Homework 7, due October 27th, 11:59pm

October 19, 2023

1. Download the image `horse025b.png` from Canvas, containing an image of a horse shape of size 84×128 pixels.

The goal of this project is to train a regression Neural Network to predict the value of a pixel $I(x, y)$ given its coordinates (x, y). We will use the square loss functions on the training examples $(x_i, y_i, I(x_i, y_i)), i = 1, ..., n$:

$$S(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(I(x_i, y_i) - f_{\mathbf{w}}(x_i, y_i))^2 \tag{1}$$

where $(x_i, y_i) \in \{1, ..., 84\} \times \{1, ..., 128\}$ are all $n = 128 \cdot 84 = 10752$ pixels of the image.

All NNs $f_{\mathbf{w}}(x, y)$ we will train have a 2D input (x, y) and a 1D output, but we will experiment with different numbers of hidden layers.

Try to use a GPU and CUDA for faster training. If you want, for better convergence, you could try to standardize the inputs (x_i, y_i) to have zero means and std 1, and the outputs to have zero mean.

a) Train a NN with one hidden layer containing 128 neurons, followed by ReLU. Train the NN for 300 epochs using the square loss (1). Use the SGD optimizer with minibatch size 64, and an appropriate learning rate (e.g. 0.003). Reduce the learning rate to half every 100 epochs. Show a plot of the loss function vs epoch number. Display the image reconstructed from the trained NN $f_{\mathbf{w}}(i, j), i \in \{1, ..., 84\}, j \in \{1, ..., 128\}$. (2 points)

b) Repeat point a) with a NN with two hidden layers, first one with 32 neurons and second one with 128 neurons, each followed by ReLU. (2 points)

c) Repeat point a) with a NN with three hidden layers, with 32, 64 and 128 neurons respectively, each followed by ReLU. (2 points)

d) Repeat point a) with a NN with four hidden layers, with 32, 64, 128 and 128 neurons respectively, each followed by ReLU. (3 points)