Computational Thinking
Sample Solutions to Exercise 11

1 Limitations of Neural Networks

A neural network can in theory approximate any continuous function given a sufficiently large number of hidden nodes. Therefore, only c) and e) cannot be represented, as those functions are not continuous.

2 VC Dimension

A linear logistic regression on two scalar inputs gives a classification boundary that can be visualized as a line in the 2-dimensional input plane. Given three points on this 2-dimensional plane (that do not lie on a line), we can always draw a line that separates the points into 2 classes. Specifically, we can do so to get a correct classifier for every possible labeling of the points. Given 4 points however, we can label points in such a way that no line can separate the classes. An example is the XOR labeling in Figure 1. Note that such a labeling can be given to any 4 points in the plane. Therefore, the VC dimension of a linear logistic regression classifier is 3, as no data set of 4 points exists that allows all labelings.

![Figure 1: The XOR function visualized in the 2-dimensional input space with the labels represented as colors (0 as red and 1 as blue). No line can separate the classes.](image)

3 An Ill-Designed Network

a) \( \hat{f}(x|a,b) = 1 \cdot \tanh(100 \cdot 0.9) = 1 \) (given numerical precision)

b) \( \frac{dL}{db} = \frac{dL}{df} \cdot \frac{df}{db} = (-\hat{f}(x) + \hat{f}(x|a,b)) \cdot \tanh(ax) = 0.1 \cdot \tanh(90) = 0.1 \)
e) \[
\frac{dL}{db} = \frac{dL}{df} \cdot \frac{df}{d\tanh(ax)} \cdot \frac{d\tanh(ax)}{d(ax)} \cdot x
\]
\[
= (-f(x) + \hat{f}(x|a,b)) \ast b \ast (1 - \tanh^2(ax)) \ast x
\]
\[
= 0.0 \text{ (since } 1 - \tanh^2(90) = 0). \tag{3}
\]

\(\text{d) } a_{\text{new}} = a, \ b_{\text{new}} = b - 0.1 \cdot \frac{dL}{da} = 0.99. \) The weight \(a\) which causes the issue did not get any update due to a vanishing gradient, which causes the problem to persist for further updates.

\(\text{e) If we do the same calculations for } x = 0.9 \text{ again we find that } \frac{dL}{da} \approx 3099.56. \) This yields \(a_{\text{new}} = a - \alpha \frac{dL}{da} \approx -308.956\) and following updates will again have the vanishing gradient problem. The first update suffers from what is called an exploding gradient here.

\[\text{[Bonus]}\] The hyperbolic tangent is close to linear around the origin, a decent approximation would therefore be given by \(0 < a << 1\) and \(b = 1/a\).

4 Gradient Descent with Momentum

\(\text{a) } \beta = 0\)

\(\text{b) Roughly at the same point where the light green cross is, as the loss surface is flat which leads to a gradient close to zero.}\)

\(\text{c) The update is much bigger into the direction of the global optimum as } m_w\text{ is dominated by the bigger gradient from the preceding step.}\)

\(\text{d) In the global optimum.}\)

\(\text{e) The large gradients in the first few iterations might dominate } m_w \text{ and drive the optimization across the global optimum up the hill into the local optimum on the right.}\)