DATA.ML.200 Pattern Recognition and Machine Learning

Exercise week 3: Perceptron and Multi-layer Perceptron (MLP)

Be prepared for the exercise sessions (watch the demo lecture). You may ask TAs to help if you cannot make your program to work, but don't expect them to show you how to start from the scratch.

1. `pen&paper` *Derive update gradients for a perceptron with three inputs*

   In your homework you derived the gradient with respect to the weights ($\frac{\nabla \mathcal{L}}{\nabla w_i}$) for two inputs ($x_1$ and x_2) and a bias (+1). Add gradient to one more input x_3.

2. `python` *Perceptron gradient descent (10 pts)*

   Write gradient descent (GD) Python code for a single neuron and train and test how well it learns the following logical functions:

   - $x_1 \wedge x_2 \wedge x_3$
   - $x_1 \vee x_2 \vee x_3$
   - $((x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)) \wedge x_3$

   For all combinations of the inputs print the ground truth values and the outputs of a trained perceptron.

   Return the following items:

   - Python code: perceptron.py that does all above.
   - PNG image: your full desktop screenshot that includes a terminal where the python file is executed and prints the results: perceptron_screenshot.png

3. `pen&paper` *Derive necessary MLP gradients*

   During the lectures and in your homework you derived `Forward pass`, `Backward pass`, and `Update rule` items for a three neuron MLP. Keep these, but add one more input x_3 to the hidder layer neurons.

4. `python` *MLP gradient descent. (20 pts)*

   Write gradient descent (GD) Python code for the MLP and train and test how well it learns the following logical functions:

   - $x_1 \wedge x_2 \wedge x_3$
   - $x_1 \vee x_2 \vee x_3$
   - $((x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)) \wedge x_3$

For all combinations of the inputs print the ground truth values and the outputs of the trained MLP.

Return the following items:

- Python code: mlp.py that does all above.
- PNG image: your full desktop screenshot that includes a terminal where the python file is executed and prints results: mlp_screenshot.png