In this file we propose the method for search of ground state energy in spin models. The variational energy is given by

$$E = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle}.$$

If the wave function depends in the set of parameters W, energy gradient reads

$$\nabla_W E = \frac{1}{\langle\psi|\psi\rangle}\left(\partial_W\langle\psi|\hat{H}|\psi\rangle - \partial_W \log\langle\psi|\psi\rangle\right).$$

Note that overall factor $\langle\psi|\psi\rangle$ is adsorbed by «learning rate» and can be later neglected. The aim of this proposal is to reformulate froblem in the way that it can be solved by standard neural network frameforks such as *tensorflow, keras, theano*. To do so, one needs to provideanalytical gradient approximation or loss approximation. Then the framework will run super-fast and efficient gradient descent. The other advantage is that one will be able so very simply change network structure (for instance, use 2D convolutional layers) without manually programming these layers operations.

Usually, the neural network training in standard frameworks is done as follows: first, a subset (batch) of training samples is taken. Using this batch, loss and gradient are approximated, and then the gradient descent iteration is done. In our case, we want the network to take N spin values ± 1 and return two numbers: real and imaginary part of the corresponding amplitude. In terms of basis states, gradient reads

$$\partial_W\left(\sum_{mn}\psi_m^* H_{mn}\psi_n - \sum_l \psi_l^*\psi_l\right).$$

The problem at this point is that the first term can not be decoupled into independent samples (states) contributions, because matrix elements H_{mn} couple different states. However, there is a solution! We suggest to approximate this gradient as follows: consider network that takes *two* basis states at once and predicts *two* amplitudes at once. This two predictions are done in parallel, but they share the same weights.

We suggest sampling *pairs* of basis states $\{n, m\}$, feeding them to this two-at-a-time predicting network, thus obtaining pairs of $\{\psi_n, \psi_m\}$ and estimating loss as

$$\sum_{\{n,m\}}\psi_m^* H_{mn}\psi_n.$$

This loss is analytically differentiable with respect to weights W, because $\psi(n, W)$ is an analytical function of weights, and gradients are calculated automatically in standard frameworks.

Finally, one also need to estimate the contribution of $\langle\psi|\psi\rangle$ to gradient. As the paper of G. Carleo suggests, efficient approximation of this term is done using Metropolis algorithm. Thus, we need to wisely choose the subset of indexes $\{l\}$ to estimate $\sum_l \psi_l^*\psi_l$. Given current weights configuration W, we apply Metropolis algorithm (desctibed in Appendix C) to generate the set of states $l_0 \to l_1 \to \ldots \to l_k$. These indexes are used for generation of ψ_{l_0}, $\psi_{l_1}, \ldots$ and thus obtaining loss

$$-\log\left(\sum_{\{l\}}\psi_l^*\psi_l\right),$$

which is again analytically differentiable with respect to current weights W.

Thus, we constructed an approach of a simple feed-forward neural network for variational ground state energy minimization. Such aproach does not require programming gradients (which are calculated automatically) and only requires efficient computation of matrix elements H_{mn}. ANother advantages of this approach is efficiency (frameworks have python interface but work super-fast because of C++ or fortran inner implementation) and ability to simple switch between various network architectures (network architecture is constructed very easily within standard frameworks).