Exercises for “Vertiefung Neuronale Netze”
SS 2018 Sheet 9

Due on: 25.5.2018

Task 9.1, LLM: Implement the LLM algorithm in a programming language of your choice (e.g. `python`). Keep your data structures extendable to other methods as the next assignments will draw on them: A network should contain a number of nodes with edges to their neighbors. Each Node is associated with multiple attributes like reference vectors in the input and output space (w_c^{in}, w_c^{out}) as well as a Jacobian A_c. Choose for yourself, which topological algorithm (SOM, GNG, ITM) you want to use in your implementation.

Consider the ability to visualize the network before choosing the programming language – you should be comfortable with the visualization tools in the selected language. Useful tutorials for `python` are for example:

- `https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`
- `https://docs.scipy.org/doc/numpy-dev/user/basics.broadcasting.html`
- `http://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html`
- `http://docs.enthought.com/traits/traits_user_manual/index.html`

Train the network with the function $z(x, y) = x \cdot y$ in the unit cube $[-1, 1]^2$ with about 100 nodes. Visualize the output of the network on the grid $[-1.5, 1.5]^2$ (using a grid spacing of 0.05) and compare the different interpolation forms:

1. simple vector quantization: $y(x) = w_n^{out}$
2. Soft-Max-Interpolation:

$$\begin{aligned} y(x) &= \sum_{c \in \mathcal{C}} g_c \cdot w_c^{out} \\ \text{with} \qquad g_c &= \frac{w_c}{N} \\ w_c &= exp\left(-\frac{\|x - w_c^{in}\|^2}{\sigma_c^2}\right) \\ N &= \sum_{c \in \mathcal{C}} w_c \end{aligned}$$

3. LLM: $y(x) = w_n^{out} + A_n(x - w_n^{in})$
4. LLM with Soft-Max-Interpolation: $y(x) = \sum_{c \in \mathcal{C}} g_c \cdot \left(w_c^{out} + A_c(x - w_c^{in})\right)$

Experiment with various neighborhood sets for soft-max interpolation:

1. direct topological neighbors: $\mathcal{C} = \mathcal{N}(w_n)$
2. $N \approx 4$ closest neighbors – measured in the input space (instead of the graph)
3. complete network: $\mathcal{C} = \mathcal{A}$

If the time until next friday should be too short, we can discuss the results on 1.6. too. Please provide feedback regarding this in the next lecture. Teams with up to two members are allowed.