Learning with general recurrent neural networks

Guy Isely
Feedforward neural networks are “timeless”
Some problems with a feedforward model of temporal processes

• Computational cost grows with temporal duration modeled

• Can’t capture long-time contextual dependencies in sequences

• Networks don’t have persistent state—“noise correlations” might be state!
Hopfield networks have fixed-point attractor dynamics

- Dynamics are gradient descent on an energy function (the Lyapunov function)
- Autonomous after initial input
- Guaranteed to converge to a stable fixed point due to symmetric connectivity matrix
Can we use gradient descent to train general RNNs?

Yes, yes we can!

...but there's a wrinkle.
Backpropagation review

\[ \Delta W_{kl} = -\eta \frac{\partial E}{\partial W_{kl}} = \eta \sum_i [T_i - z_i(x)] \frac{\partial z_i(x)}{\partial W_{kl}} \]

\[ \frac{\partial z_i(x)}{\partial W_{kr}} \frac{\partial z_i(x)}{\partial y_k} \frac{\partial y_k}{\partial W_{kl}} \]

\[ \Delta W_{kl} = \eta \sum_i [T_i - z_i(x)] \sigma'(u_{zi}) V_{ik} \sigma'(u_{yk}) x_l \]

\[ = \eta \delta_{yk} x_l \]

where

\[ \delta_{yk} = \sigma'(u_{yk}) \sum_i \delta_{zi} V_{ik} \]

It's just the chain rule!
Can we apply backpropagation directly to an RNN?

• Not exactly— the gradient of a RNN’s error function w.r.t. to the weights depends on the network’s state at all previous time steps.

• But we can unravel the network structure in time to get a feedforward network and perform backprop on this network.

• This is called backpropagation through time (BPTT).
Realtime recurrent learning (Williams and Zipser 1989)

\[
\frac{\partial y(t)}{\partial W_r} = \text{diag}(\sigma'(y(t)))W^\top \cdot \frac{\partial y(t - 1)}{\partial W_r}
\]

- We can run the recurrence relation underlying the gradient computation forward in time!
- Downside: BPTT is $O(tn^2)$ per time step but RTRL is $O(n^3)$ per time step—prohibitive for large networks!
Intermission
(aka neural network winter)
Reservoir computing

Echo State Networks
(Jaeger & Haas 2004)

Liquid State Machines
(Maass et al. 2002)
Ideas from Echo State Networks

• Use an unoptimized random sparsely connected recurrent reservoir and do a linear readout.

• Only optimize the readout weights.

• Use teacher forcing to achieve appropriately tuned the reservoir dynamics
BPTT returns (with a vengeance)
Where to next?

- Address vanishing/exploding sensitivity problem with network units designed for specific temporal dynamics (e.g. Long Short Term Memory)
- Move beyond gradient descent based approaches to optimizing network parameters
- Incorporate addition biophysical features of real networks (e.g. STDP, metabotropic receptor dynamics, gap junctions, dendritic non-linearities)