UNIVERSITÉ CÔTE D'AZUR

MSc. Data Science & Artificial Intelligence

Advanced Deep Learning

Dr Alessandro BETTI

Assignment 1

Author: Joris LIMONIER
joris.limonier@gmail.com

Due: October 5, 2022

Contents

List of Figures

1 Exercise 1

2 Exercise 2

2.1 Question 1

I was not familiar with the PyTorch library, so I had to perform some research in order to know how its methods/classes work. I consulted several sources, including the following, which were useful in order to answer this question:

1. https://discuss.pytorch.org/t/how-sgd-works-in-pytorch/8060/2
2. https://discuss.pytorch.org/t/performing-mini-batch-gradient-descent-or-stochastic-gradient-descent-on-a-mini-batch/21235
3. https://pytorch.org/docs/stable/generated/torch.optim.SGD.html

According to those sources, PyTorch's SGD actually computes a full-batch (vanilla) Gradient Descent, based on the data that is passed to it. It is my understanding that in order to perform actual mini-batch (*i.e.* where $1 <$ batch size $<$ number of observations), one simply needs to give subsets of the data a each iteration.
In our case, we use the full dataset in `outputs = net(inputs)`, which is why we perform full-batch GD, although we call the `optim.SGD` class.

2.2 Question 2

Figures 1, 2 and 3 show a comparison between train and test cross entropy loss. The leftmost diagrams have a linear scale for the vertical axis, while the rightmost diagrams have a logarithmic scale for the vertical axis. Beware that the number of epochs varies significantly between figures. Indeed, more parameters in the network implies longer training times per epoch, which in turn means that running for the same time both experiments results in different number of epochs.

We see that in both experiments the train error keeps decreasing, while the test error

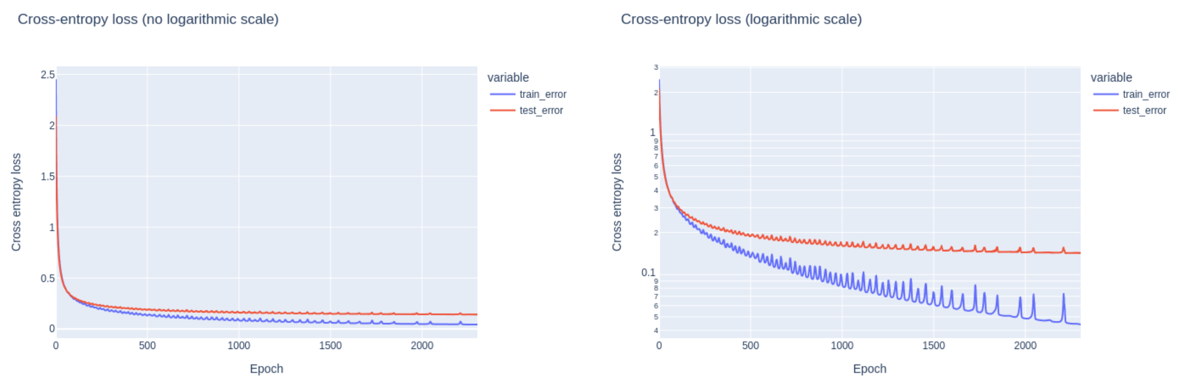

Figure 1: Cross entropy loss, train vs test (300 fully-connected).

stays constant. This means that our network is overfitting and there is no need to continue the experiment further. One way to prevent wasting time training when the network isn't

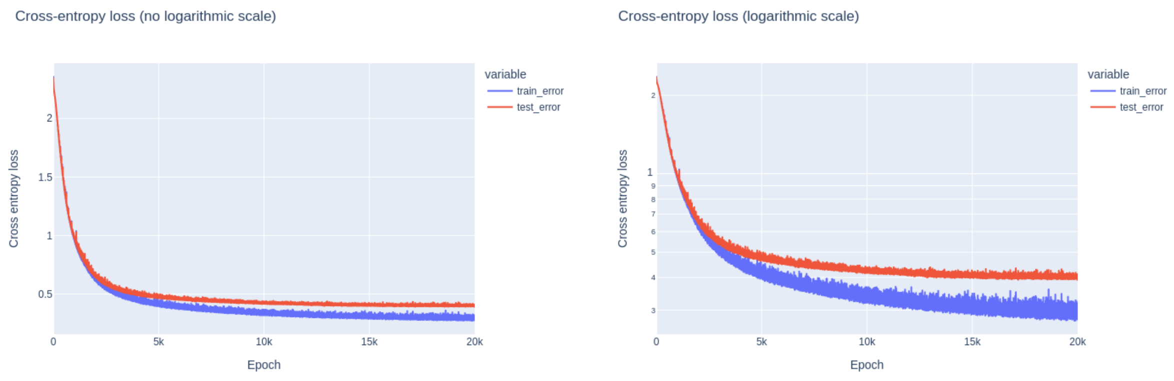

Figure 2: Cross entropy loss, train vs test $(10 - 10$ fully-connected).

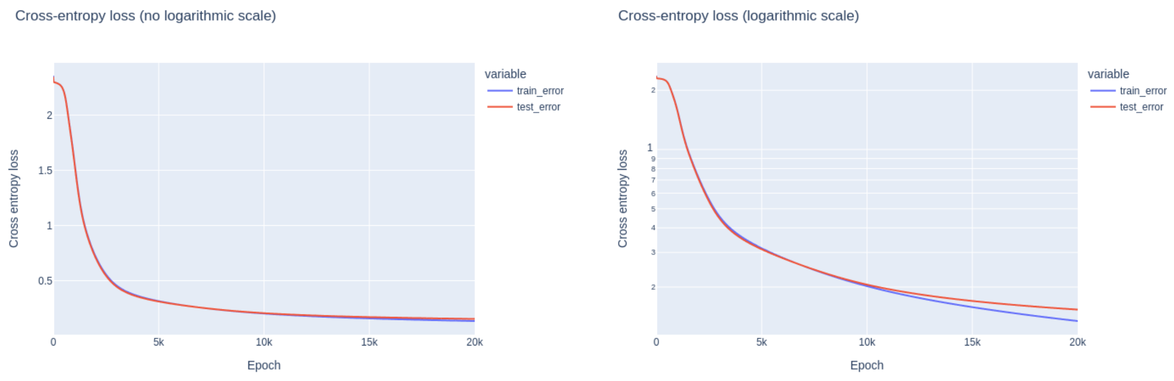

Figure 3: Cross entropy loss, train vs test $(20 - 20$ fully-connected).

making actual progress is to print the loss, or even to plot the loss curves, every couple iterations. We estimate however that the over-fitting phenomenon isn't too extreme in our case.

2.3 Question 3

Let $\beta : \mathbb{N}^{l+1} \rightarrow \mathbb{N}$ denote the number of parameters in a fully connected neural network. Then β is given by:

$$\beta(h_0, \ldots, h_l) := \sum_{i=1}^{l} (h_{i-1} * h_i) + h_i \tag{1}$$

Indeed, the multiplication term "$h_{i-1} * h_i$" accounts for the weights by counting the connections from a layer to the next. The last term "h_i" is responsible for counting the biases on each of the hidden layers, as well as the output layer.
We compute β for each of our network architectures above, with $h_0 = 28 * 28 = 784$ in all

cases, since this is the size of the input:

$$\begin{aligned}\beta(784, 10, 10, 10) &= (784 * 10 + 10) + (10 * 10 + 10) + (10 * 10 + 10)\\ &= 7850 + 110 + 110\\ &= 8070\end{aligned}$$

2.4 Question 4

2.5 Question 5

2.6 Question 6

2.7 Question 7