

of musical notes used to compose songs. [Boulanger-Lewandowski et al. \(2012\)](#) introduced the **RNN-RBM** sequence model and applied it to this task. The RNN-RBM is a generative model of a sequence of frames $\mathbf{x}^{(t)}$ consisting of an RNN that emits the RBM parameters for each time step. Unlike previous approaches in which only the bias parameters of the RBM varied from one time step to the next, the RNN-RBM uses the RNN to emit all of the parameters of the RBM, including the weights. To train the model, we need to be able to back-propagate the gradient of the loss function through the RNN. The loss function is not applied directly to the RNN outputs. Instead, it is applied to the RBM. This means that we must approximately differentiate the loss with respect to the RBM parameters using contrastive divergence or a related algorithm. This approximate gradient may then be back-propagated through the RNN using the usual back-propagation through time algorithm.

20.8 Other Boltzmann Machines

Many other variants of Boltzmann machines are possible.

Boltzmann machines may be extended with different training criteria. We have focused on Boltzmann machines trained to approximately maximize the generative criterion $\log p(\mathbf{v})$. It is also possible to train discriminative RBMs that aim to maximize $\log p(y | \mathbf{v})$ instead ([Larochelle and Bengio, 2008](#)). This approach often performs the best when using a linear combination of both the generative and the discriminative criteria. Unfortunately, RBMs do not seem to be as powerful supervised learners as MLPs, at least using existing methodology.

Most Boltzmann machines used in practice have only second-order interactions in their energy functions, meaning that their energy functions are the sum of many terms and each individual term only includes the product between two random variables. An example of such a term is $v_i W_{i,j} h_j$. It is also possible to train higher-order Boltzmann machines ([Sejnowski, 1987](#)) whose energy function terms involve the products between many variables. Three-way interactions between a hidden unit and two different images can model spatial transformations from one frame of video to the next ([Memisevic and Hinton, 2007, 2010](#)). Multiplication by a one-hot class variable can change the relationship between visible and hidden units depending on which class is present ([Nair and Hinton, 2009](#)). One recent example of the use of higher-order interactions is a Boltzmann machine with two groups of hidden units, with one group of hidden units that interact with both the visible units \mathbf{v} and the class label y , and another group of hidden units that interact only with the \mathbf{v} input values ([Luo et al., 2011](#)). This can be interpreted as encouraging