Computing Mutual Information (MI)

(notes on subtask 2 from our project plan)

The MI between two random variables A and B distributed according to $p(a, b)$ is defined as:

$$I(A; B) = \sum_a \sum_b p(a, b) \log \frac{p(a, b)}{p(a)p(b)}, \tag{1}$$

where $p(a)$ and $p(b)$ are the marginal distributions of A and B, respectively, defined as $p(a) = \sum_y p(a|b)p(b)$ (and vice versa).

We are looking to compute the quantities $I(X; T)$ and $I(T; Y)$, that is the MI between the input X or label Y and a representation of the input given by a hidden layer T, respectively.

- input: $X = x$,

  $x \in [0, 1]^n$, $n = 784$, $|X| = 60000$ in the MNIST dataset

  $x \in \{0, 1\}^n$, $n = 12$, $|X| = 4096$ in the toy dataset

- label: $Y = y$,

  $y \in \{0, 1\}^n$, $n = |Y| = 10$ in the MNIST dataset

  $y \in \{0, 1\}^n$, $n = |Y| = 2$ in the toy dataset (binary classification)

- hidden layer (input representation): $T = t$,

  $t \in \mathbb{R}^n$, $n = \#$ units in the layer, i.e. depends on the network architecture and is i.g. different for each hidden layer

Remarks on the range of values that x, y or t can take

The values of x and y are given by the dataset. In case of MNIST, the inputs are greyscale images of size 28×28 pixels, each pixel taking any value from the *continuous* interval between 0 and 1 (0 for the color white, 1 for black). The images display handwritten digits, and the learning task is to classify the images according to the 10 classes corresponding to digits 0-9. The label to each image is encoded as a "1-hot vector", that is a vector of length 10, with only the element corresponding to the assigned digit being non-zero. Example: If an image x displays the digit 3, the corresponding label is $y = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$.

In case of the toy dataset, the inputs are binary vectors of length 12, i.e. each element can take the value 0 or 1 (note the different brackets, square vs. curly, in the notation above!). The classification is binary, i.e. the labels only take two possible values (0 and 1). Thus, the "1-hot" vector y has length 2 in this case.

The possible values of t depend on the network architecture in two ways: The number of elements in the vector is given by the number of units in the layer, and the possible values of each element depend on the activation function of the respective units. For *sigmoid* and *tanh* functions, the output is a continuous number on the interval $[0, 1]$ and $[-1, 1]$, respectively.

The network architecture for the toy dataset is described in section 3 of the paper: "up to" 7 fully connected hidden layers, with 12-10-7-5-4-3-2 neurons, **hyperbolic tangent** function as activation for all neurons except for the final layer, **sigmoidal** activation for the final layer.

For calculations on the MNIST dataset, the authors unfortunately does not specify any details. The only mention of it can be found in the paper on page 17: *"[...] we examined the the SG statistics for a standard large problem (MNIST classification) and found the transition to the SGD diffusion phase."* We assume for now that they use the standard architecture from the Tensorflow tutorial.

Computing MI on the toy dataset

Section 3.2 of the paper provides the following information:

Each hidden layer of the network is treated as a singe random variable, denoted as T (or T_i, but the layer index i will be omitted to simplify notation). In order to compute the MI, we need to estimate the probability distributions $p(x, t) = p(t|x)p(x)$ and $p(t, y) = p(t|y)p(y)$.

Since all configurations are equally probable, the distribution for X is uniform:

$$p(x) = \frac{1}{4096} = const \equiv p_x. \tag{2}$$

The outputs of each neuron are discretized by binning into 30 equal intervals between -1 and 1. These discretized values are used to directly calculate $P(X, T_i)$ and $P(T_i, Y) = \sum_x P(x, Y)P(T_i|x)$.

For the following discussion we will slightly adjust the notation, using the following conventions: The input, label and each hidden layer are described by the randomvariables X, Y and T, respectively, and the lower-case letters x, y and t denote the random variable, i.e. the possible configurations (of the input or the hidden layer) or the possible labels. The probability functions will be denoted by lower-case p.

The MI between X and T is given by

$$I(X; T) = \sum_x \sum_t p(x, t) \log \frac{p(x, t)}{p(x)p(t)}. \tag{3}$$

Using the chain rule, the joint probability of x and t can be expressed through the conditional $p(t|x)$:

$$p(x, t) = p(t|x)p(x) = p(t|x)p_x. \tag{4}$$

The number of all possible configurations of T is $|T| = 30^n$, where n is the number of units in the layer, and 30 is the number of possible values each unit can take, given by the number of bins. Given a network (fixed set of weights and biases), a configuration x will be mapped to a representation t. Since t is a coarsened representation of x, it is likely that several different configurations x will be mapped onto the same representation t. However, the conditional probability for a representation t given a configuration x is either 0 or 1, i.e. a delta function:

$$p(t|x) = \delta_{t,t_x^{(\omega)}} \tag{5}$$

where $t_x^{(\omega)}$ denotes the representation t for a given x that is obtained from the network which is specified by a set of parameters ω. To keep notation slim, ω will be omitted in the following. Therefore,

$$\begin{aligned} I(X;T) &= \sum_x \sum_t p(t|x)p(x) \log \frac{p(t|x)p(x)}{p(x)p(t)} && (6)\\ &= p_x \sum_x \sum_t \delta_{t,t_x} \log \frac{\delta_{t,t_x}}{p(t)} && (7)\\ &= p_x \sum_x \log \frac{1}{p(t_x)} && (8)\\ &= -p_x \sum_x \log p(t_x) && (9) \end{aligned}$$

Similarly,

$$\begin{aligned} I(T;Y) &= \sum_y \sum_t p(y,t) \log \frac{p(y,t)}{p(y)p(t)} && (10)\\ &= \sum_y \sum_t \sum_x p(x,y)p(t|x) \log \frac{\sum_{x'} p(x',y)p(t|x')}{p(y)p(t)} && (11)\\ &= \sum_y \sum_t \sum_x p(x,y)\delta_{t,t_x} \log \frac{\sum_{x'} p(x',y)\delta_{t,t_{x'}}}{p(y)p(t)} && (12)\\ &= \sum_y \sum_x p(x,y) \log \frac{\sum_{x'} p(x',y)\delta_{t_x,t_{x'}}}{p(y)p(t_x)} && (13)\\ &= p_x \sum_y \sum_x p(y|x) \log \frac{p_x \sum_{x'} p(y|x')\delta_{t_x,t_{x'}}}{p(y)p(t_x)} && (14) \end{aligned}$$

Note that the remaining Kronecker delta in Eq. (14) restricts the sum in the argument of the log to only those configurations x' with $t_{x'} = t_x$, where x is set by the sum outside of the log.

The marginal probability of the label is

$$p(y) = \sum_x p(y|x)p(x) = p_x \sum_x p(y|x), \tag{15}$$

the conditional probability for y given x was defined in the construction the toy dataset:

$$p(y = 1|x) = \frac{1}{1 + e^{-\gamma(f(x)-\theta)}}, \tag{16}$$

with parameters $\gamma = 30.5$ and $\theta = 34$.

Thus, the remaining quantity required to calculate the MIs is the marginal probability of a representation t_x of a given x, $p(t_x)$. The representation t_x is the output of the layer's units, binned in 30 equal intervals between -1 and 1. For concreteness, consider the following example:

The hidden layer has 3 units. For a given configurations x, the output of the units is $(0.98, 0.97, -0.99)$. The first two values fall into the bin number 30, while the last value is in bin number 1. This result is saved as a list of tuples denoting the bin number and the bin count: $\{(1, 1), (30, 2)\}$.

This procedure is performed for each configuration x. The occuring configurations t are recorded and counted.

$p(t_x) = counto f t_x / numberof distincttoccuredforallx.$