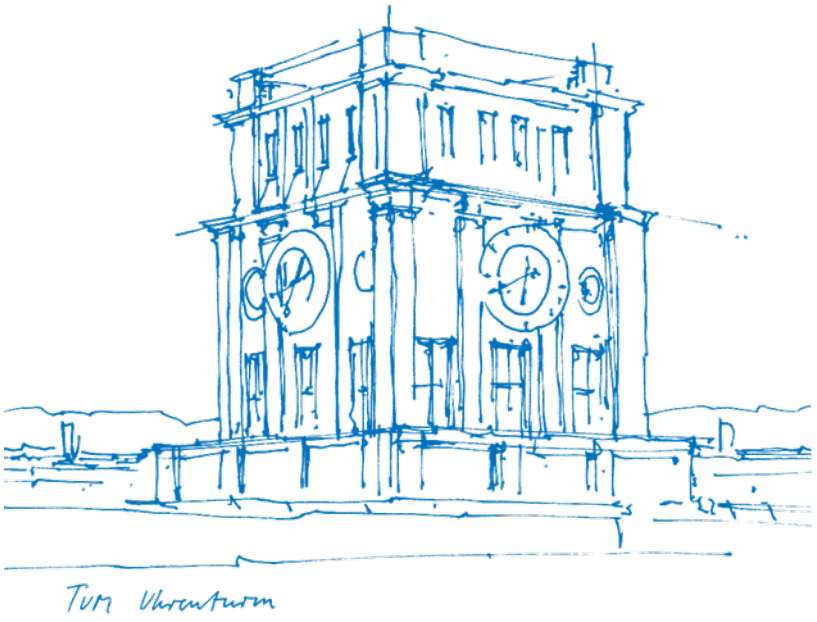
Real time evolution with neural-network quantum states by approximating the implicit midpoint method

(arXiv:1912.08831)

Christian B. Mendl
Technische Universität München
Department of Informatics
June 23, 2020

Joint work with Irene López Gutiérrez

Machine Learning for Quantum Simulation: Virtual Conference
Flatiron Institute – Center for Computational Quantum Physics

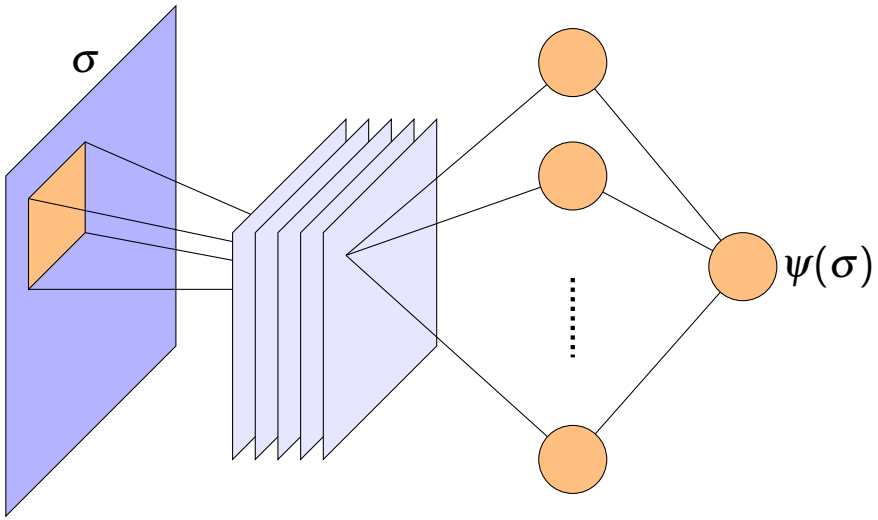


Motivation: Simulate real time evolution

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}\psi = H\psi$$

Tensor network methods limited by entanglement growth

$\rightsquigarrow$ Ansatz for ψ: neural-network quantum state (here: “classical” artificial neural network)



G. Carleo and M. Troyer, Science 355, 602–606 (2017);
I. Glasser et al., Phys. Rev. X 8, 011006 (2018) ; X. Gao and L.-M. Duan, Nat. Commun. 8, 662 (2017)

Approach: approximate numerical ODE method

Ansatz $\psi[\theta]$ with θ: network params, for each time step Δt, $\theta_n \to \theta_{n+1}$:

$$\min_{\theta_{n+1}} \left\| \psi[\theta_{n+1}] - \Phi^{\Delta t}(\psi[\theta_n]) \right\|$$

discrete flow of an ODE method

Approach: approximate numerical ODE method

Ansatz $\psi[\theta]$ with θ: network params, for each time step Δt, $\theta_n \to \theta_{n+1}$:

$$\min_{\theta_{n+1}} \left\| \psi[\theta_{n+1}] - \Phi^{\Delta t}(\psi[\theta_n]) \right\|$$

discrete flow of an ODE method

Here: $\Phi^{\Delta t}$ implicit midpoint method (*symplectic*, no intermediate quantities)

Applied to Schrödinger:

$$\psi[\theta_{n+1}] \approx \psi[\theta_n] - i\Delta t\, H\left(\frac{\psi[\theta_{n+1}] + \psi[\theta_n]}{2}\right)$$

Approach: approximate numerical ODE method

Ansatz $\psi[\theta]$ with θ: network params, for each time step Δt, $\theta_n \to \theta_{n+1}$:

$$\min_{\theta_{n+1}} \left\| \psi[\theta_{n+1}] - \Phi^{\Delta t}(\psi[\theta_n]) \right\|$$

discrete flow of an ODE method

Here: $\Phi^{\Delta t}$ implicit midpoint method (*symplectic*, no intermediate quantities)

Applied to Schrödinger:

$$\psi[\theta_{n+1}] \approx \psi[\theta_n] - i\Delta t\, H\left(\frac{\psi[\theta_{n+1}] + \psi[\theta_n]}{2}\right)$$

$\rightsquigarrow$ cost function, for input "samples" $\{\sigma^{(j)}\}$:

$$C(\theta_{n+1}) = \sum_{j=1}^{N} \left| \left(\left(I + \tfrac{i\Delta t}{2} H\right) \psi[\theta_{n+1}] - \left(I - \tfrac{i\Delta t}{2} H\right) \psi[\theta_n] \right) (\sigma^{(j)}) \right|^2$$

Approach: approximate numerical ODE method

Ansatz $\psi[\theta]$ with θ: network params, for each time step Δt, $\theta_n \to \theta_{n+1}$:

$$\min_{\theta_{n+1}} \left\| \psi[\theta_{n+1}] - \Phi^{\Delta t}(\psi[\theta_n]) \right\|$$

discrete flow of an ODE method

Here: $\Phi^{\Delta t}$ implicit midpoint method (*symplectic*, no intermediate quantities)

Applied to Schrödinger:

$$\psi[\theta_{n+1}] \approx \psi[\theta_n] - i\Delta t\, H\left(\frac{\psi[\theta_{n+1}] + \psi[\theta_n]}{2}\right)$$

$\rightsquigarrow$ cost function, for input "samples" $\{\sigma^{(j)}\}$:

$$C(\theta_{n+1}) = \sum_{j=1}^{N} \left| \left(\left(I + \tfrac{i\Delta t}{2} H\right) \psi[\theta_{n+1}] - \left(I - \tfrac{i\Delta t}{2} H\right) \psi[\theta_n] \right) (\sigma^{(j)}) \right|^2$$

For comparison: stochastic reconfiguration (SR): eventually have to solve

$$S\dot{\theta} = -iF,$$

but S can be (almost) singular, solution sensitive to pseudo-inverse cut-off

Backpropagation with complex parameters

Cost function in least squares form:

$$C(\theta) = \|A\psi[\theta] - b\|^2,$$

with A: rows of $I + \frac{i\Delta t}{2} H$ corresponding to $\sigma^{(j)}$, and

$$b_j = \left(\left(I - \tfrac{i\Delta t}{2} H\right) \psi[\theta_n]\right)\left(\sigma^{(j)}\right), \quad j = 1, \ldots, N$$

Backpropagation with complex parameters

Cost function in least squares form:

$$C(\theta) = \|A\psi[\theta] - b\|^2,$$

with A: rows of $I + \frac{i\Delta t}{2} H$ corresponding to $\sigma^{(j)}$, and

$$b_j = \left(\left(I - \tfrac{i\Delta t}{2} H\right) \psi[\theta_n]\right)\left(\sigma^{(j)}\right), \quad j = 1, \ldots, N$$

Gradients w.r.t. *complex* parameters θ via *Wirtinger formalism* ($z = x + iy$):

$$\frac{\partial}{\partial z} := \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right), \quad \frac{\partial}{\partial z^*} := \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right)$$

Applied to cost function:

$$\frac{\partial C(\theta)}{\partial \theta_\ell} = \left\langle A\psi[\theta] - b \middle| A\frac{\partial \psi[\theta]}{\partial \theta_\ell} \right\rangle$$

Backpropagation with complex parameters

Cost function in least squares form:

$$C(\theta) = \|A\psi[\theta] - b\|^2,$$

with A: rows of $I + \frac{i\Delta t}{2} H$ corresponding to $\sigma^{(j)}$, and

$$b_j = \left(\left(I - \tfrac{i\Delta t}{2} H\right) \psi[\theta_n]\right)\left(\sigma^{(j)}\right), \quad j = 1, \ldots, N$$

Gradients w.r.t. *complex* parameters θ via *Wirtinger formalism* ($z = x + iy$):

$$\frac{\partial}{\partial z} := \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right), \quad \frac{\partial}{\partial z^*} := \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right)$$

Applied to cost function:

$$\frac{\partial C(\theta)}{\partial \theta_\ell} = \left\langle A\psi[\theta] - b \middle| A\frac{\partial \psi[\theta]}{\partial \theta_\ell} \right\rangle$$

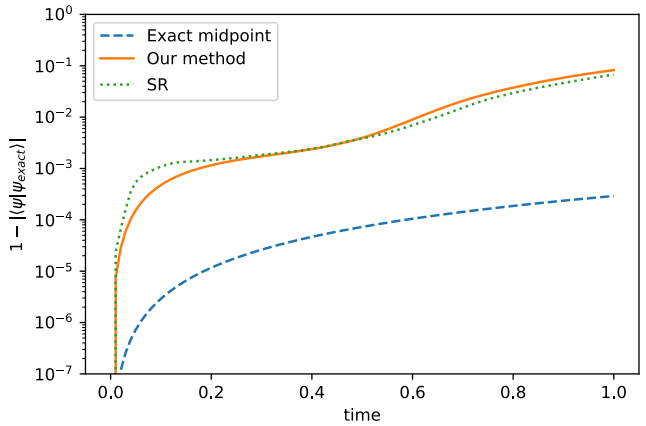
Exploit real-valued target function for Wirtinger *chain rule*

Here: operations inside neural network chosen holomorphic $\rightsquigarrow$ conventional complex derivatives $\frac{\partial \psi[\theta]}{\partial \theta_\ell}$

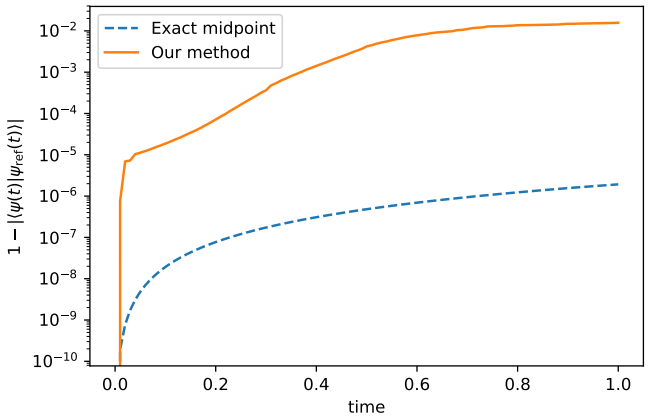
Hirose 2012; Trabelsi et al. 2018; cf. `https://fluxml.ai/Zygote.jl/latest/complex/`

Example: time evolution governed by Ising model

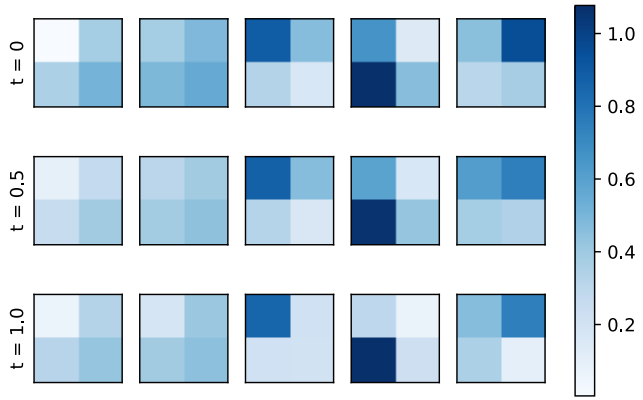
$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - h \sum_i \sigma_i^x$$



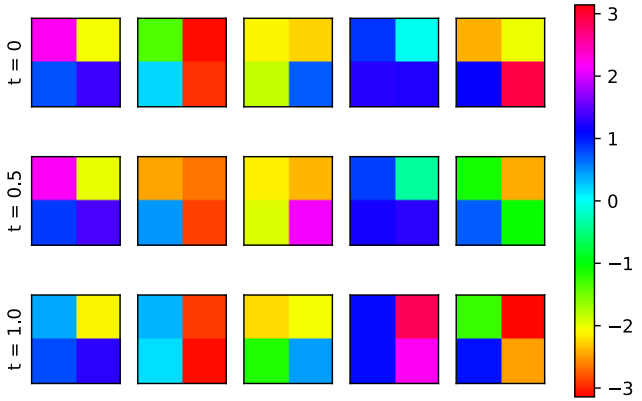
(a) 20 sites 1D lattice, RBM Ansatz



(b) 3×3 lattice, conv-layer



(c) amplitudes of filter weights



(d) arguments of filter weights

Implementation details

- (Classical) neural networks, including conv-layers via `im2col`, in C
- Construct selected rows of the Ising Hamiltonian as sparse CSR matrix
- OpenMP parallel gradient computation
- Store network (weights and biases, layer topology) in HDF5 file
- Python interface

```
//________________________________________________________________________________________________
///
/// \brief 2D convolution backward pass, using periodic boundary conditions
///
/// 'cache' must be unmodified from forward pass, will be overwritten.
///
void conv2d_backward(const struct conv2d_dims *dims, const int N, const double complex *w, const double complex *da,
struct conv2d_cache *cache, double complex *dx, double complex *dw, double complex *db)
{
    // output dimensions
    const int out_width  = dims->in_width  / dims->stride;
    const int out_height = dims->in_height / dims->stride;

    // compute gradient with respect to entries of kernel 'w' (requires 'xcol' in 'cache')
    {
        const double complex one  = 1;
        const double complex zero = 0;
        const int m = dims->kheight*dims->kwidth*dims->channels;
        const int n = dims->features;
        const int k = N*out_height*out_width;
        cblas_zgemm(CblasRowMajor, CblasTrans, CblasNoTrans, m, n, k, &one, cache->xcol, m, da, dims->features, &zero, dw,
         dims->features);
    }
```

Conclusions and outlook

- Direct ODE approximation

$$\min_{\theta_{n+1}} \left\| \psi[\theta_{n+1}] - \Phi^{\Delta t}(\psi[\theta_n]) \right\|$$

  as alternative to SR, expecting advantages for deeper networks
- Can change network architecture “on the fly”
- Our simulations: large room for improvement

  but cf. parallel work by M. Schmitt and M. Heyl, arXiv:1912.08828
- Future plans: explore “neural ordinary differential equations”

  R. T. Q. Chen et al., arXiv:1806.07366

References

- Carleo, G. and Troyer, M. (2017). “Solving the quantum many-body problem with artificial neural networks”. In: *Science* 355, pp. 602–606.
- Chen, R. T. Q. et al. (2018). “Neural ordinary differential equations”. In: *arXiv:1806.07366*. URL: https://arxiv.org/abs/1806.07366.
- Gao, X. and Duan, L.-M. (2017). “Efficient representation of quantum many-body states with deep neural networks”. In: *Nat. Commun.* 8, p. 662.
- Glasser, I. et al. (2018). “Neural-network quantum states, string-bond states, and chiral topological states”. In: *Phys. Rev. X* 8, p. 011006.
- Hirose, A. (2012). *Complex-Valued Neural Networks*. Springer-Verlag Berlin Heidelberg.
- López Gutiérrez, I. and Mendl, C. B. (2019). “Real time evolution with neural-network quantum states”. In: *arXiv:1912.08831*. URL: https://arxiv.org/abs/1912.08831.
- Schmitt, M. and Heyl, M. (2018). “Quantum dynamics in transverse-field Ising models from classical networks”. In: *SciPost Phys.* 4, p. 013.
- — (2019). “Quantum many-body dynamics in two dimensions with artificial neural networks”. In: *arXiv:1912.08828*. URL: https://arxiv.org/abs/1912.08828.
- Trabelsi, C. et al. (2018). “Deep complex networks”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H1T2hmZAb.