful comments on drafts of this paper.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A: Feature Structures, Unification and Generalisation

This appendix is a brief summary of the mathematical basis of feature structures and their relation to Bayesian optimal cognition.

A feature structure is an information structure which has several nodes, connected in a graph (typically a directed acyclic graph, or DAG) by links, or edges.

- Each node may hold several pieces of information (sometimes called slots), of different modalities, depending on the domain. A typical slot is 'colour = red', or 'size = large', etc.
- The edges carry information, which may be domain-specific – for instance 'time delay = 5 seconds ±3 seconds', or 'displacement = 5 cm left' or 'relationship = sibling' or 'node is parent of node'

The information held in nodes and edges has levels of uncertainty. Depending on the information in nodes and edges, and its uncertainty, each feature structure has an information content, typically measured as B bits.

Feature structures have a model-theoretic semantics, in that each feature structure represents a set of possible situations in the world. If a feature structure has information content B, then the set of situations it represents has probability approximately 2^{-B}. This is the basis of the relationship between feature structures and Bayesian models of cognition.

doi:10.20944/preprints202106.0635.v1

Nodes in feature structures may contain information which denotes variables, or unknown values. For instance, two nodes N1 and N2 on the same feature structure may both hold the information 'colour = ?A', where ?A denotes a variable colour. The meaning of this is 'the colour on node N1 is not known, and the colour on node N2 is unknown; but it must be the same colour on N1 and N2'.

A primary relation between feature structures is the relation of subsumption:

A feature structure A subsumes (<) another feature structure B (A < B) , if and only if all the information that is contained in A is also contained in. B.

Informally, all the nodes, slots, and edges in A are also in B; and B may have extra nodes, slots and edges. Any information in A must also be in B, so that B has equal or higher information content than A.

In terms of the model-theoretic semantics of feature structures, any situation in the world which is described by B is also described by A; but not necessarily the reverse. The set of situations described by B is a subset of the set described by A, and has smaller probability than the set described by A.

Subsumption can be used to define the important operations on feature structures, of unification and generalisation.

The unification C of two feature structures A and B (written as C = A U B) is the feature structure with smallest possible information content which satisfies both A < C and B < C.

The information content of C satisfies:

1. $I(C) \geq I(A)$
2. $I(C) \geq I(B)$
3. $I(C) \leq I(A) + I(B)$

There is an algorithm to compute the unification C from A and B, with domain-specific variations. You match pairs of nodes from A and B, trying to get the maximum match of information on paired nodes, while respecting the constraints of the edges. In the result C, you retain all the shared nodes and all the unmatched nodes which come from either A or B, allowing no contradictions. Hence the result C (if it exists) contains all the information in A, and all the information in B. The best match of nodes maximises the amount of information in C which is not duplicated, coming from both A and B.

Unification may involve both discrete optimisation (choosing matching nodes) and optimisation of continuous variables (such as distances)

The model-theoretic interpretation of unification is as follows: Any situation described by C is also described by A, and is described by B. The set of situations described by C has the highest probability (lowest information content) of situations described by both A and B.

Thus C describes the maximum likelihood set of situations consistent with both A and B. Unification is the operation of maximum likelihood inference. Put another way, unification is an operation of scene construction, which constructs the most likely scene, or most likely model of the world, given A and B. Unification is the core operation of Bayesian scene construction, in any domain.

We can use this to describe how animals apply knowledge, once it has been learned. Suppose an animal has learned some cause-effect regularity, of the form (Cause => Effect). This can be paraphrased as 'if the current situation matches the cause, then the effect is likely to follow'. Both the cause and the effect can be expressed as feature structures, and can be combined as a single rule feature structure R – with a left-hand 'cause' branch and a right hand 'effect' branch.

Then the rule is applied by trying to unify the rule R with the current situation S. If the 'cause' branch of R matches S, then the effect branch of R predicts what will happen – and because of the Bayesian model semantics of feature structures, it is the maximum likelihood prediction of what will happen – which will give the animal greatest fitness. A rule R will not unify with the current situation S if there is a contradiction between them.

Unifying the current situation S with learned rules R gives a maximum-likelihood (and so maximum fitness), model of the world.

Subsumption is also used to define the operation of generalization, which is of central importance for learning new rules R.

The generalisation D of two feature structures A and B (written as $D = A \cap B$) is the feature structure with largest possible information content which satisfies both $D < A$ and $D < B$.

The information content of D satisfies:

1. $I(D) \leq I(A)$
2. $I(D) \leq I(B)$

There is a complementary relationship between the unification $C = A \cup B$ and the generalisation $D = A \cap B$. Because of this complementarity, there is a relation between information contents:

$$I(C) + I(D) = I(A) + I(B)$$

There is an algorithm to compute D from A and B, which is similar to that used to compute the unification C. To compute $D = A \cap B$, you again match pairs of nodes from A and B, trying to get the maximum match of information on paired nodes, while respecting the constraints of the nodes and edges. For generalisation, you retain only the matched nodes, slots and edges – and throw away any parts of A and B which are not matched. The best match of nodes maximises the amount of information in D.

The operations of unification and generalisation fit together in an algebraic structure, or feature structure algebra. Typical relations of this algebra are:

$$A \cap B = B \cap A$$
$$A \cup (A \cap B) = A$$
$$A \cup (B \cup C) = (A \cup B) \cup C$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Some of these relations are not exact, but can be applied in practice. They underpin the self-consistency of feature structure operations. The relations resemble the relations of set theory, because feature structures denote sets of situations in the world.

The application of generalisation to learning is described in section 3.

References

1. Marr, D. (1982) Vision, W.H.Freeman
2. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature 521, 436–444. doi: 10.1038/nature14539
3. Friston K. (2003) Learning and Inference in the Brain, Neural Networks 16, 1325 – 1352
4. Friston K. (2010) The free-energy principle: a unified brain theory? Nature Reviews Neuroscience
5. Friston K., Kilner, J. & Harrison, L. (2006) A free energy principle for the brain. J. Physiol. Paris 100, 70–87
6. Smith R, Friston K. and Whyte C J (2021) A Step-by-Step Tutorial on Active Inference and its Application to Empirical Data
7. Hawkins, J., Lewis, M., Klukas, M., Purdy, S., and Ahmad, S. (2019). A framework for intelligence and cortical function based on grid cells in the neocortex. Front. Neural Circuits 12:121. doi: 10.3389/fncir.2018.00121
8. Hawkins J (2021) A Thousand Brains, Basic Books, New York
9. Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems, volume 30, pages 3856–3866. Curran Associates, Inc. 2, 7
10. Hinton G, (2021) How to represent part-whole hierarchies in a neural network, arXiv 2102.12627
11. Worden, R.P. (1992) Navigation by fragment fitting: a theory of hippocampal function, Hippocampus 2, 165
12. Worden, R. P. (1997) A Theory of language learning, unpublished paper on ResearchGate, to be made available on arXiv
13. Worden R. P. (2002) Linguistic Structure and the Evolution of Words, in 'Linguistic Evolution Through Language Acquisition', Briscoe E (ed.) Cambridge
14. Clocksin, W. F. and Mellish C. S. (1979) Programming in Prolog
15. Kaplan, R. M. and J. Bresnan (1981) Lexical Functional Grammar: a Formal System for Grammatical Representation

16. Gazdar, G., E. Klein. G. K. Pullum and I. A. Sag (1985), Generalised Phrase Structure Grammar, Blackwell, Oxford
17. Pollard, C. and I. Sag (1987) Head-driven Phrase Structure Grammar, University of Chicago Press
18. Shieber, S. (1986) An introduction to unification-based approaches to grammar, CSLI, Stanford, CA.
19. O'Keefe, J. (1979) A Review of the Hippocampal Place Cells. Prog. Neurobiol. 13:419-439
20. Moser, M.B., Rowland, D.C., Moser, E.I., 2015. Place cells, grid cells, and memory. Cold Spring Harbor perspectives in biology 7, a021808.
21. Chen, Z., Gomperts, S.N., Yamamoto, J., Wilson, M.A., (2014). Neural Representation of Spatial Topology in the Rodent Hippocampus. Neural Computation 26, 1-39.
22. Stachenfeld K, Botvinick M, and Gershman S (2017) The Hippocampus as a predictive map, Nature Neuroscience vol 20, no. 11
23. Anderson, J.R (1990) The Adaptive character of thought, Lawrence Erlbaum Associates
24. Worden R. P. (1995) An optimal yardstick for cognition, Psycoloquy, Vol 7. Expanded version on ResearchGate
25. Mackay D. and Peto L (1995) A Hierarchical Dirichlet Language Model, Natural Language Engineering 1 (3) 289
26. Worden, R.P. (1996) Primate Social Intelligence, Cognitive Science 20, 579 - 616.
27. Dickinson, A. (1980) Contemporary animal learning theory, Cambridge University Press
28. Chomsky, C. (1969) Acquisition of syntax in children from 5 to 10, MIT Press, Cambridge, Mass.
29. Treisman, A. and Gelade, G. (1980), A feature integration theory of attention, Cognitive Psychology, 12 (1): 97–136
30. Hinton, G. E. (1981). A parallel computation that assigns canonical object-based frames of reference. In The 7th International Joint Conference on Artificial Intelligence, Volume 2, page 683–685. Morgan Kaufmann
31. Mountcastle, V. (1978). “An organizing principle for cerebral function: the unit model and the distributed system,” in The Mindful Brain, eds G. Edelman and V. Mountcastle (Cambridge, MA: MIT Press), 7–50.
32. Bennett EL, Diamond MC, Krech D, Rosenzweig MR (1964). "Chemical and Anatomical Plasticity of the Brain". Science. 146 (3644): 610–619
33. Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2016). Active inference and learning. Neurosci. Biobehav. Rev. 68, 862–879. doi: 10.1016/j.neubiorev.2016.06.022
34. Parr, T., and Friston, K. J. (2018). The anatomy of inference: generative models and brain structure. Front. Comput. Neurosci. 12:90. doi: 10.3389/fncom.2018.00090
35. O’Reilly R, Wyatte D and Rohrlich J (2017) Deep Predictive Learning: A Comprehensive Model of Three Visual Streams, arXiv: 1709.04654v1
36. Smith, R., P. Schwartenbeck, T. Parr and K. J. Friston (2019). "An active inference approach to modeling structure learning: concept learning as an example case." bioRxiv: 633677.
37. Friston, K. J., M. Lin, C. D. Frith, G. Pezzulo, J. A. Hobson and S. Ondobaka (2017). "Active Inference, Curiosity and Insight." Neural Comput 29(10): 2633-2683.
38. Hawkins J and Ahmad S (2016) Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex, Front. Neural Circuits, 30 March 2016 https://doi.org/10.3389/fncir.2016.00023
39. Worden, R. P. (2020). An Aggregator model of spatial cognition. arXiv [Preprint]. Available online at: https://arxiv.org/abs/2011.05853.
40. Worden R, Bennett M. and Neascu V (2021) The Thalamus as a Blackboard for Perception and Planning, Front. Behav. Neurosci., 01 March 2021 | https://doi.org/10.3389/fnbeh.2021.633872
41. Schank, R.C. and R.P.Abelson (1977) Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures, Lawrence Erlbaum Associates, Hillside, New Jersey
42. Gowanlock D, Tervo R, Tenenbaum J and Gershman S (2016) Toward the neural implementation of structure learning, Current Opinion in Neurobiology 37:99–105