been implemented as a formal computational model. Therefore, instead of pointing out its limitations, we focus on mechanisms that need to be elaborated before this account can be expanded into a precise mathematical model that can be tested.

3.2 Learning in computational models

Scientific investigation begins with the observation of a physical system. The systematic recording of observations provides a list of behaviours that form the raw data. The goal of scientific enquiry is to understand the underlying reasons for the production of this data. The investigation expresses these reasons as a theory or a model that explains the data in terms of a set of processes operating over a group of representations. The next stage in research is to match the behaviour of the model with that of the physical system. The behaviour of the model is governed by its input and a set of free parameters. The investigation adjusts these free parameters so that the model closely matches the observed behaviour of the physical system. This adjustment of the free parameters as the model comes into contact with its environment is termed as *learning* (Mendel & McLaren, 1970).

Another way to look at a model is as a transformation of an input stimulus to an output or a response. The transformation describes the behaviour of the model and depends on a set of free parameters. As the model comes in contact with its environment, it generates a response but it also changes its parameters. The model not only transforms the input, but also gets transformed by the input. Various learning algorithms describe how computational models can both process information and perform learning at the same time. In this section we consider two learning paradigms that achieve this goal. In the rest of the chapter we look at specific learning algorithms and models developed to explain structural priming.

§ 3.2.1 Teachers and pupils

Let us consider a computational model that tries to approximate the stimulus-response characteristics of a physical system. As the model comes into contact with the environment, it would perform transformations on the input signals and generate an output. This output might or might not match the output that the physical system would have produced. If we want the model to learn from this episode of information processing, it will be useful to give the model information about the correctness of the output signal.

In other words, the model needs a *teacher*, that possesses knowledge of the physical system and can ascertain the correctness of the response for each given input. The learning paradigm that assumes the existence of such a teacher is called *supervised learning*.

During supervised learning, the model is trained on a pairs of inputs and target outputs. The training simulates the model with each input on the list and generates an output. A teacher compares this output with the target and generates an error. The model then uses this error to adjust its free parameters such that the error is reduced if the model was simulated again.

Of course, if we want to use this learning paradigm we have to assume that a teacher will be available that possesses enough knowledge about the environment and we can use this teacher to predict the target output for every given input. But this assumption might not always be true. Consider a system that models language comprehension. The target output of such a system in not entirely clear. Depending upon what stage of language comprehension we are modelling, the output might be a concept, a proposition or a thought. Because these are abstract cognitive constructs, it is difficult to know the target output of the model for any of these stages of comprehension. In the absence of this knowledge, we cannot use supervised learning for modelling language comprehension. As we shall see below, Chang et al. (2006) used a trick to overcome this difficulty. Instead of using any of these abstract cognitive constructs as output, their model used the input utterance itself as the output and a delayed version of the utterance as the input.

The alternative to supervised learning is learning without a teacher. This learning paradigm is, rather unimaginatively, called *unsupervised learning*. The goal of the model remains the same: to approximate the behaviour of a physical system. Because we do not have a teacher that can give us the target response for each input, we must use a heuristic for adjusting the free parameters of the model. The system may have an internally derived training signal based on the system's ability to to predict its own input, or it may be some more general measure of the quality of its internal representation (Becker, 1995). One of the earliest and most popular unsupervised learning algorithms is Hebbian learning, which postulates that the synaptic efficacy between pre- and post-synaptic neurons should increase whenever they get co-activated. Instead of comparing the response of the system to a target output, this algorithm learns patterns in the input stimuli.

Both supervised and unsupervised learning paradigms assume that the goal of the model is to approximate the input-output behaviour of a physical system. However, learning might not be targeted at achieving a particular transformation from input to output, but only at changing a system in a specific way. A finite-state transducer (FST) is a good example of this form of learning. When an FST receives an input, this input (possibly) changes the state of the FST. We can say that the FST has undergone learning. But the FST does not try to approximate an input-output behaviour as a result of this learning. The output depends not only on the input, but also on the state that the FST was in, when it received the input. Declarative memory is another example. An episode of stimulation might leave a trace in declarative memory but this trace need not approximate any input-output transformation. Rather, the trace simply acts as a record of the episode.

Well known learning algorithms such as error-correction learning and self-organised learning try to optimize a global objective function. As training proceeds, the model iteratively changes its parameters so that the sequence of changes converge to an optimal value of this objective function. Learning is governed by a global variable and changes local values. We call this form of learning the *top-down* approach to learning. In contrast, we saw the example of declarative memory where the synaptic adjustment may or may not lead to a global objective function (Becker, 1995). This form of learning can be called *bottom-up* approach to learning. One of the reasons why Hebbian learning is so popular is that it combines bottom-up synaptic learning with achieving a top-down globally optimised objective function. Later in this chapter, we will present the trailing-activation account of structural priming which falls into the bottom-up approach to learning. Although we will be using unsupervised learning algorithms to implement the trailing-activation account, the reader should remember that our aim will be to record traces of information rather than internalising an input-output transformation.

3.3 Error-based learning model

It is uncontroversial that priming is a consequence of some kind of learning. The question is which part of the cognitive system undergoes learning and what is the mechanism of such learning. In Section 3.2.1 we saw that learning could be supervised or unsupervised. In this section, we consider a previous model that uses supervised learning and tries to explain priming as a consequence of such this mechanism.

§ 3.3.1 A brief summary of Chang et al. (2006)

Chang et al. (2006) presented a language-comprehension and production model that relies on error-correction to learn the sequential structure of utterances. When this model is given an input utterance, it breaks down the utterance into a sequence of words. As each word is processed by the system, it tries to predict the next word. In order to make such a prediction, the system maintains an internal model of the structure of sentences. If the prediction is incorrect, the system generates an error. This error is backpropagated through the system so that the elements of internal model that led to the incorrect prediction are penalised. Thus, for each utterance, the system makes a sequence of predictions and adjusts its internal model based on these predictions. Chang et al. (2006) showed that a system based on these principles is able to successfully extract the abstract structure of utterances so that it can produce grammatical utterances if it is given the meaning of these utterances as its input. Figure 3.3.1 shows the flow of information through this model.

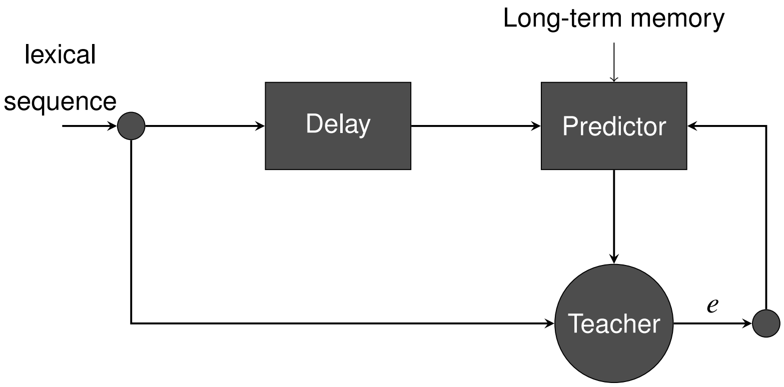

Figure 3.3.1: [Supervised comprehension] To perform comprehension using supervised-learning, Chang et al. (2006) used the lexical sequence both as the input and as the target. The predictor forecasts the next word using the given sequence of words. The teacher, then, compares this prediction with the input signal and generates an error, which is used to adjust the rules of prediction.

Internally, the model presented in Chang et al. (2006) (hereafter CDB06) consists of two parallel pathways for the flow of information. While the *meaning pathway* represents aspects of semantics that are critical to sequencing of utterances, the *sequencing pathway* consists of a sequential recurrent network (Elman, 1990) that encodes the structure of sequences themselves (Figure 3.3.2).

CDB06 implements the model using a connectionist framework. It represents words, their features, syntax and semantics as patterns of activation over units in a network. These units are connected to each other through weighted links and the system encodes its internal rules by adjusting weights on these links. The system learns most of these weights through error-backpropagation, and some of them are set manually. During comprehension and production, activation spreads in the network along these weighted connections, through both meaning and sequencing pathways. Different patterns of activation compete with each other for higher activation and the network makes a prediction based on the pattern that receives largest activation.

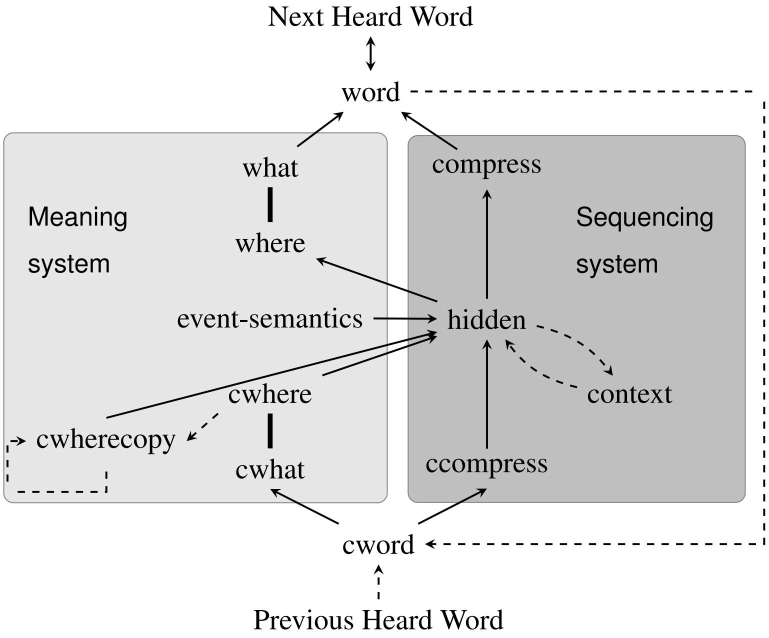

Figure 3.3.2: [Dual path model] The *Meaning system* and *Sequencing system* are parallel pathways for the flow of information in CDB06. The sequencing system performs categorisation using the *ccompress* and *compress* units and it performs sequencing using the sequential recurrent network implemented by *hidden* and *context* units. The thick lines between *what* and *where* units represents connections that are manually set at the beginning of the trial. Adapted from Chang et al. (2006).

At the heart of CDB06 is a sequential recurrent network (SRN). It is this element that allows the model to predict a word based on its context. An SRN is a feedforward network with the hidden layer connected to a recurrent layer. This recurrent layer stores the context, or the previous values, of the hidden layer. Every time the network receives an input, the output of the hidden layer is governed not just by this input, but also by the context stored in the recurrent layer. Elman (1990) showed that this setup in an SRN gives it the capability of remembering sequences of patterns. Chang et al. (2006) trained the network on a subset of the English language. They demonstrated that, after training on this subset of language, the network was able to predict words that were in grammatically correct positions 89% of the time and utterances that were semantically correct 82% of the time.

The sequential recurrent network developed by Elman (1990) was extended in two ways by Chang et al. (2006). First, they introduced a meaning pathway that runs parallel to the SRN and connects to the hidden layer of the SRN. This meaning pathway encoded the thematic roles of words and event-semantics of utterances. As a result of these connections, the SRN started to predict sequences based not just on input words, but also on the thematic and event-semantic properties of utterances. Next, Chang et al. (2006) introduced a set of *compression* units that lie between the internal and external representations of the SRN. These compression units reduced the dimensionality of external representations. As a result of this dimensionality-reduction, the SRN based its sequencing decisions on word classes rather than individual words. As the network learnt to order utterances, it also learnt to abstract away from specific words to word-classes, thus making the system less lexically specific with training. Together, the two extensions made the network more suited for natural language production and ensured that the network predicted word-categories based on the intended message.

§ 3.3.2 Testing CDB06

Chang et al. (2006) used their network to model two cognitive phenomena: Syntactic acquisition (during development) and structural priming (during a conversation). Both acquisition and priming are forms of learning. The central claim of Chang et al. (2006) was that both these phenomena rely on the same algorithm of learning: backpropagation of error generated by comparing predicted and actual input. They claimed that the cognitive system maintains an internal model of linguistic structure which it uses to make predictions during training. When the cognitive system makes predictions that do not match the external input, it adjusts its internal model. Over a long period of time, this learning algorithm leads to an internal model that correctly predicts the external input. Crucially, Chang et al. (2006) also claimed that over a shorter period of time, the same learning algorithm led to structural priming. Priming and acquisition are manifestations of the same higher cognitive principle: prediction-based learning.

To substantiate their claim, Chang et al. (2006) had to show that their model could reproduce experimental findings on priming and acquisition. Therefore, they needed to design simulations that replicated the procedure of experimental studies. Priming studies investigate how comprehension or production of utterances affect syntactic choices during subsequent utterances. Thus these experiments consist of *comprehension trials*, where subjects are required to understand an utterance that they see or hear, and *production trials* where the subjects are required to produce an utterance for a given picture or situation. Because Chang et al. (2006) wanted to replicate these experiments on the model, they defined corresponding comprehension and production trials for their model.

Definition of a production trial is simple. For subjects, production is the act of converting an abstract message into a sequence of words. The trial adopted this definition and initiated production by supplying the model with a message. It manually set the activation of conceptual, thematic and event-semantic units in the meaning pathway. It also set the strength of connections between these units. Given this message, the model was expected to produce a sequence of words. Since this was a production trial, the model received no external input and it did not perform any error correction.

Comprehension trials, however, are a bit more complicated to define. They always involve an external input, which is the sequence of words that the model is required to 'understand'. But this understanding can either proceed in the absence of any contextual information – i.e. without a given message – or in the presence of the intended message, in which case comprehension involves learning to match the utterance with the message. The latter kind of comprehension trials are essential if the model wanted to learn the connection strength between the meaning and sequential pathways. Chang et al. (2006) called these trials *situated events*, while the trials in which the model performed predictions in the absence of a message were called *messageless events*.

Chang et al. (2006) trained the model on a mixture of situated and messageless trials and tested it for messageless events followed by production trials. During both training and testing phases, the model learnt (through error-backpropagation) at the end of (both kinds of) comprehension trials. The amount of priming was measured as the difference between the percent of target structure produced after prime of the same and competing structures.

The trained model replicated results from Bock and Griffin (2000), showing a main effect of prime structure – i.e. the model was more likely to choose a syntactic structure if it had recently comprehended a prime with the same structure. As in Bock and Griffin (2000), the amount of priming persisted over lags of up to ten intervening filler trials. This result verified that error-based learning leads to both short-term and long-term changes in the system.

Chang et al. (2006) also replicated several other experimental findings which demonstrated that the model shows structural priming irrespective of thematic role overlap between prime and target. Both locatives such as *The wealthy widow drove an old Mercedes to the church* and prepositional datives such as *The wealthy widow gave an old Mercedes to the church* primed a dative target to the same extent (Bock & Loebell, 1990). Since the locatives and prepositional datives have the same structural form and since the model showed same amount of priming for both structures, Chang et al. (2006) argued that the model's sequencing system generalised over the two different thematic roles, making predictions based solely on each utterance's surface structure.

Finally, Chang et al. (2006) used the same model to explain patterns of syntactic acquisition. They tried to resolve the conflict between two contrasting observations of language acquisition by proposing that the same internal principles could manifest themselves as these contrasting observations. While the *early-syntax* theory (Fisher, 2002; Gleitman, 1990; Naigles, 2002) claims that children exhibit an ability to make structural classifications at an early age, the *late-syntax* theory (Bates & Goodman, 2001; Lieven, Behrens, Speares, & Tomasello, 2003; MacWhinney, 1987; Tomasello, 2003) claims that children's initial syntactic constructions are lexically specific and that they arrive at abstract syntactic constructions only at a later age. Early-syntax theories rely on evidence from preferential-looking data in children and late-syntax theories rely on evidence from sentence production. Chang et al. (2006) attempted to resolve this debate by showing that different estimates of syntactic competence in the model can explain observations compatible with both early and late-syntax theories. One estimate, the *error-difference score*, measured the amount of preferential looking while another estimate, *grammaticality*, measured the correctness of produced utterances. By analysing the value of these estimates at different stages of training, Chang et al. (2006) demonstrated that the model agreed with the early-syntax theory when syntactic competence is measured as the error-difference score, but with the late-syntax theory when syntactic competence is measured as grammaticality.

The major success of CDB06 is to demonstrate that the same principle of error-based learning can seamlessly explain the short-term phenomenon of structural priming and the long-term data regarding language acquisition.