

## Exercises

Machine Learning: Foundations and Applications  
MATH CS 120AI

Paul J. Atzberger  
<https://web.atzberger.org/>

Can choose to complete 1 out the following 3 problems.

- (Neural Networks) Consider a basic Multilayer Perceptron (MLP) with two inputs  $x_1, x_2$ , single output  $y$ , and a hidden layer with  $n$  units  $h_i$ . Corresponding to this MLP is the hypothesis space  $\mathcal{H} = \{q : \mathbb{R}^2 \rightarrow \mathbb{R} \mid q(x_1, x_2; \mathcal{W}) = \sum_{i=1}^n w_i^{(2)} h_i, \text{ where } h_i = \sigma(w_{i1}^{(1)} x_1 + w_{i2}^{(1)} x_2)\}$ . The output is  $y = q(x_1, x_2; \mathcal{W})$  where the  $\mathcal{W}$  denotes the collection of weights.
  - Consider the case where we set  $x_2 = 1$  and  $x_1 \in [0, 1]$  with activation the Rectified Linear Unit (ReLU)  $\sigma = \max(0, z)$ . Show that with at most  $n = k + 2$  hidden units we can exactly represent any function  $f(x_1)$  that is piece-wise linear with  $k$  internal transition points on  $[0, 1]$  and  $f(x) = 0$  for  $x \notin (0, 1)$ . For instance, show that  $f(x) = 2x$  for  $x \leq 1/2$  and  $f(x) = 2(1 - x)$  for  $x > 1/2$ , which has  $k = 1$  internal transition points, can be exactly represented on  $[0, 1]$  by a MLP with  $n = 3$  hidden units.
  - Consider approximating a general function  $f(x)$  on  $[0, 1]$  by using a gradient descent  $\dot{\mathbf{w}} = -\alpha \nabla_{\mathbf{w}} L$  to minimize the loss  $L(q) = \frac{1}{m} \sum_{i=1}^m (f(z_i) - q(z_i; \mathbf{w}))^2$ . Consider  $m$  data points  $z_i \in [0, 1]$  where we take in the MLP  $x_1 = z_i$  and  $x_2 = 1$ . State for the MLP the back-propagation method for computing the gradient in  $\mathbf{w}$ . Draw the computational graph in the case when  $n = 1$  and  $m = 1$  for both the “forward pass” and the “backward pass.”
  - Explain techniques for how you might mitigate getting stuck in local minima or overfitting the data?
- (Neural Networks) Consider a Multilayer Perceptron (MLP) with two inputs  $x_1, x_2$  and activation  $\sigma(z)$ .
  - Show a single layer Perceptron is not able to solve the XOR problem to output  $y = x_1 \otimes x_2$ .
  - Show a two layer MLP can solve the XOR problem. For inputs  $x_1, x_2 \in \{-1, 1\}$  show there is an MLP with  $\sigma(z) = \max(z, 0)$  that can give the correct output  $y = x_1 \otimes x_2$ .
  - Show for any Boolean function  $b(x_1, x_2) : \{-1, 1\}^2 \rightarrow \{-1, 1\}$  there is an MLP with two layers and activations  $\sigma(z) = \text{sign}(z)$  that can compute the output matching  $b$ . Show when the activation is  $\sigma(z) = \max(z, 0)$  the two layer MLP can also compute outputs matching  $b$ .
- (Perceptron and Gradient Descent) Consider a dataset  $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$  where  $\mathbf{x}_i \in \mathbb{R}^d$  and  $y_i \in \{-1, +1\}$ . Assume this dataset is separable in the sense that there exists a  $\mathbf{w}$  so that  $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1$ . Let  $\mathbf{w}^*$  be such a separating weight that has the minimum norm  $\|\mathbf{w}^*\|^2$ . Consider  $f(\mathbf{w}) = \max_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)$ .
  - Show that for  $\mathcal{S}$  and  $\mathbf{w}^*$  above we have  $\min_{\mathbf{w}: \|\mathbf{w}\| \leq \|\mathbf{w}^*\|} f(\mathbf{w}) = 0$ . Show that for any  $\mathbf{w}$  with  $f(\mathbf{w}) < 1$  we have the weight  $\mathbf{w}$  separates  $\mathcal{S}$ .

- (b) Compute the subgradient  $\partial f$  of the function  $f(\mathbf{w})$ . The subgradient at  $\mathbf{w}$  is the set of vectors  $\partial f(\mathbf{w})$  with  $\mathbf{g} \in \partial f(\mathbf{w})$  if  $\forall \mathbf{u}$  we have  $f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \mathbf{g} \rangle$ .
- (c) Consider the Stochastic Subgradient Descent optimization algorithm for  $f(\mathbf{w})$ . Compare this to the Batch Perceptron training algorithm discussed previously and in the exercises.