APM 598: Homework 2 (02/26)

1 Two-layers neural networks

Ex 1.

Suppose $\mathbf{x} \in \mathbb{R}^2$. We consider two-layers neural networks (n.n.) of the form (see fig. 1):

$$f(\mathbf{x}) = \mathbf{b}_2 + W_2\Big(\sigma(\mathbf{b}_1 + W_1 \cdot \mathbf{x})\Big), \tag{1}$$

where $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^2$ are *'bias'* vectors, $W_1, W_2 \in \mathcal{M}_{2\times 2}(\mathbb{R})$ are matrices (2×2) and the activation function σ is the ReLu function (i.e. $\sigma(x) = \max(x, 0)$). We denote by $\mathbf{s} = f(\mathbf{x})$ the *score* predicted by the model with $\mathbf{s} = (s_1, s_2)$ where s_1 is the score for class 1 and s_2 the score for class 2.

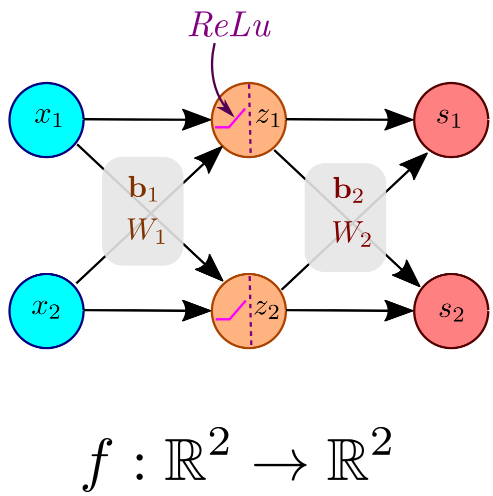

Figure 1: Illustration of a two-layer neural network using ReLu activation function.

a) Consider the points given in figure 2-left where each color correspond to a different class:

$$\begin{aligned} &\text{class 1:} \quad \mathbf{x}_1 = (1, 0) \text{ and } \mathbf{x}_2 = (-1, 0), \\ &\text{class 2:} \quad \mathbf{x}_3 = (0, 1) \text{ and } \mathbf{x}_4 = (0, -1). \end{aligned}$$

Find some parameters $\mathbf{b}_1$, $\mathbf{b}_2$, W_1 and W_2 such that the scores $\mathbf{s}$ satisfy:

$$s_1 > s_2 \text{ for } \mathbf{x}_1 \text{ and } \mathbf{x}_2 \,, \quad s_1 < s_2 \text{ for } \mathbf{x}_3 \text{ and } \mathbf{x}_4.$$

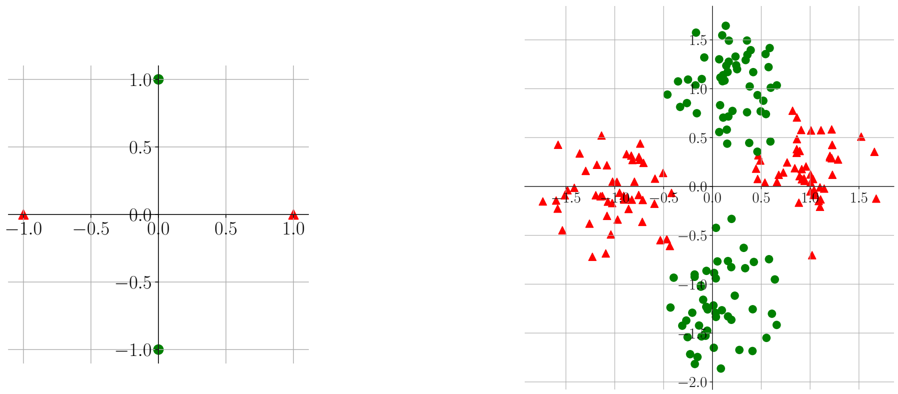

Figure 2: Data points to classify.

b) Consider now the data-set given in figure 2-right (see code below to load the data). Train a two-layer neural network of the form (1) to classify the points. Provide the accuracy of the model (percentage of correctly predicted labels).

```
##################################
##          Exercise 1          ##
##################################
import numpy as np
import pandas as pd
df = pd.read_csv('data_HW2_ex1.csv')
X = np.column_stack((df['x1'].values,df['x2'].values))
y = df['class'].values
```

Ex 2.

The goal of this exercise is to show that two-layers neural networks with ReLu activation can approximate any continuous functions. To simplify, we restrict our attention to the one-dimensional case:

$$g : [0,1] \longrightarrow \mathbb{R} \text{ (continuous)}.$$

We claim that for any $\varepsilon > 0$, there exists f two-layers n.n. such that:

$$\max_{x\in[0,1]} |g(x) - f(x)| < \varepsilon. \tag{2}$$

In contrast to **Ex 1**, f will be taken with a *large* hidden layers, i.e. $\mathbf{z} \in \mathbb{R}^m$ with $m \gg 1$ (see figure 3-left). To prove this result, we are going to show that f can be used to perform piece-wise linear interpolation (see figure 3-right).

a) Denote $y_0 = g(0)$ and $y_1 = g(1)$. Find a two layers n.n. such that $f(0) = y_0$ and $f(1) = y_1$.

b) Consider now three points: $y_0 = g(0)$, $y_1 = g(1/2)$, $y_2 = g(1)$. Find f such that: $f(0) = y_0$, $f(1/2) = y_1$ and $f(1) = y_2$.

c) Generalize: write a program that take as inputs $\{(x_i, y_i)\}_{0 \le i \le N}$ with $x_i < x_{i+1}$ and return a two layers n.n. such that $f(x_i) = y_i$ for all $i = 0 \dots N$.

Extra) Prove (2).
Hint: use that g is uniformly continuous on $[0,1]$.

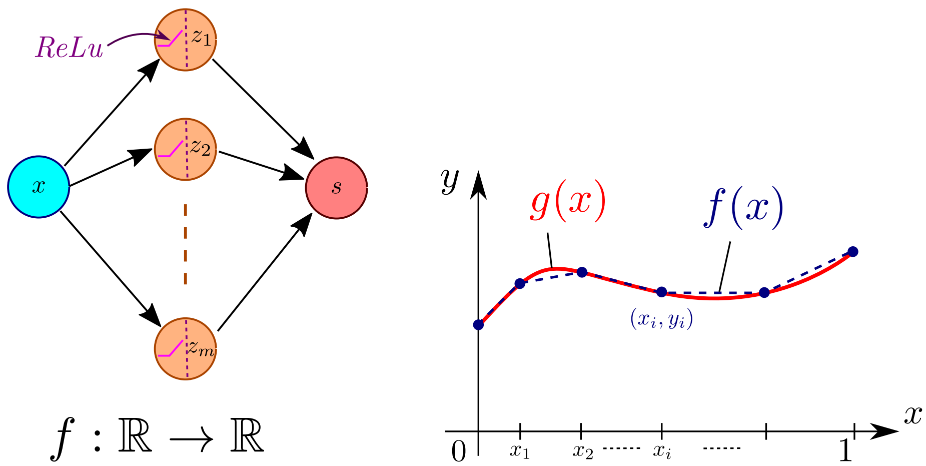

Figure 3: **Left**: two layers neural network used to approximate continuous function. The *hidden* layer (i.e. $\mathbf{z} = (z_1, \dots, z_m)$) is in general quite large. **Right:** to approximate the continuous function g, we interpolate some of its values (x_i, y_i) by a piece-wise linear function.

2 Convolutional Neural Networks

Ex 3.

Using convolutional layers, max pooling and ReLu activation functions, build a classifier for the Fashion-MNIST database (see a sketch example in figure 4). The accuracy of your network on the test set will be your score on this exercise (+5 points for the group with the highest accuracy).

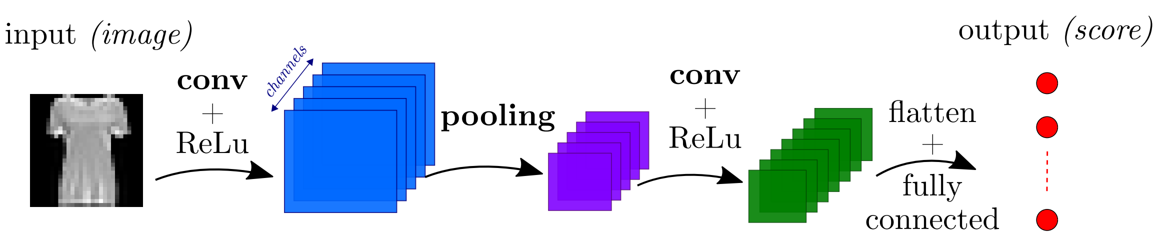

Figure 4: Schematic representation of neural network for image classification.