HW #3 CS159

Francesca-Zhoufan Li, Greg Stroot, Zhuofang Li
2136030, 2135380, 2136043

Due on April 29 2021

1 MLP sketch

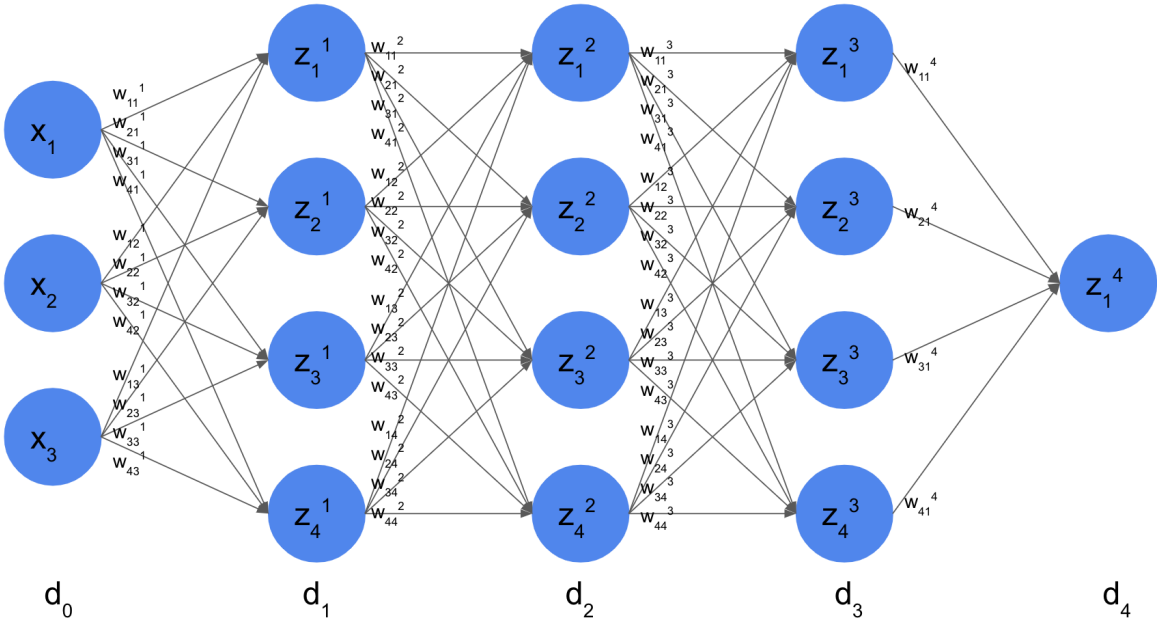

Figure 1: MLP Sketch

2 Wiring constraints

We are given:

$$y = \sum_{i=1}^{d} w_i x_i \tag{eq:2.1}$$

$$\sum_{i=1}^{d} w_i = 0 \tag{eq:2.2}$$

$$\sum_{i=1}^{d} w_i^2 = 1 \tag{eq:2.3}$$

where the input components $x_1, x_2, \ldots, x_d$ are uncorrelated random variables each with mean μ and variance σ^2 and we need to compute $\mathbb{E}[y]$ and $\mathrm{Var}\,[y]$

Plug in the expression for y from eq:2.1, we get:

$$\mathbb{E}[y] = \mathbb{E}\left[\sum_{i=1}^{d} w_i x_i\right] = \sum_{i=1}^{d} \mathbb{E}\left[w_i x_i\right] \tag{eq:2.4}$$

As w_i are constants, we can pull them out as the following:

$$\mathbb{E}[y] = \sum_{i=1}^{d} w_i \,\mathbb{E}\left[x_i\right] \tag{eq:2.5}$$

As we know that the mean of a random variable is equal to its expected value, meaning $\mathbb{E}[x_i] = \mu$, and thus we get:

$$\mathbb{E}[y] = \sum_{i=1}^{d} w_i \mu = 0 \tag{eq:2.6}$$

Now that μ is a constant, we can then pull that out from eq:2.6 with eq:2.2 to get:

$$\mathbb{E}[y] = \mu \sum_{i=1}^{d} w_i = 0 \tag{eq:2.7}$$

Explanation
Given that the coefficients of w_i sum to zero, the linear transform has an expectation of zero. Hence, if we want to have a zero-mean transformation, we should ensure that $\Sigma w_i = 0$. As discussed in class, a network with zero-mean can be optimally trained, so we should ensure we have this property at every training step.
Additionally, since we are doing linear transformations, the mean does not get effected.

Now, for the variance, we again plug in the expression for y from eq:2.1, and get:

$$\text{Var}\left[y\right] = \text{Var}\left[\sum_{i=1}^{d} w_i x_i\right]. \tag{eq:2.8}$$

This directly follows from $\mathbb{E}[y] = 0$. Since the variables are uncorrelated and we know that w_i is constant, meaning

$$\text{Cov}\left[w_i x_i, w_j x_j\right] = 0 \quad \text{for } i \neq j \tag{eq:2.9}$$

or

$$\text{Var}\left[y\right] = \sum_{i,j=1}^{N} w_i w_j \text{Cov}\left[x_i, x_j\right] = \sum_{i=1}^{d} w_i^2 \text{Var}\left[x_i\right] + \sum_{i \neq j} w_i w_j \text{Cov}\left[x_i, x_j\right] \tag{eq:2.10}$$

So we can simplify eq:2.10 to get:

$$\text{Var}\left[y\right] = \sum_{i=1}^{d} w_i^2 \text{Var}\left[x_i\right] \tag{eq:2.11}$$

Given each x_i has μ and σ^2 as its mean and variance, we have:

$$\text{Var}\left[y\right] = \sum_{i=1}^{d} w_i^2 \sigma^2 \tag{eq:2.12}$$

With σ^2 as a constant and eq:2.3, we have:

$$\text{Var}\left[y\right] = \sigma^2 \sum_{i=1}^{d} w_i^2 = \sigma^2 \tag{eq:2.13}$$

Explanation
Again when performing a linear transform, we find that if we have $\Sigma w_i^2 = 1$ then the variance of the input does not change at all. This, again, is optimal for training a neural network, so we should ensure that we have $\Sigma w_i^2 = 1$ at every training step.

3 Deep perturbation theory

For the following "deep linear network", given by,

$$y(x; W_2, W_1) \coloneqq W_2 W_1 x, \tag{1}$$

prove the following inequality, for two perturbation matricies $\Delta W_1, \Delta W_2$:

$$\frac{||y(x; W_2 + \Delta W_2, W_1 + \Delta W_1) - y(x; W_2, W_1)||_2}{||y(x; W_2, W_1)||_2} \leq C\Big[\Big(1 + \frac{||\Delta W_2||_F}{||W_2||_F}\Big)\Big(1 + \frac{||\Delta W_1||_F}{||W_1||_F}\Big) - 1\Big] \tag{2}$$

Let us start by expanding out the right hand side, per the hints:

$$C\Big[\Big(1 + \frac{||\Delta W_2||_F}{||W_2||_F}\Big)\Big(1 + \frac{||\Delta W_1||_F}{||W_1||_F}\Big) - 1\Big] \tag{3a}$$

$$= C\Big[\frac{||\Delta W_2||_F}{||W_2||_F} + \frac{||\Delta W_1||_F}{||W_1||_F} + \frac{||\Delta W_1||_F ||\Delta W_2||_F}{||W_1||_F ||W_2||_F}. \tag{3b}$$

Now that we know the form of the upper bound we are looking for, we may try to manipulate the LHS of the inequality to reach that upper bound.

$$\frac{||y(x; W_2 + \Delta W_2, W_1 + \Delta W_1) - y(x; W_2, W_1)||_2}{||y(x; W_2, W_1)||_2} \tag{4a}$$

$$= \frac{||(W_2 + \Delta W_2)(W_1 + \Delta W_1)x - W_1 W_2 x||_2}{||W_1 W_2 x||_2}$$

$$= \frac{||(W_1 \Delta W_2 + \Delta W_1 W_2 + \Delta W_1 \Delta W_2)x||_2}{||W_1 W_2 x||_2}$$

$$\leq \frac{||(W_1 \Delta W_2 + \Delta W_1 W_2 + \Delta W_1 \Delta W_2)||_F ||x||_F}{||W_1 W_2 x||_2}$$

$$\leq \frac{[||W_1 \Delta W_2||_F + ||\Delta W_1 W_2||_F + ||\Delta W_1 \Delta W_2||_F]||x||_F}{c_A[||W_1||_F ||W_2||_F ||x||_F]}$$

$$\leq \frac{||W_1 \Delta W_2||_F + ||\Delta W_1 W_2||_F + ||\Delta W_1 \Delta W_2||_F}{c_A ||W_1||_F ||W_2||_F}$$

$$\leq \frac{||W_1 \Delta W_2||_F + ||\Delta W_1 W_2||_F + ||\Delta W_1 \Delta W_2||_F}{c_A ||W_1||_F ||W_2||_F} \tag{4b}$$

By setting $C \equiv 1/c_A$, we arrive at our desired result.

Using an educated guess, we expect a form, like the following,

$$\frac{||y(x; W_3 + \Delta W_3, W_2 + \Delta W_2, W_1 + \Delta W_1) - y(x; W_2, W_1)||_2}{||y(x; W_3, W_2, W_1)||_2} \tag{5}$$

$$\leq C\Big[\Big(1 + \frac{||\Delta W_3||_F}{||W_3||_F}\Big)\Big(1 + \frac{||\Delta W_2||_F}{||W_2||_F}\Big)\Big(1 + \frac{||\Delta W_1||_F}{||W_1||_F}\Big) - 1\Big],$$

for the linear network with 3 layers instead of two.

4 Linear neuron yields a Gaussian process

We have the linear neuron $y(x) = \sum_{i=1}^{d} w_i x_i$ and a collection of n input vectors $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$
We need to show that if the weights $\{w_i\}_{i=1}^{d}$ are drawn iid $\mathcal{N}(0, \sigma^2)$,
then the outputs $y(x^{(1)}), y(x^{(2)}), \ldots, y(x^{(n)})$ are jointly Normal.

Let's start by stacking the inputs into a "Data matrix" as the following:

$$\mathbf{X} = \begin{bmatrix} \text{———} & x^{(1)} & \text{———} \\ \text{———} & x^{(2)} & \text{———} \\ & \vdots & \\ \text{———} & x^{(n)} & \text{———} \end{bmatrix} \tag{eq:3.1}$$

Now, we can write the output as:

$$\begin{bmatrix} y(x^{(1)}) \\ y(x^{(2)}) \\ \vdots \\ y(x^{(n)}) \end{bmatrix} = \begin{bmatrix} \text{———} & x^{(1)} & \text{———} \\ \text{———} & x^{(2)} & \text{———} \\ & \vdots & \\ \text{———} & x^{(n)} & \text{———} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \tag{eq:3.2}$$

or

$$\mathbf{y} = \mathbf{X}\mathbf{w} \tag{eq:3.3}$$

We know that for the weights $\{w_i\}_{i=1}^{d}$ are drawn iid $\mathcal{N}(0, \sigma^2)$, so the components of $\mathbf{w}$ are jointly Gaussian or $\mathbf{w}$ is a Guassian random vector, meaning the vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma}$ is the finite con variance
The moment generating functions of $\mathbf{w}$ with $\mathbf{t}$ as a fixed vector is:

$$\mathbb{E}[e^{\mathbf{t}^T \mathbf{w}}] = e^{\mathbf{t}^T \mathbf{0} + \frac{1}{2}\mathbf{t}^T \mathbf{\Sigma} \mathbf{t}} \tag{eq:3.4}$$

Then, the moment generating functions of $\mathbf{X}\mathbf{w}$ replacing $\mathbf{t}^T$ with $\mathbf{t}^T\mathbf{X}$ and thus $\mathbf{t}$ with $\mathbf{X}^T\mathbf{t}$:

$$\mathbb{E}[e^{\mathbf{t}^T \mathbf{X}\mathbf{w}}] = e^{\mathbf{t}^T \mathbf{X}\mathbf{0} + \frac{1}{2}\mathbf{t}^T \mathbf{X}\mathbf{\Sigma}\mathbf{X}^T \mathbf{t}} \tag{eq:3.4}$$

By uniqueness of the moment generating functions, we note from eq:3.4 that a linear transformation of a Gaussian random vector is still Gaussian, and thus $\mathbf{y}$ is also a Gaussian vector and the process is a Gaussian process:

$$\mathbf{y} = \mathbf{X}\mathbf{w} \sim \mathcal{N}(\mathbf{X}\mathbf{0}, \mathbf{X}\mathbf{\Sigma}\mathbf{X}^T) \tag{eq:3.5}$$

Thus, we can conclude that the linear neuron yields a Gaussian process.

Now we want to compute the mean $\mathbb{E}[y(x^{(1)})]$ of this Gaussian process.
We know that

$$y(x^{(1)}) = \sum_{i=1}^{d} w_i x_i^{(1)} \tag{eq:3.6}$$

and

$$y(x^{(2)}) = \sum_{i=1}^{d} w_i x_i^{(2)} \tag{eq:3.7}$$

$$\mathbb{E}[y(x^{(1)})] = \mathbb{E}\left[\sum_{i=1}^{d} w_i x_i^{(1)}\right] = \sum_{i=1}^{d} \mathbb{E}\left[w_i x_i^{(1)}\right] \tag{eq:3.8}$$

As w_i and $x_i^{(1)}$ are independent of each other, we can use $\mathbb{E}(XY) = \mathbb{E}(X) \cdot \mathbb{E}(Y)$ to split the expectation in eq:3.8 as the following:

$$\mathbb{E}[y(x^{(1)})] = \sum_{i=1}^{d} \mathbb{E}\left[w_i\right] \mathbb{E}[x_i^{(1)}] \tag{eq:3.9}$$

As we know that the weights $\{w_i\}_{i=1}^{d}$ are drawn iid $\mathcal{N}(0, \sigma^2)$ and thus $\mathbb{E}[w_i] = 0$ in eq:3.9:

$$\mathbb{E}[y(x^{(1)})] = \sum_{i=1}^{d} 0\, \mathbb{E}[x_i^{(1)}] = 0 \tag{eq:3.10}$$

Now we want to compute the second moments $\mathbb{E}[y(x^{(1)})y(x^{(2)})]$ of this Gaussian process.
We plug in eq:3.6 and eq:3.7:

$$\mathbb{E}[y(x^{(1)})y(x^{(2)})] = \mathbb{E}\left[\sum_{i=1}^{d} w_i x_i^{(1)} \sum_{j=1}^{d} w_j x_j^{(2)}\right] \tag{eq:3.11}$$

Pulling out the sum, we can modify eq:3.11 to:

$$\mathbb{E}[y(x^{(1)})y(x^{(2)})] = \sum_{ij} \mathbb{E}\left[w_i w_j\right] x_i^{(1)} x_j^{(2)} \tag{eq:3.12}$$

Note that w_i and w_j are independent with $\mathbb{E}[w_i] = \mathbb{E}[w_j] = 0$, we have $\mathbb{E}[w_i w_j]$ equal to 0 unless $i = j$:

$$\mathbb{E}[w_i w_j] = \delta_{ij}\sigma^2 \tag{eq:3.13}$$

Plug eq:3.13 into eq:3.12, we now have

$$\mathbb{E}[y(x^{(1)})y(x^{(2)})] = \sum_{i} \sigma^2 x_i^{(1)} x_i^{(2)} = \sigma^2 x^{(1)T} x^{(2)} \tag{eq:3.14}$$

5 Verifying the relu MLP covariance

The empirical covariance matrix:

$$\begin{bmatrix} 1.0376 & 1.0315 & 0.9744 & 0.9265 \\ 1.0315 & 1.0326 & 0.9691 & 0.9215 \\ 0.9744 & 0.9691 & 1.0279 & 0.8939 \\ 0.9265 & 0.9215 & 0.8939 & 1.0591 \end{bmatrix}$$

The theoretical covariance matrix:

$$\begin{bmatrix} 1.0000 & 1.0000 & 0.9771 & 0.9394 \\ 1.0000 & 1.0000 & 0.9726 & 0.9357 \\ 0.9771 & 0.9726 & 1.0000 & 0.9137 \\ 0.9394 & 0.9357 & 0.9137 & 1.0000 \end{bmatrix}$$

Those two matrices are almost equal, which confirms the claim.

Code for empirical cov:

```
covariance=torch.zeros(4,4)
for i in range(4):
  for j in range(4):

    covariance[i][j]=torch.mean(torch.mul(output[i,:],output[j,:]))-torch.mean(output[i,:])*torch.mean(output[j,:])
```

Code for theoretical cov:

```
covariance2=torch.zeros(4,4)
for i in range(4):
  for j in range(4):


    covariance2[i][j]=h(h(torch.mean(torch.mul(output[i,:],output[j,:]))))
```