plified by the fact that the word "milk" often occurs soon after the word "cow", but beyond a certain point any improvement in performance must result from a deeper understanding of the text's meaning.

Although standard RNNs are very expressive, we found that achieving competitive results on character-level language modeling required the development of a different type of RNN that was better suited to our application. This new "MRNN" architecture uses multiplicative connections to allow the current input character to determine the hidden-to-hidden weight matrix. We trained MRNNs on over a hundred of megabytes of text for several days using 8 Graphics Processing Units in parallel to perform significantly better than one of the best word-agnostic single character-level language models: the sequence memoizer (Wood et al., 2009; Gasthaus et al., 2010), which is a hierarchical nonparametric Bayesian method. It defines a prior process on the set of predictions at every conceivable context, with judiciously chosen details that make approximate inference computationally tractable. The memoizer induces dependencies between its predictions by making similar predictions at similar contexts. Although intelligent marginalization techniques are able to eliminate all but a relatively small number of the random variables (so the datastructures used scale linearly with the amount of data), its memory requirements are still prohibitively expensive for large datasets, which is a direct consequence of its nonparametric nature.

While our method performs at the state of the art for pure character-level models, its compression performance falls short of the best models which have explicit knowledge of words, the most powerful of these being PAQ8hp12 (Mahoney, 2005). PAQ is a mixture model of a large number of well-chosen context models whose mixing proportions are computed by a neural network whose weights are a function of the current context, and whose predictions are further combined with a neural-network like model. Unlike standard compression techniques, some of PAQ's context models not only consider contiguous contexts but also contexts with "gaps", allowing it to capture some types of longer range structures cheaply. More significantly, PAQ is not word-agnostic, because it uses a combination of character-level and word-level models. PAQ also preprocesses the data with a dictionary of common English words which we disabled, because it gave PAQ an unfair advantage over models that do not use such task-specific (and indeed, English-specific) explicit prior knowledge. The numerous mixture components of PAQ were chosen because they improved performance on a development set, so in this respect PAQ is similar in model complexity to the winning entry of the netflix prize (Bell et al., 2007).

Finally, language models can be used to "generate" language, and to our surprise, the text generated by the MRNNs we trained exhibited a significant amount of interesting and high-level linguistic structure, featuring a large vocabulary, a considerable amount of grammatical structure, and a wide variety of highly plausible proper names that were not in the training set. Mastering the vocabulary of English did not seem to be a problem for the MRNN: it generated very few uncapitalized non-words, and those that it did generate were often very plausible, like "homosomalist" or "un-ameliary". Of particular interest was the fact that the MRNN learned to balance parentheses and quotes over long distances (e.g., 30 characters). A character-level $N$-gram language model could only do this by modeling 31-grams, and neither Memoizer nor PAQ are representationally capable of balancing parentheses because of their need for exact context matches. In contrast, the MRNN's nonlinear dynamics enables it to extract higher level "knowledge" from the text, and there are no obvious limits to its representational power because of the ability of its hidden states to perform general computation.

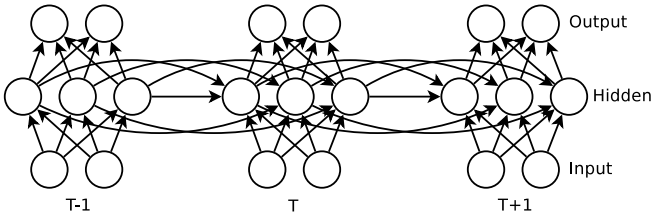


*Figure 1.* A Recurrent Neural Network is a very deep feedforward neural network whose weights are shared across time. The nonlinear activation function used by the hidden units is the source of the RNN's rich dynamics.

## 2. Recurrent Neural Networks

A Recurrent Neural Network is a straightforward adaptation of the standard feed-forward neural network to allow it to model sequential data. At each timestep, the RNN receives an input, updates its hidden state, and makes a prediction (fig. 1). The RNN's high dimensional hidden state and nonlinear evolution endow it with great expressive power, enabling the hidden state of the RNN to integrate information over many timesteps and use it to make accurate predictions. Even if the non-linearity used by each unit is quite simple, iterating it over time leads to very rich dynamics.

The standard RNN is formalized as follows: Given a sequence of input vectors $(x_1, \ldots, x_T)$, the RNN computes a sequence of hidden states $(h_1, \ldots, h_T)$ and a sequence of outputs $(o_1, \ldots, o_T)$ by iterating the following equations